

A Reconfigurable and Compact Spin-Based Analog Block for Generalizable n^{th} Power and Root Computation

Adrian Tatulian and Ronald F. DeMara

Department of Electrical and Computer Engineering
University of Central Florida, Orlando, FL 32816-2362
adrian.tatulian@ucf.edu and ronald.demara@ucf.edu

Abstract— While square and square root are critical for vector operations, several challenges exist in their computation in the digital domain including power, area, and delay overheads. While selective computation in the analog domain is a longstanding alternative, tradeoffs of increased noise and reduced accuracy are prominent challenges. Herein, we propose a reconfigurable analog circuit which is capable of performing any n^{th} -power and n^{th} -root function and may be implemented within an analog or mixed-signal field programmable array. The resulting analog block of Magnetic Tunnel Junctions (MTJs) along with FET-based sensing and amplification circuits are circuit-switched-configurable with terminal-level control. Herein the design is configured to rapidly evaluate various arithmetic operations within acceptable error tolerances for selected applications. When compared to a state-of-the-art approximate digital multiplier, our design yields 97% reduction in transistor count and stable output within a period comparable to single-cycle execution. When compared to an alternative recently proposed analog multiplier, the proposed design yields improvement in error and versatility to perform generalized operations.

Keywords—Analog computation, reconfigurable computing, Field Programmable Analog Array, Magnetic Tunnel Junction

I. INTRODUCTION

Square and square root are critical operations for algorithms in fields such as computer vision [1] and signal processing [2, 3]. In processing vector-valued data, these applications will often require repeated calls to power and root functions for calculation of vector norms. However, these functions can require significant area and delay overheads in the digital domain, requiring 12 or more clock cycles to execute [4] and hundreds of logic gates [5]. As a result of these challenges, there has recently been renewed interest in pursuing an analog approach to operations such as multiplication, square, and square root [6, 7].

Analog circuits offer reductions in area and power overheads at the expense of computational precision; this is an attractive tradeoff for error-tolerant applications [6], especially in applications where power and area are limited, e.g., Internet of Things (IoT) devices. An example of such an application in Compressive Sensing (CS). CS entails compression and transmission of a spectrally-sparse signal, and then reconstruction of the signal at the receiving end. Due to its approximate nature and suitability for IoT devices, CS is an ideal application for analog computation.

In recent years, Field Programmable Analog Arrays (FPAAs) have been proposed as a counterpart to traditional

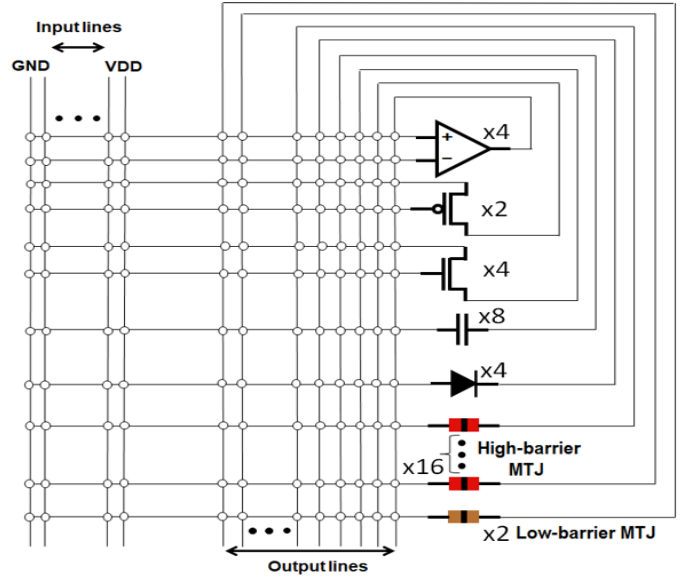


Fig. 1: FPAA fabric comprised of active and passive analog devices such as NMOS/PMOS transistors, capacitors and diodes, along with spin-based Magnetic Tunnel Junction (MTJ) devices.

digital-only FPGAs, particularly for computations involving sensor interfacing and signal processing [8]. FPAAs consist of a set of analog components, such as field-effect transistors (FETs), capacitors, resistors, and diodes integrated into a reconfigurable fabric architecture (Fig. 1). While a lack of software for FPAA programmability has been a challenge, recent developments including the Reconfigurable Analog Signal Processor (RASP) and associated high-level tools have provided a pathway for system-level analog design [9].

Analog computation can provide vast energy improvements, up to a 1000-fold improvement in computational energy efficiency [10] and thus FPAA technology has already been implemented in ultra-low power IoT sensing applications, including temperature sensors and heart rate alarms [11].

