Graphene Nanoribbon Based McCulloch-Pitts Neural Network

Dumitru, F. S.; Enachescu, M.; Antonescu, A. M.; Cucu-Laurenciu, N.; Cotofana, S. D.

DOI
10.1109/NANO61778.2024.10628801

Publication date
2024

Document Version
Final published version

Published in
2024 IEEE 24th International Conference on Nanotechnology, NANO 2024

2024 IEEE 24th International Conference on Nanotechnology (NANO)
July 8-11, 2024. Gijón, Spain

Graphene Nanoribbon based McCulloch-Pitts Neural Network

F.-S. Dumitru[1], M. Enachescu[2], A. M. Antonescu[3], N. Cucu-Laurenciu[4], S. D Cotofana[5]

***Abstract*— In the context of an artificial intelligence and machine learning landscape that is evolving at an unprecedented pace, we propose a low power, high-speed, mixed-signal graphene nanoribbon-based (GNR) McCulloch-Pitts neuron (MCPN) implementation featuring programmable synaptic weights and inhibitory inputs. By definition, a generic MCPN is comprised of two parts, a weighted summation element and a decision element, called a soma. Our summation element implementation uses three distinct non-rectangular GNR devices, biased under specific conditions, to fulfill the roles of current source, low-side and high-side switches. The programmable excitatory and inhibitory synapses were obtained leveraging GNR SRAM cells and logic gates, hence providing the flexibility needed by real-world applications. The decision element's threshold activation function was implemented using a chain of GNR inverter structures which manifest the function's characteristic in the analog domain. Modulation of the decision element's threshold is achieved indirectly by means of a configurable resistive load which is varied depending on the configuration stored in SRAM. Our benchmark results, obtained using a generic 5 by 5 pixel pattern recognition application, reveal that the GNR-based implementation achieves $3.5\times$ less power consumption, $20\times$ higher speed, while occupying $3\times$ less active area when compared to its FinFET analog circuit counterpart.**

I. Introduction

The human brain acts as an energy efficient, high-performance chip capable of comprehending, processing, and storing data. It features parallelism, plasticity, pattern recognition, learning, memory, and fault tolerance. Numerous attempts were made to create bio-inspired artificial neuromorphic systems with the aforestated properties [1].

In these *brain-chips*, billions of neurons serving as the core processing unit and the trillions of synapses that interconnect them require integration. Complex CMOS circuitry is mostly used in state-of-the-art neuromorphic architectures to build neurons and synapses [2]. However, their relatively high supply voltage and high-power consumption limit the architecture's scalability and integration density. Recently, CMOS neurons and nanoscale resistive (or memristive) synapses, or even resistive switching memory device-based neurons have been proposed due to their reconfigurability, state retention, and good scalability [3]. Nevertheless, they are subject to variability-induced unpredictable behavior, which could potentially destabilize the neuromorphic system.

Graphene has been identified as a high potential material for nano-electronics due to its exceptional characteristics such as ballistic transport, flexibility, and bio-compatibility [4]. These appealing attributes have led to the development of graphene-based Boolean logic gates [5] and implementations of spiking neurons and synapses [6–9]. Specifically, work in [5] has compared GNR-based logic gates (GNR-L) with 7 nm FinFET CMOS counterparts and proved via simulation results that GNR-L could achieve 2 orders of magnitude lower power consumption, $6\times$ smaller propagation delay, and 2 orders of magnitude smaller active area.

The basic building blocks of neural networks are artificial neurons and synapses. These serve as essential processing units and act as communication links between neurons. Hence, the first step was designing a low-voltage, low-area, 5-bit GNR-based current digital to analog converter (DAC) in [10] to confirm the feasibility of the GNR-based programmable synapses necessary for a MCPN implementation.

In this paper, we design and implement the complete MCPN functionally using GNR structures. Firstly, we present a configurable GNR-based summation element's implementation, realized by integrating a 5-bit DAC's unit current sources, high-side and low-side switches, logic gates, load resistor, and SRAM cells. Secondly, we demonstrate the operation of the graphene MCPN realized by merging the GNR-based summation element with our GNR-based decision element. Finally, we demonstrate the detection capabilities of the proposed design by considering a one layer neural network consisting of 5 neurons configured to recognize the vowels "A", "E", "I", "O", and "U" (and variations of them) represented using a 5 by 5 black and white pixel matrix.

The rest of this paper has the following structure: Section II presents an overview of the MCPN model, related work on GNR circuits, and GNR simulation framework. Section III describes the GNR-based neuron implementation and the neural network used for bench-marking. Section IV presents the results of our simulations for the GNR and FinFET MCPN networks and compares their performance. Finally, the paper ends with concluding remarks in Section V.

