

Artificial Intelligence Powered Memristive Average Value Calculator and its Real-time Applications

Sravan K. Vittapu
Department of ECE
Nalla Narasimha Reddy Education
Society's Group of Institutions-
Integrated Campus
Hyderabad, 500088, India

P Michael Preetam Raj
Department of Computer Science and
Engineering
Bharat Institute of Engineering and
Technology
Hyderabad, 501510, India,

N. Divya
Department of Computer Science and
Engineering
G. Narayanamma Institute of
Technology and Science
Hyderabad, 500104, India

Manu Gupta
Department of Electronics and
Computer Engineering
Sreenidhi Institute of Science and
Technology
Hyderabad, 501301, India

Pinapati Lakshmana Rao
Department of Computer Science and
Engineering
Koneru Lakshmaiah Education
Foundation (KLEF)
Vaddeswaram, Guntur, AP, 522502
India

Yallamati PrakasaRao
Department of Computer Science and
Engineering
Koneru Lakshmaiah Education
Foundation (KLEF)
Vaddeswaram, Guntur, AP, 522502
India

Abstract—Recently, disasters due to floods based on extreme rainfall have resulted in over 2 million deaths and US\$ 3.64 trillion loss of property over the last five decades globally. On the other side, researchers have been increasing interest in employing large language models (LLMs) to generate solutions for numerous real-time applications. This work proposes memristor-based electronic circuits for flood alert indication developed through LLM after training through an example. Initially, an average value calculator was designed to record the rainfall associated with four consecutive days. Importantly, a memristor-based electronic circuit was developed to provide extreme rainfall alert indication during heavy rain. Most importantly, the employment of LLM was advantageous in terms of ease and reduction in code complexity by 98.94%. Furthermore, the power consumed by our proposed circuit resulted in ~99.9% improvement, whereas the circuit complexity was reduced by ~99% when compared to their existing transistor-based counterparts.

Keywords—Artificial Intelligence, Disaster management, Memristor, Rainwater harvesting systems, Large Language Models.

I. INTRODUCTION (HEADING 1)

Recently, disasters due to extreme rainfall conditions have become more frequent in several parts of the world [1, 2]. In the past five decades, globally there have been over 2 million deaths and a huge loss of property above US\$ 3.64 trillion owing to floods [3]. Therefore, it is exceedingly essential to detect extreme rainfall, to avoid such natural disasters. To achieve this, transistor-based electronic circuits have been widely employed [4]. In this regard, researchers have developed an extreme rainfall detection system in which large volumes of data were gathered and analysed to generate the very important excess rainfall alert [4]. However, such work is disadvantageous since it involves storing large quantities of information and processing such data which ultimately results in undesirably high computational complexity and the requirement of a large amount of the associated hardware. In other instances, researchers have monitored water level variations in rivers using sensors and microcontrollers to

design a flood alert system using wireless sensor networks [5, 6]. This study aimed to record and send a message to the people before the calamity. Nevertheless, this work necessitated a huge amount of data that had to be amassed at even intervals of time which ultimately resulted in increased circuit complexity. The most significant drawback of the above-mentioned systems is that all these systems are based on transistor technology which has reached its termination stage. Recently transistor technology abided with discommodos due to grading of device attributes such as undesirable quantum mechanical burrowing and curt channel consequences, that mitigate device functionality and heightened power intake [7–12]. The above-named drawbacks bespeak the cease of the transistor epoch, and attempts were consecrated to explicate substitute technologies [13–15]. On the other side, the memristor has been attracting widespread interest due to its multilevel logic (92 states within a single device), high network density (~40% improvement), faster execution (within 1 ns), ameliorated power dissipation (~0.01% consumed), ability to recollect its state history, and employment of existing CMOS fabrication process without the need for newer fabrication systems [16–18]. Consequently, it is highly recommended to switch towards memristive systems, which surmounts the shortcomings consociated with transistor-based systems [19]. It is important to mention that a memristor is essentially a resistor with memory; its resistance (memristance) depends upon the amount of current that passes through it [15]. To overcome the disadvantages of transistor technology and develop advanced rainfall alert and extreme rainfall indication systems, it is very much important to propose memristor-based rainfall measurement and extreme rainfall alert systems for futuristic applications. To develop such systems, it is extremely essential to develop its fundamental version of the operational circuit. One such circuit is the average value calculator, which can be extended to develop the memristor rainfall alert system. It is well known that, the traditional method of performing average value calculation is achieved by executing the addition operation followed by division of the total number of elements. Recently researchers have developed memristor-based adder [20] circuits which have exhibited significant

advantages compared to the widely used CMOS technology in terms of power consumption (54.7%) and circuit complexity (29.4%) [21]. In a different article, researchers have developed memristor-based adder circuits which exhibited 20% improvement in computational speed. Electronic circuits that perform all four basic arithmetic operations, namely addition, subtraction, multiplication, and division were developed using memristive technology [22]. These circuits provided for high data retention compared to their existing transistor-based counterparts. Several such memristor-based arithmetic circuits were reported in the literature which exhibited numerous advantages over the conventional transistor-based circuits [23–30]. Despite the above-mentioned advancements in memristive computational circuits, unfortunately, to this day no one has yet developed a memristor-based average value calculator circuit.