Subsequently, simulations of a mixed-signal reconfigurable fabric yield 5-fold energy reduction and 26-fold reduction in area requirement, simply by performing vector-matrix multiplication (VMM) operations in analog using a crossbar array, as opposed to performing the computations in the digital domain. Part of this improvement can be attributed to the use of spin-based devices, i.e., Magnetic Tunnel Junctions (MTJs), which offer near-zero standby power

dissipation and area efficiency by allowing for reduced transistor count [12].

Herein, we propose an analog circuit based on the fabric shown in Fig. 1 which is capable of performing generalized n^{th} root and power operations. Area overhead is minimized by a) performing computations in the analog domain based on intrinsic properties of the embedded op-amps, and b) a reconfigurable architecture allowing for the realization of multiple functionalities within a single fabric. In addition, the use of low-barrier MTJs allow for stochastic computation relying on intrinsic properties of the MTJ device.

The remainder of the manuscript is organized as follows: Section II gives background and related works relating to analog computation circuits and spin-based devices. Section III introduces the proposed design. Finally, Section IV provides simulation results and Section V concludes the paper.

II. BACKGROUND AND RELATED WORKS

A. Analog Computational Circuits

Analog computation relies on the intrinsic device properties of circuit components to conduct computations. Analog computations tend to be approximate, but may be superior to digital counterparts in latency, power consumption, and area albeit at times subject to significant precision, noise, temperature, and operating voltage challenges.

Implementation of analog computation has taken a wide variety of forms within AI applications. A recent work [13] discusses analog computation in the context of one type of neural network, i.e., the multilayer perceptron (MLP). The MLP hardware utilizes MOSFET-based current mirrors along with operational transconductance amplifiers to perform vector-matrix multiplication. Two operations are performed: multiplication and addition utilizing Kirchhoff's Current Law (KCL) in order to sum the current signals at a specific node. [14] proposes a generalized non-linear function synthesizer through Taylor series approximation. The hardware implementation relies on successive application of a squaring unit (SU) based on a class AB current mirror architecture, and yields a maximum error of 10% for a 5th order polynomial. Meanwhile, [15] shows how to construct a reliable nonlinear circuit to exploit nonideal properties within a cascaded array of analog multipliers for simulation of mathematical chaos. In [16], a mixed analog and digital hybrid solution is introduced that seeks to alleviate challenges, e.g., lack of programmability, associated with a fully analog nonlinear computation stack. The

architecture developed by the authors is applied to obtain solutions to non-linear ordinary differential equations. In [17], root and power computations are implemented using time-mode circuits. Its hardware relies on the translinear principle, i.e., exponential I-V characteristics of CMOS transistors, whereby hardware with exponential and logarithmic outputs is cascaded to yield the desired result.

Several authors have sought automated hardware synthesis and optimization through the use of genetic algorithms. In [18], which is one of the pioneering works in this field, the authors present a wide variety of analog circuits produced via genetic algorithms, including a cube root implementation. Subsequently, [19] expands upon the work of [18] to synthesize a wider variety of evolutionary computation circuits, with improved output error. Meanwhile, [20] conducts iterative refinement on computational circuits including squaring, square root, and cubing circuits. From this iterative process, certain circuits that are created through genetic programming are able to be refined through the error produced from the previous best-of-run from the same circuit. [21] extends this approach to combine analog and digital computation, whereby analog outputs are refined digitally to improve computational accuracy for less error-prone applications. Finally, [22] explores synthesis of arbitrary functions through Puiseux series, using genetic algorithms to minimize error. Table I summarizes a selection of these works in the context of the research proposed herein.

B. Spin-Based Devices

Spin-based devices, e.g., Magnetic Tunnel Junctions (MTJs), have recently been researched for applications such as non-volatile memory due to several advantages, including: near-zero standby power dissipation, high endurance, and area efficiency [3]. MTJs are composed of two ferromagnetic layers, referred to as fixed layer and free layer, separated by a thin oxide barrier; the magnetization orientation of the free layer switches direction when a bidirectional current of sufficient magnitude passes through the device, and thus flips the device from the Parallel (P) state to the Anti-Parallel (AP) state or vice-versa. Based on the device properties, a low resistance is attained in the P state while a high resistance is attained in the AP state. Specifically, the device resistance is given by $R_P = R_{MTJ}$ and $R_{AP} = R_{MTJ}(1 + TMR)$ whereby

$$R_{MTJ} = \frac{t_{ox}}{Factor \times Area \sqrt{\phi}} \exp(1.025 t_{ox} \sqrt{\phi}) \quad (1)$$