This work was supported by a grant from the National Program for Research of the National Association of Technical Universities - GNAC ARUT 2023.

[1]F.-S. Dumitru is with Faculty of Electronics, Telecommunications, and Information Technology, National University of Science and Technology Politehnica Bucharest, Romania mailto:florin.dumitru2610@upb.ro

[2]M. Enachescu is with Faculty of Electronics, Telecommunications, and Information Technology, National University of Science and Technology Politehnica Bucharest, Romania mailto:m.enachescu@upb.ro

[3]A. M. Antonescu is with Faculty of Electronics, Telecommunications, and Information Technology, National University of Science and Technology Politehnica Bucharest, Romania mailto:aantonescu@upb.ro

[4]N. Cucu-Laurenciu, is with High Energy Physics Department, IMAPP, Radboud University, The Netherlands. mailto:n.cuculaurenciu@science.ru.nl

[5]Sorin Cotofana is with Faculty of Electrical Engineering, Mathematics and Computer Science, Delft University of Technology, 2600 AA Delft, The Netherlands mailto:S.D.Cotofana@tudelft.nl

2024 IEEE 24th International Conference on Nanotechnology (NANO) | 979-8-3503-8624-0/24/$31.00 ©2024 IEEE | DOI: 10.1109/NANO61778.2024.10628801

II. Background

A. McCulloch-Pitts neuron model

The McCulloch-Pitts neuron is considered one of the simplest neuron models, it simulates a binary decision-making unit, thus emulating the behavior of a biological neuron. Eq. 1 describes the output of a McCulloch-Pitts neuron model featuring only excitatory inputs, where $x_i = \{0, 1\}$ represent the values of the excitatory inputs, each input having a corresponding weight $w_i = \{0, 1\}$. If the sum of the weighted inputs, $x_i \cdot w_i$, exceeds the neuron's threshold, θ, then the output of the neuron will be 1, otherwise it will be 0.

$$f(x_1, \ldots, x_n) = \begin{cases} 1 & \text{if } \sum_{i=1}^{n} w_i \cdot x_i \geq \theta \\ 0 & \text{otherwise} \end{cases} \quad (1)$$

The complete McCulloch-Pitts neuron model expands Eq. 1 by accounting for the existence of inhibitory inputs and is described in Eq. 2, where $y_i = \{0, 1\}$ represent the inhibitory inputs. We remark that triggering even a single of the inhibitory inputs will force the neuron's output to 0.

$$\tilde{f}(x_1, \ldots, x_n; y_1, \ldots, y_m) = f(x_1, \ldots, x_n) \cdot \prod_{j=1}^{m} (1 - y_j) \quad (2)$$

We chose to implement the model from Eq. 2, because the model from Eq. 1, lacking inhibitory inputs, can be forced to falsely trigger by setting all excitatory inputs, x_i to 1.

B. GNR-based DAC for programmable synaptic weights

Configuring the geometrical parameters of the generic GNR-based device, illustrated in Figure 1, allows us to achieve analog behaviors ranging from current sources [10] to low-side and high-side analog switches, as shown later in this paper. Programmable synaptic weights can be implemented using the unit current source device, GNR_{ISRC}, from a GNR-based current DAC [10] featuring the geometrical parameters listed in Table I, and resulting in the topology shown in Figure 2a. The transfer and output characteristics of this current source device are similar to a CMOS device's saturation region [10].

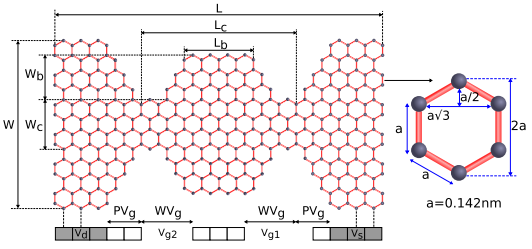


Fig. 1: GNR Geometry Description Parameters [11].

C. GNR-based SRAM cells and Boolean gates

A compact, power efficient, SRAM cell implementation is essential for the development of neural networks due to the inherent need to have synapses that are individually reconfigurable. Although the schematic of the GNR-based SRAM resembles its CMOS counterpart, it offers superior performance in several key areas. The GNR-based SRAM proposed in [5] achieves 3.6× smaller delay, 2 orders of magnitude less power consumption, and 1 order of magnitude smaller active area. Due to these advantages we chose GNR-based SRAM as the volatile medium for storing our network's configuration information.