It is a well-known fact that rainwater is usually gathered, stored, and maintained in reservoirs for public use. Recently, researchers have developed numerous electronic circuits and systems to automatically perform such functionalities [31–34]. In [31] researchers have developed a non-contact type water measurement system using microcontrollers. In [32] river water level was monitored employing microcontrollers to provide flood indication. In [33] water levels were controlled and monitored in an overhead tank; logical functionalities were implemented through flip-flops. In [34] microcontroller-based automatic water pumping was achieved and various water levels were indicated using an ultrasonic sensor. Regardless of such advancements, it is noteworthy to mention that all these reservoir maintenance systems are based on transistor technology which possesses numerous disadvantages as mentioned earlier.

At present, researchers have been increasing interest in employing large language models (LLMs) to generate solutions for numerous real-time applications [35, 36]. The same is the situation in electronic circuit design [37, 38]. Nevertheless, the existing LLMs are unable to generate accurate netlists for circuit simulation applications [39]. In this work, a memristor-based average value calculator was designed which calculated the amount of direct current (DC) components present in a given signal. This circuit was further developed to record the amount of rainfall associated with four consecutive days by converting rainfall in centimetres to its corresponding resistive values. Interestingly, heavy rainfall was represented by a wider pulse, whereas lighter rainfall was represented by a narrow pulse. To develop an application for the maintenance of a reservoir, the inflow of water into the reservoir was represented by a positive pulse, whereas the outflow of water was indicated by a negative pulse where the width of the pulse represented the amount of water. Utilizing this approach memristor was employed to maintain the record of water available in the reservoir. Most importantly, to provide a very important extreme rainfall alert indication during heavy rainfall, a memristor-based electronic circuit was developed, and the same was implemented using simulation routes. The proposed memristor-based electronic circuits were developed through LLM after training with an example.

II. MATERIALS AND METHODS

The LLM ChatGPT based on the GPT-4 architecture was employed in this work. The LTspice XVII simulator was utilized to obtain the electronic behaviour of the memristive

circuits proposed in this work. The aimed device comprises of 10 nm TiO₂ thin film in between two platinum metal electrodes. This device was first proposed in 2008 by Hewlett-Packard Labs, which is useable in LTspice [40]. Initially, library files of this memristor were involved in the LTspice directory to construct the novel memristive logic circuits.

III. RESULTS AND DISCUSSIONS

The LLM was prompted to generate SPICE netlists. As shown in Figure 1, the AI, on its own, was incapable of generating the correct circuit as per the given prompt to generate a SPICE netlist for a 1 V DC source connected across a memristor. The circuit in Figure 1 was obtained when the generated netlist was reopened as the schematic version in the SPICE environment. However, when the LLM was trained through a simple example shown in Figure 2, it got trained and generated the required netlist. The generated netlist shown in Figure 3, when simulated in the SPICE environment was 100% accurate. Interestingly, although trained for only a DC voltage source, the LLM correctly interpreted the prompt and generated a SPICE netlist for a biased sinusoidal signal. As presented in Figure 4. In this manner, one could simulate all the circuits presented in this work through the aid of AI. These results provide promising rules for LLMs on futuristic circuit designing; the following SPICE simulations were employed to validate the proposed memristive circuits. Initially, piecewise linear (PWL) based pulse signals were applied to the memristor as shown in Figure 5(a), it was noticed that while the pulse was ON the memristance gradually increased and once the pulse was low the memristance was observed to be constant (Figure 5(b)). Interestingly, it was identified that the memristive variations remained uniform whenever the area under the pulse was equal, irrespective of the variations in the height or width of the pulse. This finding validated the usefulness of the memristor to estimate the DC component i.e., the average value of pulse-based signals. These results offer compelling evidence for real-time applications, one such instance is rainfall calculation. For example, let us consider that one has to measure the rainfall for four consecutive days. In such case, the width of the pulse (in Figure 5(b)) indicated the amount of rainfall i.e., a wider pulse showed heavy rainfall, whereas a narrow pulse pointed to low rainfall, and such pulses were represented using PWL signal. When these signals were applied, the memristance gradually got updated per the rainfall (as indicated in Figure 5(b)) i.e., whenever there was heavy rainfall memristance was increased by a correspondingly large amount, whereas, for the event of low rainfall memristance increased by only a small quantity.