TABLE I. COMPARISON OF ANALOG COMPUTATION ARCHITECTURES

Work	Functionality	Mode of operation	No. of components	Highlighted Contributions
[14]	n^{th} power via Squaring Unit	Class AB current mirror	22	Arbitrary nonlinear functions in terms of Taylor series expansion
[17]	Square, cube, 4 th power	Translinear time-to-voltage and voltage-to-time convertors	~100	Nonlinear operations through the time-mode translinear principle
[18]	Cube root	Evolved computational circuit	48	Pioneer in evolutionary circuit design
[19]	Square, square root, cube, cube root	Evolved computational circuit	≤ 44	Genetic algorithms for optimizing analog circuits for non-conventional applications
Herein	Generalized n^{th} power and root; inverse functions	Op-amps in reconfigurable fabric	43	Reconfigurable design, and intrinsic stochasticity

$$TMR = \frac{TMR_0}{1 + \left(\frac{V_b}{V_h}\right)^2} \quad (2)$$

in which TMR is tunneling magnetoresistance, t_{ox} the oxide layer thickness, $Factor$ a material-dependent parameter which depends on the resistance-area product of the device, $Area$ the surface area of the device, ϕ the oxide layer energy barrier height, V_b bias voltage, and V_h the bias voltage at which TMR drops to half of its initial value.

The energy barrier, E_B , between the P and AP states of an MTJ device can be tuned based on fabrication dimensions. The device is considered to be low-barrier under the condition $E_B \ll 40kT$, in which case thermal fluctuations at room temperature are sufficient to change the state of the device. This observation has led to construction of the probabilistic bit (p-bit) device, as shown in Fig. 2. A p-bit [23, 24] takes analog input and yields a digital output whose probability of being logic 1 depends on the supplied input voltage. This functionality is due to the p-bit's structure as a voltage divider between a low-barrier MTJ and NMOS transistor. A higher voltage applied to the gate of the transistor results in reduced drain-source voltage, r_{ds} , which increases the probability of delivering sufficient voltage to the input of the inverter to yield a logic 1 output.

The p-bit output is described by the equation [23]

$$V_{out} = V_{DD} \operatorname{sgn}\{\tanh(V_{in}/V_0) + \operatorname{rand}(-1,1)\} \quad (3)$$

where sgn represents the sign function, $\operatorname{rand}(-1,1)$ represents a random number in $[-1,1]$ and V_0 is a model parameter.

III. RECONFIGURABLE ANALOG MULTIPLIER

The multiplier design presented herein is based on the translinear principle, similar to [17]. Based on the nonlinear characteristics of an operational amplifier, a three-stage design is presented whereby the output is a power function of the input, with the power determined by circuit parameters. Thus, the circuit functionality is generalizable via reconfigurability of the fabric in which it is embedded.

Fig. 3 shows the op-amp included in the FPAA fabric, which includes two cascaded stages: an input stage consisting of a differential amplifier, followed by a gain stage. The focus is on simplicity to optimize area, power consumption, and speed. The design consists of 10 transistors, which have been simulated using models from the PTM 14nm LSTP FinFet library operated at $V_{DD} = 0.8V$.

Based on the op-amp, the multiplier circuit consists of three stages, as shown in Fig. 4. The first stage is a logarithmic amplifier, with output voltage V_1 given by solving:

$$V_1 = -A_{OL}V_0 \quad (4)$$

$$-\frac{V_0 - V_{in}}{R_1} = I_{S1} \left[\exp\left(\frac{V_0 - V_1}{V_T}\right) - 1 \right] \quad (5)$$

where A_{OL} represents open-loop gain and I_{S1} represents the saturation current of diode D_1 . Eq. 4 is from general op-amp theory and Eq. 5 follows from KCL. Solving Eq. 4 and Eq. 5 simultaneously, and applying the limit of infinite open-loop gain, yields:

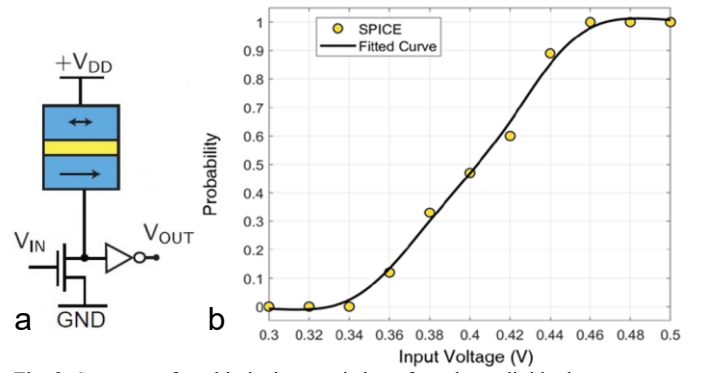


Fig. 2: Structure of a p-bit device consisting of a voltage divider between a low-barrier MTJ device and NMOS transistor (a); probability of a logic 1 output value (b).