With CMOS scaling to sub-10nm technology node, temperature variations significantly affect the reliability and performance of traditional logic gates [12]. The temperature effects on both CMOS and GNR gates have been thoroughly investigated in [13] and [11] and even in the worst-case condition they outperform their CMOS FinFET 7nm counterparts. For instance, the INV case showed 1.6× smaller delay and 185× less power consumption. These findings suggest that the GNR-based gates hold significant potential as basic building blocks for future reliable, low-power, carbon-based nano-electronics.

D. Simulation framework

The assessment of the McCulloch-Pitts neuron circuit (GNR-MCPN) is done through a hybrid simulation framework that integrates Cadence Spectre's mixed-signal environment with Matlab's parallel computing toolbox. This co-simulation methodology allows for the execution of SPICE-level circuit simulations in conjunction with precise atomistic-level GNR computations within Matlab. At the interface between these two environments we have a Verilog-A model which exchanges data between the SPICE simulation and the Matlab model.

To calculate the GNR's electronic transport properties, the tight-binding Hamiltonian approach is used to characterize the GNR structure. Additionally, we apply the Non-Equilibrium Green Function (NEGF) quantum transport model to solve the Schrödinger equation and the Landauer formalism, to compute the GNR's conductance [14] as:

$$G = \frac{q \cdot \int_{-\infty}^{+\infty} T(E) \cdot (f_0(E - \mu_1) - f_0(E - \mu_2)) dE}{h \cdot (V_d - V_s)}, \quad (3)$$

where q is the electron charge, h is the Planck constant, $T(E)$ is the transmission function, $f_0(E)$ is the Fermi-Dirac distribution function at temperature T, and $\mu_{1,2}$ denote the Fermi energy of the source and drain contacts.

The current through the GNR is illustrated by Eq. 4 as:

$$I(d, s) = V(d, s) \cdot G. \quad (4)$$

The quantum capacitance, C_q is calculated by integrating the product of the density of states, $DOS(E)$, and the thermal broadening function, $F_T(E)$, over all energy levels, where E represents the energy [15, 16]. This results in the quantum capacitance of our GNR device as a function of the distribution of the available energy states.

$$C_q = q^2 \int_{-\infty}^{+\infty} \text{DOS}(E) \cdot F_T(E - (\mu_1 - \mu_2))\, dE \quad (5)$$

By propagating the results of the conductance and quantum capacitance computations using our Verilog-A model we are able to accurately simulate the time dependent switching behavior occurring in the proposed McCulloch-Pitts neuron.

III. GNR-based McCulloch-Pitts neural network

A. GNR-based configurable synapses

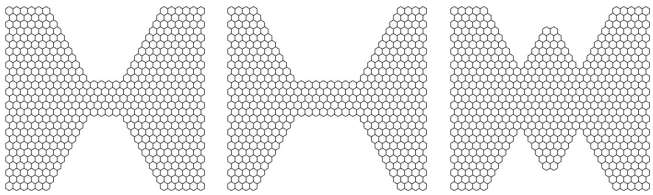
Each synapse of our MCPN implementation can find itself in one of the following three states: a) excitatory input with weight 0, b) excitatory input with weight 1, or c) inhibitory input. Therefore, storing their configuration will require two bits for each synapse. The first bit models the weight for cases a) and b), while the second bit models whether the synapse is an inhibitory input or not for case c).

The configuration data is initially loaded into SRAM at power on, an approach frequently seen in FPGA designs, and subsequently updated at any point when the neural network requires adjustment of its weights.

The proposed GNR-based configurable synapse is shown in the green rectangle in Figure 5, containing three GNR devices. From bottom to top these are the low-side switch, GNR_{DOWN}, the current source, GNR_{ISRC}, and the high-side switch, GNR_{UP}. The proposed GNR_{DOWN} and GNR_{UP} devices, have their geometries summarized in Table I, are represented in Figures 2b and 2c, and are characterized using their transfer and output characteristics in Figure 3.

The GNR_{UP} resembles a PMOS switch, exhibiting high resistance when the gate is driven to VDD and low resistance when it is driven to GND. Device GNR_{UP}'s gate is driven by a GNR-based 2 input NAND gate whose inputs are a) the pixel's state and b) synaptic weight stored in GNR-based SRAM array. If both the pixel's output signal and the weight value stored in SRAM are a logical '1', then the GNR_{UP} switch will be closed, otherwise it will be open.

Similarly, the GNR_{DOWN} device resembles an NMOS switch, exhibiting high resistance when the gate is driven to GND and low resistance when it is driven to VDD. Device GNR_{DOWN}'s gate is driven by a GNR-based 2 input AND gate whose inputs are a) the pixel's state and b) inhibitory configuration bit stored in a GNR-based SRAM array. If both the pixel's output signal and the weight value stored in SRAM are a logical '1', the GNR_{DOWN} switch will be closed, otherwise it will be open.