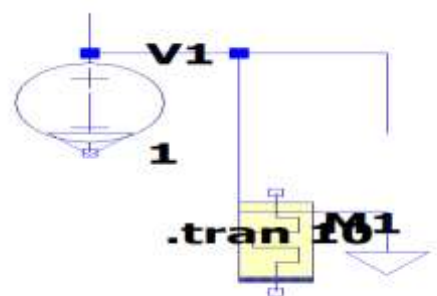


Figure 1: Schematic contents generated by circuit specifically untrained LLM. One can notice that the general training given to the LLMs was insufficient to generate the correct results for electronic circuit simulations.

```

Question:
1V=M
Answer:
Version 4
SHEET 1 880 680
WIRE 208 80 64 80
WIRE 64 112 64 80
WIRE 208 112 208 80
WIRE 64 224 64 192
WIRE 144 224 64 224
WIRE 208 224 208 192
WIRE 208 224 144 224
WIRE 144 256 144 224
FLAG 144 256 0
SYMBOL memristor 192 96 R0
SYMATTR InstName U1
SYMBOL voltage 64 96 R0
WINDOW 123 0 0 Left 0
WINDOW 39 0 0 Left 0
SYMATTR InstName V1
SYMATTR Value 1
TEXT 24 280 Left 2 !tran .7
Question:
2V=M
Answer:

```

Sure, I understand now. When you specify "1V=M," you mean you want a voltage source with a value of 1V connected across a memristor, represented in LTspice format. Here's the LTspice schematic for a 2V voltage source connected across a memristor:

Figure 2: LLM successfully trained through a simple SPICE netlist. The operation '=' represented the parallel connection between the voltage source and the memristor.

```

Version 4
SHEET 1 880 680
WIRE 208 80 64 80
WIRE 64 112 64 80
WIRE 208 112 208 80
WIRE 64 224 64 192
WIRE 144 224 64 224
WIRE 208 224 208 192
WIRE 208 224 144 224
WIRE 144 256 144 224
FLAG 144 256 0
SYMBOL memristor 192 96 R0
SYMATTR InstName U1
SYMBOL voltage 64 96 R0
WINDOW 123 0 0 Left 0
WINDOW 39 0 0 Left 0
SYMATTR InstName V1
SYMATTR Value 2
TEXT 24 280 Left 2 !.tran .7

```

Figure 3: SPICE netlist of a 1 V voltage source connected in parallel to a memristor, generated by LLM after training through an example. The content generated by LLM proved to be 100% accurate when simulated in the SPICE environment.

```

Question:
(1V+SINE(0,1))=M
Answer:

```

Here's the LTspice schematic for a voltage source with a 1V DC offset and an AC component given by a sine wave (amplitude 1V, offset 0V), connected across a memristor:

```

Version 4
SHEET 1 880 680
WIRE 208 80 64 80
WIRE 64 112 64 80
WIRE 208 112 208 80
WIRE 64 224 64 192
WIRE 144 224 64 224
WIRE 208 224 208 192
WIRE 208 224 144 224
WIRE 144 256 144 224
FLAG 144 256 0
SYMBOL memristor 192 96 R0
SYMATTR InstName U1
SYMBOL voltage 64 96 R0
WINDOW 123 0 0 Left 0
WINDOW 39 0 0 Left 0
SYMATTR InstName V1
SYMATTR Value SINE(1 1 50)
TEXT 24 280 Left 2 !.tran .7

```

Figure 4: SPICE netlist of a sinusoidal signal with DC offset generated by LLM. Although trained for only a DC voltage source, the LLM correctly interpreted the prompt and generated a biased SPICE netlist for a sinusoidal signal.

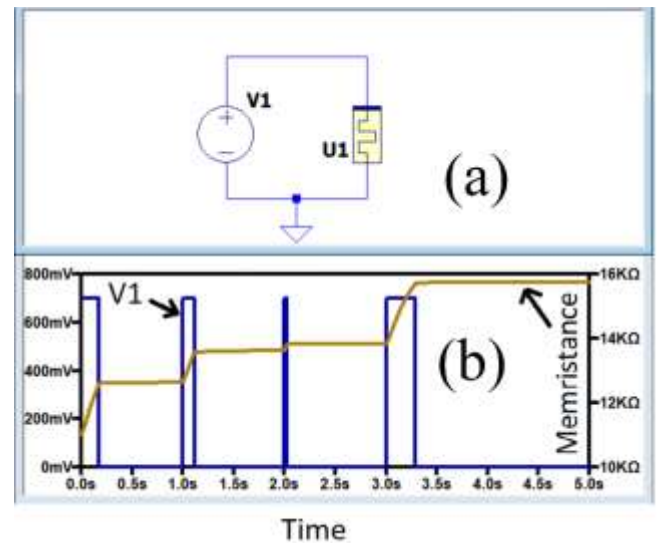


Figure 5. (a) Average signal calculator circuit and (b) its electronic characteristics. When pulse signals are given as input, the memristance increases along with the pulse, and when there is no pulse signal the memristance remains constant.

