AFRL-RI-RS-TR-2022-073

ANALOG CIRCUIT SYSTEMS FOR MEMRISTOR-BASED NEUROMORPHIC SYSTEM

SAN FRANCISCO STATE UNIVERSITY

MAY 2022

FINAL TECHNICAL REPORT

APPROVED FOR PUBLIC RELEASE; DISTRIBUTION UNLIMITED

STINFO COPY

AIR FORCE RESEARCH LABORATORY
INFORMATION DIRECTORATE

■ **AIR FORCE MATERIEL COMMAND** ■ **UNITED STATES AIR FORCE** ■ **ROME, NY 13441**

NOTICE AND SIGNATURE PAGE

Using Government drawings, specifications, or other data included in this document for any purpose other than Government procurement does not in any way obligate the U.S. Government. The fact that the Government formulated or supplied the drawings, specifications, or other data does not license the holder or any other person or corporation; or convey any rights or permission to manufacture, use, or sell any patented invention that may relate to them.

This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09. This report is available to the general public, including foreign nations. Copies may be obtained from the Defense Technical Information Center (DTIC) (http://www.dtic.mil).

AFRL-RI-RS-TR-2022-073 HAS BEEN REVIEWED AND IS APPROVED FOR PUBLICATION IN ACCORDANCE WITH ASSIGNED DISTRIBUTION STATEMENT.

FOR THE CHIEF ENGINEER:

/ S /
LISA LOOMIS
Work Unit Manager

/ S /
GREGORY HADYNSKI
Assistant Tech Advisor
Computing & Communications Division
Information Directorate

This report is published in the interest of scientific and technical information exchange, and its publication does not constitute the Government's approval or disapproval of its ideas or findings

REPORT DOCUMENTATION PAGE

PLEASE DO NOT RETURN YOUR FORM TO THE ABOVE ORGANIZATION.

1. REPORT DATE	2. REPORT TYPE	3. DATES COVERED	
		START DATE	END DATE
MAY 2022	FINAL TECHNICAL REPORT	JULY 2018	DECEMBER 2021

4. TITLE AND SUBTITLE

Analog Circuit Systems for Memristor-based Neuromorphic System

5a. CONTRACT NUMBER	5b. GRANT NUMBER	5c. PROGRAM ELEMENT NUMBER
FA8750-18-2-0123	N/A	62788F
5d. PROJECT NUMBER	**5e. TASK NUMBER**	**5f. WORK UNIT NUMBER**
		R2JA

6. AUTHOR(S)

Jiang, Hao

7. PERFORMING ORGANIZATION NAME(S) AND ADDRESS(ES)

San Francisco State University
1600 Holloway Ave.,
San Francisco CA 94132

8.PERFORMING ORGANIZATION REPORT NUMBER

9. SPONSORING/MONITORING AGENCY NAME(S) AND ADDRESS(ES)	10.SPONSOR/MONITOR'S ACRONYM(S)	11.SPONSOR/MONITOR'SREPORT NUMBER(S)
Air Force Research Laboratory/RITB 525 Brooks Road Rome NY 13441-4505	RI	AFRL-RI-RS-TR-2022-073

12. DISTRIBUTION/AVAILABILITY STATEMENT

Approved for Public Release; Distribution Unlimited. This report is the result of contracted fundamental research deemed exempt from public affairs security and policy review in accordance with SAF/AQR memorandum dated 10 Dec 08 and AFRL/CA policy clarification memorandum dated 16 Jan 09.

13. **SUPPLEMENTARY NOTES**

14. ABSTRACT

In this project, an innovative circuit system was designed and implemented. It consists of a global circuit cell that supports the entire system and many local cells that support massively parallel I/O of the memristor crossbar arrays (MCAs). The I/O circuits use the pulse width modulated (PWM) signals as the computation variable. Specifically, performers designed and implemented the global delay-locked loop (DLL) and local phase detector (PD) to generate the PWM signals, and the active current mirror, the trans-impedance amplifier (TIA), and the integrated successive approximation register analog-to-digital convertor (SAR-ADC) to scale, measure and digitize the current from the MCA. Performers also develop a printed circuit board (PCB) to evaluate MCA.

15. SUBJECT TERMS

memristor, neuromorphic, computing, architecture, memory, resistive

16. SECURITY CLASSIFICATION OF:			17. LIMITATION OF ABSTRACT	18. NUMBER OF PAGES
a. REPORT	b. ABSTRACT	C. THIS PAGE		
U	**U**	**U**	**SAR**	**26**

19a. NAME OF RESPONSIBLE PERSON	19b. PHONE NUMBER *(Include area code)*
LISA LOOMIS	**N/A**

PREVIOUS EDITION IS OBSOLETE.

STANDARD FORM 298 (REV. 5/2020)
Prescribed by ANSI Std. Z39.18

TABLE OF CONTENTS

LIST OF FIGURES

Figure **Page**

1.0 SUMMARY

The overarching research goal is to develop energy and area efficient analog *input* and *output* (I/O) circuitry to facilitate the neuromorphic computing using the recently developed *memristor crossbar array* (MCA) technology. In this project, we designed and implemented an innovative circuit system. It consists of a global circuit cell that supports the entire system and many local cells that support massively parallel I/O of the MCAs. In this project, the I/O circuits use the *pulse width modulated* (PWM) signals as the computation variable. Specifically, we designed and implemented the global *delay-locked loop* (DLL) and local *phase detector* (PD) to generate the PWM signals, and the active current mirror, the *trans-impedance amplifier* (TIA), and the integrated *successive approximation register analog-to-digital convertor* (SAR-ADC) to scale, measure and digitize the current from the MCA. We also develop a *printed circuit board* (PCB) to evaluate MCA.

2.0 INTRODUCTION

The neuromorphic system has the potential to realize high-efficiency computing [1]. Traditionally, the neuromorphic computing system is implemented in conventional digital integrated circuits and requires a significant amount of power and chip area [2].

The recently-developed memristor technology [3] provides a promising path to have a large memory without consuming a large amount of power or size [4, 5]. Several research groups have explored the hardware implementation of a neuromorphic system using the *memristor crossbar array* (MCA) [6-8]. The power/area efficiency of the I/O circuits, which facilitate the MCA based neuromorphic computation, significantly impact on the overall computation efficiency [8]. The objective of this project is to develop power/area efficient I/O circuits.

In this two-year project, the team at San Francisco State University (SFSU) has designed and implemented high-efficient analog circuits that are part of the neuromorphic processor chip developed by Duke University (Duke) and University of Massachusetts Amherst (UMass). Duke has designed and implemented the MCA based neuromorphic controller systems, while UMass has developed MCA technology. We designed and implemented the global *delay-locked loop* (DLL) and local *phase detector* (PD) to generate the PWM signals, and the active current mirror, the *trans-impedance amplifier* (TIA), and the integrated *successive approximation register analog-to-digital convertor* (SAR-ADC) to scale, measure and digitize the current from the MCA.

3.0 METHODS, ASSUMPTIONS, AND PROCEDURES

The memristor crossbar array technology has the potential to build a massively parallel, high-efficiency neuromorphic computing system [9]. When implemented in hardware, the required energy and chip area of the analog I/O circuits could dramatically impact the system's overall performance. When the size or the number of layers of MCAs increase, the required power and chip area of the I/O circuits could significantly bring down the overall computation efficiency.