(a) GNR_{ISRC} (b) GNR_{DOWN} (c) GNR_{UP}

Fig. 2: Graphene-based device topologies

B. GNR-based summation element

A simplified, one pixel row, representation of the proposed GNR-based summation element and its FinFET counterpart is illustrated in Figure 4. The nominal voltage of the devices in the GNR-based circuit translates into a VDD of 0.2 V, while in the FinFET circuit's case the VDD is 0.7 V. In the GNR-based implementation each pixel is handled by the previously described group of three devices with analog roles and two logic gates, this arrangement is repeated 5 times to handle one 5 pixel row of the 5 by 5 pixel arbitrary symbol we are monitoring with our analog neuron. We remark that the FinFET counterpart shown in Figure 4b has a similar implementation, except for requiring a diode connected PMOS device biased at a constant current to act as reference for the current source PMOS devices. The GNR_{ISRC} device behaves similarly to a current source when the gate is driven to VDD and enters a high resistance state when the gate is driven to GND. The geometry of the GNR_{ISRC} device is represented in Figure 2a. The high resistance state exhibits a leakage current in the order of several tens of nA, leading us to insert the previously described GNR_{UP} device in series so as to reduce the off-state current of the GNR_{ISRC} device. By contrast with a CMOS current source, the GNR_{ISRC} device does not require a reference device for biasing and can be driven using a logic gate or hardwired to VDD.

The function of the summation element, when dealing only with excitatory inputs, can be implemented as the addition of the currents of each individual synapse of weight '1' with an active pixel. The total current resulting from this summation then flows through resistor R_{LOAD} and determines the value of the voltage of net $SOMMA_{OUT}$.

However, if any of the synapses are configured to be inhibitory and their pixel becomes active, they will shunt the R_{LOAD} resistor and prevent net $SOMMA_{OUT}$ from exceeding the threshold of the activation function, as the threshold is calibrated to only be reached when all the pixels of the symbol match. Thus, any shunting of the R_{LOAD} resistor prevents neuron the output going high.

The simplicity of this analog approach takes advantage of the proven high-speed GNRs and will translate into a fast response time MCPN. By contrast, other GNR-based spiking neurons implementation report times in the range of ms [6, 8, 9] before triggering spikes.

TABLE I: Dimensions of GNR-based Neuron Structures

	(W,L)	(W_c,L_c)	(W_b, L_b)	(P_{V_G}, W_{V_G})
GNR_{ISRC}	$(41, 27\sqrt{3})$	$(8, 4\sqrt{3})$	$(0, 0)$	$(2\sqrt{3}, 6\sqrt{3})$
GNR_{UP}	$(41, 27\sqrt{3})$	$(14, 8\sqrt{3})$	$(9, 2\sqrt{3})$	$(12\sqrt{3}, 6\sqrt{3})$
GNR_{DOWN}	$(41, 27\sqrt{3})$	$(8, 8\sqrt{3})$	$(0, 0)$	$(3\sqrt{3}, 6\sqrt{3})$

C. GNR-based activation function circuit

In our analog implementation we use the structure shown in Figure 5b), resembling a chain of inverters, to achieve the threshold activation function. Ideally, the threshold function is a step function, where the output is 1 if the weighted sum exceeds a threshold, θ, and 0 otherwise. We see that the input signal (red) and the output signal (blue) are in phase and measure a delay of approximately 4 ps.

The ability to trigger the MCPN for symbols comprised of different numbers of active pixels necessitates a configurable activation function threshold, θ. Since the activation function element's implementation shown in Figure 5b, has a fixed threshold, configurability must be achieved indirectly.

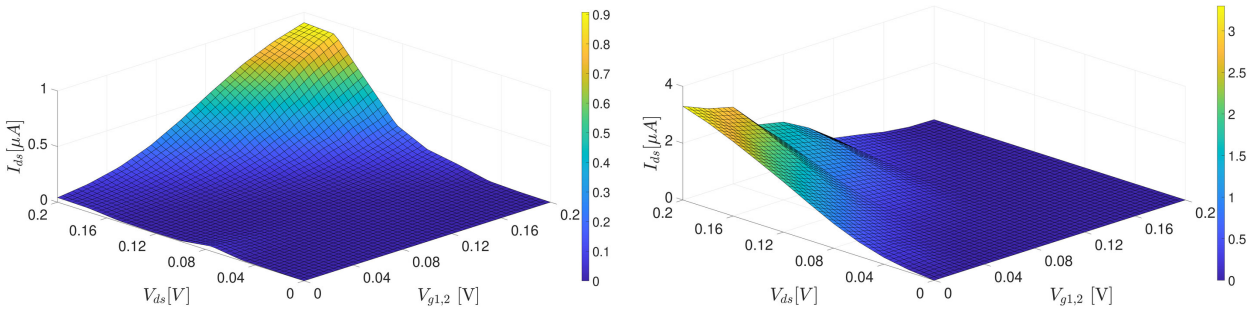


Fig. 3: Transfer and output characteristics for GNR_{DOWN} (left) and GNR_{UP} (right) devices

