High-Throughput Dynamic Time Warping Accelerator for Time-Series Classification With Pipelined Mixed-Signal Time-Domain Computing

Zhengyu Chen, *Member, IEEE*, and Jie Gu, *Senior Member, IEEE*

***Abstract*— 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 work presents a mixed-signal DTW accelerator utilizing mixed-signal time-domain (TD) computing 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 65-nm CMOS test chip was implemented and measured. The results show more than 9× improvements in throughput compared with prior work on TSC. As most existing TD designs suffer from the lack of TD storage elements, this work utilizes sequential circuit elements in TD computing extending the capability of time-based circuits.**

***Index Terms*— Dynamic programming, dynamic time warping (DTW), energy efficient computing, machine learning, mixed-signal time-domain (TD) computing (MSTC), time flip-flop (TFF), time-series classification (TSC).**

I. Introduction

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 U.S. $6 billion in 2018 to U.S. $90 billion in 2025 specially contributed by the edge devices [1]. Compared with general-purpose CPU or microcontroller, the rapid growth in special purpose application-specified integrated circuit (ASIC) accelerators is attributed by several factors from the current technology trends. First of all, the computing efficiency of general-purpose processors such as CPUs does not meet the heavy computation demand from many modern machine learning algorithms or similar special purpose computing algorithms due to the overhead of instruction encoding/decoding support as well as the memory-related operations in general-purpose microprocessors [2]. As an example, for the deoxyribonucleic acid (DNA) sequencing tasks, special ASICs are shown to achieve hundreds of times of performance enhancement in comparison (CMP) with CPU or GPU [3]. Such a performance loss from general-purpose processors is sometimes intolerable for the real-time classification of time-series signals, for instance, human motion classification in a feedback control system [4]. Second, transistors (excluding the most advanced technology) have become cheaper and cheaper enjoying 20%–30% cost reduction leading to favorable adaption of special purpose ASIC designs with power and performance advantages [5]. Third, 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 large justifying the cost of special purpose accelerator chips with examples of tensor processing unit (TPU) from Google, Mountain View, CA, USA [6], Amazon Web Services (AWSs) Inferential chip from Amazon, Seattle, WA, USA [7], self-driving AI chip from Tesla, Palo Alto, CA, USA [8].

The fast development of application-specific accelerators also creates new opportunities in design space where non-conventional computing methodology is being explored in search of higher computing efficiency. The energy improvement of conventional digital circuits has reached a bottleneck because 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. Besides, the leakage power of digital design also contributes significantly to total power consumption and the leakage power is also mainly determined by the technology in use. As a result, it is urgent to find alternative computing methods that can bring efficiency beyond the conventional digital approach. 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 (NN) exploited efficient analog computation and digital intra-network communication for feature extraction and classification with 7.5× more energy efficient than an equivalent digital design [9]. A switched capacitor-based analog matrix multiplication design was proposed to perform multiply-accumulate-operation (MAC) operations efficiently for machine learning tasks with similar accuracy compared with digital counterpart [10]. In addition (ADD), memristor or RRAM-based computing explores the voltage, current, and resistance relationship to achieve much higher efficiency

Manuscript received March 18, 2020; revised July 23, 2020; accepted August 20, 2020. Date of publication September 17, 2020; date of current version January 28, 2021. This article was approved by Associate Editor Edith Beigne. This work was supported in part by the National Science Foundation under Grant CCF-1846424. *(Corresponding author: Zhengyu Chen.)*

The authors are with the Department of Electrical and Computer Engineering, Northwestern University, Evanston, IL 60208 USA (e-mail: zhengyuchen2015@u.northwestern.edu).

Color versions of one or more of the figures in this article are available online at https://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/JSSC.2020.3021066

on multiplier-accumulator (MAC) operations for deep neural network (DNN) applications [11]. One weakness of analog computing is its sensitivity to process variation and error-prone operations. However, as already being well studied, analog computing, which incorporates analog building blocks such as amplifiers and analog-to-digital converters, suffers from requirement of headroom, static leakage, and poor compatibility with digital circuits [12].

To address the above challenges, in this work, we present a time-domain (TD) design methodology to conduct emerging applications with higher throughput and low energy consumption. More specifically, a dynamic time warping (DTW) engine for time-series classification (TSC) using TD computing is proposed. Through a special design of TD flip-flop as a TD memory, this work realizes an efficient and high-throughput TD pipelined architecture.

A. Related Work

More recently, mixed-signal computing techniques using time-based circuits have drawn significant interests in some application regimes, e.g., signal processing and machine learning [13]–[20]. By using digital circuits to encode and process information in TD rather than in voltage domain, mixed-signal TD computing (MSTC) shows promise in many applications with high efficiency while staying mostly compatible with digital circuits [13], [14]. This leads to the benefit of both technology scalability as well as compatibility with large-scale digital design methodology, e.g., synthesis and place and route (P&R) with regular electronic design automation (EDA) tools [15]. Similar to analog computing where the multi-bit information is densely encoded in a single signal, MSTC preserves the benefits of energy efficiency in analog space but also suffers from process variation and low resolutions. Interestingly, in analog and mixed-signal design space, despite error-prone operations, a desirable error resilient feature is also observed where the most-significant-bit (MSB) has the least possibility for errors as compared with the opposite trend in a digital design [16]. This enables a favorable and flexible accuracy and performance tradeoff for the analog and mixed-signal computing. In this work, we explore mixed-signal TD design for a special purpose time-series analysis demonstrating significant benefits of such non-conventional computing techniques.

Several demonstrations have been developed in recent years using MSTC for realizing emerging applications [13]–[20]. For instance, a TD low-density parity-check (LDPC) design was demonstrated with 2× reduction in area compared with the digital implementation [17]. A swarm robotic system incorporating a TD reinforcement learning accelerator was implemented with over 30% saving of energy compared with the digital counterpart [14]. A TD convolutional NN (CNN) engine showed 12× improvement for energy efficiency compared with the other state-of-the-art digital implementations [18]. A TD accelerated image processing engine was delivered with 40% area and energy improvement compared with the digital counterpart [19]. A highly efficient time-based in-memory computing graph ASIC chip was realized using wavefront expansion and 2-D gradient control for solving single-source shortest path problems [20]. An NN-based cardiologist-level arrhythmia detection and classification engine was implemented in [31]. In ADD, for one edge per line-based time of arrival encodings, a memristor-based temporal memory design was recently published in [33].

However, there are still some limitations in the existing demonstrations.

1) There is a lack of memory in TD 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) Majority of existing works are confined to low-bit precisions, e.g., 1–4 bits [17]–[21].
4) Also, partially due to the lack of TD memory, most prior works suffer from excessive time-to-digital or digital-to-time conversion, leading to significant speed degradation 14], [18].

In this work, we present a DTW engine for TSC using TD computing. Through a special design of TD flip-flop as a TD memory, this work realizes an efficient and high-throughput TD pipelined architecture.

Note that, there is an interesting time-register design proposed in [32]; however, such a design cannot fulfill the computation demand in TD DTW design as described in the following.

