

Received January 31, 2018, accepted March 5, 2018, date of publication March 12, 2018, date of current version April 25, 2018.

Digital Object Identifier 10.1109/ACCESS.2018.2814625

Two-Dimensional Multiply-Accumulator for Classification of Neural Signals

YU-CHIEH CHEN^{1,2}, (Member, IEEE), HSIN-CHI CHANG¹,
AND HSIN CHEN¹, (Member, IEEE)

¹Department of Electrical Engineering, National Tsing Hua University, Hsinchu 300, Taiwan

²Instrument Technology Research Center, NARL, Hsinchu 300, Taiwan

Corresponding author: Hsin Chen (hchen@ee.nthu.edu.tw)

This work was supported in part by the Chip Implementation Center, The National Science Council, Taiwan, and in part by the Ministry of Science and Technology, Taiwan, under Grant 106-2622-8-007-014-TA.

ABSTRACT Automatic spike detection and classification have been used for a neuroelectronic interface to reduce data amount or even to interact with neurons in a closed loop. While conventional neuroelectronic interfaces employ voltage-mode circuits to amplify neural signals and convert the signals into binary data, the dynamic range and signal-to-noise ratio of these circuits are directly limited by the supply voltage. To release this constraint, this paper proposes an analog-to-time converter (ATC), which uses positive feedback to convert analog neural signals into a sequence of pulse trains. Custom-designed digital circuits, including two types of time-to-digital converters (TDCs) and a 2-D multiply-accumulator (2-D-MAC), are further proposed for processing such time-mode signals. The ATC is implemented with the standard 0.35- μm CMOS technology and proved able to convert analog voltages into pulse-width-modulated signals with a resolution of 6 bits. The TDCs and 2-D-MAC are realized in FPGA and compared to the standard digital IPs. The comparison indicates that the TDC based on dual counters minimizes area consumption and the other based on delayed clocks minimizes power consumption. The 2-D-MAC further facilitates parallel computation of partial products and allows data to be classified without summing up all partial products. Finally, the application of the proposed time-mode system is demonstrated as classifying neuronal spikes.

INDEX TERMS Analog-to-time converter (ATC), multiply-accumulate (MAC) operation, time-mode signal processing.

I. INTRODUCTION

The microelectronics technology has been exploited to interface with neurons at a high spatiotemporal resolution [1]. This is helpful not only for advancing neuroscience research but also for developing novel neural prostheses. At a neuroelectronic interface, the ability to detect and classify spikes automatically is essential for interacting with neurons in a closed-loop manner. For neural prostheses, automatic spike detection and sorting further help to reduce the power and bandwidth of wireless data transmission, as well as to facilitate delivering bio-feedbacks in real time. Therefore, many embedded systems able to detect and sort spikes automatically have been proposed [2], [3]. While spike detection is achievable by either analog or digital circuits, spike sorting mainly relies on digital signal processing and thus necessitates converting neural signals into digital data. Conventional neuroelectronic interfaces usually employ a low-noise amplifier to amplify miniature neural signals (ranging from

several microvolts to several millivolts [4]). The amplified signals are then converted by a rail-to-rail, analog-to-digital converter (ADC). This architecture faces the following drawbacks. First, a large amplification gain is required to exploit the full resolution of the ADC, but large gain unavoidably consumes extra power or extra chip area (for the input capacitance of a capacitive amplifier). Secondly, as the supply voltage reduces with the technology to decrease power consumption, the achievable signal-to-noise ratio (resolution, dynamic range) of a voltage-mode ADC is reduced simultaneously.

Under the above concerns, this paper investigates the feasibility of using time-mode computation with custom-designed digital representation to release the constraints. Fig. 1 shows the proposed architecture, mainly consisting of an analog-to-time converter, a time-to-digital converter (TDC), and a two-dimensional multiply-accumulate (2D-MAC) operator. Neural signals are amplified to only several tens of millivolts

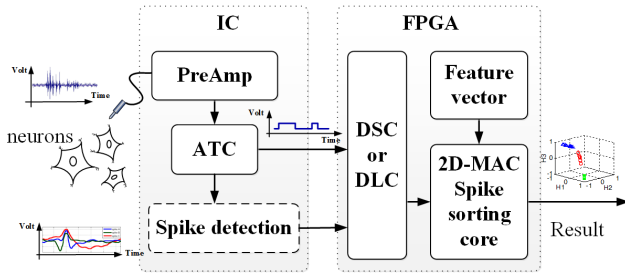


FIGURE 1. The architecture of the proposed time-mode system for neural recording and spike sorting.

by the preamplifier. Subsequently, the amplified signal is converted into a “time-mode signal” by the analog-to-time converter (ATC).

The ATC utilizes positive feedback to convert each analog sample into a digital pulse efficiently, and the pulse width is proportional to the analog level. The TDC then converts each pulse into two binary data, representing one coarse and one fine estimates of the pulse width, by either the dual-scale counter (DSC) or the delay-line counter (DLC). Finally, as many data classification methods (e.g. PCA, LDA) are based on computing the inner product between data and feature vectors, the 2D-MAC is designed to compute the inner product of the TDC output and feature vectors to achieve spike sorting.

Compared to conventional architectures, the ATC avoids the need for a high amplification gain and uses positive feedback to translate analog values into pulse widths efficiently. The pulse-width representation further allows the dynamic range to be independent of the supply voltage. This feature is particularly useful as the technology development continues to shrink transistor sizes. While the supply voltage has to reduce the transistor size, the cutoff frequency of transistors increases and favors time-mode data conversion [5], [6]. Nevertheless, the pulse-width-coded data need to be converted into binary data to facilitate signal processing (e.g., spike sorting). Although the most straightforward method is using a counter to measure the pulse duration, the counter will need a high-frequency clock and consume remarkable power to achieve high resolution. Also, the pulse-width and the counting clock have to be synchronized before the counter starts to count.

To mitigate these drawbacks, two methods are proposed and compared in this paper. The first is using a dual-scale counter (DSC) with two different clock frequencies [7]. A low-frequency clock is first used to obtain a coarse count (measurement) of the pulse duration. Only the remaining part of the pulse width is measured by a high-frequency clock. The second method is based on the Vernier delay line proposed in [8] and [9]. The high-frequency clock is replaced by a set of low-frequency clocks generated from interpolating delay lines. The delayed clock edges are used as the edges of a high-frequency clock. However, a wide dynamic range would require a large number of delay lines, which consume

not only extra power and area but also exhibit significant variations [10]. The proposed delay-line counter releases this constraint by digitalizing most of the pulse duration by a low-frequency clock and using delay lines to measure only the remaining duration. This helps to achieve both a high resolution and a wide dynamic range without numerous delay lines. The conversion efficiency of two proposed TDCs is compared and discussed in this paper. Finally, as multiply-accumulate (MAC) operators consume most power in digital signal processing, the proposed two-dimensional (2D)-MAC operator employs two-dimensional, parallel processing to reduce both power consumption and time complexity. The performance of the 2D-MAC will be compared to the conventional MAC.

To facilitate the comparison between the proposed digital circuits and those in the standard digital library, a mixed-signal system consisted of a custom-designed chip and a field-programmable-gate-array (FPGA) is set up in this study. As indicated by Fig. 1, the ATC is realized by the 0.35 μm 2P4M CMOS technology provided by the Taiwan Semiconductor Manufacturing Company (TSMC), while the proposed digital circuits including the DSC, the DLC, and the 2D-MAC operator, are implemented in the FPGA (Altera MAX V-5M2210ZF256C4). The FPGA provides the flexibility of realizing and comparing different types of TDCs, as well as comparing the proposed 2D-MAC with conventional MAC regarding power and area consumption.

Following the introduction, Sec. II Sec.IV introduce the design of the ATC, TDCs, and 2D-MAC, respectively. The measurement of these circuits is then presented and discussed in Sec. V. The full system’s ability to classify neural spikes is further demonstrated in Sec. VI. Finally, Sec. VII concludes the findings and future works.