Two methods are demonstrated in this project to minimize I/O circuits' power consumption and chip area. The first method is to employ the constant amplitude PWM signal as the computation variable. The second method is to separate the traditional I/O circuits into two parts: the global cell that is shared by the entire system and the individual local cell that serves each input and output. Both methods aim to reduce the I/O circuits' power consumption and chip area.

The general approach to develop these analog circuit blocks is (1) conceptualize the circuit topology, (2) SPICE simulation to validate the concept, (3) integrate the analog blocks with the rest of the circuit system, and (4) evaluate the proposed circuit system to recognize hand-written digits.

3.1. The Design Concept

Memristor crossbar array has been developed to carry out matrix vector multiplication in neuromorphic algorithms as the *Dot Product Engine* (DPE). By referring to the circuit diagram in Figure 1, the matrix-vector multiplication can be realized as follows: (1) represent the input vector as a set of input voltage signals to *wordlines* (WLs) of the memristor crossbar; (2) transfer the mathematical matrix to the resistance state of the memristors in the array; and (3) collect the currents at the *bitlines* (BLs) as the output vector of the matrix-vector multiplication operation.

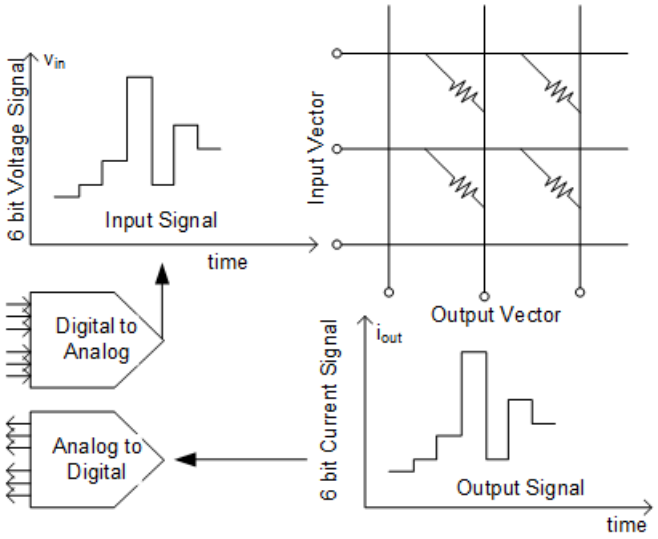

Figure 1: An overview of the MCA based DPE

3.1.1 PWM System

The implementation of the MCA based DPE has been extensively discussed in [8] and [10], as depicted in Figure 1. In [8, 10], *Amplitude Modulated* (AM) voltage signals are treated as the computation variable, as depicted in Figure 2 (b). A *Digital-to-Analog Convertor* (DAC) is required for each WL. When the size and number of MCAs increase, a large

number of DACs are needed. The power consumption and chip-area of arrays of DACs could impair the overall computation efficiency [8].

The PWM based signal, instead of traditional AM based signal, has been recently proposed [11]. As depicted in Figure 2 (a), the duty cycle of a constant amplitude voltage signal is used as the computation variable in the PWM circuit system. The PWM signals can be generated by a *phase-detector* (PD) with the reference clock and a shifted clock signal chosen by a digital selector. Thus, the input circuit for each WL only consists of a PD and a selector. Its power consumption and chip-area are much less than that of a DAC.

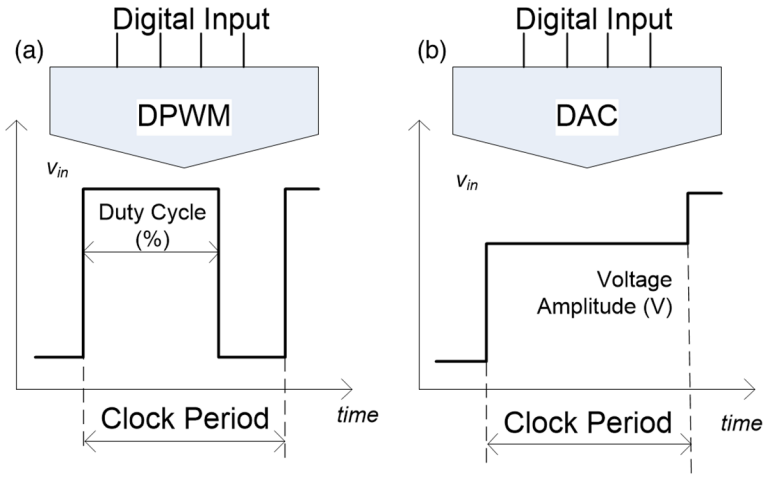

Figure 2: Illustration of the computation variable

3.1.2 Global and Local Circuit

To support massively parallel computation, arrays of repetitive I/O circuits are required to facilitate the neuromorphic computing MCAs. The power consumption and chip-area of I/O circuits could go up as the size and number of MCAs increase.

The global and local circuit system aims to take the power-hungry components out of the individual I/O circuit and form a global cell that is shared by the entire system. The practice is to further reduce the power and chip-area of repetitive local I/O circuit. At the input, the 16-tap DLL is the global cell. Each local cell selects one of the taps to produce the corresponding 4-bits PWM signals. At the output, 3 DACs form a global cell to produce 3 voltage references. Each local cell compares the output voltage to these voltage references using SAR logic circuit to get the 4-bits digital word. A MLP (Multi-Layer Perceptron) neural network is constructed to identify the hand-written MNIST dataset as shown in Figure 3. The 28x28 image is first reduced to 12x12. The MLP has two layers, 144x64 and 64x10.