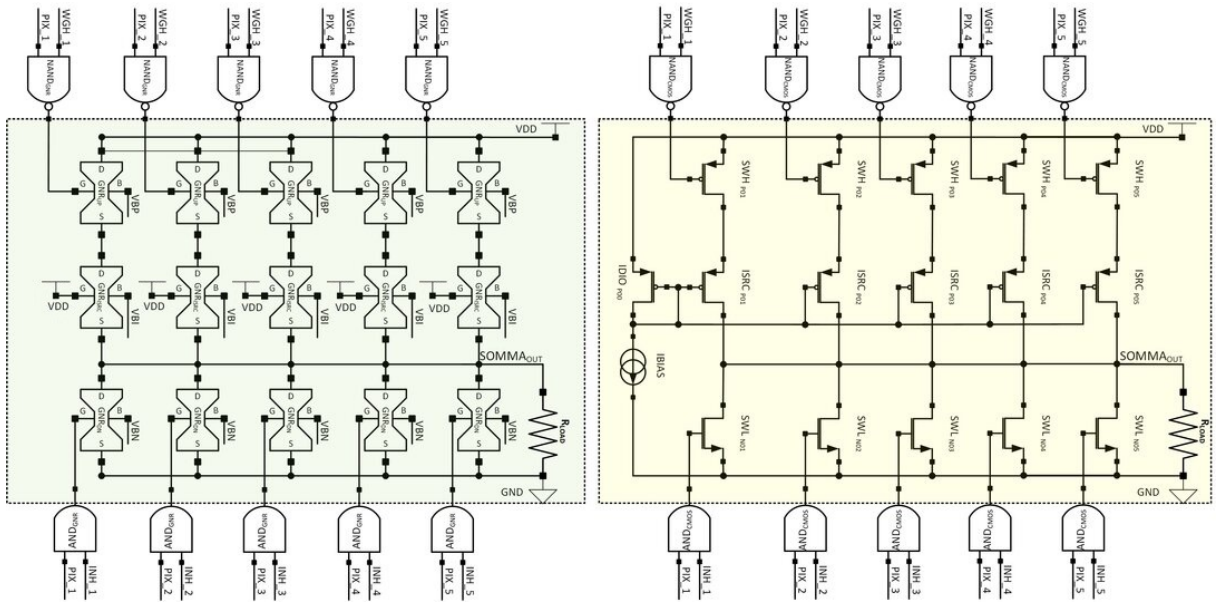


Fig. 4: Analog summation element implementation handling a single row of 5 pixels a) GNR (left) b) FinFET (right)

Noting that the neuron triggers when $SOMMA_{OUT} = I_{TOT} \cdot R_{LOAD}$ is equal to θ, we use the fact that modulating R_{LOAD} is equivalent with modulating θ to achieve a configurable threshold. The configurable R_{LOAD} is implemented by reusing the thermometric decoder from [10] and applying it to a string of resistors. A GNR_{DOWN} switch is inserted between GND and each intermediate net of the resistive string, resulting in a linearly configurable R_{LOAD}.

D. GNR-based neuron implementation

The complete GNR-MCPN implementation is shown in Figure 5. As each neuron features 25 synapses connected to the $SOMMA_{OUT}$ net and two SRAM cells, one for the excitatory weight bit and one for the inhibitory configuration bit, 50 SRAM cells are required for every neuron.

In Figure 5 it also becomes clear that the neuron will trigger once enough excitatory input synapses of weight 1 are activated such that their total current, I_{TOT} multiplied by R_{LOAD} exceeds the activation function's θ. At the same time, the neuron's output will be forced to zero if any of the inhibitory inputs are triggered.

While the McCulloch-Pitts neuron comes with clear limitations due to having only binary inputs and weights, for certain applications, such as black and white pattern detection it can be an adequate solution. On the other hand, we remark that having 5-bit synapse weights would translate into having to store 1550 bits for a single 25 synapse neuron.