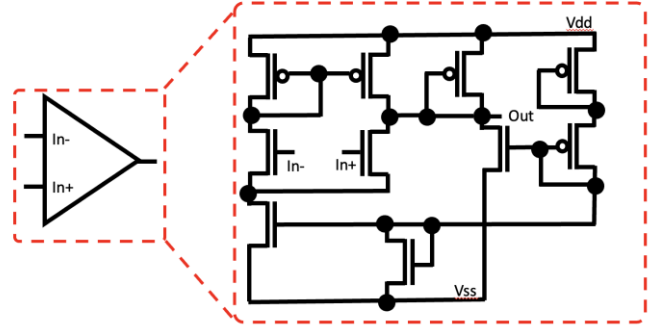


Fig. 3: Op-amp comprised of 10 MOSFETs offering high speed and compact area.

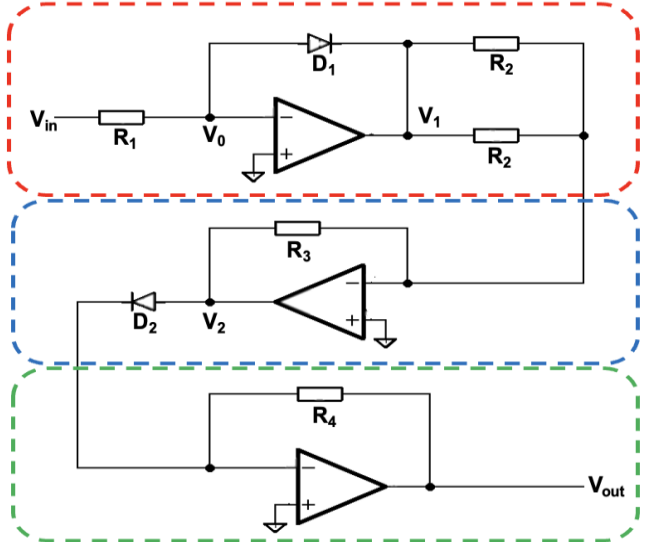


Fig. 4: Squaring Circuit Design. The first, second, and third stage are outlined in red, blue, and green, respectively.

$$V_1 = -V_T \ln\left(\frac{V_{in}}{R_1 I_{S1}}\right) \quad (6)$$

which is an approximation that holds for sufficiently large V_{in} .

The second stage is an analog adder, whereby a similar analysis yields $V_2 = -\frac{2V_1 R_3}{R_2}$. Finally, the third stage is an anti-log amplifier with output approximately given by:

$$V_{out} = -R_4 I_{S2} e^{\frac{V_2}{V_T}}. \quad (7)$$

TABLE II. ERROR, BANDWIDTH, AND DELAY DATA

V_{in2}	Non-linearity Error ¹	-3-dB bandwidth ²	Delay ³
0.3V	0.48%	195MHz	3.8ns
0.4V	0.11%	191MHz	3.9ns
0.5V	0.25%	186MHz	4.1ns
0.6V	0.43%	178MHz	4.4ns
0.7V	0.55%	174MHz	5.0ns

¹ V_{in1} is DC with magnitude swept between 0.3V and 0.7V.
² V_{in1} is sinusoidal with offset of 0.45V and amplitude of 0.25V.
³ $V_{in1} = V_{in2}$

Overall, it is simple to see that the output of this circuit is given by:

$$V_{out} = -\frac{R_4 I_{S2}}{(R_1 I_{S1})^a} (V_{in})^a \quad (8)$$

where $a = 2R_3/R_2$.

According to this theory, the circuit shown in Fig. 4 can be used to implement any positive power function of the input voltage by modifying the diode characteristics and resistances in the various stages. Furthermore, by inserting a standard inverting amplifier before the final exponentiation, any inverse power function can be implemented as well. It is important to note that Eq. 6 – Eq. 8 hold only for infinite open-loop gain which is not attained in practice. Thus, the equations provide a starting point for the design, after which parameters must be adjusted to minimize output errors. Final parameters are: $R_1 = 3500k\Omega$, $R_2 = 50k\Omega$, $R_3 = 150k\Omega$, $R_4 = 75k\Omega$, $I_{S1} = 50nA$ and $I_{S2} = 5.4nA$. In addition, a load capacitance of 100fF and load resistance of 1000k Ω is included at the output stage of each op-amp.