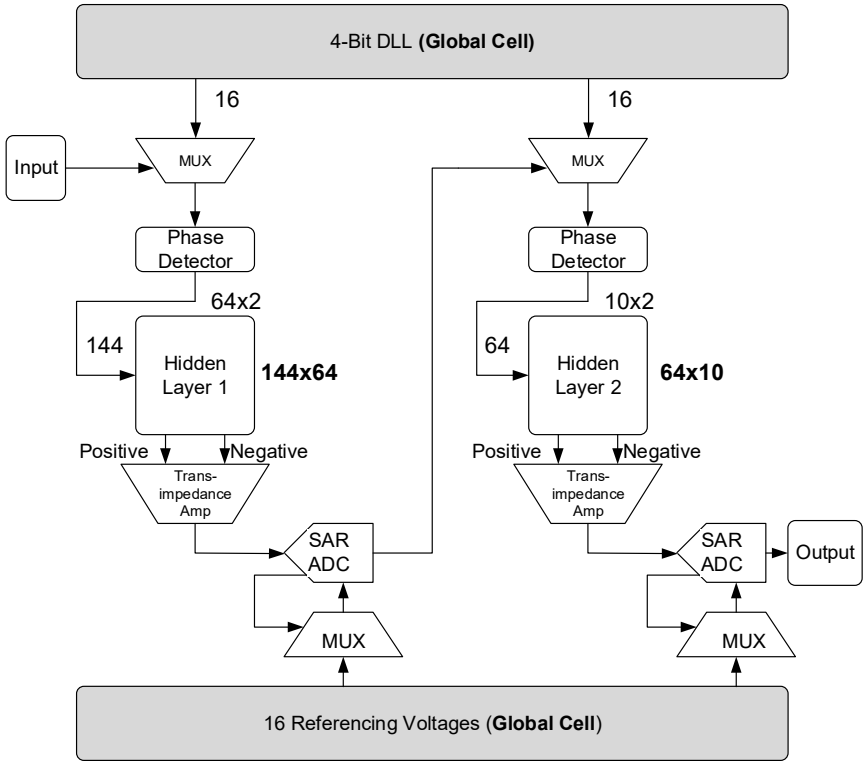

Figure 3: The schematic of global/local circuit system

3.2. Input Circuits

The PWM based DPE has a global DLL for every DPE in the neural networks, and a set of local cells as the write and read circuits for each row (word line) and column (bit line) of the MCA, as depicted in Figure 1. The global DLL produces a set of clock signals whose delay is evenly distributed within a clock period [12]. In this design, the DLL is designed to generate $2N-1$ delayed clocks, where N represents the number of bits that each PWM signal is going to represent. At the input of each row, one of the $2N-1$ delayed clocks, the m-th delayed clock, is selected by a N-to-1 multiplexer according to the N-bit digital input. By comparing the selected m-th clock to the reference clock, a PWM driver, which is made of two flipflops (similar to the phase-frequency detector in a regular *Phase-Locked Loop* (PLL) [13]), is able to produce a voltage signal whose duty cycle is proportional to $m/2^N$ with a constant amplitude (0 to constant Vin). When the constant amplitude PWM voltage signal is applied to each row of the MCA, the averaged current over one clock period at the output of each column is proportional to the overall conductance of the specific column (i.e., weights of the DPE) and the duty-cycle of all input voltage signals (i.e., input of the DPE). Based on the input voltage and the output current that are averaged over one clock period, the PWM based system is equivalent to the AM based counterpart.

3.2.1 Global Delay Locked Loop (DLL)

A traditional DLL is used to produce 2^4-1 evenly delayed clocks to support the 4-bit operation [12]. As depicted in Figure 14, the DLL consists of 8 differential CML (Current Mode Logic) delay stages whose delay time can be controlled by the control voltage (V_C), a phase-

frequency detector that detects the phase-frequency difference between the last delayed clock and the reference clock, a charge pump and a loop filter to provide v_C to control the delay time of every delay stage. The phase-frequency detector, and loop filter are widely used in PLLs [13].

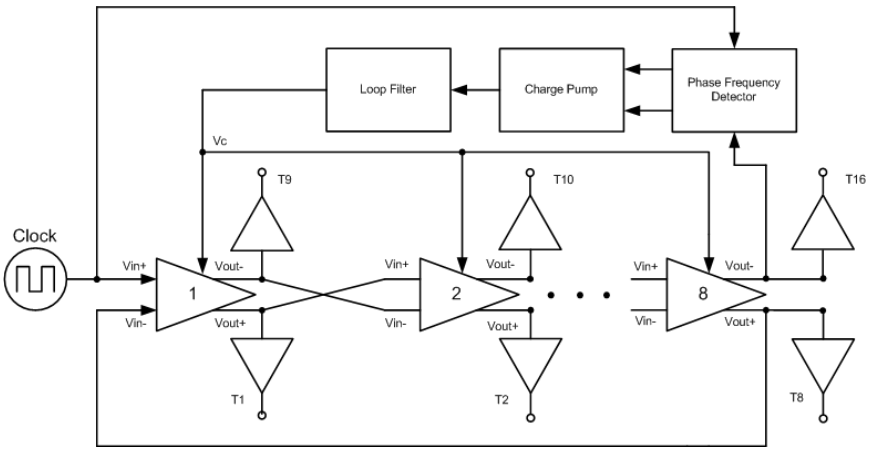

Figure 4: The circuit schematic of the DLL (Delay Locked Loop)

The DLL is implemented in Cadence. The DLL consumes about 850 uW when it is operating at 25 MHz, including the buffers at each tap. Figure 5 depicts the reference clock with the output taps. A delayed signal of 25MHz is shown on the 11th output tap.

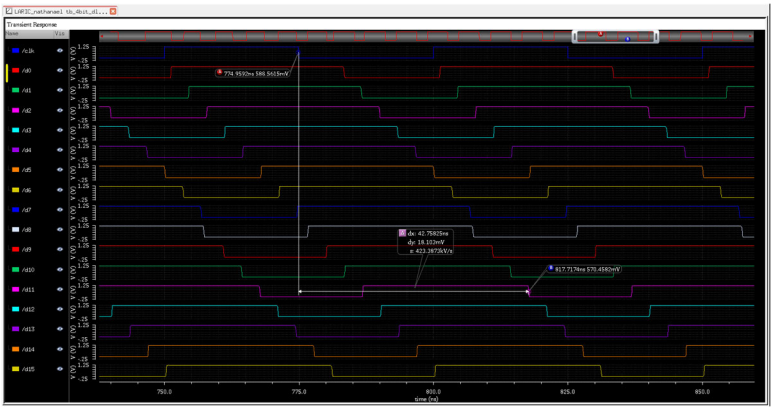

Figure 5: The transient response of a 4-Bit Delayed-Locked Loop

3.2.2 Local Input Circuit

An input circuit is needed for each row (word-line) of the MCA to translate the digital input to the PWM signal. In the described 4-bit PWM system, a 4-to-1 multiplexer is used to select one of $2^4 - 1$ delayed clocks according to the 4-bit digital input, as depicted in Figure 6 and Figure 7. In the circuit implementation, a PWM driver that is made of two D-type flip-flops is used to generate a signal whose duty cycle is proportional to the digital input. A digital buffer is added to the output of the *phase detector* (PD) so that it is able to drive a large-size MCA with many parallel-connected memristors. In this implementation, the input circuit consists of a 4-to-1 MUX, two D-flipflops and a digital buffer.