E. GNR-based neural network

The proposed neural network shown in Figure 6 features one fully-connected layer of 5 neurons, each neuron being configured to detect the pattern associated with a vowel. Due to it being a fully-connected one layer network, our neurons perform this simple form of detection independently of each other. The recognition quality is inherently limited due to the simple topology of the neural network, but is sufficient to demonstrate the GNR-MCPN's ability to recognize characters.

IV. Simulation Results

To demonstrate the suitability of the proposed reconfigurable GNR-based McCulloch-Pitts neural network (GNR-MCP-NN) architecture for various application scenarios, and to validate the synaptic weight training approach effectiveness, we have tailored a GNR-MCPN for character recognition, mapped it on the proposed reconfigurable GNR-MCP-NN architecture, and evaluated it using SPICE simulations.

A. GNR-based single Neuron Behavior Evaluation

To validate the operation of the proposed GNR-MCPN, a SPICE simulation was run on a neuron with its synapses

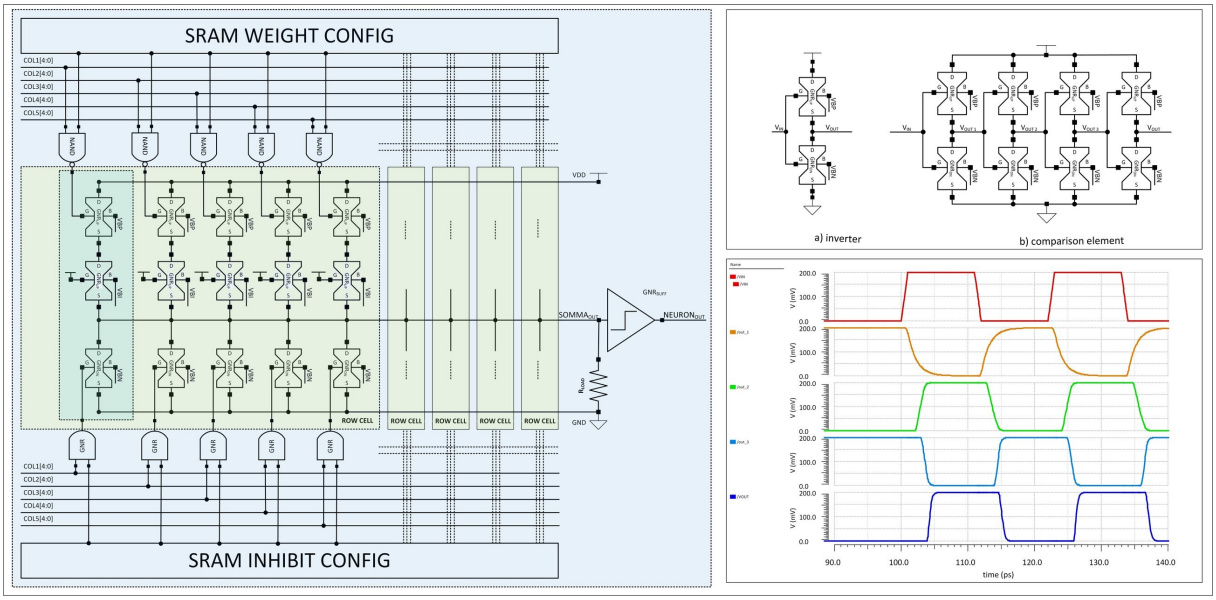


Fig. 5: GNR-based McCulloch-Pitts neuron

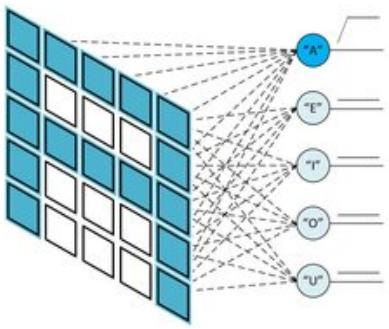


Fig. 6: McCulloch-Pitts neural network

configured to detect the vowel 'U', as shown in Figure 7.

The neuron's configuration bits are represented on the left side in Figure 7. Here, the pixel array labeled $Weight$ represents the excitatory inputs, where a $green$ pixel represents a weight of 1 and a $white$ pixel represents a weight of 0. Similarly, the pixel array labeled $Inhibitors$ represents the inhibitory inputs, where an $orange$ pixel represents a weight of 1 and a $white$ pixel represents a weight of 0.