The mathematical relation for a continuum of memristive updates (from its initial value of 11 kΩ, when a DC source of 0.7 V is applied) is presented in Equation (1)

$$M = 11172 + (8058 * (t)) \quad (1)$$

where M represents memristance in Ω, and t denotes time in seconds. However, the relation for the memristive updates shown in Figure 5(b) is expressed based on its previous values as given in Equation (2) shown below.

$$M = (8058 * (t - tp)) + Mp \quad (2)$$

where Mp denotes the previous memristance in Ω , and tp denotes the time instance when a new pulse has been applied to update the memristance. For example, as depicted in Figure 5(b), the initial value of memristance is 11 k Ω . i.e., $tp = 0$, and $Mp = 11$ k Ω . Building upon this the memristance at time $t = 0.1616$ s was calculated as 12.53 k Ω and the same is reflected in the graph shown in Figure 5(b). Depending upon the relation in Equation (2) the update of memristance per the pulse signals is presented in Table 1. In this case, when a pulse signal of 170.3 ms was applied to the memristor, its memristance got updated to 12.6 k Ω from its initial value of 11 k Ω . Consequently, when a pulse signal of 110.2 ms was supplied to the memristor, its value was incremented to 13.6 k Ω from its previous value of 12.6 k Ω . Subsequently, the memristance was updated to 13.8 k Ω and 15.7 k Ω for the pulse signals of duration 21.5 ms and 285 ms, respectively.

Table 1. Memristive value updates in accordance with pulse signals

PULSE DURATION (MS)	MEMRISTANCE (K Ω)
170.3	12.6
110.2	13.6
21.5	13.8
285.1	15.7

To showcase the practical application, let us consider the case where rainwater is stored in dams; this same water gets supplied to people to meet their day-to-day water requirements. It is noteworthy to mention that in the case of a water dam, there is an inlet and an outlet water flow. To maintain the record of the amount of water available in the reservoir, the inlet flow was represented by a positive pulse such that a wider pulse represented heavy water flow whereas a narrow pulse represented lower water flow (Figure 6(a)). On the other side, the outlet water flow instances were represented by negative pulses as shown in Figure 6(a). Since memristors respond to the applied DC signal, it was employed to track the amount of water available in the water dam. It can be seen in Figure 6(b) that whenever there was a heavy inflow of water the memristance increased accordingly. However, when there was no inflow or outflow of water the memristance retained its value. Nevertheless, when there was an outflow of water the memristance got reduced accordingly. In this manner, the memristor maintained the record of the amount of water available in the reservoir. The mathematical expression for the memristive updates depicted in Figure 6(b) (negative pulse signals) is expressed based on its previous values as given in Equation (3)

$$M = (-16468 * (t - tp)) + Mp \quad (3)$$

where Mp is the previous memristance in Ω , and tp is the time instance when a new pulse has been applied to the memristor.

Relying on Equation (3) the memristive update in agreement with the pulse signals is given in Table 2. Here, when a positive pulse signal of 170.3 ms was applied to the memristor, its memristance was updated to 12.6 k Ω from its initial value of 11 k Ω . Consequently, when a negative pulse signal of 99.1 ms was supplied to the memristor, its value was

decreased to 11.7k Ω from its previous value of 12.6 k Ω . Subsequently, the memristance was updated to 12.7 k Ω , 12.9 k Ω , 12.1 k Ω , and 14.5 k Ω for the remaining pulse signals as exhibited in Table 2.

Table 2. Memristance updates for input pulse signals

PULSE DURATION (MS)	MEMRISTANCE (K Ω)
170.3 (positive pulse)	12.6
99.1 (negative pulse)	11.7
110.2 (positive pulse)	12.7
21.5 (positive pulse)	12.9
97.8 (negative pulse)	12.1
285.1 (positive pulse)	14.5