II. THE ANALOG-TO-TIME CONVERTER

Fig. 2 shows the architecture of the proposed ATC, whose circuit design has been detailed in [11]. V_{in} represents the pre-amplified neural signal. The transistor M_{sub} operating

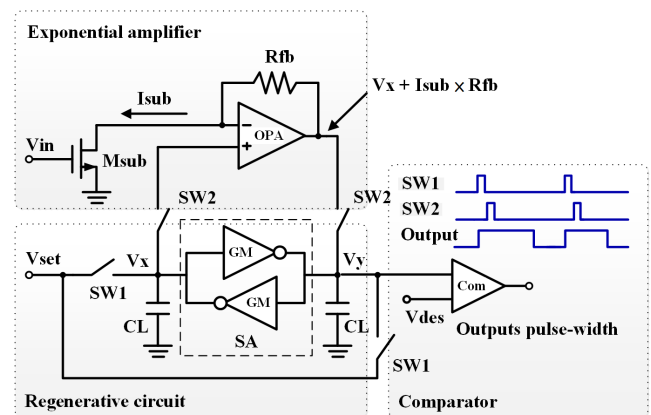


FIGURE 2. The circuit architecture and the timing diagram of the analog-to-time converter.

in the subthreshold region converts V_{in} into a current I_{sub} exponentially proportional to V_{in} . Subsequently, the operational amplifier (OPA) together with R_{fb} transform I_{sub} into an output voltage equaling to $(V_x + I_{sub} \times R_{fb})$, where V_x is the voltage at the positive input of the OPA as the switches SW2 are on. The output of the OPA is then sampled by the regenerative circuit, consisting of two transconductances (G_m) amplifiers and two capacitors (CL) connected in a positive-feedback loop. In our design, each transconductance amplifier is simply realized by an inverter, so that the regenerative circuit functions as the sense amplifier in the conventional DRAM. The switches SW1 and SW2 are controlled by two non-overlapping clocks during the analog-to-time conversion. As the switches SW1 are turned on, both V_x and V_y are reset to V_{set} . Afterward, the switches SW1 are turned off and the switches SW2 are turned on.

V_y becomes $(V_x + I_{sub} \times R_{fb})$. Let $V_{xy} = V_x - V_y$. As both SW1 and SW2 are turned off, the initial voltage $V_{xy}(0)$ is given as (1), and the dynamics of $V_{xy}(t)$ is governed by (2).

$$V_{xy}(0) = I_{sub} \cdot R_{fb} = R_{fb} \cdot I_0 \cdot e^{\kappa V_{in}/U_T} \quad (1)$$

$$V_{xy}(t) = \frac{CL}{G_m} \cdot \frac{dV_{xy}(t)}{dt} \quad (2)$$

I_0 is a process-dependent parameter and is also proportional to the transistor size W/L . κ is the coupling coefficient for subthreshold operation, and $U_T = kT/q$ is the thermal voltage. The rising edge of an output pulse is triggered as soon as both SW1 are turned on. The comparators then compares V_y with V_{des} . As soon as $V_y = V_{des}$, the falling edge of the pulse is triggered. Let $\tau_s = CL/G_m$. The time t_s required for $V_y = V_{des}$ can be derived by solving (2) as

$$\int_0^{t_s} \frac{1}{\tau_s} \cdot dt = \int_0^{t_s} \frac{1}{V_{xy}(t)} \cdot dV_{xy}(t)$$

$$\frac{t_s}{\tau_s} = \ln |V_{xy}(t_s)| - \ln |V_{xy}(0)|$$

$$t_s = \tau_s \cdot \ln \left(\frac{V_{xy}(t_s)}{V_{xy}(0)} \right) \quad (3)$$

Substituting the initial condition in (1) for $V_{xy}(0)$ then gives

$$t_s = \tau_s \cdot \ln \frac{V_{xy}(t_s)}{I_0 \cdot R_{fb}} - \tau_s \cdot \frac{\kappa V_{in}}{V_T} \quad (4)$$

As the first term is a constant, t_s is linearly proportional to V_{in} .

III. THE TIME-TO-DIGITAL CONVERTERS

Two methods are proposed to convert the pulse signal into binary data for digital signal processing. The first is based on the dual-scale counter (DSC), and the second is based on the delay-line counter (DLC). The DSC requires less chip area, while the DLC consumes lower power for achieving a high resolution and a wide dynamic range. The following subsections detail their design concepts and FPGA implementations.

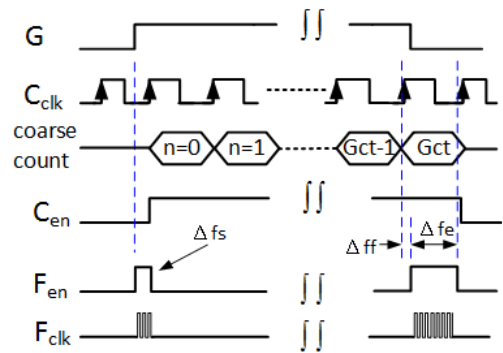


FIGURE 3. The timing diagram of the signals during the time-to-digital conversion based on the dual-scale counter.

A. THE DUAL-SCALE COUNTER

Fig. 3 illustrates the operation of the proposed DSC. G is the pulse signal to be digitalized. G is first re-sampled by the coarse-count clock (C_{clk}) to generate the C_{en} signal, and the duration of C_{en} is measured by the coarse-count clock as G_{ct} counts. Taking the exclusive-OR between G and C_{en} then gives the F_{en} signal, which indicates the remaining duration of G at the beginning, Δfs , and the over-estimated duration at the end, Δfe . Therefore, F_{en} triggers the fine-count clock F_{clk} to measure the duration of Δfs and Δfe as G_{fs} and G_{fe} counts, respectively. Let the frequency of F_{clk} be k times higher than that of C_{clk} . The total duration of G is written as

$$G = (G_{fs} - G_{fe}) + G_{ct} \times k. \quad (5)$$

B. THE DELAY-LINE COUNTER

Fig. 4 illustrates the operation of the proposed DLC, which replaces the fine-count clock in DSC by the interpolated signals, $D1 \sim D8$. The interpolated signals are generated

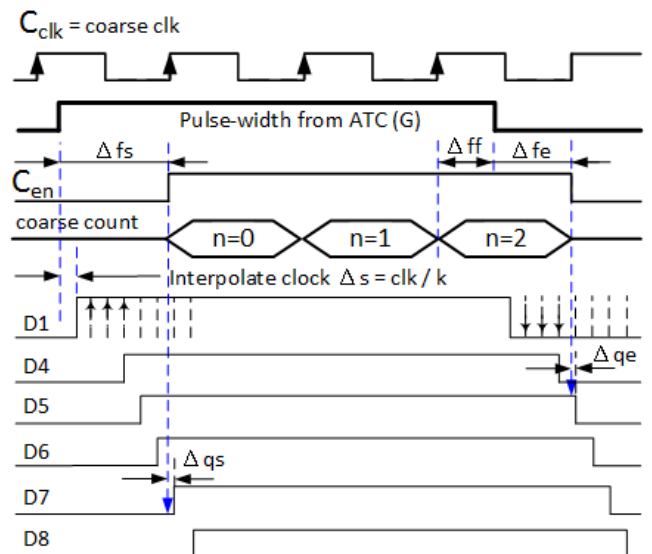


FIGURE 4. The timing diagram of the signals during the time-to-digital conversion based on the delay-line counter.

by simply passing the pulse signal G through a set of delay lines, moreover, the delay between consecutive signals is designed to be one- k th cycle of the coarse-count clock. As the pulse signal G arrives, its rising edge still triggers the coarse counter to count its duration, and C_{en} is generated by re-sampling G . At the same time, the rising edge of G also triggers the counting of the number of rising edges of the interpolated signals till the onset of C_{en} . This counting result, G_{fs} , indicates the duration of Δfs . On the other hand, the falling edge of G triggers the counting of the number of falling edges of interpolated signal till the falling edge of C_{en} . This counting result, G_{fe} , indicates the duration of Δfe . Therefore, the total duration of G is still given as (5). To simplify hardware implementation, the negative term can be removed by rewriting (5) as

$$\begin{aligned} G &= [G_{fs} + (k - G_{fe})] + (G_{ct} - 1) \times k \\ &= [G_{fs} + G_{ff}] + (G_{ct} - 1) \times k \\ &= G_f + G_c \times k \end{aligned} \quad (6)$$

The minimum delay achievable in our study is the delay for a datum passing through a D-FlipFlop and the routing path to the next stage. To estimate the minimum delay, the delays of 100 routing paths (randomly generated) involving D-FlipFlops in our FPGA device (Altera MAX V-5M2210ZF256C4) are measured. The statistics indicate that, although the routing paths vary significantly, approximately 45% of the delays equal 2.5 ns and more than 75% of the delays are smaller than 3 ns. The minimum delay achievable is thus estimated to be 3 ns, much smaller than the duration of the pulse signal ($> 4\mu s$ in Fig. 12). Therefore, the errors Δqs and Δqe in Fig. 4 are ignored in this study.

C. THE FPGA IMPLEMENTAION

This section presents the DSC and DLC hardware architectures, which was implemented in an FPGA device.

