



A subranging nonuniform sampling memristive neural network-based analog-to-digital converter

Hao You^a, Amirali Amirsoleimani^{b,*}, Jianxiong Xu^a, Mostafa Rahimi Azghadi^c, Roman Genov^a

^a Department of Electrical and Computer Engineering, University of Toronto, 27 King's College Cir, Toronto, ON M5S, Canada

^b Department of Electrical Engineering and Computer Science, 4700 Keele St, Toronto, ON M3J 1P3, Canada

^c James Cook University, 1 James Cook Dr, Douglas QLD 4811, Australia

ARTICLE INFO

Keywords:

Analog to digital converter (ADC)
Memristor
Artificial neural network (ANN)
Nonuniform sampling (NUS)
Subranging

ABSTRACT

This work presents a novel 4-bit subranging nonuniform sampling (NUS) memristive neural network-based analog-to-digital converter (ADC) with improved performance trade-off among speed, power, area, and accuracy. The proposed design preserves the memristive neural network calibration and utilizes a trainable memristor weight to adapt to device mismatch and increase accuracy. Rather than conventional binary searching, we adopt quaternary searching in the ADC to realize subranging architecture's coarse and fine bits determination. A level-crossing nonuniform sampling (NUS) is introduced to the proposed ADC to enhance the ENOB under the same resolutions, power, and area consumption. Area and power consumption are reduced through circuit sharing between different stages of bit determination. The proposed 4-bit ADC achieves a highest ENOB of 5.96 and 5.6 at cut-off frequency (128 MHz) with power consumption of 0.515 mW and a figure of merit (FoM) of 82.95 fJ/conv.

1. Introduction

As machine learning (ML) develops rapidly in the recent era, more and more complex algorithms with a large number of parameters have emerged, and the need for highly efficient hardware accelerators such as in-memory or neuromorphic platforms have become eminent [1]. The emergence of resistive switching (RS) memory technologies brings new opportunities to rethink the design of the current computing circuits and systems. Not only using these emerging technologies, specifically memristor crossbar arrays can significantly accelerate vector–matrix multiplication (VMM) as the most important and widely used mathematics operations within current ML algorithms [2], but also utilizing them as the new circuit component and building block within the design of the expensive power and area hungry CMOS peripheral circuits like digital to analog converter (DAC) and analog to digital converter (ADC) will create more efficient mixed-signal circuits [3]. To make a smooth and accurate input–output transition between real and virtual worlds, the analog-to-digital converter (ADC) is an indispensable linked component. With the continuing requirements of fast, low-powered, and precise mixed-signal devices from their applied field (oscilloscopes, high-resolution display, headphones, etc.), an adequate high-speed, and low-powered ADC with high accuracy is necessary [4].

Conventional uniform sampling ADCs inevitably have some inherent trade-offs among various factors. High-speed ADCs, like flash and pipeline, are limited in accuracy and effective resolutions, affected by the mismatch of resistor ladder or capacitors [5,6]. SAR ADC with high resolutions has a lower processing speed due to its binary searching mechanism and consumes more power if the sampling speed exceeds 100 MS/s [7].

Recently, a novel efficient ADC [8] has been proposed by having a trade-off on the speed, power, and accuracy utilizing the memristor-based neuromorphic architecture to ensure the ADC's accuracy through training of the memristive weights. Using the neuromorphic architecture, this ADC can achieve high-speed conversion with high accuracy by its controllable weights and training algorithm. However, in Fig. 1(a), with the problems that whenever the outputs bits need to be doubled, the number of weights increases quadratically, and the number of computation neurons also needs to be doubled, which consumes significant area and will be challenging to incorporate into ultra-dense memory arrays for sensing applications. The binary searching mechanism makes the weights and neurons' number increase quadratically and linearly according to their resolutions, making the area and power increase substantially [8].

Additionally, another factor that limited the conventional ADCs' effective resolutions is its large quantization noise caused by uniform sampling. While the input signal is replicated to the output

* Corresponding author.

E-mail addresses: hao.you@mail.utoronto.ca (H. You), amirsol@yorku.ca (A. Amirsoleimani), jianxiong.xu@mail.utoronto.ca (J. Xu), mostafa.rahimiazghadi@jcu.edu.au (M.R. Azghadi), roman@eecg.utoronto.ca (R. Genov).

<https://doi.org/10.1016/j.memori.2023.100038>

Received 10 November 2022; Received in revised form 3 March 2023; Accepted 12 March 2023