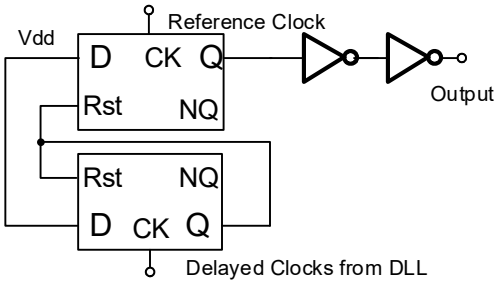

Figure 6: The schematic of the PWM generator

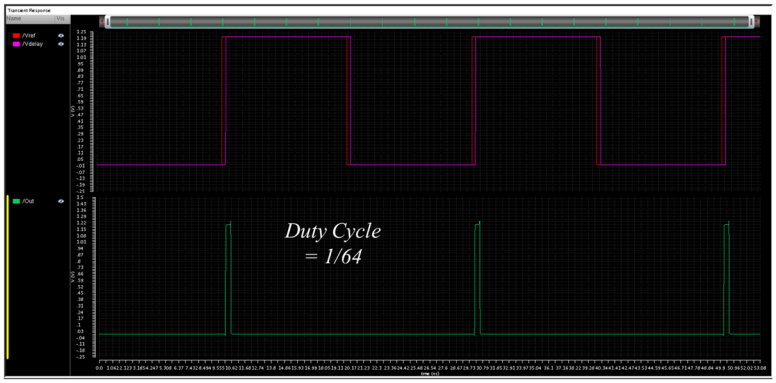

Figure 7: PWM signal produced by the PWM generator

3.3. Output Circuit

The output circuit converts an array of analog input waveforms into an array of digital words, as depicted in Figure 8. The output circuit consists of two circuitry blocks: global and local cell. The global cell provides the array static reference voltages. If the resolution of the integrated *Successive Approximation Register Analog-to-Digital Convertor* (SAR-ADC) array is N bit, the number of the static reference voltages is $2^{N/2}$-1. These reference voltages are shared with each individual local cell. The local cell consists of $2^{N/2}$-1 comparators, an encoder, a *Successive Approximation Register* (SAR) logic, $2^{N/2}$ equal value resistors and a selector. The integrated SAR-ADC array has two consecutive operating steps in time. The input analog voltage is first compared to $2^{N/2}$-1 reference voltages by $2^{N/2}$-1 comparators. The outputs of the comparators generate the first half of the total bits by the encoder, like a traditional Flash ADC. The outputs of the comparators are also fed into a SAR logic to select the 2 adjacent reference voltages that are close to the analog input, like a traditional SAR ADC. The 2nd sets of the reference voltages can be produced by applying the 2 selected adjacent reference voltages to $2^{N/2}$ equal value resistors, like a resistive voltage divider. The input analog voltage is then compared to the 2nd set of $2^{N/2}-1$ reference voltages using the same comparators. The encoder then generates the second half of the total bits. The local cell consists of comparators and digital logic circuit without any operational amplifiers and capacitors. The power consumption and the chip-area of the local cell can be easily minimized. Comparing to traditional SAR ADC, the power-hungry DAC, which is responsible for generating the reference voltages, is shared by the

entire ADC array. Thus, the power efficiency and the chip area of the integrated SAR-ADC array can be minimized by the described global-local arrangement.

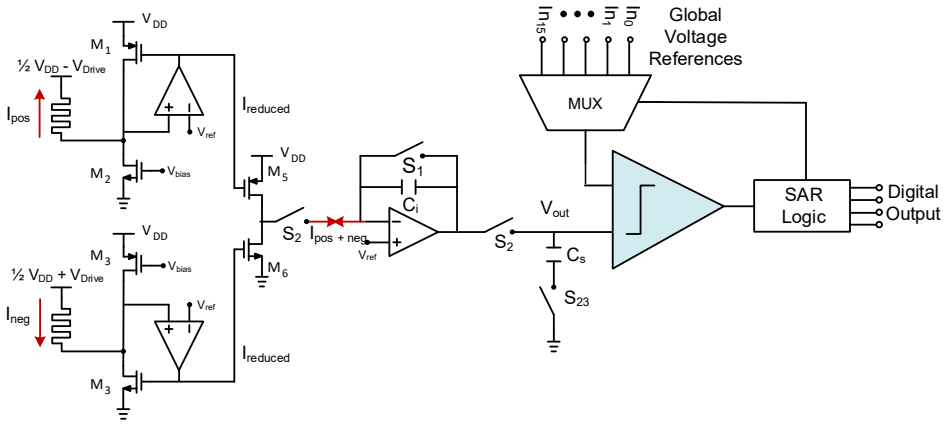

Figure 8: The schematic of the output circuits

3.3.1 Voltage References (Global Cell)

A current-steering DAC is used to generate voltage references, as depicted in Figure 9. In a current-steering DAC, each bit turns on or off a branch of a binary weighted current. As depicted in Figure 9, a constant gate voltage is applied to a set of NMOS with the weighted channel width [14]. Consequently, each branch produces a weighted current. By controlling the ON or OFF of each weighted branch current, the sum of the output current (I_{DAC}) is proportional to the control register. A cascode current mirror is used to duplicate the same amount of I_{DAC} to be applied to an operational amplifier, which is used to convert the input current from the cascoded current mirror and produce an output reference voltage.

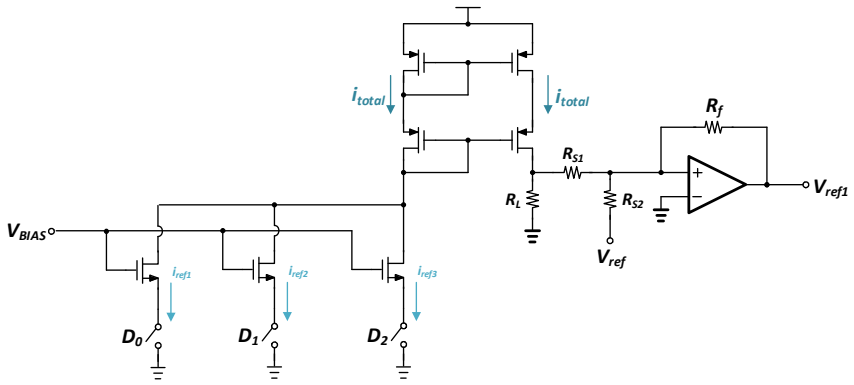

Figure 9: The schematic of voltage references

3.3.2 Amplifier and Sample-and-Hold (Local Cell)

3.3.2.1 Current Amplifier (Local Cell)

The current amplifier receives the current readout from the memristor crossbar array and reduces the current. The current amplifier, shown in Figure 8, is an active feedback current

mirror current amplifier. The operational amplifier creates negative feedback to pin the voltage of the input current node to the reference voltage, and to control the current flow of the current mirror. The current mirror of the circuit is constructed with M1 and M5. The input current to output current ratio is determined by the current mirror sizing between M1 and M5. The current mirror of M2 and M6 amplifies the current from the MCA that represents the negative weights.

3.3.2.2 Current Integrator (Local Cell)

The current integrator averages the current over a fixed period and convert the averaged current into a voltage level. The current integrator is depicted in Figure 8. Once switch S_1 is turned off and both S_2 and S_{23} are turned on, the capacitor (Ci) will accumulate the charge and the operational amplifier convert the accumulated charge into an output voltage (V_{out}).

3.3.2.3 Sample and Hold Circuit (Local Cell)

The sample-and-hold circuit samples the voltage output of the current integrator and holds the voltage for ADC readout, as depicted in Figure 8. The passive bottom-plate sample and hold circuit is used here. S_{23} is turned off slightly earlier than both S_2 switches. Once the S3 is turn off, the charge at the cap can only be affected by the V_{in}. After S_{23} and two S_2 are turned off, the voltage held at C_S will be isolated from any charge injection coming from the ground and the V_{in}. The held voltage of C_s is passed to the ADC. Then, S_1, both S_2 and S_{23} are turned on so that Vin and the voltage held at C_S are all reset to $V_{DD}/2$.

3.3.2.4 Output Amplifier and Sample and Hold Circuit Implementation

The output amplifier and sample and hold circuit is implemented in Cadence as depicted in Figure 10. The combination of the current amplifier, current integrator, and passive sample-and-hold circuit. Figure 11 depicts the simulation result of the full circuit simulation with 250μA input current over 40ns. The first stage is the current integration. V_{Int} charges up linearly over time. Sample voltage V_S follows V_{int}. The second stage is hold. V_S maintains the voltage and V_{int} resets. The last stage is reset. Both V_{int} and V_S are reset back to half of V_{DD}. Figure 12 depicts the transimpedance of the circuit with 40ns integration time. The transimpedance of the circuit is $2.29V/mA$, with the integrating capacitor of 600fF.