1) The previous time-register design is used for time-to-digital converter (TDC) for phase-locked loop (PLL) design, while we focus on digital application leading to many different requirements, e.g., operation range, precision, and operation sequence.
2) The previous time register cannot be directly used as a TD register file for pipelining since in the output, the time delay is $T_{\text{out}} = T_{\text{full}} - T_{\text{in}}$. Additional conversion circuits are needed in order to realize the function of $T_{\text{out}} = T_{\text{in}}$. On the other hand, our time flip-flop (TFF) has a quite different design which is based on ring-based inverter chains. By nature, our proposed design can generate the output pulse of $T_{\text{out}} = T_{\text{in}}$.
3) The previous time-register design cannot deal with the overflow issue, as they need to reset the time-register manually when the whole capacity is reached for storing the TD information. On the other hand, due to the ring-based structure, our proposed TFF can automatically reset the ring when the ring is full. This special feature enables the capability of cascading several TFF into larger-bit TFF modules.

Prior work on DTW operation suffer from low throughput and high power. For example, the work from [3] and [21] have relatively low throughput due to a combinational circuit nature of the mixed-signal design and the ASIC design from [25]–[27] suffer from the high power and long delay of each CMP module. Compared with the previous state-of-the-art mixed-signal analog implementation [3], we introduced the TD pipeline computing structure and improved the throughput of GCUP by 2.4×–47× compared with the prior chip implementation in both analog and digital domains [21], [25]–[27]; the low throughput of prior work low throughput is due to a combinational circuit nature of the mixed-signal

design [3], [21] and the high power and long delay of each CMP module in ASIC design and the high power and long delay of each CMP module in ASIC design [25]–[27].

Hence, in this work, we implemented a low-power high-throughput TD DTW accelerator by utilizing pipelined architecture and systolic array-based data streaming scheme with novel TFF design. Overall, our design has improved the throughput of GCUP by 2.4×–47× compared with prior chip implementation in both analog and digital domains [21], [25]–[27].

B. Contribution of This Work

As extended from the previous publication in [13], in this work, we deliver a novel pipelined TD computing design to realize a commonly used algorithm for time-series analysis, i.e., DTW algorithm. More specifically, the contributions of this article are highlighted as follows.

1) *At Circuit Level:* We developed a special TD storage cell, namely, TFF. TFF can not only store multiple-bit TD information, i.e., 6 bit in this work, but also can work as a TD accumulator. TFF can be further cascaded into a wider TFF with 10 bit or more precision. In ADD, we also developed special circuits in TD, e.g., absolute (ABS) and minimum (MIN) modules to realize the TD DTW operation.
2) *At Architecture Level:* We presented a pipelined architecture with TFF circuit. The pipelined operation leads to an order-of-magnitude improvement in throughput and a scalable processing capability for time-series data.
3) *At System Level:* We realized a TD acceleration solution for DTW algorithm for TSC. A special data streaming flow was utilized to support pipelined operation. A highly automated design methodology was utilized in this work to eliminate the manual layout effort for the mixed-signal circuit design. Also, a systematic calibration scheme was applied to deal with process variations.

The proposed techniques were implemented to conduct DTW algorithm for TSC, e.g., electrocardiogram (ECG) classification, gesture recognition, DNA sequencing, and so on. We demonstrated such techniques in a 65-nm test chip with results showing orders of magnitude improvement compared with the state-of-the-art implementations. The remainder of this article is organized as follows. Section II introduces overall the background of TSC including the DTW algorithm. Section III presents the TD circuit technique and acceleration method. Section IV introduces the architecture level design methodology of the TD DTW engine. Chip implementation and measurement results are discussed in Section V.

II. Background

A. Time-Series Classification

A time series is a series of data points indexed, listed, or graphed in time order [22]. 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, ECG signal, and so on. The classification of time-series signals, e.g., an ECG signal, is commonly used for detection of special events or operational anomaly. However, 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 [21], [22]. Three conventional classification methods are being developed including the distance-, model-, and feature-based methods [22], [23]. The model- and feature-based methods are case-specific and complex to implement. For example, the hidden Markov model (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. Especially, DTW, a variant of the dynamic programming algorithm, has been widely used for TSC. In ADD, as machine learning introduced promising results in dealing with classification and detection workloads, a few NN-based works for TSC were implemented showing good classification results [31]. 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 also usually consumes more area and energy compared with the succinct distance-based methods. The strong capability for distance measurement for variable-speed temporal sequences makes DTW a popular method for TSC in broad applications, such as ECG diagnosis, motion detection, voice recognition, stock prediction, and so on. [22] In ADD, a similar dynamic programming-based approach is also being used in DNA sequencing for CMP of similarity between DNA pairs [21]. 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 (GCUP) throughput at 70-mW power consumption [21].

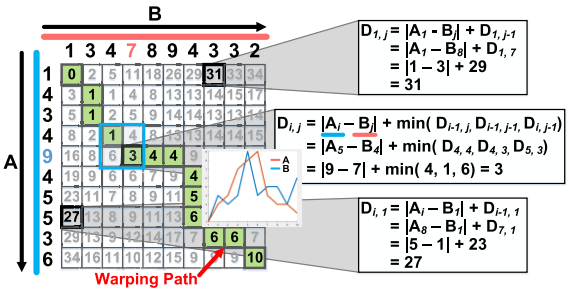

Fig. 1. DTW algorithm.

B. Dynamic Time Warping

Fig. 1 shows the basic principle of DTW, which detects similarities among temporal signals with variable speed. As shown in Fig. 1, for two time series A and B, $D_{i,j}$ can be formulated as the summation of ABS difference $|A_i - B_j|$ and the MIN 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 denote the ith and jth elements of A and B, respectively, and $D_{i,j}$ denotes the DTW value at

node $(i,\ j)$. The equation is written as follows:

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

A "warping path" is produced in order to align the two signals in time, as highlighted in Fig. 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, TD 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 TD computing to accelerate the DTW operations.

The time complexity of DTW algorithm implemented on this chip (matrix level) is $O(m \times n)$ where m and n represent the length of each time sequence. In each DTW module, it realizes the operation of $\text{ABS}(x - y) + \text{MIN}(a, b, c)$ whose time complexity is $O(1)$.

III. TD Acceleration Technique

A. Mixed-Signal TD Computing

The basic concept of MSTC is to represent data/information in the format of delay or length of time pulses and then process the information in TD with special mixed-signal circuits. More specifically, in this work, the information is encoded as the pulsewidth of the data-carrying time pulses. Many computation tasks, e.g., ADD, subtraction (SUB), and nonlinear logic operation, e.g., maximum (Max), MIN, and Compare, can be efficiently carried out in TD [16]. As shown in Fig. 2(a), a digital-to-time converter (DTC) or also referred to as time encoder is used to convert digital information into TD. Correspondingly, TDC carries the job to convert TD information back into digital domain. The circuit examples of DTC and TDC are depicted in Fig. 2(b) and (c).

In MSTC design, digital information is encoded in a linear fashion which is indeed a drawback in terms of information density in some perspectives. However, such an encoding scheme introduces the unique energy/area efficiency for TD computing in some computation especially non-linear operations, e.g., MAX, MIN, CMP, and so on. When it comes into the case that we need to deal with larger bit group operations, e.g., 8-bit or more, we partition the large bit group into several small bit groups. For example, we can partition 8-bit multiplication into 4 of the 4-bit multiplication to improve the area energy and throughput. In this design, we partition the 12-bit data path into 2 of the 5-bit data path.