converting the rainfall to pulse signals (at the terminal rainfall-based pulse width modulation (RPWM)) and the other circuit for accepting those pulse signals as input to provide the extreme rainfall alert system. In the first subsequent, rainfall for four consecutive days was represented in terms of electronic resistances. These resistive values were obtained by measuring the resistance of the volume of rainwater gathered in a container. The dimensions of this container were such that its radius was 4.7 cm, with a height of 15 cm. The cross-sectional area of this cylindrical structure was 69.36 cm². Utilizing this arrangement, the rainfall measured for four consecutive days was 8.8 cm, 5.9 cm, 2.7 cm, and 12.6 cm respectively. As expected, heavy rain was represented by high resistance, whereas low rainfall was indicated by smaller resistance. The resistances observed these four days were such that the first-day resistance was 2.5 k Ω , whereas 1.7 k Ω for the second day. Consequently, third- and fourth-day resistances were 770 Ω and 3.6 k Ω respectively. Each of these resistances namely R3, R4, R5, and R6 were given as consecutive inputs to the circuit by employing single pole double throw (SPDT) switches U1, U3, and U4 through controlled timing-based V4, V5, and V6 pulse signals. The switching was such that in the first instance, 2.5 k Ω was transferred to the location in between the positive terminal of U1 and the ground. Consequently, 1.7 k Ω , 770 Ω , and 3.6 k Ω were passed on during the second, third, and fourth instances respectively. A voltage divider network was constructed which consisted of DC input voltage V1 of 1 V, resistance R1 of 100 k Ω , and the abovementioned resistance value switched into the circuit. Subsequently, the voltage across the newly switched resistor appeared at the positive terminal of the comparator U2. The operation of the voltage divider network was such that it gives an output voltage corresponding to the resistance received from the SPDT switches and the 100 k Ω resistor. Such obtained different voltage levels were compared with a sawtooth waveform through U2 to generate the RPWM pulse signals that followed the rainfall. Thus, the rainfall represented in the form of pulse signals was obtained at the terminal RPWM. The logic of using a sawtooth signal is that whenever the value of the sawtooth signal is lower than the voltage signal at the positive terminal of U2, the resultant signal is high; otherwise, the resultant waveform is in its low state, resulting in pulse signals. These pulse signals were scaled down 5 times through the employment of the dependent voltage source B1 in the second subsequent. This was because

a low voltage (~ 1 V) is sufficient for memristive write operations. On the second circuit, as shown in Figure 7(a), the $33\ \mu\text{F}$ capacitor C1 was employed to suppress the unwanted spike signals at the output of B1. It is noteworthy to mention that a very high value of capacitance leads to smoother waveform; however, the size of the capacitor will be large. On the other side smaller value of capacity will give improper smoothing. Therefore, an optimum value of $33\ \mu\text{F}$ was employed in this work. Furthermore, two memristors U8 and U10 were employed in which U8 updated its memristance following the rainfall-based pulses, whereas U10 was employed for isolating U8 and the voltage source B1 such that the voltage across the memristor is different from that of the source voltage.

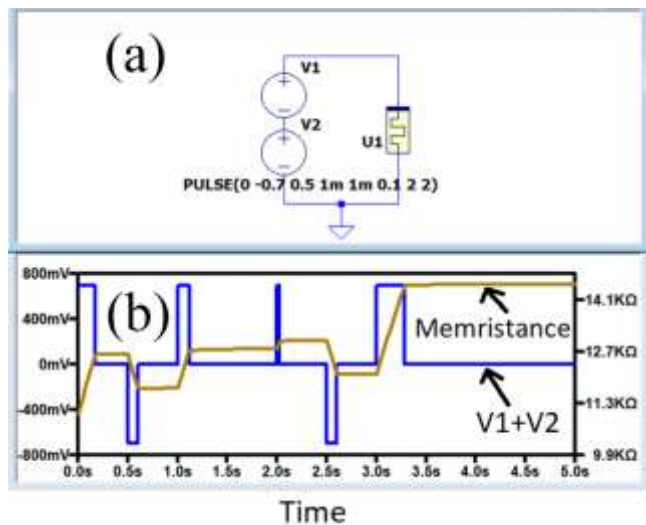


Figure 6. (a) Average signal calculator circuit with both positive and negative PWL signals. V1 was the same as that of the previous circuit, while V2 was connected to provide negative pulses. (b) The variations of memristance per that of the applied PWL signal.

The op-amp comparator U9 compares the voltage across U8 with the pre-set reference voltage which was set at V10 to 0.96 V. Based on the accumulation of the highest rainfall detected (here in this case it is the fourth day) an extreme rainfall alert signal was produced at the output terminal (LED QTLP690C) following the reference voltage. Therefore, as shown in Figure 7(b) whenever there was heavy rainfall (the end of the fourth day in our case) an alert signal was generated by this circuit. Thus, this circuit was used as an excess rainfall indicator.

When compared to the existing circuits (which employ thousands of transistors [4–6]) the proposed memristive circuits achieved $\sim 99\%$ improvement in circuit complexity. Importantly, the power consumed by our proposed circuit was $62.7\ \mu\text{W}$, resulting in $\sim 99.9\%$ improvement in power consumption.