Since MTJs take dual resistance values, resistances are implemented using high-barrier MTJs available in the FPAA fabric to further reduce area. In addition, the proposed FPAA components allow for construction of up to 2 p-bit devices. The device outputs can be integrated using a capacitor to generate tunable random inputs to the multiplier, e.g., to simulate noise in signal processing applications.

IV. SIMULATION RESULTS

A. Analog Multiplier

The design in Fig. 4 can be adjusted to operate as a multiplier, if two logarithmic amplifiers with separate inputs are used in the first stage. This design is evaluated in terms of DC transfer characteristics, frequency response, and Total Harmonic Distortion (THD).

Fig. 5 shows DC transfer characteristics, in the case where one input voltage, V_{in1} , is swept across the operational range and the second input voltage, V_{in2} , is static. The output voltage is shown for $V_{in2} = 0.3V$ to $V_{in2} = 0.7V$, i.e., 50% of the total available range. The average non-linearity error, i.e., percentage deviation from linear regression line, is computed in each case and listed in Table II. It is seen that the maximum error of 0.55% occurs for $V_{in2} = 0.7V$, and also that minimum errors are attained in the middle of the operational range.

Next, Fig. 6 shows the frequency response of the circuit, in the range from 100MHz to 1GHz. The data are obtained by

TABLE III. THD WITH ONE DC AND ONE SINUSOIDAL INPUT

Frequency	Amplitude = 0.25V	Amplitude = 0.05V
10kHz	0.80%	0.76%
100kHz	0.81%	0.77%
1MHz	0.81%	0.75%
2MHz	1.08%	1.10%
3MHz	1.82%	1.61%

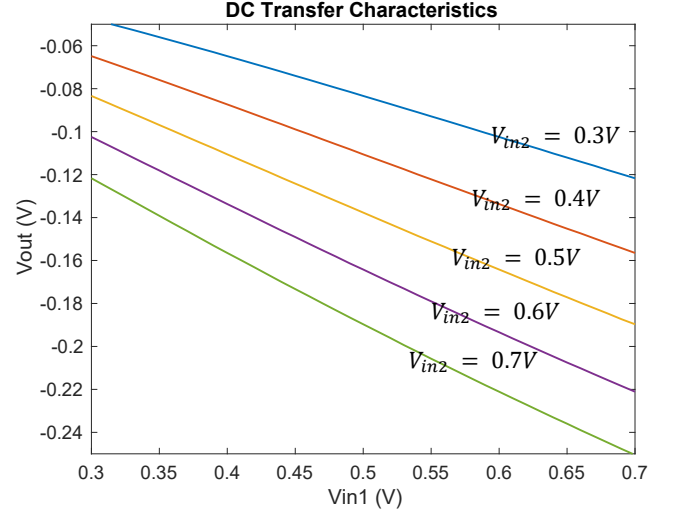


Fig. 5: DC transfer characteristics for the proposed multiplier, with one input fixed and the second input varying across the operational range.

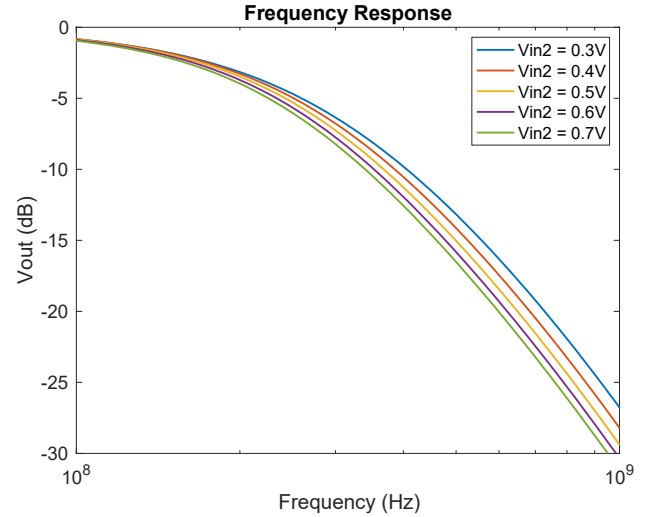


Fig. 6: Frequency response, with one input fixed and the second input sinusoidal with offset of 0.45V and amplitude of 0.25V.

setting V_{in1} as a sinusoidal signal with offset of 0.45V and amplitude of 0.25V. Again, V_{in2} is fixed and the curves represent data for $V_{in2} = 0.3V$ to $V_{in2} = 0.7V$. The -3-dB bandwidth, as noted in Table II, is slightly less than 200MHz in each case and is assuming a capacitance of 100fF connected to the output stage of each op-amp. While this bandwidth may not be suitable for applications with limited signal-to-noise ratio, the capacitance can simply be replaced with a higher value to limit bandwidth if necessary.