Note that, among existing demonstrations of TD computing techniques including this work, floating-point (FP) operations have not been supported due to the complexity of some of the FP operations, such as shift and ADD operations.

The operation waveform of TDC is shown in Fig. 2(c). The basic concept is to delay the input pulse (T_{in}) by multiple times of TD single bit resolution (T_{s}) and compare the delayed input pulse with the reference time pulse (T_{ref}) to generate the digital output ($D[2^n - 1 : 0]$). In Fig. 2(d), the operation of $\text{CMP}(A + B,\ C + D)$ is implemented in simple TD circuits consisting of only tunable delay cells and standard cell circuits

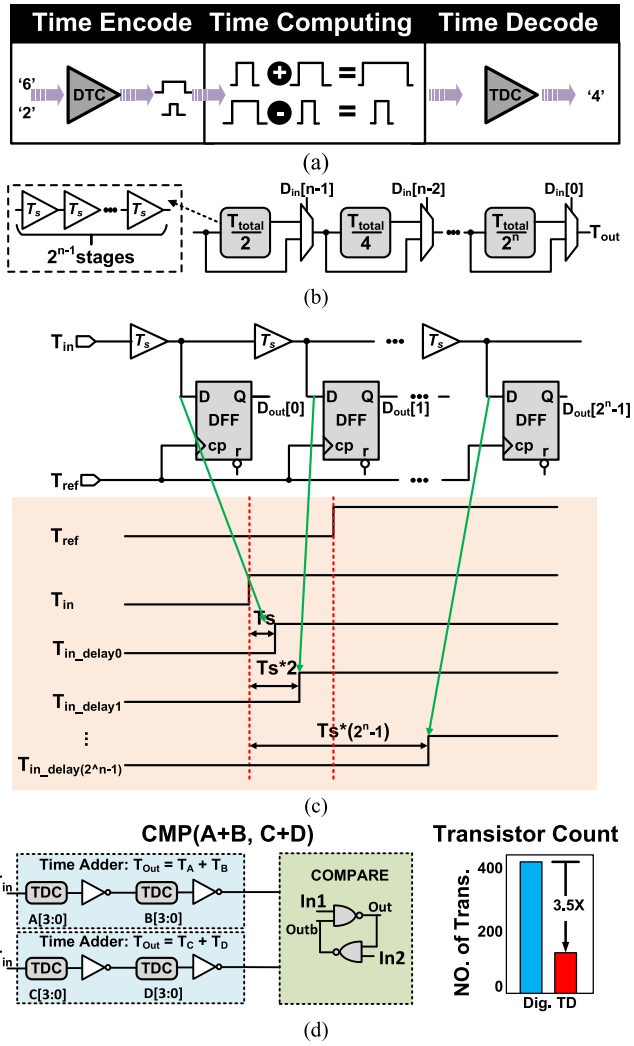

Fig. 2. (a) Overview of TD computing. (b) DTC. (c) TDC and waveform. (d) TD implementation example and transistor count CMP.

rendering 3.5× reduction in terms of transistor count. The detail of basic TD operations as well the corresponding circuits will be introduced in Section III-B.

As the nature of analog mixed-signal design, on-chip variation plays an important role in the MSTC design. We addressed the variation concern in the following three steps.

1) A comprehensive study on the impact of both global process voltage temperature (PVT) variation and local mismatch variation impact was conducted in [16]. Compared with local mismatch, global PVT variation has very relaxed impact for TD design since the relative delay among bits matter more compared with the operation. Hence, linearity matters the most rather than the ABS delay values. Local mismatch poses more challenges to TD design which is addressed below.
2) During the circuit design phase, the variation margin was carefully budgeted based on the application requirement. Since we typically target on error-tolerant applications, 1- or 2-bit error of final results would not cause significant accuracy degradation (∼2% degradation). We conduct variation analysis through the entire data path

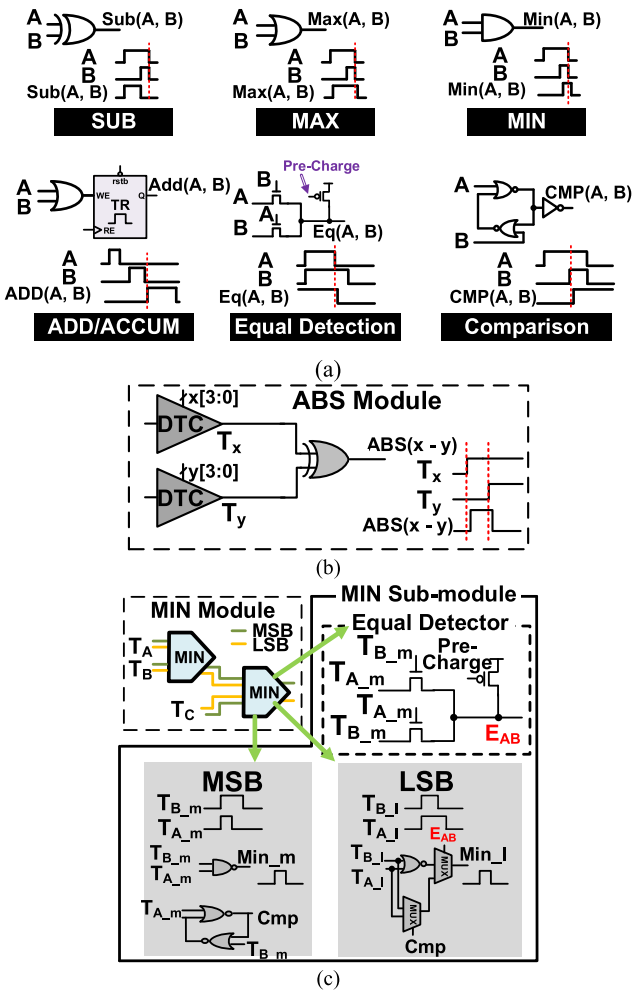

Fig. 3. Circuit details of TD circuits implemented in this work. (a) Basic TD circuits. (b) ABS module. (c) Three-input MIN module.

based on Monte Carlo simulation (under three sigma configuration) to understand the error caused by local mismatch. We then utilized such information to decide the single bit resolution in TD design, e.g., 40 ps per digital bit, to guarantee the final error is within 1 LSB in TD.

3) Moreover, a calibration scheme was introduced in this design by integrating calibration capability inside of some variation vulnerable modules based on Monte Carlo simulation on each individual module. We first find out which sub-module introduces the most variation and then implement calibration circuits to realize the best efficiency in terms of calibration. More information and testing results are introduced in Section V-B.