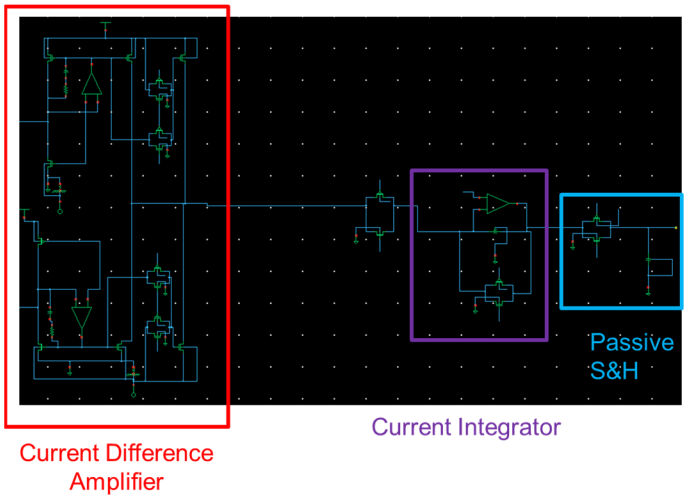

Figure 10: The schematic of output amplifier and sample and hold circuit

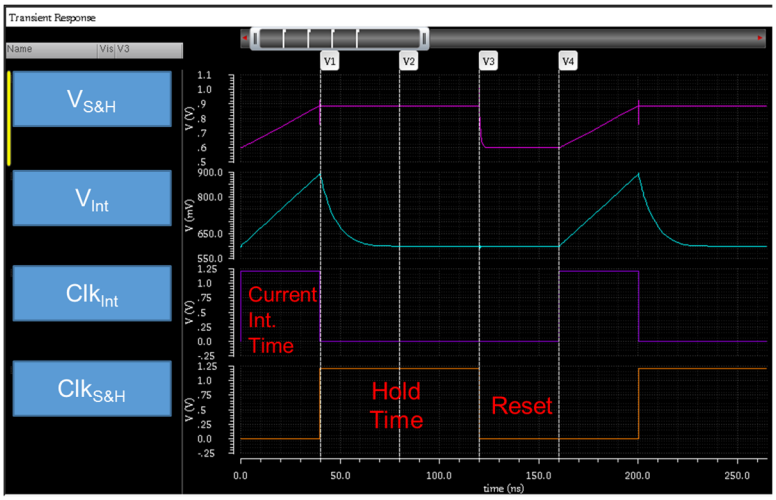

Figure 11: The transient simulation results of output amplifier and sample-and-hold circuit

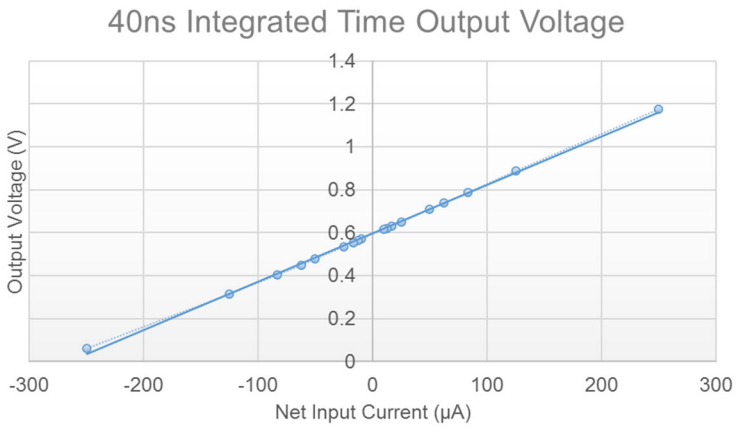

Figure 12: The output hold voltage versus net output current

3.3.3 Integrated SAR-ADC (Local Cell)

The operating principle of the proposed 4-bit ADC array circuit is illustrated in Figure 13. The global block is responsible for generating 3 ($=2^2-1$) reference voltages that are equally spaced inside of the ADC input window. The 3 reference voltages will be used by every local cell to produce the first 2-bit. 2 adjacent reference voltages from the 3 reference voltages and the ADC input start and stop voltages will be selected by the local cell to produce another 3 equally spaced reference voltages between the 2 selected adjacent reference voltages. Each local cell has 3 comparators. The comparator operates at two time-consecutive steps as depicted in Figure 13(a) and (b).

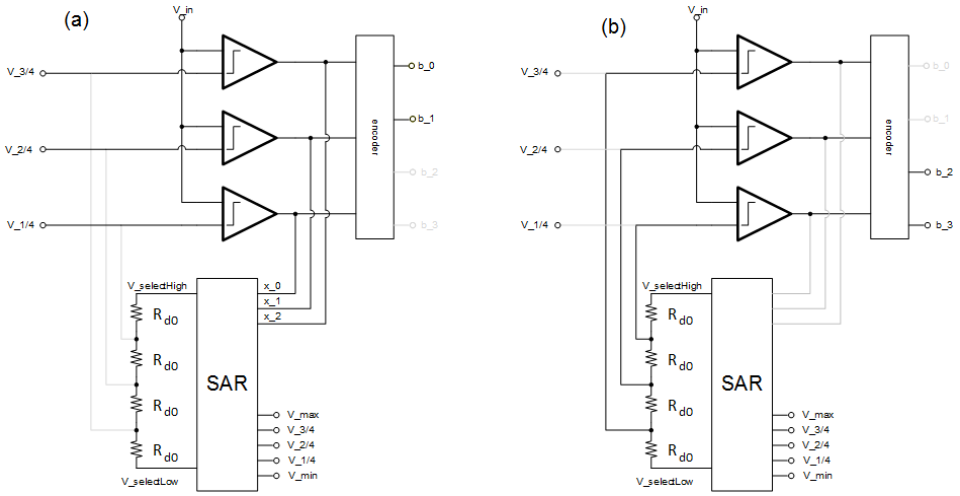

Figure 13: Operating principle and schematic of described Integrated ADC array

The logic operation has two steps:

- Step 1. The analog input voltage is compared to 3 reference voltages generated by the global cell. The 3 reference voltages are equally spaced in the ADC input window. The output of the 3 comparators is fed into an encoder to produce the first 2 bit, like a traditional Flash ADC, as shown in Figure 13(a). The output of the 3 comparators is also fed into a logic circuit and a selector. The logic circuit with a selector chooses the 2 reference adjacent voltages that are higher and lower than the analog input voltage, as V_selectHigh and V_selectLow as shown in Figure 13 (a). When the selected 2 reference voltages are applied to 4 equal value resistors, the 2nd set of 3 equally spaced reference voltages between the 2 selected adjacent are generated.

- Step 2. The analog input voltage is compared to the 2nd set of 3 reference voltages generated by the 4 equal-value resistors in the local cell. The output of the 3 comparator produces the last 2-bits as shown in Figure 13(b).

The local cell has 3 comparators, a SAR logic, a selector and 4 equal value resistors, as depicted in Figure 14. The local cell doesn't need capacitor that takes large amount of chip

area. Thus, the local cell's power consumption and chip area can be minimized. Comparing to the array made of individual ADCs, the described integrated ADC array's power efficiency can be significantly improved when the size of the array increases.