TABLE IV. COMPARISON OF RESULTS

	Herein	Herein	Herein	[5]	[7]	[14]	[17]
Mode	Analog	Analog	Analog	Digital	Analog	Analog	Analog
Operation	Cube root	Square root	Square	Multiplier	Multiplier	Square	Square
Tech node	14nm	14nm	14nm	28nm	130nm	500nm	180nm
V_{DD}	0.8V	0.8V	0.8V	1V	0.6V	1.5V	1.3V
No. of components	43	43	43	~1000	35	12	~100
Power	123 μ W	122 μ W	126 μ W	126 μ W	23 μ W	600 μ W	149 μ W
Mean Error	0.50%	0.66%	1.30%	1.6%	9.1%*	N/A	0.24%**

*RMS noise vs. max. output
 **At $V_{in} = 0.4V$

In conjunction with bandwidth, Table II also provides the delay necessary to attain 90% of the target voltage in the case where $V_{in1} = V_{in2}$; the data show that the circuit produces outputs in $\leq 5ns$. This figure is in agreement with the bandwidth data presented in the same table and serves to characterize the high-speed capabilities of our design. High speed serves to limit total energy consumption, which is critical in IoT applications.

Finally, Table III gives THD in the case where one input is 0.45V DC, and the second input is sinusoidal with amplitudes of 0.05V and 0.25V. It is seen that THD is roughly constant and within 1% up to a frequency of approximately 1MHz, indicating practical functionality of the system.

B. n^{th} Power and Root

The proposed circuit is capable of executing any n^{th} root and power function, in accordance with the theory presented in Section II. Simulation results have shown that n^{th} root functions can be implemented with high accuracy; power functions beyond squaring are challenging due to saturation-related errors. However, it is possible to obtain these functions using a squaring unit by iteratively applying the identity: $(A + B)^2 - (A - B)^2 = 4AB$. For example, the cubing function is implemented by substituting x^2 for A and x for B ; in this manner, any n^{th} power function, $n \geq 2$, can be computed by using an architecture such as that described by [14]. The authors of [14] were able to compute a 5th order polynomial function within 10% error through this approach.

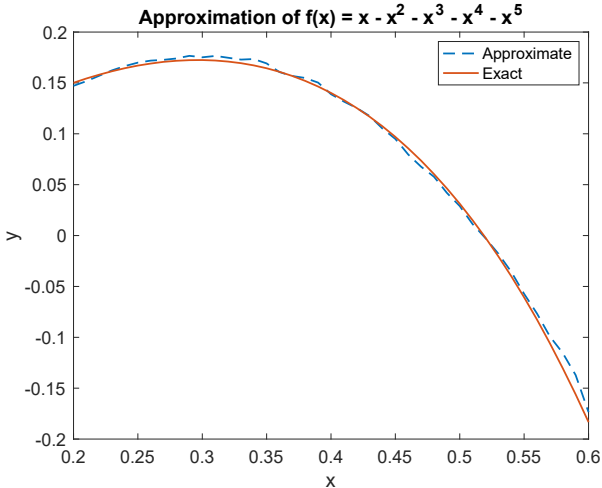


Fig. 7: Approximation of a 5th order polynomial function using the proposed hardware, showing agreement with an error-free implementation.

Table IV gives details on cube root, square root, and squaring circuits implemented using the proposed design, including technology node, supply voltage, total number of elementary components, power dissipation, and mean error over an input range of 0.2V – 0.6V. Comparing to the approximate digital multiplier described in [5], at the design point giving nearly-identical power consumption, the analog circuit described herein yields slightly improved mean error across the operational range. Furthermore, the approximate digital design requires an estimated 250 logic gates, corresponding to a transistor count on the order of 1000. Thus, our design achieves a 97% reduction in transistor count and 96% reduction in number of components, indicating a significant area reduction.