Fig. 5 shows the circuit architecture of the DSC in FPGA. The counter #3 is the coarse counter, and its output is shifted upwards by four bits because of $k = 8$ in this implementation example. The other two counters measure the duration of Δfs and Δfe in Fig. 3. The enabling signal of up counter #1 is obtained by $F_{en} \wedge \overline{C_{en}}$, so that the output of counter #1 corresponds to G_{fs} in (5). Similarly, the enabling signal of

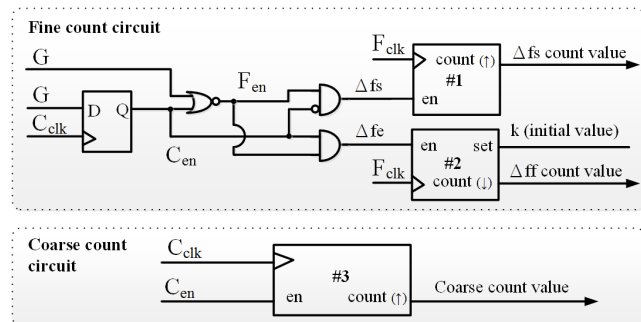


FIGURE 5. The circuit architecture of the proposed DSC in FPGA.

down counter #2 is obtained by $F_{en} \wedge C_{en}$, which down counts from k and outputs the result which corresponds to G_{ff} in (5). Both fine counts are added to the shifted coarse count as the final result. Nevertheless, to simplify the circuit and to reduce the power consumption, the fine-count and coarse-count results are processed directly by the 2D-MAC described in next section.

Fig. 6 shows the circuit architecture of the DLC in FPGA. The delay chain contains two series of D-FlipFlops. The top series delay the rising edge of G , while the bottom series delay the falling edge of G . The delayed signals are transmitted to the D-flipflops in the edge-sensing circuit. Subsequently, the top row of D-flipflops latches the states of the delayed signals at the rising edge of C_{en} , whereas the bottom row latches the states at the falling edge of C_{en} . The latched states are then sent to the decoder to generate the G_{fs} and G_{ff} in (6). Although the total duration of G can be derived from G_{fs} , G_{ff} and G_{ct} , the DLC simply outputs $G_f = G_{fs} + G_{ff}$ and $G_c = G_{ct} - 1$ for the 2D-MAC operator to classify data directly.

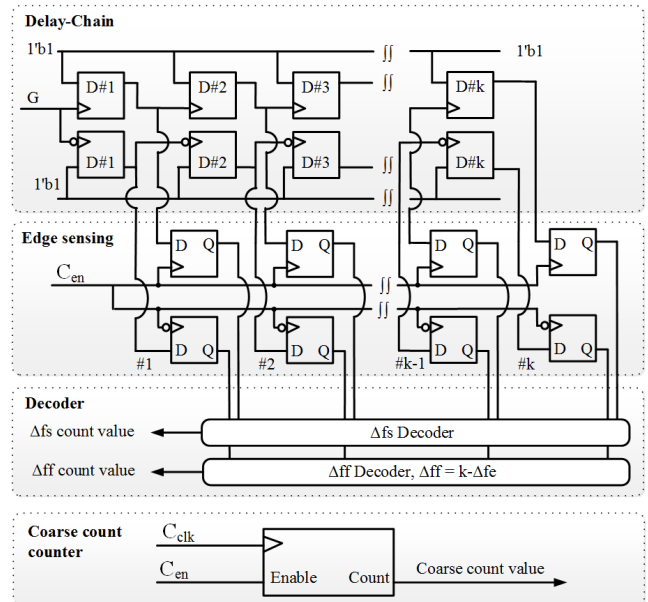


FIGURE 6. The circuit architecture of the proposed DLC in FPGA.

IV. THE TWO-DIMENSIONAL MULTIPLY-ACCUMULATOR

A. DESIGN CONCEPT

Let G and H represent the binary-coded multiplicand and multiplier, respectively. Conventional multiplication process needs to compute the partial products and accumulate them to obtain the final product value, as illustrated by the top-left panel in Fig. 7.

Instead of calculating the final product directly, we propose keeping the partial products in a two-dimensional (2D) register array, as shown by the bottom-left panel in Fig. 7. In this example, both G and H have four bits plus one sign bit. The 2D register array R thus contains 4×5 bits. Each row i

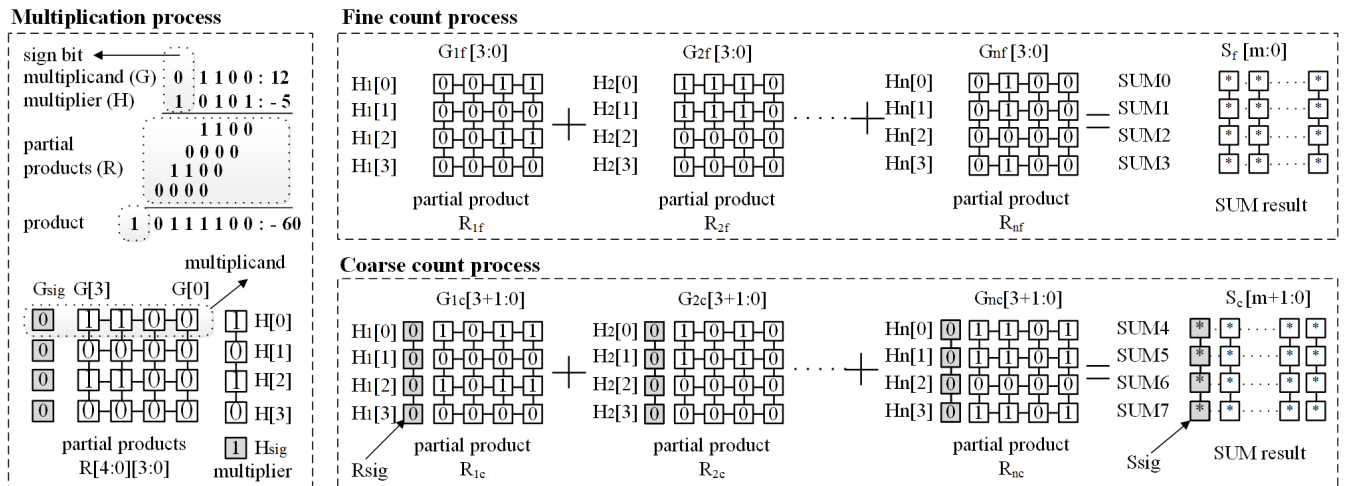


FIGURE 7. The operational concept of the proposed 2D-MAC. (Left-top) the conventional MAC computation; (Left-bottom) the proposed 2D-MAC computation. (Right) The accumulation process for computing the inner product with the 2D-MAC. The n sets of the partial-product registers store the multiplication between G_l and H_l for $l = 1 \sim n$. $H_1[3+1:0] = 0_0101$, $H_2[3+1:0] = 0_0011$, ..., $H_n[3+1:0] = 0_1011$; $G_1[7+1:0] = \{G_{1c} = 0_1011, G_{1f} = 0011\}$, $G_2[7+1:0] = \{G_{2c} = 0_1010, G_{2f} = 1110\}$ and $G_n[7+1:0] = \{G_{nc} = 0_1101, G_{nf} = 0100\}$. Both H_l and G_l have additional one bit as a sign bit at the MSB. The accumulated result is stored in the summing register $SUM[m+1:0][7:0]$ with the $(m+1)$ -th column storing the sign bits S_{sig} .

stores the partial product of G and $H[i]$, and the MSB of each row stores the sign bit as $R_{sig} = G_{sig} \oplus H_{sig}$. Computing the partial product is extremely easy and hardware inexpensive. G is copied to the i -th row of R if $H[i] = 1$, while i -th row of R is set to zero if $H[i] = 0$.