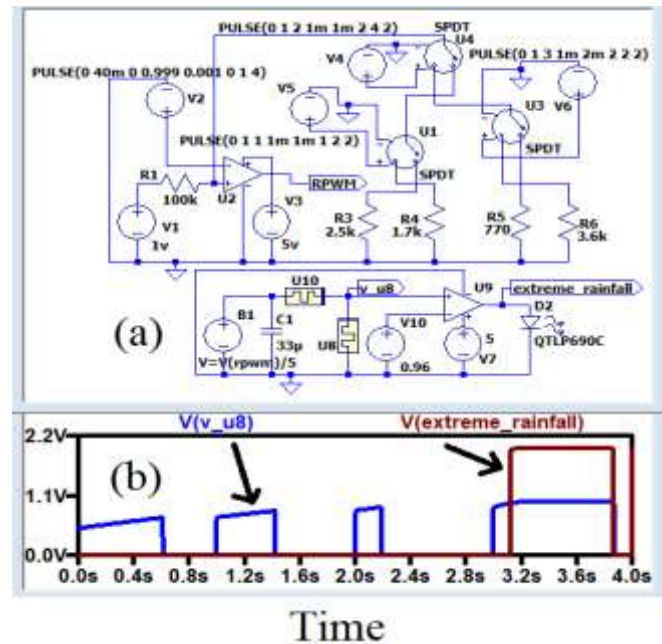


Figure 7. (a) Circuit for converting rainfall into pulse signals and providing extreme rainfall alerts. The resistance measured for the four days is converted into pulse signals utilising SPDT switches and the obtained signals are given as input to the behavioural voltage source to obtain the excess rainfall alert.

IV. CONCLUSION

In this work, memristor-based circuit simulations were performed through training LLM to generate a SPICE netlist. The memristive average value calculator was designed to calculate the average value present in a given signal. This circuit was further developed to record the amount of rainfall associated with four consecutive days. A memristor was employed to maintain the record of rainwater available in the reservoir. Importantly, in order to provide the very important extreme rainfall alert indication during heavy rainfall, a memristor based electronic circuit was developed and the same was implemented using simulation routes. When compared to the existing circuits the proposed circuits achieved $\sim 99\%$ improvement in circuit complexity. Importantly, the power consumed by our proposed circuit was $62.7\ \mu\text{W}$, resulting in $\sim 99.9\%$ betterment in power consumption. Most importantly, the employment of LLM was advantageous in terms of ease and reduction in code complexity by 98.94% . The ideas proposed in his work pave the way for futuristic automatic circuit designing through AI.

REFERENCES

- [1] Lahsen, M., & Ribot, J. (2022). Politics of attributing extreme events and disasters to climate change. *Wiley Interdisciplinary Reviews: Climate Change*, 13(1), e750. <https://doi.org/10.1002/WCC.750>
- [2] Dietze, M., & Ozturk, U. (2021). A flood of disaster response challenges. *Science*, 373(6561), 1317–1318. <https://doi.org/10.1126/SCIENCE.ABM0617/ASSET/F335B388-0C1B-40F5-A8C5-F5B0D567FC1E/ASSETS/SCIENCE.ABM0617.FP.PNG>
- [3] Proske, D. (2021). Estimation of the Global Health Burden of Structural Collapse. *Lecture Notes in Civil Engineering*, 153 LNCE, 327–340. https://doi.org/10.1007/978-3-030-73616-3_24
- [4] Mazzogio, P., Laio, F., Sandu, C., & Boccardo, P. (2019). Assessment of an Extreme Rainfall Detection System for Flood Prediction over Queensland (Australia). *Proceedings 2019*, Vol. 18, Page 1, 18(1), 1. <https://doi.org/10.3390/ECRS-3-06187>