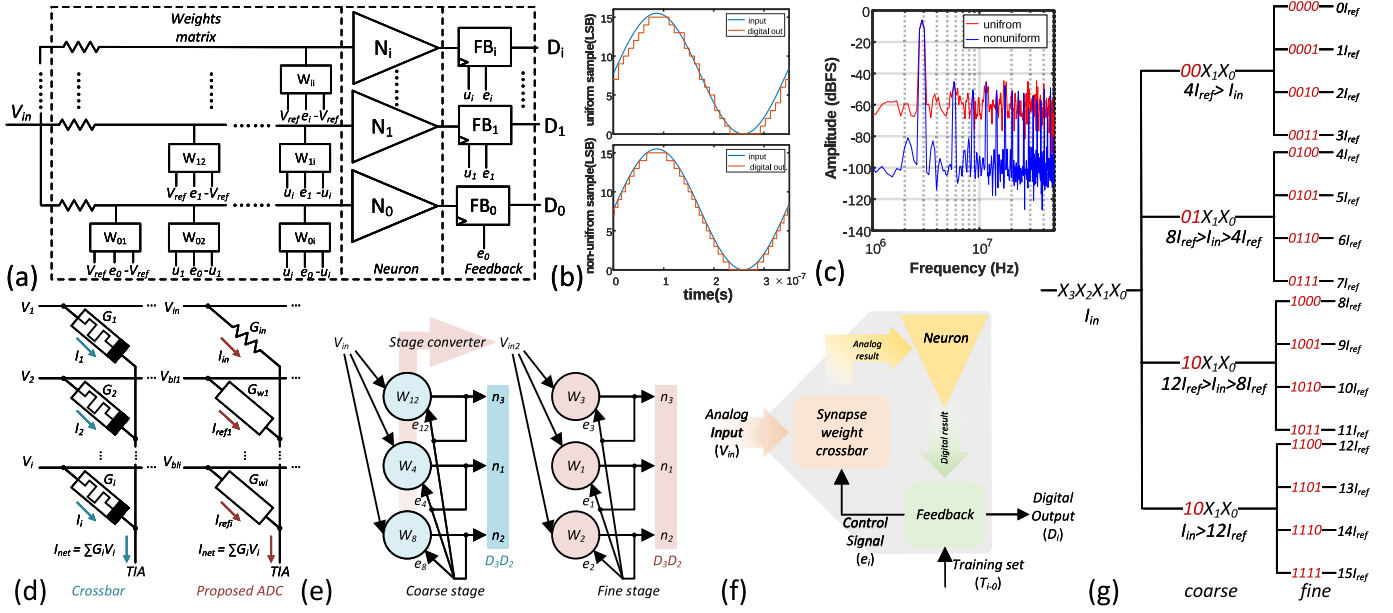


Fig. 1. (a) Binary search ANN structure pipeline ADC [8]. (b) The time domain digital output by ideal uniform sampling and nonuniform sampling 4 bit ADC, where the nonuniform one has a lower noise floor than the uniform one (d) Memristor crossbar (e) The proposed model's neural network with the relation between each weight, W_j , n_j and inputs. (f) General ANN structure of proposed ADC. (g) The quaternary search mechanism with all possible combinations for 4 bit ADC.

at every integer time of the sampling period, some samples are inevitably redundant or missing the edge crossing of input, as shown in Fig. 1(b) [9]. Rather than sampling in a uniform frequency, the non-uniform sampling (NUS) technique is proposed [9] and a sub-ranging based architecture of which is implemented [10]. The NUS's unique property allows the ADC to sample whenever the input signal is changed and thus makes the sampling more efficient and accurate. This leads to an alias-free spectral and higher signal-to-noise ratio (SNR) compared to the conventional ADC under the same number of quantization levels as long as it meets the Nyquist rate [11], which is shown in Fig. 1(c) that the ideal non-uniform sampling spectrum has a much lower noise floor compared to uniform sampling. On the other hand, sub-ranging architecture can further alleviate the power and area consumption by repeatedly utilizing the computation neurons and their coarse and fine stage at different quantization levels.

Intriguing from the existing solutions, the proposed 4 bits ADC is designed as a general-purpose ADC and is aimed to provide another optimal solution for solving the ADC's speed, power, and accuracy tradeoff through:

- (1) Memristive weight crossbar and ANN calibration (Training) for fabrication mismatch and adapt environmental variation,
 - (2) Level-crossing NUS technique with oversampling for a high effective number of bits (ENOB) under the same resolutions [12],
 - (3) Sub-ranging architecture and quaternary search mechanism to speed the design while maintaining the low area and power consumption.
- The rest of the paper will first introduce each part of the circuit and its mechanism, describe the main working and training process, conduct various evaluations to validate the proposed design, and generalize the proposed ADC for higher resolutions.

2. Mechanism

2.1. Quaternary search and weight crossbar

The proposed design adopts quaternary searching that determines two bits together to achieve coarse and fine bits determination. Using

4 bits for example, the bits D_3 and D_2 are coarse quantize while D_1 and D_0 are fine quantize,

$$\begin{cases} \mathbf{D}_3 = \mu(I_{in} - 8I_{ref}) \\ \mathbf{D}_2 = \mu(I_{in} - 12I_{ref}) \parallel (\mu(I_{in} - 4I_{ref}) \& !\mathbf{D}_3) \end{cases} \quad (1)$$

$$\begin{cases} I_{in2} = I_{in} - 8\mathbf{D}_3 I_{ref} - 4\mathbf{D}_2 I_{ref} \\ \mathbf{D}_1 = \mu(I_{in2} - 2I_{ref}) \\ \mathbf{D}_0 = \mu(I_{in2} - 3I_{ref}) \parallel (\mu(I_{in2} - I_{ref}) \& !\mathbf{D}_1) \end{cases} \quad (2)$$

where I_{in} is the input current that represents $D_3 D_2 D_1 D_0$ equal to the analog input voltage V_{in} divide R_{in} and I_{ref} is the reference current that represent one LSB. $\mu(\cdot)$ is the sigmoid function that is 1 if the parameter is positive and 0 if the parameter is negative. The full-scale currents of a 4 bits ADC are $16I_{ref}$. $D_3 D_2$, at the 2nd stage, can be determined independently without other bits by comparing I_{in} with quarter, half, or three-quarters of the full-scale current (coarse). Then corresponding $8 \times D_3 + 4 \times D_2$ times of I_{ref} will be minus from the input current to generate an intermediate current I_{in2} and is replicated through the stage converter, composed by a trans-impedance amplifier (TIA) and an inverter which both use the cascode inverter op-amp in Fig. 2(f) with C_{os} , for multiple branches used. $D_1 D_0$ at 1st stage is determined similarly as $D_3 D_2$ except using I_{in2} to compare with the one-sixteenth, one-eighth, and three-sixteenth of full-scale current (fine). To achieve the current summation, the proposed design adopted the crossbar-like weight architecture shown in Fig. 1(d) to sum the currents from input and different weights. The detailed implementation is shown in Fig. 2(a). Fig. 1(g) shows this gradual approximation with all of the possible combinations for 4 bits.