The right panels of Fig. 7 further illustrates how to use the data in the 2D register to compute the inner product between two n -dimensional vectors, $\mathbf{G} = [G_1, G_2, \dots, G_n]$ and $\mathbf{H} = [H_1, H_2, \dots, H_n]$. In our experiment, \mathbf{G} represents the recorded neural spikes, and \mathbf{H} represents the feature vector for spike sorting. Let the proposed TDC convert each $G_l, l = 1 \sim n$ into one fine-count value G_{lf} and one coarse-count value G_{lc} . G_{lf} and G_{lc} are multiplied with H_l , and the partial products are stored in the registers R_{lf} and R_{lc} , respectively. In the example in Fig. 7, both G_{lf} and G_{lc} contains four bits, and the additional MSB of G_{lc} is G_{sig} . H_l is an four-bit data plus its sign bit. Let $G_{lf}[p:0]$ denote that G_{lf} contains $p+1$ bits and $G_{lc}[q:0]$ denote that G_{lc} contains $q+1$ bits, the additional MSB at G_{lc} is the sign bit ($p = q = 3$ in Fig. 7). The computation of partial products can be formulated as

$$\begin{aligned} R_{lf}[i] &= G_{lf}[p:0] \wedge H_l[i] \\ R_{lc}[i] &= G_{lc}[q:0] \wedge H_l[i] \end{aligned} \quad (7)$$

To accumulate the partial products, two summing register arrays, S_f and S_c , are employed to sum up the register values in R_{lf} and R_{lc} , respectively. The summing register also has a 2D structure. The i -th row of S_f stores the summation of the values in the i -th row of $R_{lf}, l = 1 \sim n$. The same relationship applies to S_c and $R_{lc}, l = 1 \sim n$. As G_{lf} contains $p+1$ bits, the maximum summation value for each row is $(2^{(p+1)} - 1) \cdot n$. Therefore, each row of S_f should contain m bits with

$$m = \log_2 \left((2^{(p+1)} - 1) \cdot n \right)$$

The proposed 2D-MAC has the following advantages over conventional MAC:

- 1) The accumulator of the 2D-MAC requires much fewer bits than that of the conventional MAC because the 2D-MAC only accumulates row data in the partial-product register while conventional MAC needs to accumulate all partial products.
- 2) In applications that entail computing inner products of vectors, the 2D-MAC can compute the multiplication of multiple elements simultaneously, as illustrated by Fig. 8. The simultaneous processing saves considerable time as the vector size is large.
- 3) For classification tasks, data can be classified by simply looking at specific elements in the summing registers of the 2D-MAC rather than deriving the final product value. This allows the power, area, and time to be saved from converting 2D data into 1D data. This feature will be detailed in Sec. VI-C.

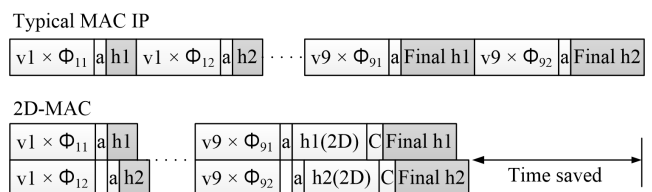


FIGURE 8. The computational flow of a typical MAC and a 2D-MAC when calculating (11). σ indicates the processing time required by the accumulator.

B. THE FPGA IMPLEMENTATION

Fig. 9 shows the circuit architecture for implementing the proposed 2D-MAC in FPGA. Although the inner product $\mathbf{G} \cdot \mathbf{H}$ involves the multiplication and accumulation of n elements. The architecture only requires two partial-product

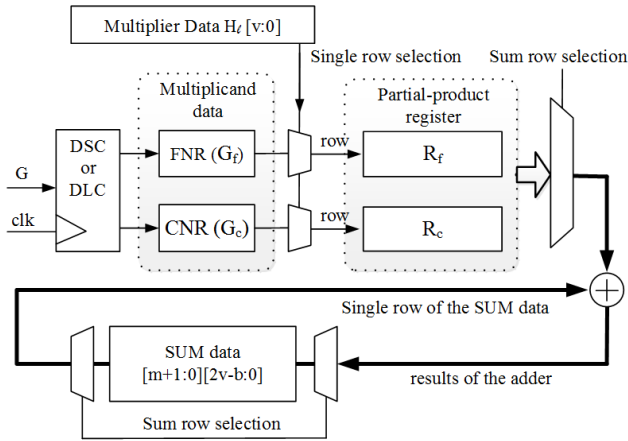


FIGURE 9. The circuit architecture of the proposed 2D-MAC in FPGA.

registers, R_f and R_c , for storing the l -th partial product under computation. As the pulse train of G arrives, each pulse is converted by the DSC or the DLC into a coarse-count value G_c and a fine-count value G_f , which are stored in the coarse-number register (CNR) and the fine-number register (FNR), respectively. To compute the partial product between G_l and H_l , the outputs of CNR and FNR are connected to the row-selection multiplexers. As $H_l[i] = 1$, the G_f and G_c values are stored into the i -th row of the partial-product registers R_f and R_c , respectively. Conversely, zeros are filled into the i -th row of the partial-product registers as $H_l[i] = 0$. The multiplexer behind the partial product register then selects one row at a time. The selected i -th row of R_f or R_c is added to the corresponding row of the summing register SUM .

V. MEASUREMENT RESULTS

A. THE ANALOG-TO-TIME CONVERTER

A prototype ATC was implemented with the $0.35\mu\text{m}$ 2P4M CMOS technology provided by the TSMC. The chip micro-photograph is shown in Fig. 10. The digital circuit generates the control signals depicted in Fig. 2 automatically. As $V_{in} = 480\text{ mV}$, the measured pulse output and corresponding control signals are shown in Fig. 11. The pulse width is $7.2\mu\text{s}$, and the magnified view of the falling edge indicates that the jitter is around 100 ns in maximum. By varying V_{in} from 340 mV to 540 mV with a step size of 10 mV , the corresponding pulse widths are measured and plotted in Fig. 12. The pulse width decreases from 10.7 to $4.3\mu\text{s}$ linearly with V_{in} for $400\text{ mV} \leq V_{in} \leq 530\text{ mV}$. As $V_{in} > 530\text{ mV}$, the initial voltage $V_{xy}(0)$ across the sense amplifier is large, so that the transconductance of the two inverters differ significantly and cause the nonlinearity. As $V_{in} < 390\text{ mV}$, the nonlinearity mainly comes from the dependence of κ in (1) on the gate-source voltage of M_{sub} , i.e. V_{in} . In subthreshold operation, κ approximates [12]

$$\frac{C_{ox}}{C_{ox} + C_{dep}} \quad (8)$$

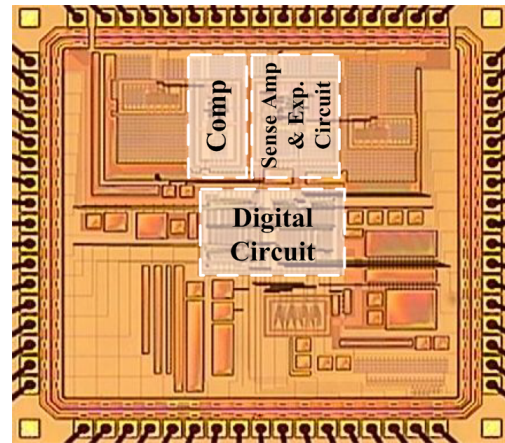


FIGURE 10. The photo of the proposed ATC implemented in the standard CMOS $0.35\mu\text{m}$ technology.

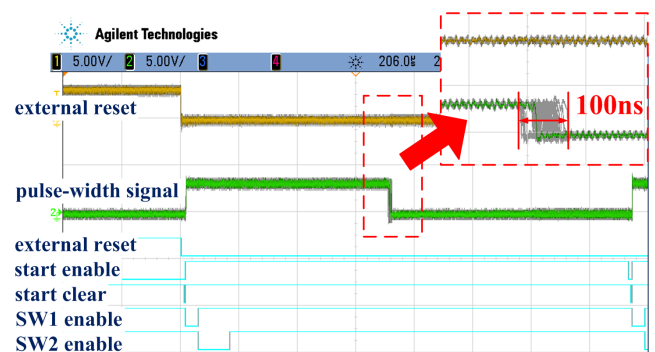


FIGURE 11. The measured ATC output and its control signals. As the trigger of the oscilloscope is set to the rising edge of the pulse-width signal (green). The top-right inset reveals the jitter at the falling edge.

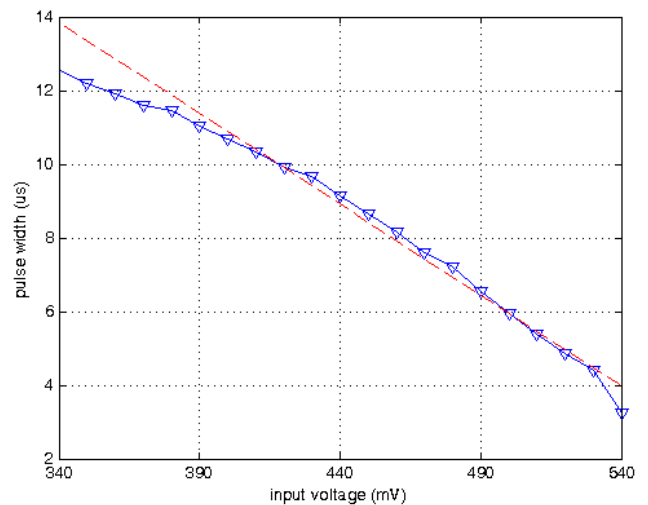


FIGURE 12. The measured output pulse width (triangular symbols) versus the input voltage V_{in} for the ATC. The regression curve (blue) and the ideal linear line (red) are also depicted.

where C_{ox} is gate-oxide capacitance, and C_{dep} the depletion capacitance of the body, which is nonlinearly-dependent on the gate-source voltage. As V_{in} increases from 340 mV to

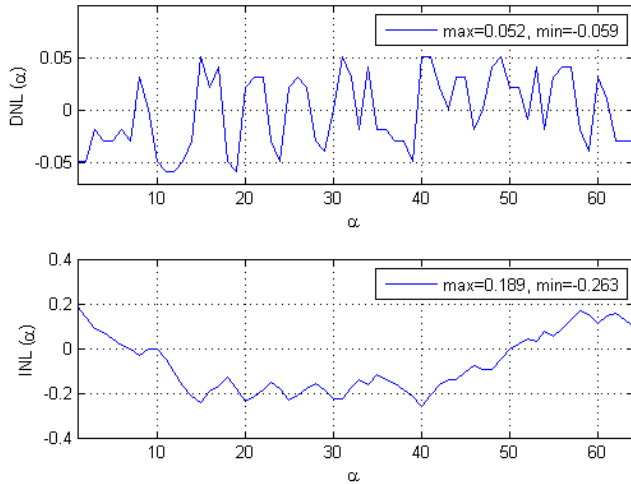


FIGURE 13. The effective differential non-linearity and integral non-linearity of the ATC with $V_{in} = \alpha \cdot 2mV + 400mV$.