The input pattern applied to the 5 by 5 array of pixels goes through 9 phases illustrated at the top of the figure. This pattern gradually increases the analog output of the summation element during the first 7 patterns, as seen on traces $Somma_{out}GNR$ and $Somma_{out}FinFET$. At the 6th pattern θ is exceeded, causing $Neuron_{out}GNR$ and $Neuron_{out}FiNFET$ to toggle high at 11 active pixels. The 7th and 9th patterns hit inhibitory pixels which cause the output of the summation element to get pulled down. The 8th pattern doesn't trigger any inhibitory pixel but is at the threshold of detection with only 10 active pixels. Therefore, the threshold for letter 'U' pattern is between 9 and 10 pixels of the 11 pixels that constitute the letter.

B. GNR vs FinFET neural network performance

We validate the behavior of the neural network shown in Figure 6 by applying an alternating series of vowels and intermediate symbols updated every 1 ns as shown in Figure 8. We note that while the GNR-based neuron's output toggles very fast with the change of the inputs, taking $8-20$ ps, the FinFET implementation takes $140-360$ ps. Therefore, in terms of speed the GNR-based variant is approximately $20\times$ faster.

Since static power dominates this current source approach and equal bias currents of the GNR-based and FinFET current sources were used, we achieve about $3.5\times$ better power efficiency due to the lower operating voltage.

In terms of the occupied area, we will separately evaluate the analog and the SRAM and digital parts of the implementation. The analog portion of the GNR-based design occupied approximately $3\times$ less active area, while the SRAM and control logic occupied approximately $9\times$ less active area.

V. Conclusions

In this paper, we investigated the potential of GNR devices to implement a configurable analog MCPN. The proposed MCPN was evaluated using both GNR-based and 7 nm FinFET technologies within a generic neural network symbol recognition application where it achieved $3.5\times$ less power consumption, $20\times$ higher speed, while occupying $3\times$ less active area when compared to the FinFET analog implementation, and $9\times$ less active area when comparing the SRAM and logic gates.

References

[1] Yoeri van de Burgt et al. "Organic electronics for neuromorphic computing". In: *Nature Electronics* 1 (2018), pp. 386–397.

[2] Paul A. Merolla et al. "A million spiking-neuron integrated circuit with a scalable communication network and interface". In: *Science* (2014), pp. 668–673.

[3] Dmitri B.. Strukov et al. "The missing memristor found". In: *Nature*. 2008, pp. 80–83.

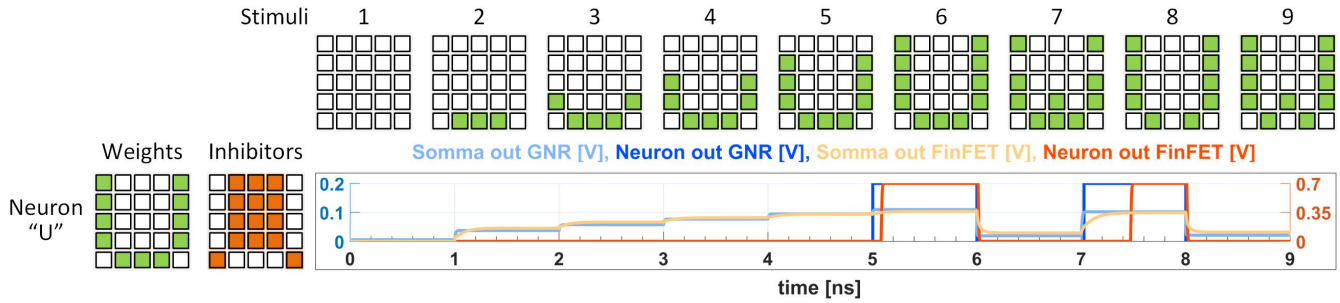


Fig. 7: Evaluate vowel 'U'