B. Basic TD Computing Circuits

As the fundamental building blocks, basic TD operations, i.e., SUB, MAX, MIN, ADD/accumulation, equal detection (EQ), and CMP, are specially designed with high energy and area efficiency, as depicted in Fig. 3(a). As shown in Fig. 3(a), some of the input signals are required to be overlapped while

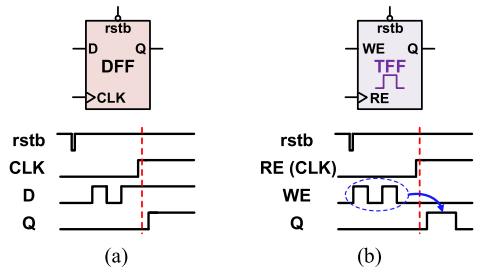

Fig. 4. Differences between (a) DFF and (b) TFF.

others are not. In order to guarantee the correctness of TD 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 and 2) we have special technique to make sure the rising or falling edges of two input TD signals are aligned in order to conduct the operation correctly. For example, by using the proposed TFF to latch TD signals, the output TD pulses are aligned by falling edge. The operations, such as CMP, MAX, and MIN, can be easily implemented in TD 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. Fig. 3(b) and (c) shows the MIN and ABS modules used in this work. 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 three-input MIN module consists of a two-input MIN module and one equal detector module. The data path is divided into MSB and LSB paths. As shown in Fig. 3(b) and (c), both MSB and LSB MIN modules are built by simple NAND, NOR, and MUX gates with corresponding waveform depicted.

As mentioned in Section II, the existing TD demonstrations suffer from excessive digital and TD conversion and the lack of internal storage. Missing the storage mechanism in TD causes a lack of TD sequential logic which is required for high-throughput pipelined structure or design of finite state machines in non-combinational circuits [21]. Thus, in this article, a novel TD storage cell, namely TFF, is introduced in Section III-C.

C. TFF Circuit

The proposed TFF takes time pulse as inputs and generates time pulse as the output triggered by the read enable signal. As shown in Fig. 4, compared with digital D-type flip-flop (DFF), the proposed TFF operates in a similar fashion but has some advanced features: 1) TFF can store multi-bit information in TD; 2) TFF takes multiple time pulses as input in a sequential order; and 3) accumulation operation can be naturally realized—the output pulsewidth equals to the width summation of input pulses.

Fig. 5(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 TD information with

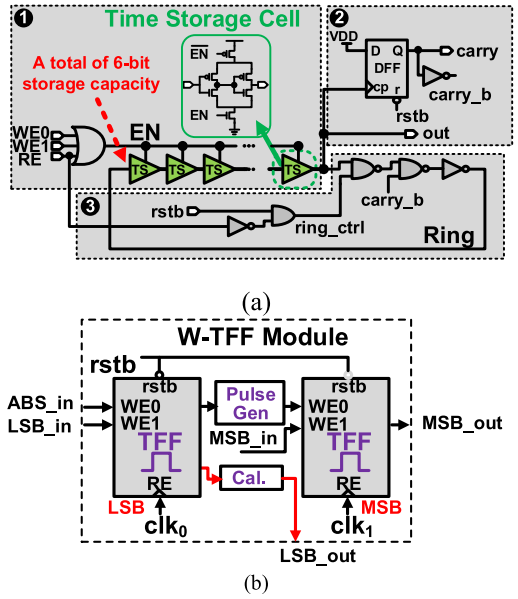

Fig. 5. TD flip-flop designs. (a) Circuit diagram of TFF. (b) Circuit diagram of the WTFF module.

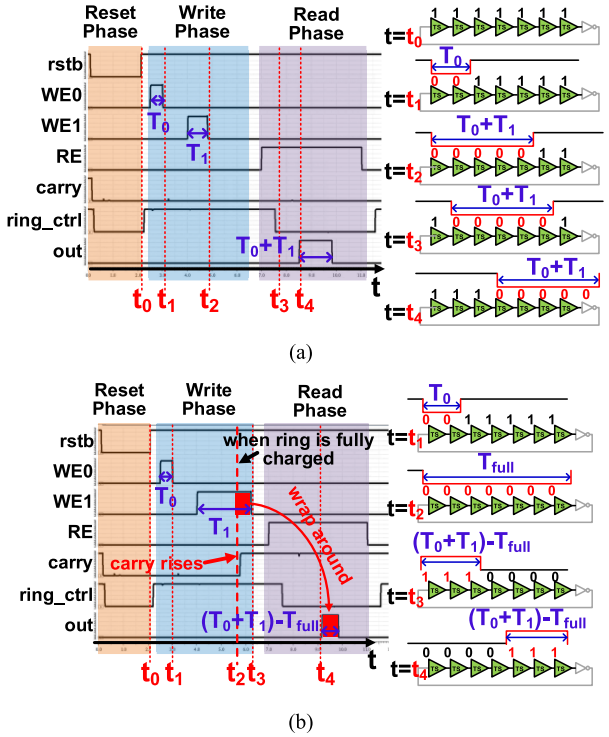

Fig. 6. Simulated waveform of TFF when (a) ring is not fully filled and (b) ring is fully filled.

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 TD signal and two TFFs are used to construct a 10-bit TD values separated into MSB and LSB units, leading to a wide-TFF (WTFF) module, as shown in Fig. 5(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 ADD, a MIN 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 Fig. 6. In the scenario when the input pulses are not large enough to fully fill the ring (overflow), the simulated waveform is shown in Fig. 6(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 pulsewidth 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 Fig. 6(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 ($t = t_3$–t_4). The "rotation" operation conveniently allows cascading TFFs into multi-bit groups rendering a scalable large numerical range of TFF.

IV. TD DTW Architecture

A. TD DTW Algorithm Mapping

As shown in (1), the core computations of DTW contain two non-linear operations—the ABS and MIN. Such operations can be efficiently realized in TD. The corresponding TD waveform for node $D_{i,j}$ of (1) is depicted in Fig. 7(a). The MIN value of its three ancestor nodes is carried by TD signal $T(\min(D_{i-1,j}, D_{i,j-1}, D_{i-1,j-1}))$ which is generated by the TD MIN module. The ABS difference is carried by TD signal $T(|A_i - B_j|)$ which is generated by the TD 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 TD implementation is shown in Fig. 7(b) with succinct topology and data path.

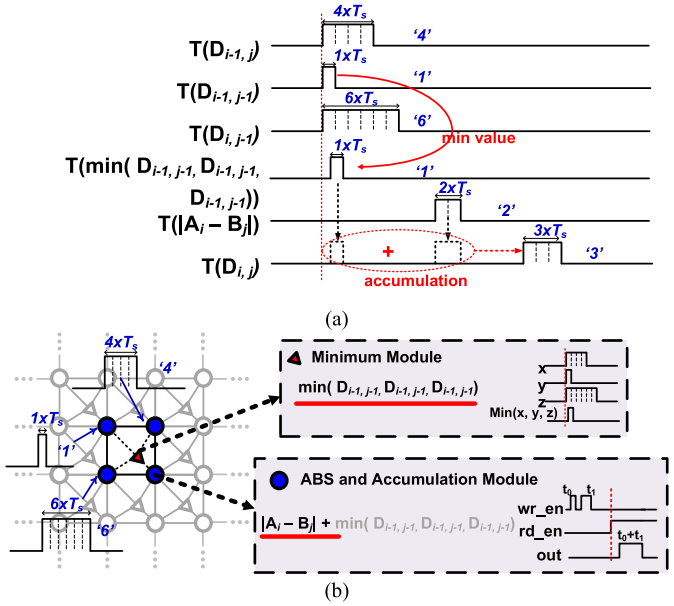

Fig. 7. TD DTW algorithm. (a) Waveform of TD DTW. (b) TD implementation of DTW.