390 mV, C_{dep} decreases and causes κ to increase. Therefore, according to (10), the decrement of the pulse width increases with V_{in} in this region. After V_{in} is greater than 400 mV, inversion charges grow more significantly so that the depth of depletion region and C_{dep} becomes nearly independent of V_{in} . Furthermore, the time t_s required for $V_y = V_{des}$ can be described from (9) to (10). The curve in Fig. 12 induced the logarithmic term $\ln(V_{des})$ in (10) which affects the nonlinearity. To solve the logarithmic issue while apply to wide dynamic range applications, we recommend that the pulse width time t_s defines as when $V_{xy} = V_{des}$.

$$\begin{aligned} V_x(t) &= \tau_s(V_{set} + I_{sub} \cdot R_{fb}) \cdot (1 - e^{(1/\tau_s)^2 t}) + V_{set} \\ V_y(t) &= (V_{set} + I_{sub} \cdot R_{fb}) \cdot e^{(1/\tau_s)^2 t} \end{aligned} \quad (9)$$

The time which $V_y = V_{des}$ in (10) is

$$t_s = \tau_s^2 (\ln(V_{des}) - \ln(V_{set} + I_0 \cdot e^{\kappa V_{in}/U_T} \cdot R_{fb})) \quad (10)$$

As the measured jitter is 100 ns in maximum, the effective resolution achieved by the proposed ATC (within its linear range) is computed as $\log_2((10.7 - 4.3)/0.1) = 6$ bits. One least significant bit corresponding to 100 ns for the pulse-width-modulated signal. To further quantify the non-linearity of the ATC, V_{in} is swept from 400mV to 530mV with a step size of 2mV, and the corresponding pulse width increment is normalized with respect to 100ns to derive the effective differential and integral nonlinearity shown in Fig. 13.

B. TIME-TO-DIGITAL CONVERTERS AND THE 2D-MAC OPERATOR

As the function of the DSC in FPGA has been verified and presented in [7], this section mainly presents and discuss the functions of the proposed DLC and 2D-MAC in FPGA. The measurement result is displayed in Fig. 14.

The two pulses of the G signal are represented two different values to be multiplied by the two unsigned multipliers, $H_1 = 0011b$ and $H_2 = 0111b$, respectively. As C_{en} first goes high, the DLC detects seven rising edges of interpolated signals and thus sets $fs = 7$. At the same time, the coarse counter (Pcounter in Fig. 14.) is triggered to measure the duration of G with C_{clk} . The falling edge of G triggers the DLC to measure the duration of Δfe . As C_{en} goes low, the DLC sets $ff = 2$, indicating that fourteen falling edges of interpolated signals are detected during Δfe ($k = 16$ in this experiment). At the same time, the coarse counter counts to 17 and stores the value in CNR . The fine-count value equaling to $(fs + ff)$ is stored in FNR . In the next clock cycle, the FNR value is copied into the registers SUM0 and SUM1 because $H_1[0] = H_1[1] = 1$, and the CNR value is copied into registers SUM4 and SUM5 because each coarse count value needs to be multiplied by 16. All other SUM registers remain zero because the corresponding $H_1[i] = 0$. Similarly, the DLC measures the second pulse and gives $FNR = 5$ and $CNR = 10$. Afterward, the FNR value is added

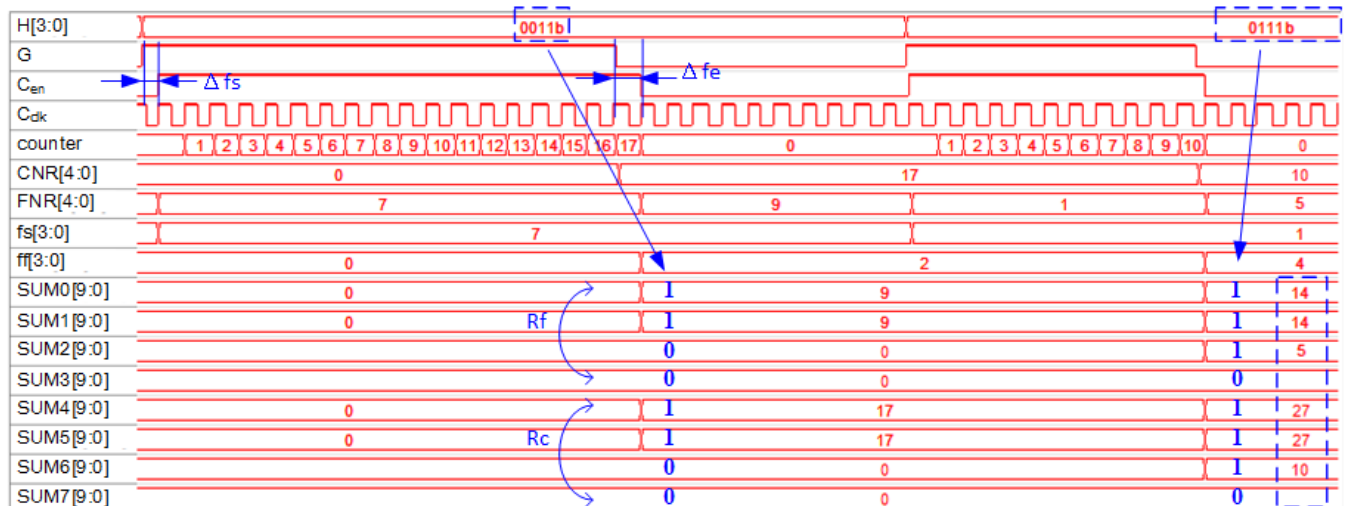


FIGURE 14. The measured operational signals of the 2D-MAC with DLC during the computation of $(H_1 \times G_1) + (H_2 \times G_2)$.

to the registers SUM0, SUM1, and SUM2, while the CNR value is added to the registers SUM4, SUM5, and SUM6.

C. POWER AND AREA TRADEOFF FOR THE DLC METHOD

Let k represent the number of interpolated signals in the DLC method ($k = 8$ in Fig. 4). The conversion resolution of the DLC method can be increased by either increasing k or increasing the frequency of the coarse-count clock. Table 1 compares the power and area consumptions for different combinations of k and coarse-count clock frequency. The value in each bracket indicates the resolution achieved by the corresponding combination. Obviously, increasing k requires more delay lines and thus a larger chip area, while increasing the clock frequency consumes more power. Increasing the clock frequency also helps to reduce the chip area because the unit delay time reduces with the clock period, so does the required number of cascaded delay units. This comparison reveals the great flexibility of the proposed DLC method. A designer can choose to minimize the power consumption, or to minimize the area, or to identify an intermediate tradeoff optimal for a specific application.

TABLE 1. The power and area consumption required for different computation resolutions, which are achieved by different combinations of k values and coarse-count frequencies.