While the realistic circuit is not a linear-time-invariant (LTI) system and various conditions (device mismatch, temperature etc.) affects the converter's accuracy, an appropriate method that generates n times of I_{ref} to make neuron have expected decision is crucial. Inspired by recent discoveries on memristor neural networks [13,14], a 2T1R memristor weight is utilized to bias different V_{bli} to different times of I_{ref} as shown in Fig. 2(c). Where for the weight W_j , $I_{W_j} = j \times I_{ref}$, and e_j is the weight's control signal. Assumed both MOSFETs are in the linear region,

$$I_{W_j} = V_{bli} \cdot G_j(s) \quad (3)$$

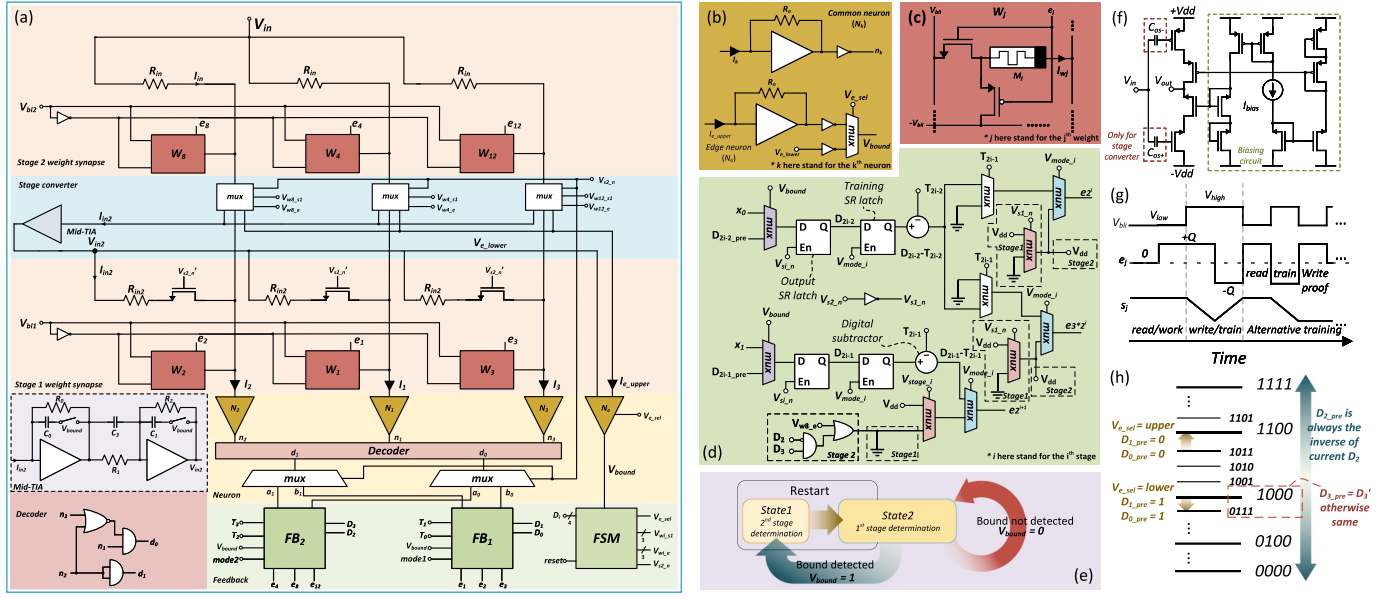


Fig. 2. Over Fig. 2, i stands for the i th stage, j stands for the j th weight, and k stands for the k th neuron. (a) The detailed structure of ADC with synapse weight W_j , computation neuron N_k , and feedback FB . (b) Neuron circuit with common neuron N_k and edge detection neuron N_e . (c) 2T1R weight unit circuit W_j (d) Feedback circuit of stage i . (e) The finite state machine (FSM) of ADC with two states each for one stage. (f) Circuit of cascade inverter amplifier used in neuron and stage converter, where C_{os} is only for stage converter to boost the gain. (g) The state variable s_j 's change with respect to different e_j and V_{bli} . (h) The prediction mechanism of D_{3-0_pre} during the level crossing. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

where V_{bli} is the corresponding i th stage bit line voltage, $G_j(s)$ is the conductance of memristor in W_j , and s is the state variable control the memristor conductance. The conductance of MOSFET is ignored here since it is much bigger than that of the memristor. Rather than utilizing the memristor by directly programming it to certain discrete stable states (which depend on the memristor's precision), the memristor here is described in the VTEAM model where its state variable, s , is continuous [15]. Hence, the memristor can be biased to a certain desired state (and desired resistance) from the initial state by large writing voltage on two sides of the memristor and writing time.

The weights are controlled by V_{bli} and e_j from the feedback circuit. And when different V_{bli} and e_j are operated on the MOSFET, different voltages will be applied on the memristor to implement different requirements. The feedback control signal e_j and V_{bli} will generally be in these four states:

- (1) $e_j = 0$, $V_{bli} = V_{low}$, both MOSFETs are nonconducting, the voltage on two sides of the memristor is approximately 0, the state variable of the memristor does not change, $I_{W_j} = 0$.
- (2) $e_j = +Q$, $V_{bli} = V_{low}$, only NMOS are conducting in the linear region, the voltage on two sides of the memristor is greater than V_{on} but smaller than 0, state variable of memristor does not change, $I_{W_j} < 0$.
- (3) $e_j = +Q$, $V_{bli} = V_{high}$, only NMOS are conducting in the linear region, the voltage on two sides of the memristor is smaller than V_{on} , state variable of memristor change, $I_{W_j} < 0$.
- (4) $e_j = -Q$, $V_{bli} = V_{high}$, only PMOS are conducting in the linear region, the voltage on two sides of the memristor is greater than V_{off} , state variable of memristor change, $I_{W_j} > 0$.

Fig. 2(g) shows the state variable changes under different e_j and V_{bli} .

2.2. Feedback circuits

The proposed design's network structure is shown in Fig. 1(e), which mainly contains two layers. The first layer (coarse stage) computes the result through input, and the first layer weights to get D_3D_2 as coarse classification on the analog input. The second layer (fine stage)

will then utilize the output from the first stage after the stage converter to finely classify the input voltage to get D_1D_0 . Fig. 1(f) shows the structure of the circuit in another way to explain how different parts of the circuit collaborate. The analog input will be computed by the weights crossbar and sent the analog computing result to neurons. The neurons will then convert their input to digital results. Finally, the digital result will be given to the feedback to generate the digital output. As the controlling of the ADC, the feedback will send either the control signal or the training signal to the weight crossbar during working or training mode.

