A CORDIC based Configurable Activation Function for ANN Applications

Gopal Raut[†], Shubham Rai[*], Santosh Kumar Vishvakarma[†] and Akash Kumar[*]
[†] Discipline of Electrical Engneering, Indian Institute of Technology Indore Indore, India
[*]CfAED Technische Universität Dresden, Germany
<phd1701102005, skvishvakarma>@iiti.ac.in, <shubham.rai, akash.kumar>@tu-dresden.de

Abstract*—An efficient ASIC-based hardware design of activation function (AF) in neural networks faces the challenge of offering functional configurability and limited chip area. Therefore an area-efficient configurable architecture for an AF is imperative to fully harness the parallel processing capacity of an ASIC in contrast to a general-purpose processor. To address this, we propose a configurable AF based on the shift-and-add algorithm, collectively known as *Co-ordinate Rotation Digital Computer* (CORDIC) algorithm. The proposed versatile configurable activation function is designed using CORDIC architecture and implements both tan hyperbolic and sigmoid function. The derived model is synthesized and verified at 45nm technology. Further, in order to address leakage issues at lower technology nodes, we exploit the power-gating technique for the proposed AF based on CORDIC architecture. Our circuit design is extracted in cadence virtuoso and simulated for all physical parameters. With respect to the state-of-the-art, our design architecture shows improvement by 29% in area, 42% in power dissipation and 20% in latency. The used power gating technique saves 30% static power with minimal area overhead. The *Monte-Carlo* simulations for process-variations and device-mismatch are performed for both the proposed model and the state-of-the-art to evaluate expectations of functions of randomness in dynamic power variation. The dynamic power variation for our design shows that mean and σ deviation are $180.73\mu W$ and $51.7\mu W$ respectively which is 60% of the state-of-the-art.

I. Introduction

An artificial neural network (ANN) is a popular choice for a computational model like pattern recognition, prediction, and classification. An ANN implementation has the following major components under consideration – a computational unit, an activation function(AF), data precision, data types, and a design platform. The computation entails multiply-and accumulate units (MAC) followed by a nonlinear mathematical transformation called AF. The single neuron with multiple inputs and weights for MAC operation followed by a nonlinear transformation is shown in Fig. 1.

Hardware implementation of an ANN can exploit the underlying parallelism of the network to further optimize performance. Efficient VLSI architectures have been proposed, targeting diverse applications based on ANN [1], [2]. However, from an application point of view, the performance and accuracy of a neural network primarily depend upon the data precision required [1] [2]. Specifically, in case of a hardware implementation of neural networks, higher precision generally comes with high area and power overheads. This trade-off gets even more complicated when configurable architectures are required.

While FPGAs offer configurable hardware designs, they often require more chip area as compared to ASICs [3]. From the design point of view of a neural network, a configurable activation function implementing various types of non-linear transfer functions such as $sigmoid$, $hyperbolic\ tangent$ $(tanh)$, $exponential$ is often desirable [4]. To address this trade-off, we propose an optimized CORDIC-based architecture to enable efficient yet configurable computations required in the neural networks.

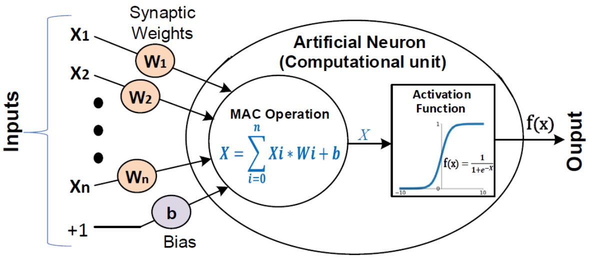

Synaptic Weights
Inputs
X1
X2
Xn
+1
W1
W2
Wn
b
Bias
Artificial Neuron (Computational unit)
MAC Operation
$X = \sum_{i=0}^{n} Xi * Wi + b$
X
Activation Function
$f(x) = \frac{1}{1+e^{-X}}$
1
0
-10
10
f(x)
Ouput

Fig. 1. Single neuron with multi inputs followed by activation function.