Power (mW)						
Coarse clk	$k = 4$	$k = 8$	$k = 16$	$k = 32$	$k = 64$	$k = 128$
17.06 MHz	0.24(10)	0.24(11)	0.24(12)			
8.54 MHz	0.16(9)	0.16(10)	0.16(11)	0.16(12)		
4.27 MHz	0.08(8)	0.08(9)	0.08(10)	0.08(11)	0.08(12)	
2.13 MHz		0.07(8)	0.07(9)	0.07(10)	0.07(11)	0.07(12)

Area (LE)						
Coarse clk	$k = 4$	$k = 8$	$k = 16$	$k = 32$	$k = 64$	$k = 128$
17.06 MHz	57(10)	72 (11)	104 (12)			
8.54 MHz	92(9)	106 (10)	136 (11)	160 (12)		
4.27 MHz	160(8)	172 (9)	179 (10)	208 (11)	268 (12)	
2.13 MHz		312 (8)	331 (9)	364 (10)	420 (11)	540 (12)

D. COMPARISON WITH THE STANDARD FPGA INTELLECTUAL PROPERTY

The power and area consumptions of the proposed TDCs and 2D-MAC are further compared to the MegeCore intellectual property (IP) library provided by Altera Corporation. To coincide with the ATC proposed in this paper, the pulse width of the time-mode signal ranges from 4.3 to 10.7 μ s. The toggle rate of each method is then averaged to derive its average power consumption. Fig. 15 and Table 2 reveal the comparison for converting time-mode signals and computing their inner products at different resolutions. Fig. 15 shows that the power consumption of the standard IP and the 2D-MAC without a DSC or a DLC grows rapidly with the required resolution. On the contrary, the 2D-MAC with DLC remains at a very low power consumption of 0.08 mW for all resolutions. This intriguing property of the DLC method comes from the fact that its resolution can be increased by simply increasing the number (k) of delay signals, as discussed in Sec.V-C.

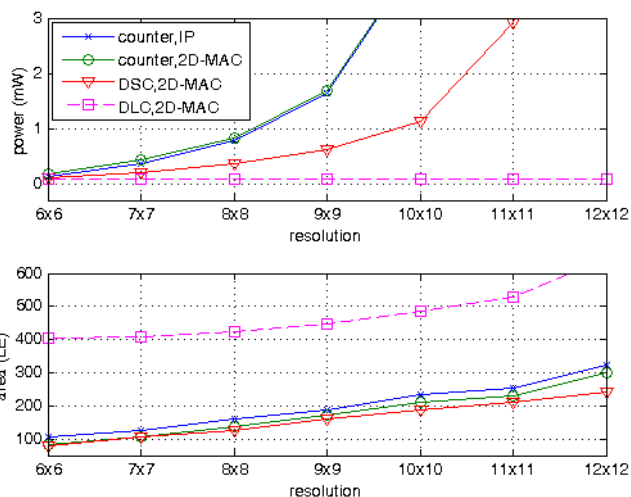


FIGURE 15. The power and area required by different methods for achieving different computation resolutions.

TABLE 2. The comparison among different methods in different resolutions.

Power (mW)	6x6	7x7	8x8	9x9	10x10	11x11	12x12
counter + IP	0.12	0.37	0.78	1.65	4.11	8.98	25.94
counter + 2D-MAC	0.18	0.43	0.83	1.69	4.14	9.01	26.06
DSC + 2D-MAC	0.11	0.19	0.37	0.61	1.13	2.92	6.54
DLC + 2D-MAC	0.08	0.08	0.08	0.08	0.08	0.08	0.08

Area (LEs)	6x6	7x7	8x8	9x9	10x10	11x11	12x12
counter + IP	107	127	161	189	233	255	322
counter + 2D-MAC	83	105	136	171	210	231	300
DSC + 2D-MAC	79	103	127	161	186	212	243
DLC + 2D-MAC	403	410	422	446	486	529	645

Table 2 further obviates the significant improvement achieved by the proposed methods. Taking the 12-bit resolution as an example, the standard IP consumes 25.94 mW. The proposed DSC reduces the power consumption to only 6.54 mW. The DLC method further helps to reduce the power by more than 320 times, while consuming only two times larger chip area than the standard IP library. For the 2D-MAC, although the 2D-MAC with the standard counter IPs consumes a little more power (0.4%) than the fully-standard IPs, the 2D-MAC helps to reduce the chip area by 7.3% (from 322 to 300 LEs). On the other hand, Fig. 15 reveals that the 2D-MAC with the DLC requires a larger area for all resolutions. Nevertheless, the required area is at most five times greater than those required by all other methods. Therefore, the 2D-MAC with the DSC is suitable for minimizing area consumption, and the 2D-MAC with the DLC is suitable for minimizing power consumption.

Table 3 further compares the proposed TDC with recently-published TDC methods which exhibit small chip area and low power consumption. It is notable that increasing the dynamic range, especially for a large pulse width, results in a considerably larger chip area and higher power consumption. The comparison in Table 3 indicates the proposed TDC (based on DLC) still offers competitive performance

TABLE 3. The comparison of the DLC-based TDC with the recently-published TDCs.

	[8] JSSC '10	[13] TCAS2 '11	[9] JSSC '12	[14] TCAS2 '14	This work
Type	Vernier Ring	Triple Slope	Vernier + GRO	Delay latch chain	Delay line counter (DLC)
Range [ns]	32	1460	40	0.73	15600
Dynamic Range	4000:1	4089:1	6897:1	128:1	4096:1
Number of bits	11	12	12	7	12
Power [mW]	7.5	1.22	3.6	1.14	0.07
Area [mm^2]	0.26	0.12	0.027	0.004	540 (LEs)
Technology [nm]	130	350	90	65	180
Year	2010	2011	2012	2014	2015

regarding high dynamic range and low power consumption, even though the proposed DLC is only implemented in an FPGA fabricated with the $0.18\mu m$ technology.

VI. EXPERIMENTS WITH REAL BIOMEDICAL DATA

The capability of the proposed circuit in processing real biomedical data is further tested with neuronal spikes recorded by a micro-electrode in the primary motor cortex of a rat [15]. Assume all detected spikes are aligned with their maximum peaks as Fig. 16. The raw data contain three types of spikes. There are 160 samples for each type of spikes, and each spike consists of 64 samples across time. To simplify the hardware implementation, all spike data are down-sampled to be 9-dimensional data. After the spike data are converted into pulse-width modulated signals, the proposed TDCs generate the coarse-count and fine-count values for each pulse (sample) signal. The 2D-MAC then sorts spikes based on the feature vectors learned by the generative model called Continuous Restricted Boltzmann Machine (CRBM) [16], [17].

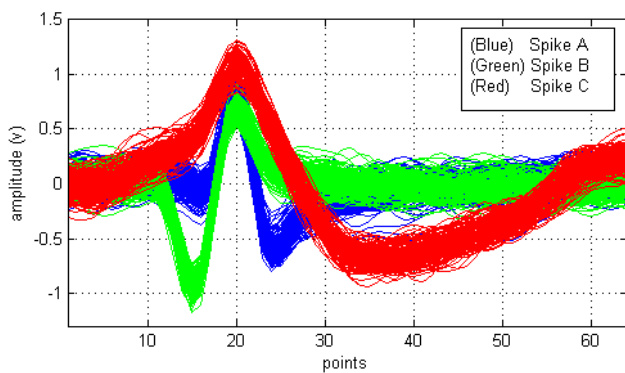


FIGURE 16. Three types of neuronal spikes recorded by a single electrode in rat cortex.

A. THE CRBM ALGORITHM

Fig. 18 illustrates the architectures of the CRBM, consisting of one visible and one hidden layer of neurons. The number of visible neurons equals the dimension of the data to be classified, while the number of hidden neurons are chosen in accordance with data complexity. In this experiment, the CRBM with nine visible and two hidden neurons is employed to learn the feature vectors of the neuronal spikes. V_0 and H_0 in Fig. 18 represent biasing neurons whose values always

equal one. Let w_{ij} represent the weight connection between neurons v_i and h_j . The CRBM learns to capture feature vectors as the weight vectors connecting between each hidden neuron and all visible neurons (e.g., w_{i1} connected to h_1). The three types of spikes in Fig. 16 are divided into a training dataset of 99 spikes and a testing dataset of 381 spikes. After training the CRBM by the minimizing-contrastive-divergence (MCD) algorithm [17] for 60,000 epochs, the parameters w_{ij} and a_j converge to their optimal values, allowing the CRBM to regenerate the three types of spikes. a_j is the scaling factor for neuron j . To classify the spikes with the trained CRBM, the hidden neurons' outputs are computed according to

$$h_j = \varphi_j(a_j \cdot \sum_{i=0}^n w_{ij} \cdot v_i) \tag{11}$$

where $n = 9$ in this experiment, φ_i is the sigmoid function, and the summation term actually denote the inner product of the weight vector $w_1 = w_{i1}$ and the spike datum $v = v_i$. Fig. 17(a) plots the calculated hidden-neuron outputs (H_1, H_2) for the testing dataset. The trained CRBM is proved able to project the three types of spikes into three distinct clusters in the hidden-neuron space.