The feedback circuit of the proposed design is organized per stage to provide the control and error signals for the corresponding synapse weight. As described in Fig. 2(d), a feedback circuit of stage i is organized as follows: two output SR-latch storing bits D_{2i-1} and D_{2i-1+1} with enable signal $V_{sl,n}$ which only receive new digital output during their stage, two training SR-latch with enable signal $V_{mode,i}$ to store the reading result during the training mode to output steady difference, two digital subtractors calculating the difference between actual results (D) and expected results (T), blue multiplexers for mode selections (change between training and working mode), pink multiplexers for operating stages selections (stage 1 or 2), and purple multiplexers to have a prediction on next digit when the level is crossing.

2.3. Computation neuron

The computation neuron N_k in Fig. 2(b) comprises a TIA and an inverter. It transforms the summing current I_k that represents the analog calculation result from the weight crossbar to digital output. The edge detection neuron, N_e , has one more multiplexer that can be used to select which edge it needs to detect: upper or lower, through the control of V_{e_sel} . More information about circuit detail is shown in Table 1.

3. Working process

The proposed ADC has two modes, with the working mode that transforms input analog signal to digital output continuously, and the training mode that uses the alternative training mechanism to train the synapse weights to the desired value.

Table 1
Circuit parameter.

Circuit part	Type	Parameter	Value	
Weight	NMOS	W/L	10	
		PMOS	W/L	20
	Memristor [16]	$V_{on/off}$	-0.3/0.4 V	
		$R_{on/off}$	2k/100 k Ω	
		$k_{on/off}$	-4.8/2.8 mm/s	
		$\alpha_{on/off}$	3/1	
		Precision	6	
Neuron	Amplifier	V_{dd}	0.6 V	
		I_{bias}	16 μ A	
		Gain	59.2 dB	
		BW	1.42 GHz	
Settling time	Amplifier	T_s	3 ns	
Stage converter	Amplifier	V_{dd}	0.9 V	
		I_{bias}	120 μ A	
		$C_{os+/-}$	10pF ic = -0.25/0.15 V	
		Gain	54.5 dB	
		BW	7.46 GHz	
		Settling time	T_s	6 ns
		ADC	Input resistor	R_{in1}
R_{in2}	11.1 k Ω			
Bitline voltage	V_{bl1}		0.1/0.43 V	
	V_{bl2}		0.05/0.49 V	
FSM	T_{state1}		3 ns	
	T_{state2}		9 ns	
NMOS	T_{train}		2 μ s	
	W/L	20/3		
Reference current	I_{ref}	1 μ A		

3.1. Working mode

The proposed ADC in working mode is controlled by the finite state machine (FSM), as stated in Fig. 2(e). It mainly has two states: state 1 for 2nd stage bits D_3D_2 determinations (coarse) and state 2 for 1st stage bits D_1D_0 determinations (fine). The circuit is restarted first by running state 1 and then state 2. As shown in Fig. 3, during state 1, $V_{s2,n} = +V_{dd}$ indicates that all summing currents to neurons are from 2nd stage and 2nd stage's SR latch is enabled to receive the digital outputs of D_3D_2 . All 1st stage's MOSFET is closed so that I_{in2} and all I_{wj} at 1st stage is zero to avoid interruption. Proceeding to the state 2, $V_{s1,n} = +V_{dd}$ and $V_{s2,n} = -V_{dd}$. At stage converter's multiplexers, corresponding $V_{wj,s1} = +V_{dd}$ and I_{in2} is calculated by summing the I_{wj} and I_{in} . I_{in2} is then replicated through the amplifiers and R_{in2} for D_1D_0 determinations. 1st stage's SR latch is enabled at state 2 and continuously receives the digital output of D_1D_0 . After restart, the FSM will stick in state 2 to track the D_1D_0 bits until a level-crossing is detected, where $V_{bound} = 1$ as illustrated in Fig. 2(e). During the tracking of D_1D_0 , an appropriate edge is selected and tracked whether the edge is crossed, as shown in Fig. 3,

- (1) When $D_1D_0 = 11$, an upper edge is going to be touched, $V_{e,sel} = upper$, and the edge of current D_3D_2 plus one is selected. This edge comparison is calculated by borrowing the result from 2nd stage, where corresponding $V_{wj,e} = +V_{dd}$ at stage converter multiplexers and the summing currents of I_{wj} and I_{in} flow to the edge detection neuron N_e .
- (2) When $D_1D_0 = 00$, an lower edge is going to be touched, $V_{e,sel} = lower$. The lower edge comparison is conducted by comparing the V_{in2} with 0.
- (3) Otherwise, D_1D_0 is away from the edge and V_{bound} is set to 0.

When a level crosses and $V_{bound} = 1$, the circuit will jump back to state 1 and then state 2 to redetermine all 4 bits, and V_{bound} will be held in 1 until state two is stabilized. The selection signals at stage converter multiplexers are all $-V_{dd}$ if they are not explicitly stated. During the level-crossing, where state 1 and state 2 are re-executed, the output SR-latch in Fig. 2(d) will receive predicted bits $D_{3-0,pre}$ rather than getting