Comparing to previous analog designs, [7] demonstrates reduced power consumption but significantly higher error and a relatively low bandwidth of 51.2kHz. [17] describes a similar design to the one introduced herein, relying on the translinear principle to implement n^{th} power functions by combining hardware with logarithmic and exponential output characteristics; however, [17] relies on time-mode circuitry, which leads to intrinsic time delays, on the order of microseconds.

To further evaluate our square and square root hardware, a simulation is performed in the context of Compressive Sensing signal reconstruction, using three reconstruction algorithms: Orthogonal Matching Pursuit (OMP), Compressive Sampling Matching Pursuit (CoSaMP), and Approximate Message Passing (AMP). These algorithms, similar to many others in the signal processing domain, require iterative computation of vector norms. For each algorithm, the number of measurements necessary to attain a -60dB reconstruction error is determined, using our hardware for square and square root computations, and a digital implementation as the baseline. The results indicate no change for OMP and CoSaMP, and a 2.5% increase for AMP.

C. Generalized Functions

The proposed hardware can be used to implement generalized functions beyond power and root. As noted in Section III, inverse power and root functions can be computed by adding an inverting amplifier before the final stage; a $1/\sqrt{x}$ function designed in this manner yields average error of 0.4%. Exponential and logarithmic outputs are obtained through the output and input stages, respectively. Other generalized

functions can be implemented using a Taylor series approximation.

Fig. 7 shows an approximation of the function $f(x) = x - x^2 - x^3 - x^4 - x^5$ based on the proposed analog squaring unit. This simulation includes squaring errors, but does not include errors in addition, subtraction, and voltage rescaling; the resulting approximation is seen to be in agreement with the target function, giving an average error of 4.83% over the tested range. This demonstrates the feasibility of generating higher-order polynomial functions, and generalized functions through Taylor series, using our analog approach.

V. CONCLUSIONS

Herein we have demonstrated a reconfigurable analog circuit designed for computation of n^{th} power and root functions. The circuit is based on an FPAA fabric, containing passive and active analog devices, in addition to spin-based devices. Compared to previous approaches, our design demonstrates high speed operation, reduced error, and reduced number of components. The reconfigurable nature of the design allows for a wide range of functionalities using a small set of hardware components. Furthermore, the bandwidth of the circuit can be tuned through proper selection of capacitors in the FPAA fabric. Finally, the presence of low-barrier spin-based devices allows for built-in stochastic computation, e.g., for simulation of noise in certain applications.

While the present focus has been in IoT, future work may also extend the design to server applications, e.g., verification and validation computations, which have previously been conducted digitally using FPGAs. One possibility would be to use analog computation in conjunction with p-bit devices to conduct simulations involving process variation, and then apply a digital refinement approach for error reduction. Future work may also include the use of automated techniques such as genetic algorithms for further design optimization, including error reduction.

ACKNOWLEDGMENTS

This work was supported in part by the Center for Probabilistic Spin Logic for Low-Energy Boolean and Non-Boolean Computing (CAPSL), one of the Nanoelectronic Computing Research (nCORE) Centers as task 2759.006, a Semiconductor Research Corporation (SRC) program sponsored by the NSF through CCF-1739635, and by NSF through ECCS-1810256.