To fit the hardware implementation of the 2D-MAC, the effective weight values $\Phi_{ij} = w_{ij} \times a_j$ is substituted into (11), and the nonlinear sigmoid function is omitted. Therefore, the 2D-MAC computes

$$h_j = \sum_{i=0}^n (\Phi_{ij} \cdot v_i) \tag{12}$$

In addition, both Φ_{ij} and v_i are simplified to fixed-point numbers. Fig. 17(b) plots the hidden-neuron outputs calculated according to the above simplification. The three types of spikes are still projected into three distinct clusters. Compared to Fig. 17(a), the separation among the clusters become

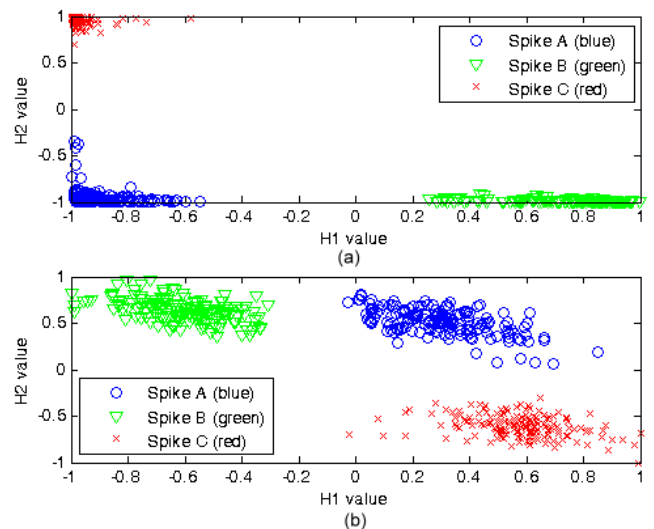


FIGURE 17. The hidden-neuron outputs of the trained CRBM in response to different types of spikes. The outputs are computed according to (a) (11) (b)(12) and normalized into $[-1, 1]$.

smaller mainly because of the lack of the sigmoid function for enlarging the difference between different clusters.

B. IMPLEMENTING THE SPIKE-SORTING CORE

To implement the CRBM-based spike-sorting core in the proposed circuit in the FPGA, the dynamic ranges of visible states (v_i), hidden states (h_j), and weight values (w_{ij}) are re-scaled according to Table 4. Each visible state is a 6-bit unsigned datum that meets the resolution of the proposed ATC. For the TDCs, both fine-count and coarse-count values are set to be three bits, i.e. $k = 8$ in (6). The effective weight values Φ_{ij} of the trained CRBM are represented by signed, 5-bit binary data and stored in the FPGA memory for the 2D-MAC.

TABLE 4. The Mapping of Parameters Between MATLAB Simulation and FPGA Implementation.

Parameters	MATLAB	FPGA	Data type
v_i	[-1.0,1.0]	[0,64]	One dimension (6 bits unsigned data)
h_j	[-1.0,1.0]	[-10240,10240]	Two dimension (7×15 bits)
Φ_{ij}	[-9.0,5.0]	[-16,16]	One dimension (4+1 bits signed data)

Note: The range of Φ_{ij} parameter in Matlab is defined based on the trained result.

Fig. 18 illustrates the partial-product registers and summing registers used by the 2D-MAC during the computation of (12). The fine-count and coarse-count values are filled into the i -th row of R_f and R_c , respectively, according to the i -th bit of Φ_{ij} . The sign column R_{sig} is derived from $G_{sig} \oplus H_{sig}$ where $G_{sig} = 0$ in this case. As R_f and R_c store the partial products of Φ_{ij} and v_i , the hidden-neuron output H_1 is derived by accumulating the row data of the R_f and R_c . The accumulated result is then stored in the 2D summing registers SUM which combines S_f and S_c registers in Fig. 19.

C. EXPERIMENTAL RESULT

Fig. 20 shows the total sum of the 2D-MAC in response to different types of spikes. Obviously, the three types of spikes can be distinguished by simply setting a threshold value between 140 and 180 for H_1 and a threshold value between 290 and 440 for H_2 . Nevertheless, the spikes can be classified even without computing the total sum from the 2D SUM registers.

As Fig. 19 illustrates, each element in the SUM corresponds to specific powers-of-two values. Elements are having the same powers-of-two value located on a specific diagonal line in the array. Depending on the sign bit in $S[7]$ of each row, the element values in the corresponding row are either positive or negative. Let a, b, c, \dots, m denote the number of positive elements corresponding to $2^0, 2^1, 2^2, \dots, 2^{12}$, respectively. Let a', b', c', \dots, m' denote the number of negative elements corresponding to $-2^0, -2^1, -2^2, \dots, -2^{12}$, respectively. The total sum is calculated as

$$SUM = 2^{12}[m - m'] + 2^{11}[l - l'] + 2^{10}[k - k'] + \dots + 2^3[d - d'] + 2^2[c - c'] + 2^1[b - b'] + 2^0[a - a'] \quad (13)$$

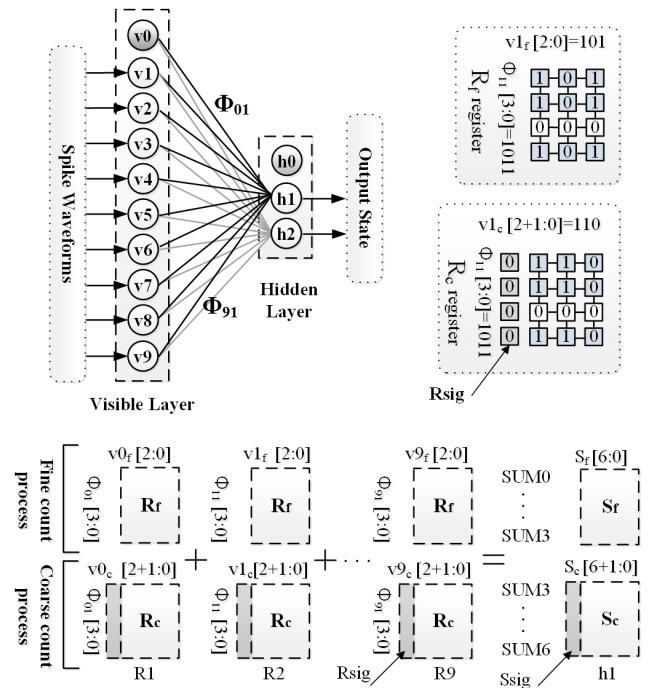


FIGURE 18. (Top-Left) The architecture of the CRBM with nine visible and two hidden neurons. The inner product of the weight vector Φ_{i1} and spike waveform v is computed by the proposed 2D-MAC. (Top-right) Each v_i is a 6-bit binary value, and the proposed TDCs convert each v_i as $v_i[5+1:0] = v_{if}[2:0] + v_{ic}[2+1:0]$. The product of v_1 and $\Phi_{11}[3+1:0] = 0_1011$ is illustrated. (bottom) Accumulating the partial products to obtain the h_1 value in the summing register.

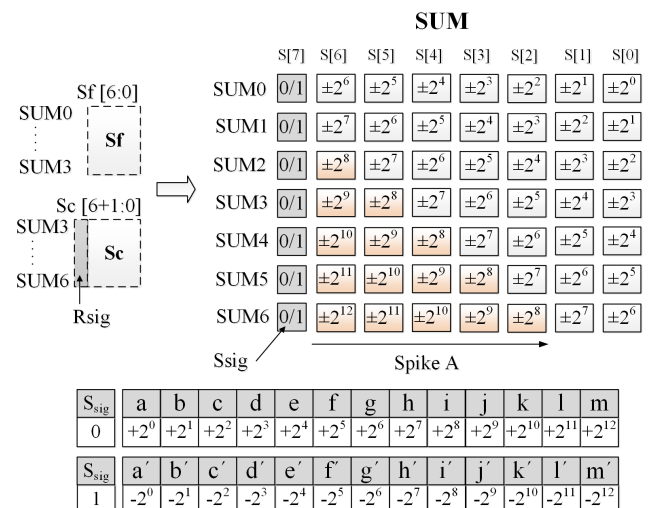


FIGURE 19. The powers-of-two exponents for the elements in a 2D summing register array.