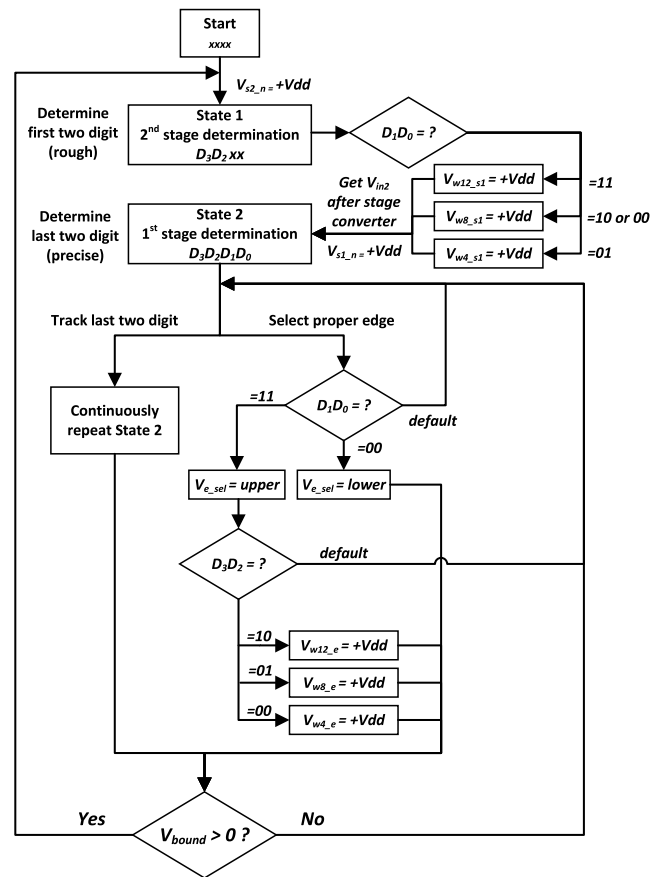


Fig. 3. Flow chart of ADC under working mode, with the detailed description on state 1&2 and edge detection.

x_1x_0 from neuron to avoid glitches due to the destabilization between states. The prediction logic is shown in Fig. 2(h).

3.2. Training mode

The proposed ADC adopts an alternative training mechanism to train the synapse weights to successively approximate the desired value where the training and working mode is switched alternatively to make sure the training result is correct. Each weight is trained separately and follows the same process. To train the weight W_j at 2nd stage, corresponding training sets V_{in} and T_3T_2 is input to weight crossbar and FB_2 , and all MOSFET in 1st stage is closed to avoid interruption. Firstly, described in Fig. 4(a), $V_{mode2} = +V_{dd}$ and $V_{bl2} = V_{low}$ to run the ADC in state 1 on working mode to read the weight's current calculation result. The digital subtractor will evaluate the difference $D_3 - T_3$ and $D_2 - T_2$ simultaneously. If the difference is 0, the training is finished. Otherwise, the circuit will enter training mode, $V_{mode2} = -V_{dd}$. At this point, the difference by digital subtractor will return through e_j to the corresponding synapse weight and $V_{bl2} = V_{high}$ to ensure the voltage on two sides of the memristor exceeds one of its threshold voltage. After a unit training period T_{train} , the memristor's state variable s will change bit based on equation [8,15],

$$\Delta s = \int_0^{T_{train}} k_{on/off} \left(\frac{\pm V_{high}}{V_{on/off}} - 1 \right)^{\alpha_{on/off}} \cdot f_{win} dx \quad (4)$$

$$f_{win} = s \cdot (s - 1) \quad (5)$$

where f_{win} is the window function for limiting state variable. And hence for memristor resistance, it will vary by [8],

$$\Delta R = (R_{off} - R_{on})\Delta s \quad (6)$$

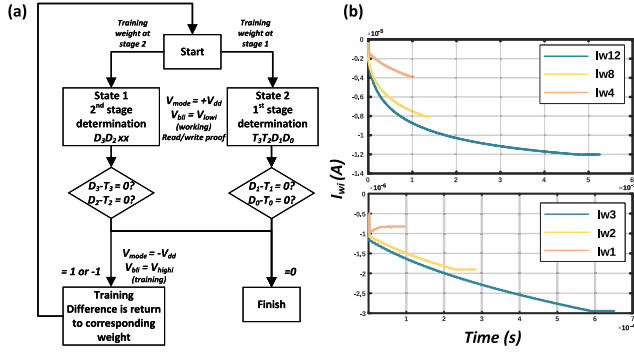


Fig. 4. (a) Flow chart of ADC under training mode. (b) The training process of $I_{w,j}$.

The circuit will then return to state 1 to have writing proof. The difference will be evaluated, and if it is not zero, the alternative training will be repeated until the zero difference is reached. The synapse weights training on 1st stage is similar to the 2nd stage weights except the state 2 is executed in every training cycle, the training sets should include T_{3-0} , and the 2nd stage should remain in working mode to provide stable I_{in2} . Edge detection is disabled during all training processes. The training results of each weight are shown in Fig. 4(b).

4. Evaluation

The proposed 4 bit ADC is simulated in SPICE software (LTspice), using 180 nm MOSFET technology (PTM BSIM3 modal) [17], and VTEAM model [15] for memristor [16]. After AD conversion, the digital output is sampled again using a 5GSPS digital sampler for a large oversampling ratio (OSR) and filtered by a digital filter. The digital sampler and filter and FSM are considered as share components and not included in the evaluation.

4.1. Power evaluation

The power consumption of the proposed ADC is analyzed and mainly comes from these sources:

- (1) Amplifiers in stage converters that cost 216 μ W each, with total dissipate 432 μ W. The three feedback resistors dissipate 1 μ W in total on average.
- (2) Amplifiers in neurons that cost 19.2 μ W each, with total dissipate 76.8 μ W. The feedback resistor R_n dissipates power in 10th of nW and is thus negligible.
- (3) Synapse weights, where W_{12} dissipate 1.11 μ W, W_8 dissipate 0.8 μ W, W_4 dissipate 0.4 μ W, W_3 dissipate 0.13 μ W, W_2 dissipate 0.1 μ W, and W_1 dissipate 0.05 μ W.

Other parts of circuits' (logic, FSM, multiplexers) power dissipation is negligible, where mostly they are around or below nW.