- [5] Castillo-Effen, M., Quintela, D. H., Jordan, R., Westhoff, W., & Moreno, W. (2004). Wireless sensor networks for flash-flood alerting. *Proceedings of the IEEE International Caracas Conference on Devices, Circuits and Systems, ICCDCS*, 142–146. <https://doi.org/10.1109/ICDCS.2004.1393370>
- [6] Hughes, D., Ueyama, J., Mendiondo, E., Matthys, N., Horré, W., Michiels, S., ... Guan, S. U. (2011). A middleware platform to support river monitoring using wireless sensor networks. *Journal of the Brazilian Computer Society* 2011 17:2, 17(2), 85–102. <https://doi.org/10.1007/S13173-011-0029-3>
- [7] Young, K. K. (1989). Short-channel effect in fully depleted SOI MOSFETs. *IEEE Transactions on Electron Devices*, 36(2), 399–402. <https://doi.org/10.1109/16.19942>
- [8] Troutman, R. R. (1979). VLSI limitations from drain-induced barrier lowering. *IEEE Journal of Solid-State Circuits*, 14(2), 383–391. <https://doi.org/10.1109/JSSC.1979.1051189>
- [9] Ng, K. K., & Taylor, G. W. (1983). Effects of hot-carrier trapping in n- and p-channel MOSFETs. *IEEE Transactions on Electron Devices*, 30(8), 871–876. <https://doi.org/10.1109/T-ED.1983.21229>
- [10] Jeong, M., Narayanan, V., Singh, D., Topol, A., Chan, V., & Ren, Z. (2006). Transistor scaling with novel materials. *Materials Today*, 9(6), 26–31. [https://doi.org/10.1016/S1369-7021\(06\)71540-1](https://doi.org/10.1016/S1369-7021(06)71540-1)
- [11] Suzuki, K., Tanaka, T., Tosaka, Y., Horie, H., & Arimoto, Y. (1993). Scaling theory for double-gate SOI MOSFET's. *IEEE Transactions on Electron Devices*, 40(12), 2326–2329. <https://doi.org/10.1109/16.249482>
- [12] Lo, S.-H., Buchanan, D. A., Taur, Y., & Wang, W. (1997). Quantum-mechanical modeling of electron tunneling current from the inversion layer of ultra-thin-oxide nMOSFET's. *IEEE Electron Device Letters*, 18(5), 209–211. <https://doi.org/10.1109/55.568766>
- [13] Theis, T. N., & Wong, H.-S. P. (2017). The End of Moore's Law: A New Beginning for Information Technology. *Computing in Science & Engineering*, 19(2), 41–50. <https://doi.org/10.1109/MCSE.2017.29>
- [14] Williams, R. S. (2017). What's Next? [The end of Moore's law]. *Computing in Science & Engineering*, 19(2), 7–13. <https://doi.org/10.1109/MCSE.2017.31>
- [15] Strukov, D. B., Snider, G. S., Stewart, D. R., & Williams, R. S. (2008). The missing memristor found. *Nature*, 453(7191), 80–83. <https://doi.org/10.1038/nature06932>
- [16] Knag, P., Lu, W., & Zhang, Z. (2014). A Native Stochastic Computing Architecture Enabled by Memristors. *IEEE Transactions on Nanotechnology*, 13(2), 283–293. <https://doi.org/10.1109/TNANO.2014.2300342>
- [17] Yilmaz, Y., & Mazumder, P. (2017). A Drift-Tolerant Read/Write Scheme for Multilevel Memristor Memory. *IEEE Transactions on Nanotechnology*, 16(6), 1016–1027. <https://doi.org/10.1109/TNANO.2017.2741504>
- [18] Zidan, M. A., Fahmy, H. A. H., Hussain, M. M., & Salama, K. N. (2013). Memristor-based memory: The sneak paths problem and solutions. *Microelectronics Journal*, 44(2), 176–183. <https://doi.org/10.1016/j.mejo.2012.10.001>
- [19] Jeong, D. S., Kim, K. M., Kim, S., Choi, B. J., & Hwang, C. S. (2016). Memristors for Energy-Efficient New Computing Paradigms. *Advanced Electronic Materials*, 2(9), 1600090. <https://doi.org/10.1002/aelm.201600090>
- [20] Revanna, N., & Swartzlander, E. E. (2017). Memristor based adder circuit design. *Conference Record - Asilomar Conference on Signals, Systems and Computers*, 162–166. <https://doi.org/10.1109/ACSSC.2016.7869016>
- [21] Khalid, M., Mukhtar, S., Siddique, M. J., & Ahmed, S. F. (2019). Memristor Based Full Adder Circuit for Better Performance. *Transactions on Electrical and Electronic Materials* 2019 20:5, 20(5), 403–410. <https://doi.org/10.1007/S42341-019-00135-5>
- [22] Merrikh-Bayat, F., & Shouraki, S. B. (2011). Memristor-based circuits for performing basic arithmetic operations. *Procedia Computer Science*, 3, 128–132. <https://doi.org/10.1016/J.PROCS.2010.12.022>
- [23] El-Slehdar, A. A., Fouad, A. H., & Radwan, A. G. (2015). Memristor based N-bits redundant binary adder. *Microelectronics Journal*, 46(3), 207–213. <https://doi.org/10.1016/J.MEJO.2014.12.005>