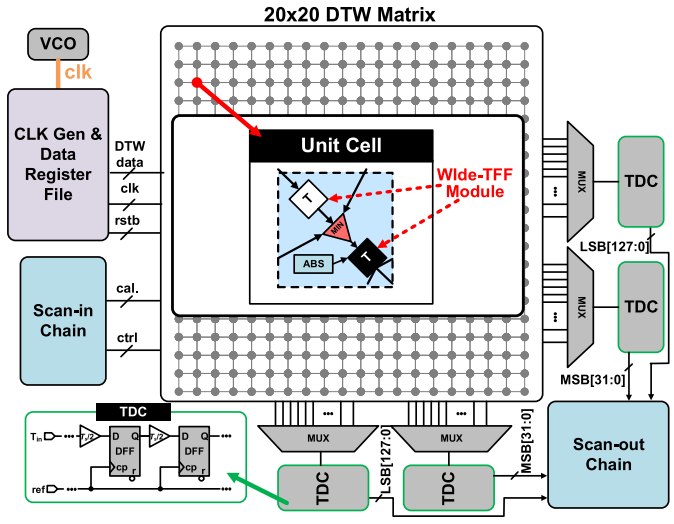

Fig. 8. Architecture diagram of implemented pipelined TD DTW.

B. Pipelined TD DTW Algorithm

The TD implementation of DTW described in Section IV-A 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.

One key element to enable the TD pipelined design is the TD information storage cell, i.e., TFF, as introduced in Section III-C. By inserting TFF to every node of the DTW matrix, the pipelined architecture can be realized. Fig. 8 shows

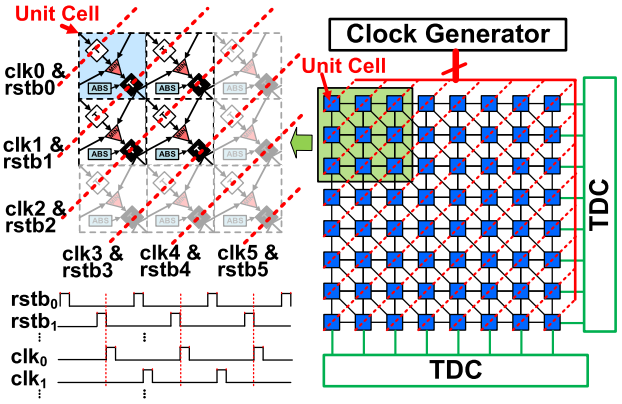

Fig. 9. Diagonal data path and pipeline stage structure of DTW engine.

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 Fig. 8, contains two 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 $(i-1,\ j-1)$ are one pipeline stage earlier than the nodes $(i-1,\ j)$ and $(i,\ j-1)$.

A 4-bit DTC is implemented inside ABS to convert input digital values into TD 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 as will be introduced in Section IV-C.

C. Pipelined Structure and Data Streaming Flow

Due to the use of the TFF, in every clock cycle, the TD pulses are propagated along the diagonal direction of the matrix, as depicted in Fig. 9. 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 with the non-pipeline mode.

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 Fig. 10(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 [see Fig. 10(b)]. Such a flow is similar to a systolic dataflow in other accelerators, e.g., Google's TPU design [6]. 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 × 4 b into

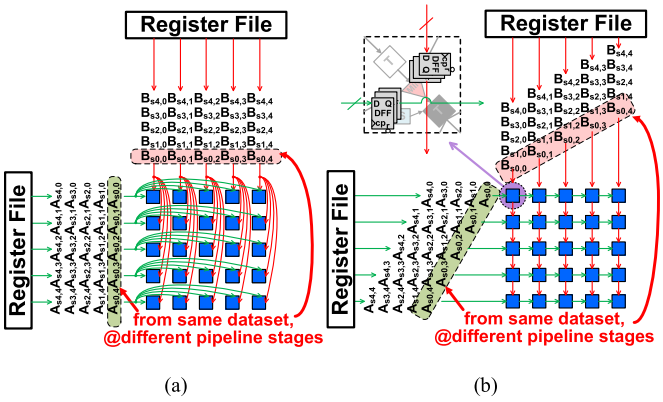

Fig. 10. Data streaming flow CMP between (a) brute-force data streaming flow and (b) systolic data streaming flow.

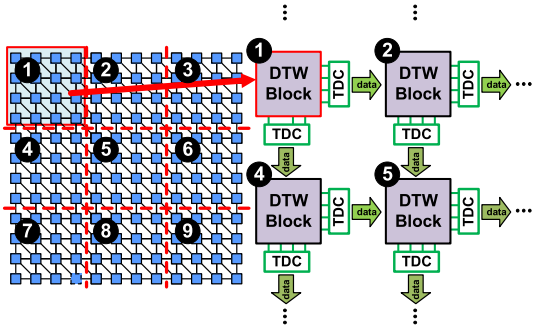

Fig. 11. Unfolding mode of the proposed DTW engine.

$2 \times 20 \times 4$ b at $20\times$ reduction. However, some calibration signals still need to be explicitly routed into each DTW node which makes the total reduction into $15\times$.

D. Unfolding DTW Operation

The pipelined operation allows fixed dimensions of the DTW engine to be unfolded for longer data sequences, as shown in Fig. 11. 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 TDCs every clock cycle and re-sent back for further operations.

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 this 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.

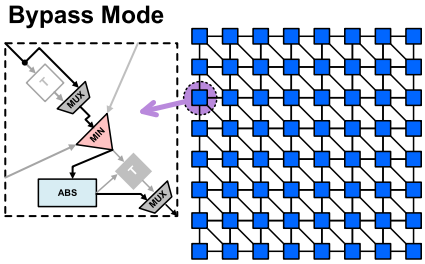

Fig. 12. Architecture diagram of non-pipelined DTW mode.

E. Non-Pipelined DTW Mode

The pipelined mode is essentially designed for accelerating multi-bit TSC. 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 TD operation, the pipelined mode is not efficient for low-resolution TSC, 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 Fig. 12.Different from pipelined case, in non-pipelined case, we encode information by the delay of rising edges instead of the pulsewidth of time pulses (similar to prior work [3]). Note that, the rising edge is naturally accumulated through the combinational block for "ADD" operation, as depicted in Fig. 7(b).

F. 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 TD design automation technique is utilized, as shown in Fig. 13 [15].

In the local module level, the implemented automation technique includes both the synthesis and P&R parts. The synthesis process involves two steps: 1) the register-transfer level (RTL) with customized syntax for TD logics is utilized to perform a special MSTC logic synthesis process which generates an initial gate-level netlist and 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 P&R process utilizes an adjacent constraint graph-based placement algorithm to realize the special signal mapping requirement in TD [15]. As a result, majority of the modules are automated except critical local cells, e.g., ring core of TFF.