4.2. Area evaluation

The scaling of the neural network ADC is improved by the proposed ADC through subranging architecture and component sharing. As shown in Fig. 5, as the resolution (N) increases, the required number of synapse weights now equals to $3 * N/2$ rather than quadratic proportional to resolution $N(N + 1)/2$ [8]. And the number of computation neurons for bit calculation is fixed (by sharing neurons among different stages) regardless of resolutions rather than equal to resolutions [8]. Except that an extra edge detection neuron is needed when one more synapse weights stage is added.

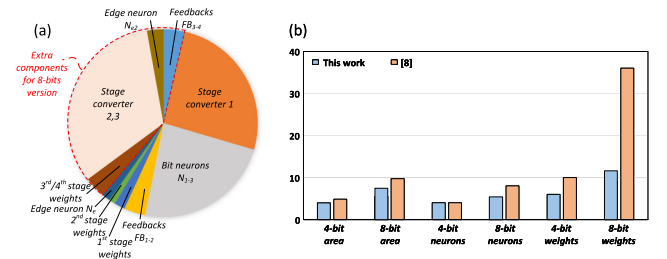


Fig. 5. (a) Area consumption by 4 bit and 8 bit version of proposed ADC (red part are the additional needed area for 8 bit proposed ADC) (b) Comparison of area consumption between this work and [8]. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

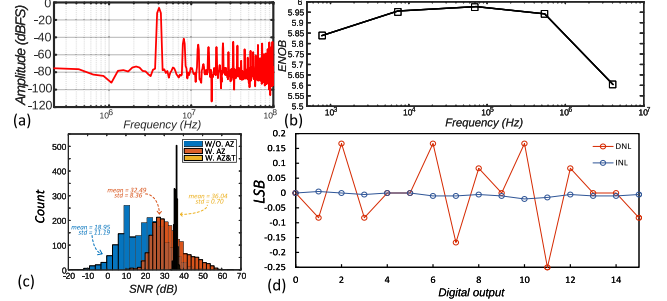


Fig. 6. (a) Frequency response of the digital output under 4 MHz sin wave input (b) Frequency response of ENOB (c) DNL and INL of ADC. (For interpretation of the references to color in this figure legend, the reader is referred to the web version of this article.)

4.3. Accuracy & speed evaluation

Fig. 6(a) shows the output spectrum of ADC under cutoff frequency (128 MHz) when a 4 MHz sine wave is input. After oversampling by 5GSPS digital sampler and filtering by a digital filter, a 35.41 dB signal-to-noise ratio (SNR) is acquired. Fig. 6(b) shows the frequency response of ENOB, which is calculated through SNR at different frequencies. The ENOB is first improved when frequency increases due to the gradually spread out of the noise floor and decrease near the ADC bandwidth due to the phase lagging of the stage converter's amplifiers.

4.4. Calibration evaluation

A proposed ADC of initial weights, with heavy transistor and resistor mismatch (10% under normal distribution), is simulated. The resulting SNR distribution is shown in Fig. 6(c) in blue color. The orange color and yellow color distribution are fellow simulation results representing the circuit with auto-zero and auto-zero with weight calibration. Since both mean and standard deviation is improved, the proposed ADC is demonstrated to have an effective calibrating mechanism. Fig. 6(d) shows the DNL and INL of ADC after training. The Variation in the memristor device's parameter generally will not influence the accuracy of the proposed ADC. As long as desired memristor's resistance is in the range of $R_{on/off}$, the alternative training will ultimately push the memristor's state to a state with desired resistance by the feedback. However, the variation of the $k_{on/off}$ and $\alpha_{on/off}$ will influence the unit training step by Eqs. (4)(5)(6), and hence influence the training speed.

4.5. Comparison with existing works

Table 2 shows the comparison between different neural networks or non-uniform sampling ADC with the proposed ADC. The proposed ADC is highlighted in the area and power efficiency. Among Table 2,

Table 2
Comparison with neural network and NUS ADC.

	This work ^c	[18]	[19]	[8]	[10]	[20]
Memory technology	RRAM	RRAM	RRAM	RRAM	N/A	N/A
CMOS Node (nm)	180	130	130	180	65	65
Area (mm ²)	3.96/5.56e-3	0.02	0.01	4.85e-3	0.7	0.13
f_s (Hz)	NUS	1G	1G	1.66G	NUS	NUS
Power (mW)	0.515/0.977	25	18	0.1	28	19.7
Bandwidth (Hz)	128M/83M ^d	1G	0.5G	1.668G	20M	200M
Resolution (bits)	4/6 ^e	8	6	4	18	4
ENOB (bits) ^a	5.6/9.24	8	5.95	3.7	10.8	10.4
FoM_w (fJ/conv) ^b	82.95/19.5	97.7	291	8.25	476.2	71.6
Trainable?	Yes	Yes	Yes	Yes	No	No

^aENOB = (SNDR-1.76)/6.02.
^b FoM_w = Power/(2^{ENOB}*BW) for NUS ADC. FoM_w = Power/(2^{ENOB} * f_s) for Nyquist ADC.
^cThis work, [18], [19], and [8] are simulation results, while [10] and [20] are measurement results.
^dBW is approximated as 2^{resolution+1}* Maximum input frequency.
^e6 bits version of the proposed ADC is estimated based on the 4 bits version by the mathematical model.

except [8], the proposed ADC generally consumes less power and area per bit.

(1) Compared with [18,19], Nyquist neural network ADC with sub-ranging architecture, the proposed ADC utilized non-uniform sampling by level crossing method. Combined with the oversampling technique, the proposed ADC is able to acquire a higher effective number of bits under the same resolution. [18] is a pipeline ADC and its output has 8 sampling periods latency to the input. The proposed ADC on the other hand always has real-time output. To increase power and area efficiency, the proposed ADC does sacrifice part of the speed. But overall on the FoM, the proposed design has better metrics.

(2) Compared with [8], except by applying nonuniform sampling, the proposed ADC solves the problem of area boosting when bits increase by sub-ranging architecture and weight sharing as described in Section 4. Evaluation, part B Area evaluation. The FoM of [8] is considered out of scope as described in [18] that [8]’s FoM is evaluated under a low input frequency (44 kHz).