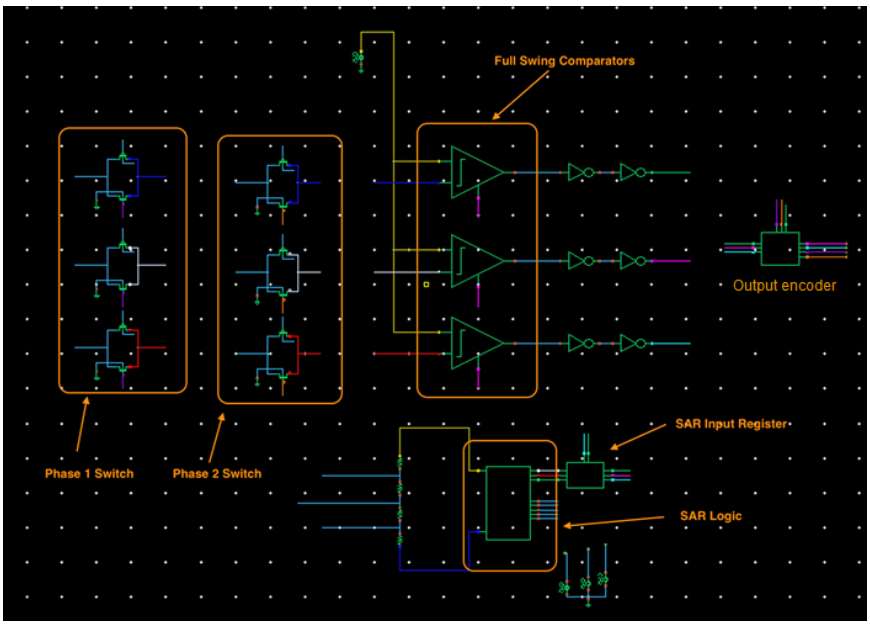

Figure 14: Circuit diagram of the SAR-Flash ADC in Cadence Virtuoso

The SAR-Flash ADC consists of comparator, encoder and SAR logic. The comparator and encoder follow the traditional approach [15]. The SAR logic has to be modified to facilitate this specific operation. A transient simulation is carried out to validate the design with the input voltage is 600 mV. The simulation results are shown in Figure 15. In Step 1, the output of the comparator is (0, 1, 1) as expected. In Step 2, the selected voltage of the SAR logic is 747 mV (750mV as designed) and 503 mV (500mV as designed). The new set of reference voltages are 564 mV (562.5 mV as designed), 625 mV (625 mV as designed) and 685 mV (687.5 mV as designed). Since all the reference voltages are corrected, the comparator outputs (1, 0, 0) are also as expected. The designed SAR-Flash ADC runs at 12.5 MHz. The power consumption is ~330 mW during the conversion. Most of the power is consumed by 3 comparators. The design does not contain any operational amplifiers and capacitors, thus, the power consumption and the required chip area can be minimized.

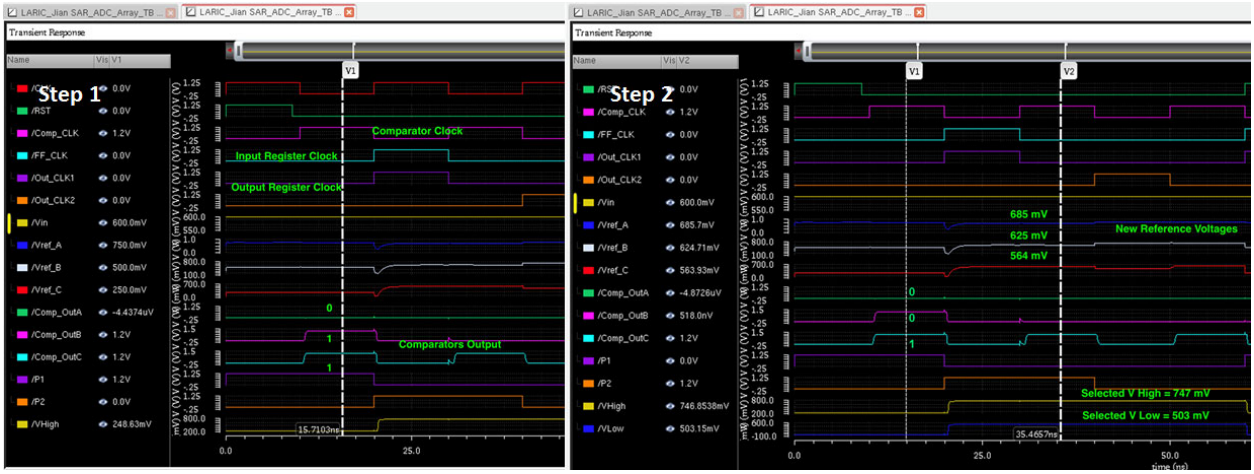

Figure 15: The transient simulation

4.0 RESULTS AND DISCUSSION

4.1. System-level Simulation

A two-layer Multi-Layered Perceptron (MLP) was trained in MATLAB to classify the MNIST dataset, as shown in Figure 316. The MLP was designed to process full-sized MNIST images (28×28) pixels and reduced sized MNIST images (12×12) pixels. The MLP networks each featured a single hidden layer with 64 rectified linear unit (RELU) neurons. Each MLP network output layer contained 10 SoftMax neurons, where each output neuron represents the probability of the input image being that class (digit).

The MATLAB OCEAN script generator is used to quickly and efficiently generate Cadence OCEAN scripts based on declared simulation parameters. An OCEAN script is used to control Cadence's simulator parameters and environment such as run time, simulator (Spectre, APS, etc..), result directory, analysis, design variables, but most importantly allows results to be saved into various files. The software system is shown in Figure 16.

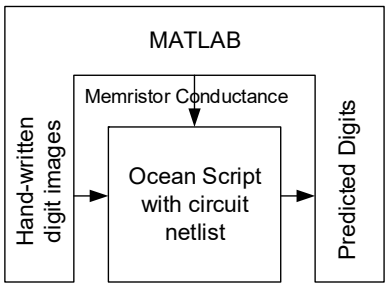

Figure 16: The verification system

The described PWM duty-cycles can be generated using a global delay lock loop (DLL) and local DPEs, as seen in [16]. As described in [16], a global n bit DLL generates $2^n - 1$ evenly spaced delay clocks within a clock period. The DPE circuit in [16] is designed from

a 2^n-to-1 multiplexer and two flipflops. The multiplexer is used to select a delay clock from the DLL, and the two flipflops generate a duty-cycle that is proportional to the phase difference between a reference clock and the selected delay clock.

To generate the duty-cycles for the modified PWM method, a slight modification to the traditional PWM DPE, Figure 17(a) is required. The modified DPE shown in Figure 17(b), requires the delay taps from the DLL to be connect to the 2^n-to-1 multiplexer in a new sequence. The output of the multiplexer and reference clock swaps inputs to the flipflops. The modified DPE output has the same duty-cycle D, however, it is delayed by $1 - D$. The power consumption for the modified PWM and Traditional PWM is the same. The modified PWM circuit does not require additional hardware as shown in Figure 17.