According to Fig. 20, a threshold value, H_1^* , between 140 and 180 allows us to distinguish between spike A and spike B. As $2^7 < H_1^* < 2^8$, we can compute only the coefficients of $2^{12} 2^8$ in (13), i.e. the elements in orange color in Fig. 19, for classifying between spike A and spike B. As long as the coefficients indicate that the summation from 2^{12} to 2^8 is positive, the spike belongs to type A. Therefore, the spikes can be classified without computing the total sum

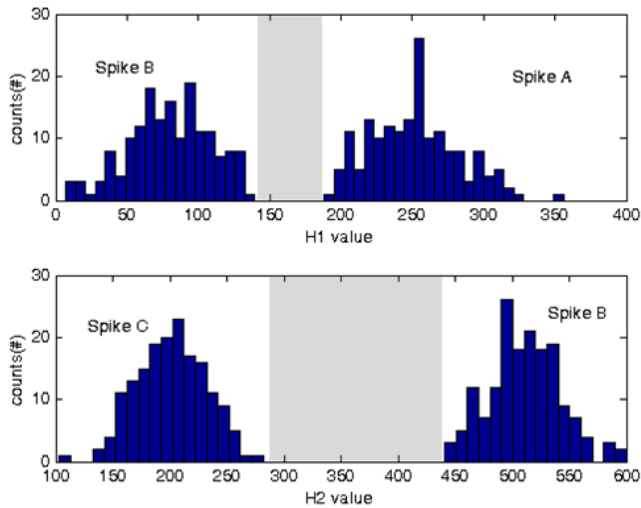


FIGURE 20. The histogram of hidden-neuron outputs in response to different types of spikes. The output values are computed according to (13). Three types of spikes can be classified by setting two threshold values, $H1^*$ and $H2^*$, in the gray region for $H1$ and $H2$, respectively.

in (13). Compared to conventional MAC, this feature avoids the need for a large register for storing the total summation of partial products.

VII. CONCLUSION

This paper presents an analog-to-time converter (ATC) together with time-mode, signal-processing circuits for classifying neuronal spikes at neuro-electronic interfaces. The ATC is verified with chip implementation, and the time-mode processing circuits are realized in FPGA and compared to standard digital IPs. The measurement result of the ATC demonstrates that the ATC is able to use positive feedback to convert analog voltages (within a dynamic range of 130mV) into pulse-width-modulated (PWM) signals with a resolution of six bits. This feature helps the neuro-electronic interface to avoid the need for a high-gain amplifier, so as to prevent the amplifier's output from being saturated by stimulation or motion artifacts. On the other hand, the ATC resolution could be further improved by designing a constant-transconductance regenerative circuit instead of simply using inverters. The PWM output of the ATC is further converted into digital data by the time-to-digital converters (TDC) based on either a dual-scale counter (DSC) or a delay-line counter (DLC). Comparing the proposed TDCs with standard IPs reveals that the DSC method minimizes area consumption, while the DLC method minimizes power consumption or provides an alternative tradeoff between power and area consumption. These features allow designers to optimize time-to-digital conversion for different applications.

REFERENCES

[1] A. V. Nurmikko et al., "Listening to brain microcircuits for interfacing with external world—Progress in wireless implantable microelectronic neuroengineering devices," *Proc. IEEE*, vol. 98, no. 3, pp. 375–388, Mar. 2010.

- [2] B. Yu et al., "Real-time FPGA-based multichannel spike sorting using hebbian eigenfilters," *IEEE J. Emerg. Sel. Topics Circuits Syst.*, vol. 1, no. 4, pp. 502–515, Dec. 2011.
- [3] Z.-H. Huang et al., "The principle of the micro-electronic neural bridge and a prototype system design," *IEEE Trans. Neural Syst. Rehabil. Eng.*, vol. 24, no. 1, pp. 180–191, Jan. 2016.
- [4] R. R. Harrison, "A versatile integrated circuit for the acquisition of biopotentials," in *Proc. IEEE Custom Integr. Circuits Conf. (CICC)*, Sep. 2007, pp. 115–122.
- [5] J. Daniels, W. Dehaene, M. S. J. Steyaert, and A. Wiesbauer, "A/D conversion using asynchronous delta-sigma modulation and time-to-digital conversion," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 9, pp. 2404–2412, Sep. 2010.
- [6] M. M. Elsayed, V. Dhanasekaran, M. Gambhir, J. Silva-Martinez, and E. Sanchez-Sinencio, "A 0.8 ps DNL time-to-digital converter with 250 MHz event rate in 65 nm CMOS for time-mode-based $\Sigma\Delta$ modulator," *IEEE J. Solid-State Circuits*, vol. 46, no. 9, pp. 2084–2098, Sep. 2011.
- [7] Y.-C. Chen, C.-T. Tang, H.-C. Wu, and H. Chen, "A compact multiply-accumulate architecture for clustering pulse-width coded biomedical signals," in *Proc. 4th Int. Symp. Bioelectron. Bioinf.*, Oct. 2015, pp. 252–257.
- [8] J. Yu, F. F. Dai, and R. C. Jaeger, "A 12-bit Vernier ring time-to-digital converter in 0.13 μm CMOS technology," *IEEE J. Solid-State Circuits*, vol. 45, no. 4, pp. 830–842, Apr. 2010.
- [9] P. Lu, A. Liscidini, and P. Andreani, "A 3.6 mW, 90 nm CMOS gated-Vernier time-to-digital converter with an equivalent resolution of 3.2 ps," *IEEE J. Solid-State Circuits*, vol. 47, no. 7, pp. 1626–1635, Jul. 2012.
- [10] G. W. Roberts and M. Ali-Bakhshian, "A brief introduction to time-to-digital and digital-to-time converters," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 57, no. 3, pp. 153–157, Mar. 2010.
- [11] H.-C. Chang, Y.-D. Wu, and H. Chen, "An analog-to-time converter with positive feedback for amplifying miniature neural recordings," in *Proc. IEEE Biomed. Circuits Syst. Conf. (BioCAS)*, Nov. 2011, pp. 86–89.
- [12] S. C. Liu, J. Kramer, G. Indiveri, T. Delbruck, R. Douglas, and C. A. Mead, *Silicon and Transistors*. Cambridge, MA, USA: MIT Press, 2002, p. 5.
- [13] M. Kim, H. Lee, J.-K. Woo, N. Xing, M.-O. Kim, and S. Kim, "A low-cost and low-power time-to-digital converter using triple-slope time stretching," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 58, no. 3, pp. 169–173, Mar. 2011.
- [14] N. U. Andersson and M. Vesterbacka, "A Vernier time-to-digital converter with delay latch chain architecture," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 61, no. 10, pp. 773–777, Oct. 2014.
- [15] V. Vigneron, H. Chen, Y. T. Chen, H. Y. Lai, and Y. Y. Chen, "Dictionary-based classification models. Applications for multichannel neural activity analysis," in *Engineering Applications of Neural Networks* (Communication in Computer and Information Science), vol. 43. Berlin, Germany: Springer-Verlag, Aug. 2009, pp. 378–388.
- [16] C. C. Lu, C. Y. Hong, and H. Chen, "A scalable and programmable architecture for the continuous restricted Boltzmann machine in VLSI," in *Proc. IEEE Int. Symp. Circuits Syst. (ISCAS)*, May 2007, pp. 1297–1300.
- [17] H. Chen, P. C. D. Fleury, and A. F. Murray, "Continuous-valued probabilistic behavior in a VLSI generative model," *IEEE Trans. Neural Netw.*, vol. 17, no. 3, pp. 755–770, May 2006.



YU-CHIEH CHEN (S'13–M'14) received the B.S. degree in mechanical engineering from the Yunlin University of Science and Technology, Yunlin, Taiwan, in 2002, and the M.S. degree in mechanical and electro-mechanical engineering from Tamkang University, Taipei, Taiwan, in 2005. He is currently pursuing the Ph.D. degree with the Neuro-Engineering Laboratory, National Tsing Hua University, Hsinchu. He is currently an Associate Researcher with National Applied Research

Laboratories, Instrument Technology Research Center, where he also is an Associate Researcher. His research interests include low-power digital signal processing and mixed-signal VLSI for biomedical applications and control theory.



HSIN-CHI CHANG received the B.S. degree in electrical engineering from the University of Wisconsin, Madison, in 2004, and the M.S. degree in electronics engineering from National Tsing Hua University, Taiwan, in 2011. Since 2011, she has been with Faraday Technology, Taiwan, where she has been in analog and mixed-signal circuit design.



HSIN CHEN (M'04) received the Ph.D. degree in electronics from The University of Edinburgh, Edinburgh, U.K., in 2004. He joined the Department of Electrical Engineering, National Tsing Hua University, Hsinchu, Taiwan, where he is appointed as an Associate Professor. He was an Invited Researcher with the IMS Laboratory, Bordeaux University, Talence, France, in 2009, the University of Évry Val d'Essonne, Évry, France, in 2011, and the Institute of Neuroinformatics, Swiss Federal Institute of Technology, Zurich, Switzerland, in 2014. He has led the Neuro-Engineering Laboratory, looking to explore neuro-inspired computation and its hardware implementation. Dr. Chen is a member of the IEEE Engineering in Medicine and Biology Society, the IEEE Circuits and Systems Society, and the Taiwan IC Design Society. He received the Outstanding Young Researcher Award from the Taiwan Integrated Circuit Design Society in 2013.

• • •