(3) Compared with [10,20], traditional non-uniform sampling ADC, the proposed ADC utilized trainable memristor weight as a solution for the mismatch during fabrication. An alternative training mechanism is utilized for calibrating the weight to adapt the mismatch and hence increase the accuracy. [20]’s ENOB is considered out of scope due to its delta-sigma noise shaping after quantization.

5. Generalization

After the demonstration of the proposed ADC’s feasibility, the proposed ADC can be used as a unit building block to extend and generalize for a higher resolution ADC. Fig. 7 shows the general structure of the extended version of the proposed ADC based on Fig. 2(a). For each extra 2 bits, the proposed ADC structure will be extended one more stage, with the corresponding increase in components described in Table 3. Every two resolutions increase requires another three weights for determining bits, an extra stage converter for converting the residual to the next stage, and an extra edge detection neuron for edge detection of this stage’s two bits. Due to the linear increase of electrical components, power consumption (which is dominated by the stage converter’s TIA and neuron’s TIA) and area increase linearly with respect to the resolutions.

To maintain the level-crossing non-uniform sampling and real-time output, the FSM of the extended ADC is also modified accordingly. Fig. 7(b) shows the modified FSM for the ADC in Fig. 7(a). The ADC is reset from state 1 and goes through to state i. After that, the ADC is maintained in the last state, state i, as long as no edge is detected. Once an edge is detected in the kth stage, the ADC will return to the corresponding state and go through all states till the last. As described

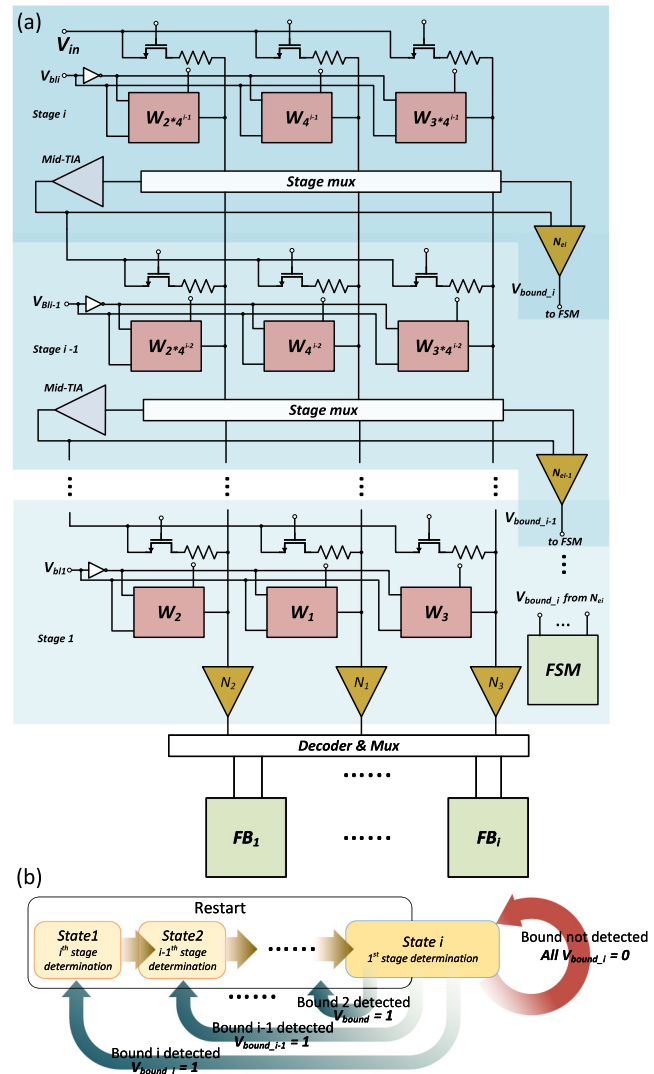


Fig. 7. (a) The general architecture of proposed ADC with i stages for 2i bits. It is a simplified but extended graph of Fig. 2(a). (b) The FSM for 2i bits version of the proposed ADC extended from Fig. 2(e). Each state k corresponds to the operation on the stage i-k+1.

in Fig. 7(b), the longest set of states that needs to go through is linearly increasing with respect to the resolutions. This will increase the average bits’ redetermination period when the edge is detected and thus decrease the bandwidth of ADC for no distortion output. The linearly increasing regularity of maximum interrupt time is described in Table 3, 5th column. Based on Table 2, 3 ns corresponds to the settling time of neurons, where for every first state, its analog result only needs to pass neurons. For every middle state, the analog result from the last stage not only needs to pass the stage converter but also neurons, which cause 3 + 6 (stage converter settling time), 9 ns in total. For the last state, the FSM only needs to maintain the interrupt until the stage converter is settled to avoid extra interruption caused by the oscillation, which is 6 ns. However, if the input signal’s slew rate is predictable, FSM operation can be simplified by jumping a certain number of states when the edge is detected and predicting the jumping bits based on a similar mechanism in Fig. 2(h). A 6 bits version of the proposed ADC is simulated based on the mathematical model with the corresponding circuit parameters, and the simulation outputs are listed in Table 2.

Table 3
Generalization of proposed ADC.

# Bits	Area			Max interrupt time (ns)	Power (mW)
	#Weight	#Stage converter	#Neuron, #Feedback		
4	6	1	3+1, 2	3+6	0.515
6	9	2	3+2, 4	3+9+6	0.977
2n	3n	n-1	3+n-1, 2n	3+9(n-2)+6	≈ 0.5(n-1)

6. Conclusion

This paper proposed a new subranging nonuniform memristor-based ANN ADC that achieves another improvement in ADC's speed, power, area, and accuracy tradeoff through multiple techniques. The proposed ADC: (1) Preserve the memristor-based structure and trainable ANN calibration to reduce inaccuracy bring by the device mismatch and make the circuit adjustable to adopt environment variation, (2) Introduce memristor and circuit sharing by subranging ADC architecture to improve the power and area efficiency, (3) Utilize quaternary search to speed ADC's bits determination process, (4) And achieve high ADC ENOB and SNR under same resolutions and area/power consumption through nonuniform sampling. Through intensive simulations on circuits, we demonstrate the proposed design is able to have enough calibration ability among different device mismatches and have stable performance over a wide range of input frequencies.