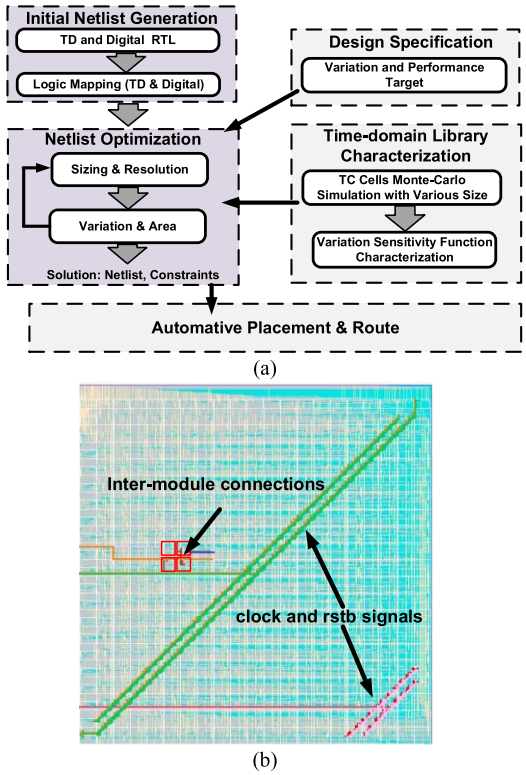

Fig. 13. Design automation techniques used in this work. (a) Design automation flowchart. (b) Layout result of 20 × 20 DTW matrix.

In the higher level, we developed placement script and utilized digital tool to conduct the layout as such an example shown in Fig. 13(b). The neighbor DTW nodes are placed right close 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 with hand layout.

G. DTW Matrix Calibration Scheme

Similar to analog computing, variation is also a significant concern in TD computing [16]. To relieve such an issue, special calibration scheme is introduced to calibrate the 20 × 20 DTW matrix, as shown in Fig. 14. A 2-b tunable delay cell is implemented in each unit cell to tune the output pulsewidth, compensating for process variations.

The DTW nodes are calibrated through each diagonal path following a center-to-side order, as depicted in Fig. 14(a). On each diagonal path, the nodes are calibrated from bottom-right to top-left, as shown in Fig. 14(b) and (c), and the calibration is performed node by node. The basic idea is to construct special input sets which make the warping path (as marked in green in Fig. 1) to lie into the particular diagonal path and to be calibrated. By specially manipulating the input data pattern, each node is further calibrated in that particular diagonal path one by one. Once the diagonal path is properly

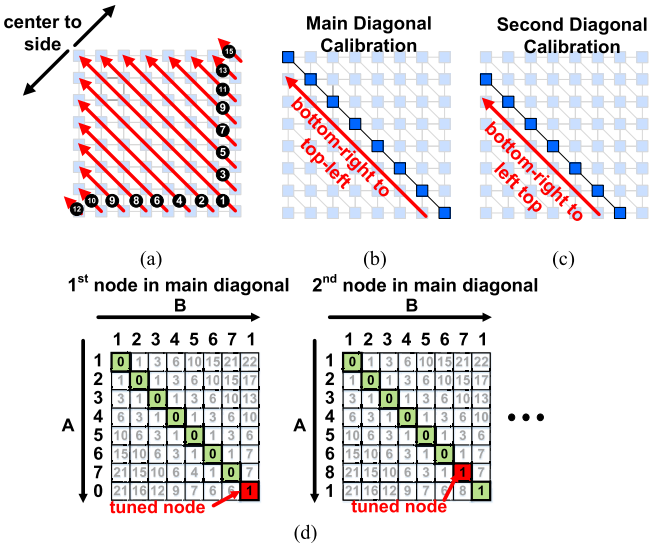

Fig. 14. Calibration scheme of the 20 × 20 DTW matrix. (a) Calibration order through different diagonals. (b) Calibration order of each DTW node on the main diagonal. (c) Calibration order of each DTW node on the second diagonal. (d) Example of special input sets to enable the calibration of different nodes on the main diagonal.

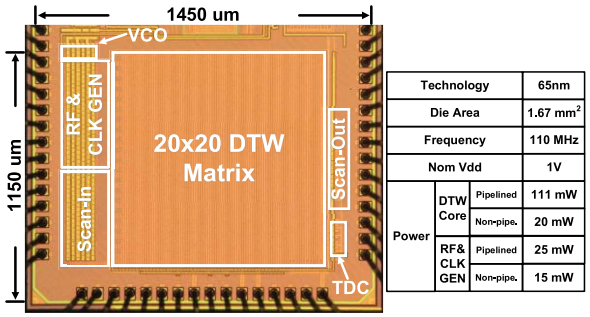

Fig. 15. Die photograph and chip specification.

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 Section V.

V. Experimental Results

A. Test Chip Setup

A test chip of the proposed DTW accelerator engine was implemented in a 65-nm CMOS process with die photo and specification table shown in Fig. 15. The chip is running at 110 MHz with a nominal supply voltage of 1 V. Two sets of TDCs, based on Vernier delay chains, are placed at the right and bottom sides to decode TD signals at the boundaries. A single-bit resolution of 40 ps 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.

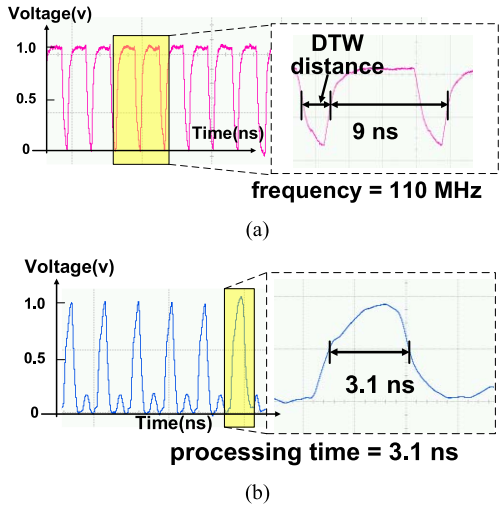

Fig. 16. Measured waveform of (a) pipelined mode and (b) non-pipelined mode.

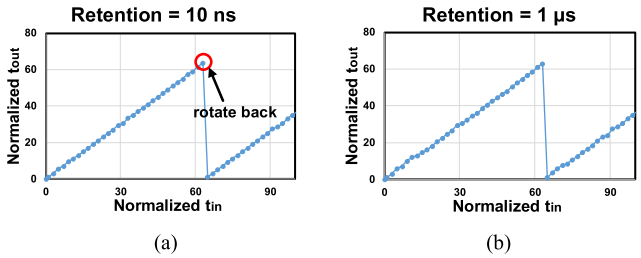

Fig. 17. Linearity measurement of TFF at nominal 1.0 V with (a) retention time of 10 ns and (b) retention time of 1 μs.

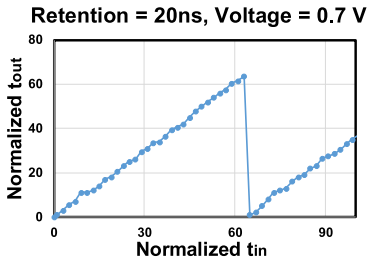

Fig. 18. Linearity measurement of TFF in low-voltage case (0.7 V) with retention time is 20 ns.

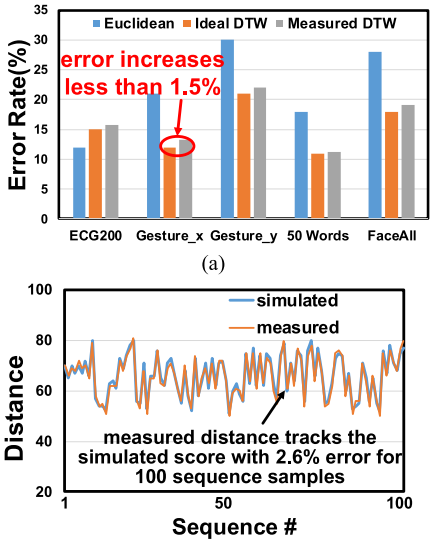

Fig. 19. Measurement results of different applications. (a) DTW classification error rate of UCR archive (pipelined mode). (b) Simulated versus measured DNA alignment distance (non-pipe mode).