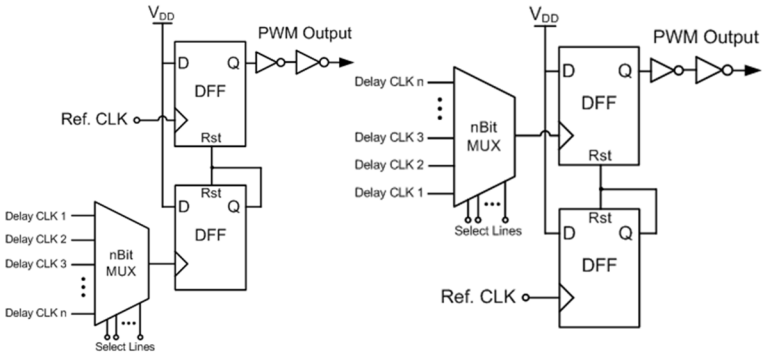

Figure 17: Circuit implementation of the traditional PWM and the modified PWM

A trained MCA for the MNIST dataset was used to evaluate traditional and modified PWM based DPEs. The size of the MCA is 144×64, where 144 represents the number of pixels in a 12×12 MNIST image and 64 represents the number of hidden neurons. The weights of the MCA have 3-bit precision, and the DPEs, (traditional and modified) have 4-bit precision, operate at 25MHz, and have an output voltage swing of 100mV. Sample 12×12 PWM based MNIST images were fed into the MCA by traditional and modified DPEs and the output current was monitored. As depicted in Figures (5)-(9), the modified PWM method distributed the input PWM signals over the entire 40nS period, whereas the traditional PWM method distributed the inputs over $\frac{2^n-1}{2^n} \approx 93.75\%$ of the period. The initial current spike that occurs at time 0 to approximately $\frac{1}{2^n}$ of the period in traditional PWM was eliminated in the modified PWM method. By implementing the modified PWM method, we reduce the current range by an average of 58.65% of the sampled bitlines, as shown in Table I. The overall testing accuracy is 96.66%.

Table I: MCA output current range comparison between traditional and modified PWM as the input variable

Digit	MCA Bitline #	Trad. PWM MCA Output Current Range [uA]	Mod. PWM MCA Ouput Current Range [uA]	Difference [%]
0	1	327.5	147.94	- 54.82
0	13	311.58	96.04	- 69.17
0	54	109.96	53.28	- 51.54
2	1	345.95	159.5	- 53.87
2	13	304.33	129.34	- 57.49
2	54	119.89	32.49	- 72.89
4	1	322.76	218.08	- 32.43
4	13	177.49	79.68	- 55.1029
4	54	80.18	30.76	- 61.63
7	1	200.08	77.56	- 61.23
7	13	115.19	37.99	- 67.013
7	54	70.25	13.63	- 80.59
9	1	424.49	271.45	- 36.04
9	13	254.65	101.29	- 60.22
9	15	100.03	34.29	- 65.72

4.2. System-level Testing Board Development

A system level printed circuit board (PCB) is built to test out 32×32 MCA as shown in Figure 18. The design of the PCB is depicted in Figure 19. A computer is used to first initialize the communication between the MATLAB and the microcontroller. The input for each row is sent from the MATLAB to the microcontroller. The microcontroller controls elements on the board to (1) turn on the relays and MCA selectors, (2) send the input to the transmitter, (3) sample the output at each column at the same time exactly, (4) turns off the relays and MCA selectors, and (5) send data from the receivers back to MATLAB.

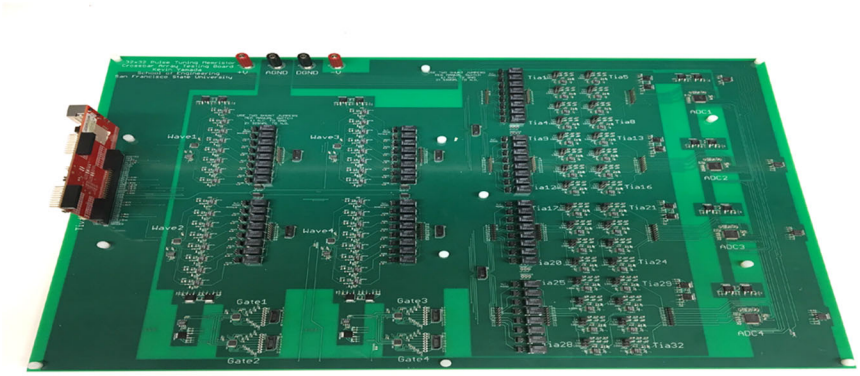

Figure 18: The printed circuit board for testing memristor crossbar array

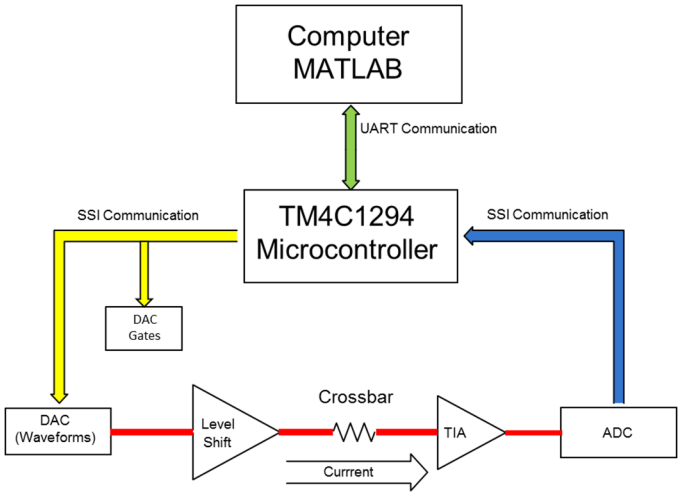

Figure 19: The system of the PCB

The transmitter has four 8-channel DACs with 32 level-shifting operational amplifiers to achieve 32 bipolar input waveforms. At the output of each DAC, a low-pass filter is used to smooths the overshoots and switching noise as shown in Figure 20. The input waveform can be defined from −2.5V to +2.5V.

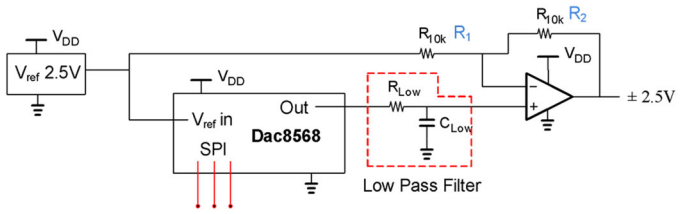

Figure 20: The schematic of the PCB transmitter

The receiver consists of *trans-impedance amplifier* (TIA) and ADC. 32 TIAs that connected to MCA's 32 columns convert input current to output voltage. The 32 output voltages are sampled by four 8 channel ADCs simultaneously. The four ADCs are linked in a daisy chained configuration allowing for simultaneous sampling between the range of 0V to 5V

The TIA and ADC both feature good linear trends with minimal error. The 1K ohm feedback resistor generates the least amount of error, while the 10k and 100k ohm feedback resistor has an average error of approximately 8.2%, as shown in Figure 21. The error is mainly from the test resistor and the small errors from each stage within the system.

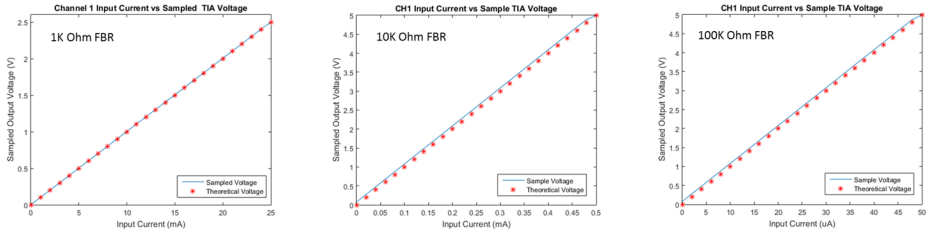

Figure 21: Measured output voltage from ADC versus the input current extracted based on the DAC input

5.0 CONCLUSIONS

In this project, we investigated and implemented the global *delay-locked loop* (DLL) and local *phase detector* (PD) to generate the PWM signals, and the active current mirror, the *trans-impedance amplifier* (TIA), and the integrated *successive approximation register analog-to-digital convertor* (SAR-ADC) to scale, measure and digitize the current from the MCA. We also develop a *printed circuit board* (PCB) to evaluate recently developed MCA.