The proposed ADC discussed in the paper is only one locally optimal solution of speed, power, accuracy, and area tradeoff. The design of the proposed ADC however prioritizes the power, accuracy, and area and put the speed on the last. Thus, the speed performance has more or less been sacrificed. With a larger power and area combined with the latest technology, we believe the proposed architecture can be optimized to a higher speed and even with a better FoM.

Declaration of competing interest

The authors declare the following financial interests/personal relationships which may be considered as potential competing interests: Amirali Amirsoleimani reports financial support was provided by Natural Sciences and Engineering Research Council of Canada (DGECR-2022-00101).

Data availability

Data will be made available on request.

References

- [1] A. Sebastian, M. Le Gallo, R. Khaddam-Aljameh, E. Eleftheriou, Memory devices and applications for in-memory computing, *Nature Nanotechnol.* 15 (7) (2020) 529–544.
- [2] A. Amirsoleimani, F. Alibart, V. Yon, J. Xu, M.R. Pazhouhandeh, S. Ecoffey, Y. Beilliard, R. Genov, D. Drouin, In-memory vector-matrix multiplication in monolithic complementary metal-oxide-semiconductor-memristor integrated circuits: Design choices, challenges, and perspectives, *Adv. Intell. Syst.* 2 (11) (2020) 2000115.
- [3] X. Guo, et al., Modeling and experimental demonstration of a Hopfield network analog-to-digital converter with hybrid CMOS/Memristor circuits, *Front. Neurosci.* 9 (2015) 488.
- [4] R.J. van de Plassche, *CMOS Integrated Analog-to-Digital and Digital-to-Analog Converters*, Springer Science & Business Media, 2013.
- [5] Y. Du, Y. Li, A/D converter architectures for energy-efficient vision processor, *CoRR*, 2017, 1703.01681.
- [6] P.J. Quinn, A.H.M. van Roermund, Accuracy limitations of pipelined ADCs, in: 2005 IEEE International Symposium on Circuits and Systems, vol. 3, 2005, pp. 1956–1959.
- [7] B. Dinesh Kumar, Sumit K. Pandey, Navneet Gupta, Hitesh Shrimali, Design of hybrid flash-SAR ADC using an inverter based comparator in 28 nm CMOS, *Microelectron. J.* (ISSN: 0026-2692) 95 (2020) 104666.
- [8] L. Daniai, N. Wainstein, S. Kraus, S. Kvatinsky, Breaking through the speed-power-accuracy tradeoff in ADCs using a memristive neuromorphic architecture, *IEEE Trans. Emerg. Top. Comput. Intell.* 2 (5) (2018) 396–409.
- [9] Tzu-Fan Wu, S. Dey, M.S.-W. Chen, A nonuniform sampling ADC architecture with reconfigurable digital anti-aliasing filter, *IEEE Trans. Circuits Syst. I. Regul. Pap.* 63 (10) (2016) 1639–1651.
- [10] Tzu-Fan Wu, M.S.-W. Chen, A subranging-based nonuniform sampling ADC with sampling event filtering, *IEEE Solid State Circuit. Lett.* 1 (4) (2018) 78–81.
- [11] F. Marvasti, *Nonuniform Sampling: Theory and Practice*, Springer, New York, NY, USA, 2001.
- [12] N. Sayiner, H.V. Sorensen, T.R. Viswanathan, A level-crossing sampling scheme for A/D conversion, *IEEE Trans. Circuits Syst.* 43 (4) (1996) 335–339.
- [13] B. Crafton, M. West, P. Basnet, E. Vogel, A. Raychowdhury, Local learning in RRAM neural networks with sparse direct feedback alignment, in: 2019 IEEE/ACM International Symposium on Low Power Electronics and Design, ISLPED, 2019, pp. 1–6.
- [14] D. Soudry, D. Di Castro, A. Gal, A. Kolodny, S. Kvatinsky, Memristor-based multilayer neural networks with online gradient descent training, *IEEE Trans. Neural Netw. Learn. Syst.* 26 (10) (2015) 2408–2421.
- [15] S. Kvatinsky, M. Ramadan, E.G. Friedman, A. Kolodny, VTEAM: A general model for voltage-controlled memristors, *IEEE Trans. Circuits Syst. II Express Briefs* 62 (8) (2015) 786–790.
- [16] J. Sandrini, B. Attarimashalkoubek, E. Shahrabi, I. Krawczuk, Y. Leblebici, Effect of metal buffer layer and thermal annealing on HfOx-based ReRAMs, in: 2016 IEEE International Conference on the Science of Electrical Engineering, ICSEE, 2016, pp. 1–5, <http://dx.doi.org/10.1109/ICSEE.2016.7806101>.
- [17] Y. Cao, T. Sato, D. Sylvester, M. Orshansky, C. Hu, New Paradigm of Predictive MOSFET and Interconnect Modeling for Early Circuit Design, *CICC*, 2000, pp. 201–204.
- [18] W. Cao, L. Ke, A. Chakrabarti, X. Zhang, Evaluating neural network-inspired analog-to-digital conversion with low-precision RRAM, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* 40 (5) (2021) 808–821.
- [19] W. Cao, X. He, A. Chakrabarti, X. Zhang, NeuADC: Neural network-inspired synthesizable analog-to-digital conversion, *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.* 39 (9) (2020) 1841–1854.
- [20] Tzu-Fan Wu, M.S.-W. Chen, A noise-shaped VCO-based nonuniform sampling ADC with phase-domain level crossing, *IEEE J. Solid-State Circuits* 54 (3) (2019) 623–635.