- [24] Fouad, A. H., & Radwan, A. G. (2019). Memristor-based quinary half adder. *AEU - International Journal of Electronics and Communications*, 98, 123–130. <https://doi.org/10.1016/j.aeue.2018.10.033>
- [25] El-Slehdar, A. A., Fouad, A. H., & Radwan, A. G. (2013). Memristor-based balanced ternary adder. 2013 25th International Conference on Microelectronics, ICM 2013. <https://doi.org/10.1109/ICM.2013.6735002>
- [26] Shaloot, A. H., & Madian, A. H. (2012). Memristor based carry lookahead adder architectures. *Midwest Symposium on Circuits and Systems*, 298–301. <https://doi.org/10.1109/MWSCAS.2012.6292016>
- [27] Liu, G., Zheng, L., Wang, G., Shen, Y., & Liang, Y. (2019). A Carry Lookahead Adder Based on Hybrid CMOS-Memristor Logic Circuit. *IEEE Access*, 7, 43691–43696. <https://doi.org/10.1109/ACCESS.2019.2907976>
- [28] Karimi, A., & Rezai, A. (2018). Novel design for a memristor-based full adder using a new IMPLY logic approach. *Journal of Computational Electronics* 2018 17:3, 17(3), 1303–1314. <https://doi.org/10.1007/S10825-018-1198-5>
- [29] El-Slehdar, A. A., Fouad, A. H., & Radwan, A. G. (2015). Memristor-based redundant binary adder. *ICET 2014 - 2nd International Conference on Engineering and Technology*. <https://doi.org/10.1109/ICENGTECHNOL.2014.7016820>
- [30] Guckert, L., & Swartzlander, E. (2017). Optimized memristor-based ripple carry adders. *Conference Record - Asilomar Conference on Signals, Systems and Computers*, 1575–1579. <https://doi.org/10.1109/ACSSC.2016.7869644>
- [31] Mohammed, S. L., Al-Naji, A., Farjo, M. M., & Chahl, J. (2019). Highly Accurate Water Level Measurement System Using a Microcontroller and an Ultrasonic Sensor. *IOP Conference Series: Materials Science and Engineering*, 518(4), 042025. <https://doi.org/10.1088/1757-899X/518/4/042025>
- [32] Sulistyowati, R., Sujono, H. A., & Musthofa, A. K. (2016). A river water level monitoring system using android-based wireless sensor networks for a flood early warning system. *Lecture Notes in Electrical Engineering*, 365, 401–408. https://doi.org/10.1007/978-981-287-988-2_43/COVER/
- [33] Getu, B. N., & Attia, H. A. (2017). Automatic water level sensor and controller system. *International Conference on Electronic Devices, Systems, and Applications*. <https://doi.org/10.1109/ICEDSA.2016.7818550>
- [34] Okhaifoh, J. E., Igbinoba, C. K., & Eriagana, K. O. (2016). Microcontroller Based Automatic Control For Water Pumping Machine With Water Level Indicators Using Ultrasonic Sensor. *Nigerian Journal of Technology*, 35(3), 579–583. <https://doi.org/10.4314/njt.v35i3.16>
- [35] Chang, Y., Wang, X., Wang, J., Wu, Y., Yang, L., Zhu, K., ... Xie, X. (2024). A Survey on Evaluation of Large Language Models. *ACM Transactions on Intelligent Systems and Technology*, 15(3). <https://doi.org/10.1145/3641289>
- [36] Raiaan, M. A. K., Mukta, M. S. H., Fatema, K., Fahad, N. M., Sakib, S., Mim, M. M. J., ... Azam, S. (2024). A Review on Large Language Models: Architectures, Applications, Taxonomies, Open Issues and Challenges. *IEEE Access*, 12, 26839–26874. <https://doi.org/10.1109/ACCESS.2024.3365742>
- [37] Chang, C.-C., Shen, Y., Fan, S., Li, J., Zhang, S., Cao, N., ... Zhang, X. (n.d.). LaMAGIC: Language-Model-based Topology Generation for Analog Integrated Circuits.
- [38] Lu, Y., Liu, S., Zhang, Q., & Xie, Z. (2024). RTLLM: An Open-Source Benchmark for Design RTL Generation with Large Language Model. *Proceedings of the Asia and South Pacific Design Automation Conference, ASP-DAC*, 722–727. <https://doi.org/10.1109/ASP-DAC58780.2024.10473904>
- [39] Ho, T.-Y., Khan, S., Liu, J., Li, Y., Liu, Y., Shi, Z., ... Zeng, X. (2024). The Dawn of AI-Native EDA: Opportunities and Challenges of Large Circuit Models. Retrieved from <https://arxiv.org/abs/2403.07257v2>
- [40] Biolk, Z., Biolk, D., & Biolková, V. (2009). Spice Model of Memristor With Nonlinear Dopant Drift. *Radioengineering*, 18(2), 210–214. Retrieved from <http://citeseerx.ist.psu.edu/viewdoc/summary?doi=10.1.1.497.8869>