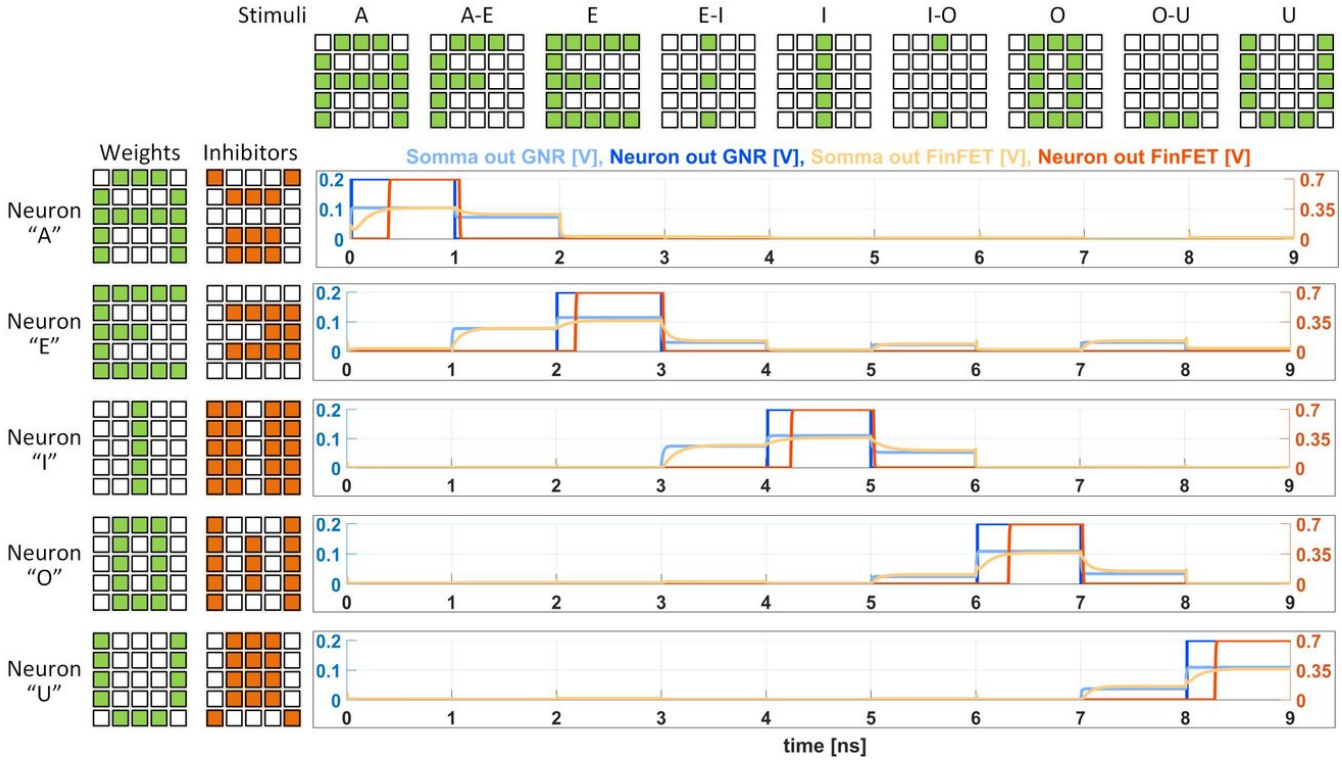


Fig. 8: Evaluate all vowels

[4] P. Avouris et al. "Graphene: synthesis and applications". In: *Materials Today* 15.3 (2012), pp. 86–97.

[5] Yande Jiang et al. "Graphene Nanoribbon Based Complementary Logic Gates and Circuits". In: *IEEE Transactions on Nanotechnology* 18 (2019), pp. 287–298.

[6] H. Wang et al. "Ultra-Compact, Entirely Graphene-Based Nonlinear Leaky Integrate-and-Fire Spiking Neuron". In: *IEEE ISCAS*. 2020, pp. 1–5.

[7] He Tian et al. "Graphene dynamic synapse with modulatable plasticity". In: *Nano letters* (2015).

[8] He Wang et al. "Graphene nanoribbon-based synapses with versatile plasticity". In: *NANOARCH*. 2019, p. 1.

[9] He Wang et al. "Compact graphene-based spiking neural network with unsupervised learning capabilities". In: *IEEE Open Journal of Nanotechnology* 1 (2020).

[10] F.-S. Dumitru et al. "Graphene Nanoribbons Based 5-Bit Digital-to-Analog Converter". In: *IEEE Transactions on Nanotechnology* (2021), pp. 248–254.

[11] Y. Jiang et al. "A Study of Graphene Nanoribbon-based Gate Performance Robustness under Temperature Variations". In: *IEEE-NANO*. 2020, pp. 62–66.

[12] Fábio G. Rossato G. da Silva et al. "Impact of Near-Threshold and Variability on 7nm FinFET XOR Circuits". In: *25th IEEE ICECS*. 2018, pp. 573–576.

[13] Mitsuhiko Igarashi et al. "Study of Local BTI Variation and its Impact on Logic Circuit and SRAM in 7 nm Fin-FET Process". In: *IEEE IRPS*. 2019, pp. 1–6.

[14] S. Datta. *Lessons From Nanoelectronics: A New Perspective On Transport.* World Sci. Publ., 2012.

[15] M. Mousavi-Khoshdel et al. "First-Principles Calculation of Quantum Capacitance of Codoped Graphenes as Supercapacitor Electrodes". In: *J. Phys. Chem. C* (2015), pp. 26290–26295.

[16] S. Datta. *Quantum Transport: Atom to Transistor.* Cambridge University Press, 2005.