6.0 REFERENCES

[1] D. Floreano and C. Mattiussi, *Bio-Inspired Artificial Intelligence*. MIT Press, 2008.

[2] H. Shayani, P. J. Bentley, and A. M. Tyrrell, "Hardware Implementation of a Bio-plausible Neuron Model for Evolution and Growth of Spiking Neural Networks on FPGA," in *Adaptive Hardware and Systems, 2008. AHS '08. NASA/ESA Conference on*, 22-25 June 2008 2008, pp. 236-243, doi: 10.1109/AHS.2008.13.

[3] D. B. Strukov, G. S. Snider, D. R. Stewart, and R. S. Williams, "The missing memristor found," *Nature,* vol. 453, no. 7191, pp. 80-83, May 1st 2008, doi: http://dx.doi.org/10.1038/nature06932.

[4] L. Beiye, H. Miao, L. Hai, C. Yiran, and X. Chun, "Bio-inspired ultra lower-power neuromorphic computing engine for embedded systems," in *Hardware/Software Codesign and System Synthesis (CODES+ISSS), 2013 International Conference on*, Sept. 29 2013-Oct. 4 2013 2013, pp. 1-1, doi: 10.1109/CODES-ISSS.2013.6659010.

[5] Z. Wang *et al.*, "Fully memristive neural networks for pattern classification with unsupervised learning," *Nature Electronics,* vol. 1, no. 2, pp. 137-145, 2018/02/01 2018, doi: 10.1038/s41928-018-0023-2.

[6] W. Yi *et al.*, "Feedback write scheme for memristive switching devices," (in English), *Appl. Phys. A,* vol. 102, no. 4, pp. 973-982, 2011/03/01 2011, doi: 10.1007/s00339-011-6279-2.

[7] W. Wang, T. T. Jing, and B. Butcher, "FPGA based on integration of memristors and CMOS devices," in *Circuits and Systems (ISCAS), Proceedings of 2010 IEEE International Symposium on*, May 30 2010-June 2 2010 2010, pp. 1963-1966, doi: 10.1109/ISCAS.2010.5537010.

[8] A. Shafiee *et al.*, "ISAAC: A Convolutional Neural Network Accelerator with In-Situ Analog Arithmetic in Crossbars," in *2016 ACM/IEEE 43rd Annual International Symposium on Computer Architecture (ISCA)*, 2016, pp. 14-26.

[9] M. Hu, H. Li, Q. Wu, and G. S. Rose, "Hardware realization of BSB recall function using memristor crossbar arrays," in *Design Automation Conference (DAC), 2012 49th ACM/EDAC/IEEE*, 3-7 June 2012 2012, pp. 498-503.

[10] M. Hu, J. P. Strachan, L. Zhiyong, R. Stanley, and Williams, "Dot-product engine as computing memory to accelerate machine learning algorithms," in *2016 17th International Symposium on Quality Electronic Design (ISQED)*, 15-16 March 2016 2016, pp. 374-379, doi: 10.1109/ISQED.2016.7479230.

[11] H. Jiang *et al.*, "Pulse-Width Modulation based Dot-Product Engine for Neuromorphic Computing System using Memristor Crossbar Array," presented at the 2018 IEEE International Symposium on Circuits and Systems (ISCAS), 2018. [Online]. Available: https://doi.org/10.1109/ISCAS.2018.8351276.

[12] J. G. Maneatis, "Low-jitter process-independent DLL and PLL based on self-biased techniques," *IEEE Journal of Solid-State Circuits,* vol. 31, no. 11, pp. 1723-1732, 1996, doi: 10.1109/JSSC.1996.542317.

[13] F. M. Gardner, 3rd, Ed. *Phaselock Techniques*. 2005.

[14] M. Gustavsson, J. J. Wikner, and N. Tan, Nick, *CMOS Data Converters for Communications*. Kluwer Academic Publishers, 2002.

[15] S. Babayan-Mashhadi and R. Lotfi, "Analysis and Design of a Low-Voltage Low-Power Double-Tail Comparator," *Very Large Scale Integration (VLSI) Systems, IEEE Transactions on,* vol. 22, no. 2, pp. 343-352, 2014, doi: 10.1109/TVLSI.2013.2241799.

[16] H. Jiang *et al.*, "Pulse-Width Modulation based Dot-Product Engine for Neuromorphic Computing System using Memristor Crossbar Array," in *2018 IEEE International Symposium on Circuits and Systems (ISCAS)*, 27-30 May 2018 2018, pp. 1-4, doi: 10.1109/ISCAS.2018.8351276.

APPENDIX A – PUBLICATION

1) Hao Jiang, Kevin Yamada, Zizhe Ren, Thomas Kwok, Fu Luo, Qing Yang, Xiaorong Zhang, J. Joshua Yang, Qiangfei Xia, Yiran Chen, Hai Li, Qing Wu, and Mark Barnell, "Pulse-Width Modulation based Dot-Product Engine for Neuromorphic Computing System using Memristor Crossbar Array," in 2018 IEEE International Symposium on Circuits and Systems (ISCAS), 27-30 May 2018, pp. 1-4

APPENDIX B – PRESENTATIONS

1) Meeting name: IEEE International Symposium on Circuits and Systems (ISCAS) 2018
Purpose: Public conference
Location: Florence, Italy
Date: May 2018
Attendees from this project: Hao Jiang
Presentation: "Pulse-Width Modulation based Dot-Product Engine for Neuromorphic Computing System using Memristor Crossbar Array"

APPENDIX C – ABSTRACT

In this project, an innovative circuit system was designed and implemented. It consists of a global circuit cell that supports the entire system and many local cells that support massively parallel I/O of the memristor crossbar arrays (MCAs). The I/O circuits use the pulse width modulated (PWM) signals as the computation variable. Specifically, performers designed and implemented the global delay-locked loop (DLL) and local phase detector (PD) to generate the PWM signals, and the active current mirror, the trans-impedance amplifier (TIA), and the integrated successive approximation register analog-to-digital convertor (SAR-ADC) to scale, measure and digitize the current from the MCA. Performers also develop a printed circuit board (PCB) to evaluate MCA.

7.0 LIST OF SYMBOLES, ABBREVIATIONS, AND ACRONYMS

ADC	Analog-to-Digital Conversion
AFRL	Air Force Research Lab
AM	Amplitude Modulated
BL	Bitline
CML	Current Mode Logic
DAC	Digital-to-Analog Conversion
DLL	Delay Locked Loop
DPE	Dot Product Engine
MCA	Memristor Crossbar Array
MLP	Multi-Layer Perceptron
MNIST	Modified National Institute of Standards and Technology
PCB	Printed Circuit Board
PD	Phase Detector
PWM	Pulse Width Modulated
SAR	Successive Approximation Register
SAR-ADC	Successive Approximation Register Analog-to-Digital Convertor
SHA	Sample and Hold Amplifier
SPICE	Simulation Program with Integrated Circuit Emphasis,
TIA	Trans-Impedance Amplifier
WL	Wordline