B. Measurement Results

Fig. 16(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 TD. Fig. 16(b) shows 3.1-ns processing time in DNA-sequencing non-pipelined mode.

The linearity of the TFF is key to the accuracy of the DTW computation. Also, the retention capability of TFF for TD signals is important since the degradation of TD 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 Fig. 17(a) and (b), the TFF is verified to retain data for over 1 μs at a supply voltage of 1 V, 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 7 ns. The linearity of TFF is also verified at a lower supply voltage of 0.7 V. As shown in Fig. 18, the linearity loss is 1.5 LSBs which results to classification error increase (2%) in the low-voltage operation.

Fig. 19(a) shows measurement results on classification error using the fabricated DTW chip. University of California, Riverside (UCR) TSC databases were used with five databases from four typical applications including ECG signal classification, gesture recognition, words recognition, and face detection [24]. The measured error rate for classification by the DTW engine is only 1.5% higher than ideal DTW operation (FP results in software). The increased error rate is mainly due to quantization loss (contributing about 0.5%) and process variation effect (contributing about 1%).

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 CMP between ideal DTW operation and measurement results. As shown in Fig. 19(b), the measured distance closely tracks the ideal results, having an error within 2.6%.

As shown in Fig. 20, in order to test the robustness of the chip, the chip was verified at different supply voltages in pipelined mode down to 0.7 V, with a 2.3% increase in error rate compared with ideal DTW operation on the UCR database.

Fig. 21 shows the chip calibration results before and after calibration operations. In this experiment, a 20 $\times$ 20 TSC task was conducted with 4-bit inputs. The scale for the figure is the measurement distance error in the unit of LSB. The final ABS computation different is 1 LSB. After calibrating the 20 $\times$ 20 DTW matrix, the MAX DTW distance computation error drops from 5 to 1.5 LSBs. Fig. 22 shows the CMP with

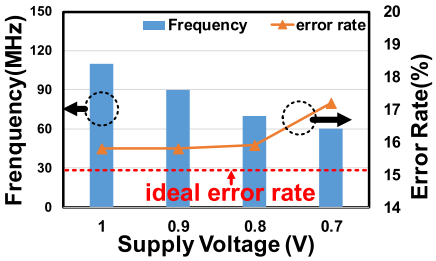

Fig. 20. Chip operating frequency and error rate measurement under different supply voltages.

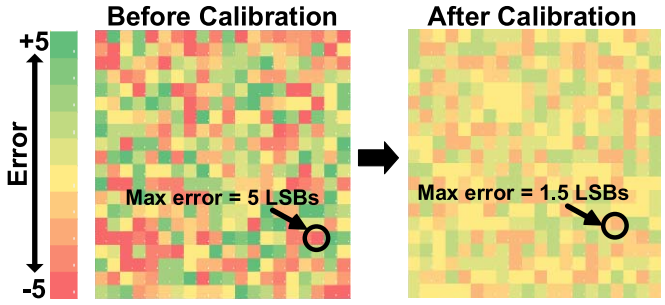

Fig. 21. DTW node error measurement before and after calibration.

prior work. A throughput of 140 GCUPS for DNA sequencing is achieved with 9× improvement over previous work [21]. 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 [21]. 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 TD circuit technique in special operations, e.g., CMP, MAX, and MIN. Overall, 1.5×–50× improvement of energy per GCUP is realized in this work compared with prior chip implementations. Over 20× and 18× improvements on inference per second per mm^2 and inference per second per watt are achieved, respectively.

In order to form an apple-to-apple CMP, the technology scaling effect is also taken into consideration, compared with [21], 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 AMS design), and our technology advances about three 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 with [27], we further scale down the process impact by 0.7 (from 90 to 65 nm) and the bit precision impact (from 32 to 4 bit), and this leads to a throughput improvement of about 11× for the ASIC implementation of [27].

In ADD, the use of TFFs enables the first pipelined architecture for TD design which not only improves the throughput but also increases the hardware utilization. Compared with 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.

	[25]	[26]	[27]	[21]	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)	9.5x10^4	2x10^5	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)	3.2x10^4	1x10^3	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.

Fig. 22. CMP table with prior work. *In DNA application, single-bit non-pipeline mode with input length of 20 is utilized for fair CMP with prior work. ** Technology scaling is considered and is further discussed in the above paragraph.

In ADD to the fabricated prior test chips, Li *et al.* [28] 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. Sundaresan *et al.* [29] introduced parallel DTW algorithm. Xu *et al.* [30] proposed a memristor-based DTW accelerator design. Compared with the digital implementations in [27] and [28], our design improved the throughput by over 4×. Compared with the analog and mixed-signal design in [29], we realized a throughput improvement over 200×.

VI. Conclusion

In this work, a general-purpose DTW engine using TD computing is designed for TSC. A special TD storage cell, namely, TFF, 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 utilizing the TFF circuits. A 65-nm CMOS test chip was fabricated and tested. The measurement shows a throughput improvement of more than

9× compared with prior works. In ADD, 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.

REFERENCES

[1] Duncan Stewart. *Edge AI Chips Come Into Their Own*. Accessed: Dec. 9, 2019. [Online]. Available: https://www2.deloitte.com/us/en/insights/industry/technology/technology-media-and-telecom-predictions/2020/ai-chips.html

[2] R. Hameed *et al.*, "Understanding sources of inefficiency in general-purpose chips," in *Proc. 37th Annu. Int. Symp. Comput. Archit. (ISCA)*, 2010, pp. 37–47.

[3] A. Madhavan, T. Sherwood, and D. Strukov, "Race logic: A hardware acceleration for dynamic programming algorithms," in *Proc. ACM/IEEE 41st Int. Symp. Comput. Archit. (ISCA)*, Jun. 2014, pp. 517–528.

[4] Y.-S. Jeong, M. K. Jeong, and O. A. Omitaomu, "Weighted dynamic time warping for time series classification," *Pattern Recognit.*, vol. 44, no. 9, pp. 2231–2240, Sep. 2011.

[5] T. Yan, "Cost-effective integrated RF power transistor in 0.18-μmCMOS technology," *IEEE Electron Device Lett.*, vol. 27, no. 10, pp. 856–858, Oct. 2006.

[6] N. P. Jouppi *et al.*, "In-datacenter performance analysis of a tensor processing unit," in *Proc. ACM/IEEE 44th Annu. Int. Symp. Comput. Archit. (ISCA)*, Jun. 2017, pp. 1–12.