REFERENCES

- [1] R. N. Strickland, T. Draelos, and Z. Mao, "Edge detection in machine vision using a simple L1 norm template matching algorithm," *Pattern Recognition*, vol. 23, no. 5, pp. 411-421, 1990.
- [2] Y. Shi, S. Xia, Y. Zhou, and Y. Shi, "Sparse signal processing for massive device connectivity via deep learning," in *2020 IEEE International Conference on Communications Workshops (ICC Workshops)*, pp. 1-6, IEEE, 2020.
- [3] A. Tatulian, S. Salehi, and R. F. DeMara, "Mixed-signal spin/charge reconfigurable array for energy-aware compressive signal processing," in *2019 International Conference on ReConfigurable Computing and FPGAs (ReConFig)*, pp. 1-8, IEEE, 2019.
- [4] A. Hasnat, T. Bhattacharyya, A. Dey, S. Halder, and D. Bhattacharjee, "A fast FPGA based architecture for computation of square root and inverse square root," in *2017 Devices for Integrated Circuit (DevIC)*, pp. 383-387, IEEE, 2017.
- [5] H. Jiang, C. Liu, F. Lombardi, and J. Han, "Low-power approximate unsigned multipliers with configurable error recovery," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 66, no. 1, pp. 189-202, 2018.
- [6] N. Arya, T. Soni, M. Pattanaik, and G. Sharma, "Area and energy efficient approximate square rooters for error resilient applications," in *2020 33rd International Conference on VLSI Design and 2020 19th International Conference on Embedded Systems (VLSID)*, pp. 90-95, IEEE, 2020.
- [7] A. J. S. de Sousa *et al.*, "A very compact CMOS analog multiplier for application in CNN synapses," in *2019 IEEE 10th Latin American Symposium on Circuits & Systems (LASCAS)*, pp. 241-244, IEEE, 2019.
- [8] R. B. Wunderlich, F. Adil, and P. Hasler, "Floating gate-based field programmable mixed-signal array," *IEEE Transactions on Very Large Integration (VLSI) Systems*, vol. 21, no. 8, pp. 1496-1505, 2012.
- [9] C. Schlottmann and P. Hasler, "FPAA empowering cooperative analog-digital signal processing," in *2012 IEEE International Conference on Acoustics, Speech and Signal Processing (ICASSP)*, pp. 5301-5304, IEEE, 2012.
- [10] Y. Huang, "Hybrid analog-digital co-processing for scientific computation," Columbia University, 2018.
- [11] B. Rumberg and D. W. Graham, "A low-power field-programmable analog array for wireless sensing," in *Sixteenth International Symposium on Quality Electronic Design*, pp. 542-546, IEEE, 2015.
- [12] S. Salehi *et al.*, "Energy-aware adaptive rate and resolution sampling of spectrally sparse signals leveraging VCMA-MTJ devices," *IEEE Journal on Emerging and Selected Topics in Circuits and Systems*, vol. 8, no. 4, pp. 679-692, 2018.
- [13] S. Abden and E. Azab, "Multilayer perceptron analog hardware implementation using low power operational transconductance amplifier," in *2020 32nd International Conference on Microelectronics (ICM)*, pp. 1-4, IEEE, 2020.
- [14] M. T. Abuelma'Atti and A. M. Abuelma'Atti, "A new current-mode CMOS analog programmable arbitrary nonlinear function synthesizer," *Microelectronics Journal*, vol. 43, no. 11, pp. 802-808, 2012.
- [15] A. Buscarino, C. Corradino, L. Fortuna, M. Frasca, J. C. Sprott, "Nonideal behavior of analog multipliers for chaos generation," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 63, no. 4, pp. 396-400, 2015.
- [16] N. Guo *et al.*, "Energy-efficient hybrid analog/digital approximate computation in continuous time," *IEEE Journal of Solid-State Circuits*, vol. 51, no. 7, pp. 1514-1524, 2016.
- [17] R. J. D'Angelo and S. R. Sonkusale, "A time-mode translinear principle for nonlinear analog computation," *IEEE Transactions on Circuits and Systems I: Regular Papers*, vol. 62, no. 9, pp. 2187-2195, 2015.
- [18] J. R. Koza, F. H. Bennett, D. Andre, M. A. Keane, and F. Dunlap, "Automated synthesis of analog electrical circuits by means of genetic programming," *IEEE Transactions on Evolutionary Computation*, vol. 1, no. 2, pp. 109-128, 1997.
- [19] Y. A. Sapargaliyev and T. G. Kalganova, "Open-ended evolution to discover analogue circuits for beyond conventional applications," *Genetic Programming and Evolvable Machines*, vol. 13, no. 4, pp. 411-443, 2012.
- [20] M. J. Streeter, M. A. Keane, and J. R. Koza, "Iterative refinement of computational circuits using genetic programming," in *Proceedings of the 4th Annual Conference on Genetic and Evolutionary Computation*, pp. 877-884, 2002.
- [21] S. D. Pyle, V. Thangavel, S. M. Williams, and R. F. DeMara, "Self-scaling evolution of analog computation circuits with digital accuracy refinement," in *2015 NASA/ESA Conference on Adaptive Hardware and Systems (AHS)*, pp. 1-8, IEEE, 2015.
- [22] V. Thangavel, Z. X. Song, and R. F. DeMara, "Intrinsic evolution of truncated Puiseux series on a mixed-signal field-programmable soc," *IEEE Access*, vol. 4, pp. 2863-2872, 2016.
- [23] K. Y. Camsari, S. Salahuddin, and S. Datta, "Implementing p-bits with embedded MTJ," *IEEE Electron Device Letters*, vol. 38, no. 12, pp. 1767-1770, 2017.
- [24] S. Datta, "p-Bits for probabilistic computing," in *2019 Device Research Conference (DRC)*, pp. 35-36, IEEE, 2019.