[7] E. Beck, W. Zhou, R. Schlüter, and H. Ney, "LSTM language models for LVCSR in first-pass decoding and lattice-rescoring," 2019, *arXiv:1907.01030*. [Online]. Available: http://arxiv.org/abs/1907.01030

[8] James Vincent. *Tesla's New AI Chip Isn't a Silver Bullet for Self-Driving Cars*. Accessed: Apr. 24, 2019. [Online]. Available: https://www.theverge.com/2019/4/24/18514308/tesla-full-self-driving-computer-chip-autonomy-day-specs

[9] F. N. Buhler, P. Brown, J. Li, T. Chen, 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 *Proc. Symp. VLSI Circuits*, Jun. 2017, pp. C30–C31.

[10] 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 *Proc. IEEE Asian Solid-State Circuits Conf. (A-SSCC)*, Nov. 2016, pp. 21–24.

[11] S. Yu *et al.*, "Binary neural network with 16 Mb RRAM macro chip for classification and online training," in *IEDM Tech. Dig.*, Dec. 2016, pp. 16–22.

[12] R. Cutler and L. S. Davis, "Robust real-time periodic motion detection, analysis, and applications," *IEEE Trans. Pattern Anal. Mach. Intell.*, vol. 22, no. 8, pp. 781–796, Aug. 2000.

[13] Z. Chen and J. Gu, "A scalable pipelined time-domain DTW engine for time-series classification using multibit time flip-flops with 140giga-cell-updates/s throughput," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2019, pp. 324–326.

[14] N. Cao *et al.*, "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 ISSCC Dig. Tech. Papers*, Feb. 2019, pp. 222–224.

[15] Z. Chen, H. Zhou, and J. Gu, "Digital compatible synthesis, placement and implementation of mixed-signal time-domain computing," in *Proc. 56th Annu. Design Autom. Conf.*, Jun. 2019, pp. 1–6.

[16] Z. Chen and J. Gu, "Analysis and design of energy efficient time domain signal processing," in *Proc. Int. Symp. Low Power Electron. Design (ISLPED)*, 2016, pp. 100–105.

[17] D. Miyashita *et al.*, "An LDPC decoder with time-domain analog and digital mixed-signal processing," *IEEE J. Solid-State Circuits*, vol. 49, no. 1, pp. 73–83, Jan. 2014.

[18] A. Sayal *et al.*, "All-digital time-domain CNN engine using bidirectional memory delay lines for energy-efficient edge computing," in *IEEE ISSCC Dig. Tech. Papers*, Feb. 2019, pp. 228–230.

[19] Z. Chen and J. Gu, "An image recognition processor with time-domain accelerators using efficient time encoding and non-linear logic operation," in *Proc. IEEE Asian Solid-State Circuits Conf. (A-SSCC)*, Nov. 2018, pp. 257–260.

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

[21] A. Madhavan, T. Sherwood, and D. Strukov, "A 4-mm^2 180-nm-CMOS 15-Giga-cell-updates-per-second DNA sequence alignment engine based on asynchronous race conditions," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Apr./May 2017, p. 1–4.

[22] H. Ding *et al.*, "Querying and mining of time series data: Experimental comparison of representations and distance measures," in *Proc. VLDB*, 2008, pp. 1287–1300.

[23] H. I. Fawaz *et al.*, "Deep learning for time series classification: A review," 2018, *arXiv:1809.04356*. [Online]. Available: https://arxiv.org/abs/1809.04356

[24] *UCR Archive*. Accessed: Jun. 2015. [Online]. Available: http://www.cs.ucr.edu/~eamonn/time_series_data

[25] M. Farrar, "Striped smith-waterman speeds database searches six times over other SIMD implementations," *Bioinformatics*, vol. 23, no. 2, pp. 156–161, Jan. 2007.

[26] Y. Liu, A. Wirawan, and B. Schmidt, "CUDASW++ 3.0: Accelerating smith-waterman protein database search by coupling CPU and GPU SIMD instructions," *BMC Bioinf.*, vol. 14, no. 1, Dec. 2013.

[27] N. Neves, N. Sebastiao, D. Matos, P. Tomas, P. Flores, and N. Roma, "Multicore SIMD ASIP for next-generation sequencing and alignment biochip platforms," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 23, no. 7, pp. 1287–1300, Jul. 2015.

[28] K. F. Li *et al.*, "Dynamic time warping in hardware," in *Proc. 14th Int. Conf. Inf. Integr. Web-Based Appl. Services (IIWAS)*. Victoria, BC, Canada: Univ. Victoria, Dec. 2012, pp. 132–137.

[29] V. K. Sundaresan, S. Nichani, N. Ranganathan, and R. Sankar, "A VLSI hardware accelerator for dynamic time warping," in *Proc. 11th IAPR Int. Conf. Pattern Recognit. Conf. D: Architectures Vis. Pattern Recognit.*, vol. 4, 1992, pp. 27–30.

[30] X. Xu *et al.*, "Accelerating dynamic time warping with memristor-based customized fabrics," *IEEE Trans. Comput.-Aided Design Integr. Circuits Syst.*, vol. 37, no. 4, pp. 729–741, Apr. 2018.

[31] A. Y. Hannun *et al.*, "Cardiologist-level arrhythmia detection and classification in ambulatory electrocardiograms using a deep neural network," *Nature Med.*, vol. 25, no. 1, pp. 65–69, Jan. 2019.

[32] K. Kim, W. Yu, and S. Cho, "A 9 bit, 1.12 ps resolution 2.5 b/Stage pipelined Time-to-Digital converter in 65 nm CMOS using time-register," *IEEE J. Solid-State Circuits*, vol. 49, no. 4, pp. 1007–1016, Apr. 2014.

[33] A. Madhavan and M. D. Stiles, "Storing and retrieving wavefronts with resistive temporal memory," 2020, *arXiv:2003.09355*. [Online]. Available: http://arxiv.org/abs/2003.09355

Zhengyu Chen (Member, IEEE) received the B.S. degree in electrical engineering from Southeast University, Nanjing, China, in 2013, and the M.S. degree in computer engineering from Cornell University, Ithaca, NY, USA, in 2015. He is currently pursuing the Ph.D. degree in computer engineering with Northwestern University, Chicago, IL, USA.

He is currently an Aspiring Researcher with Northwestern University, Evanston, IL, doing research in the area of ultra-low-power design/algorithm for VLSI, mixed-signal ICs, and emerging device. He is currently focusing on the low-power algorithm design like time-domain signal processing and accelerator design of machine learning algorithms.

Jie Gu (Senior Member, IEEE) received the B.S. degree from Tsinghua University, Beijing, China, in 2001, the M.S. degree from Texas A&M University, College Station, TX, USA, in 2003, and the Ph.D. degree from the University of Minnesota, Minneapolis, MN, USA, in 2008.

He was an IC Design Engineer with Texas Instruments, Dallas, TX, from 2008 to 2010, focusing on ultralow-voltage mobile processor design and integrated power management techniques. He was a Senior Staff Engineer with Maxlinear, Inc., Dallas, from 2011 to 2014, focusing on low-power mixed-signal broadband system-on-chip (SoC) design. He is currently an Assistant Professor with Northwestern University, Evanston, IL, USA. His current research interests include emerging mixed-signal computing circuit, the design of machine learning capable edge devices, and ultra-dynamic clock and power management for microprocessor and accelerators.

Dr. Gu was a recipient of the NSF CAREER Award. He has served as a Program Committee and Conference Co-Chair for numerous low-power design conference and journals, such as ISPLED, DAC, ICCAD, and ICCD.