The **Co**-ordinate **R**otation **DI**gital **C**omputer (CORDIC) algorithm has a linear convergence and uses minimal resources [5]. It is a desirable choice where the cost-to-performance ratio is critical [6]. The CORDIC architecture uses only shift-and-add operations and can perform several computing tasks such as trigonometric, hyperbolic and logarithmic functions [7]. Since all these non-linear functions are typically used (calculation of various activation functions) in an ANN application, CORDIC offers an interesting choice. CORDIC algorithm requires only addition, subtraction, and bit shift operations which can be easily implemented in an hardware [9].

In this work, we investigate design strategies and optimizations of activation functions (AF) for an ANN architecture. We employ CORDIC architecture in hyperbolic rotation mode to realize both $tanh$ and $sigmoid$ functions. The CORDIC-based architecture requires only shift-and-add operation which can lead to area and power savings with better data precision. The proposed architecture realizes sigmoid as well as tanh function using the same hardware resources. The major contribution is summarized in the following points:

- Design of a configurable architecture for multiple activation function using the CORDIC-based algorithm.
- Optimization of exiting CORDIC-based architecture in terms of area, power, and delay.
- Analyze the impact of technology scaling on circuit's physical parameters like area and power, and proposed power-gating approach for additional power savings.

The multiple AF design is realized using the CORDIC architecture at 45nm technology node. The configurable activation function using CORDIC is designed for low-cost hardware requirements. The power gating technique is further explored to reduce static power dissipation. For a 16-bit precision activation function, CORDIC based architecture takes only 3.5% of the area and consumes only 12.0% of the dynamic power as reported by the state-of-the-art combinational logic design. Additionally, The proposed architecture with power gating technique saves 30% static power.

The rest of this paper is organized as follows. Section II gives the related works and motivation. Section III describes the CORDIC architecture in detail. Section IV provides the proposed design architecture and its working. Section V gives experimental setup followed by simulation results and discussion in Section VI. The concluding remarks are given in section VII.

II. Motivation and Related Works

Hardware realization of ANNs can be classified into three groups–Application Specific Integrated Circuits (ASIC), Digital Signal Processing (DSP), and Field Programmable Gate Arrays (FPGA) implementations. An ASIC implementation has a performance and area advantage over the other choices [10]. The ASIC design, however, is fixed and can not be modified to cater to different applications. Additionally, the ASIC implementation of a large deep neural network is resource-hungry. Hence, realizing high precision multiple activation functions for a neural network is generally avoided. Furthermore, efficient design technique for lowering the supply voltage is essential for a power-efficient realization of an AF in ANN architecture [11].

The conventional architecture of a neuron with multiple activation functions, proposed in [13] is shown in Fig 2. It has some major drawbacks like area overhead by using separate hardware paths (path-A and path-B) for individual activation function as well as increased data propagation delay (due to MUX) and power dissipation (static power dissipation) due to the unused hardware. The dedicated hardware for the individual activation function is not the desired choice as only one activation function is activated at one time. The higher data precision requirement for these non-linear functions further translates into higher area overhead in hardware implementation.

To overcome these major drawbacks, an area and power-efficient configurable architecture is desirable at all technology nodes. In this regard, CORDIC architecture is investigated for activation function implementation [9]. The CORDIC architecture provides interesting application opportunities to calculate many transcendental algebraic functions such as multiplication, division, hyperbolic tangent, and sigmoid. Various works were proposed to exploit either the underlying parallelism or optimizing design parameters like area, delay or power using CORDIC design techniques. Configurable activation functions are realized using CORDIC design by scaling the input by two in [12]. The scaling of inputs by two implies a double rotation technique in polar coordinates. However, it requires an extra multiplier and subtractor for tanh calculations. Moreover, for computing tanh function, the authors in [12] used an additional multiplication step to the underlying architecture for calculating sigmoid function by carrying out one more multiplication step. These extra steps increase the delay and area of the overall design.

The hyperbolic tangent and sigmoid are generally the most used nonlinear activation functions. Activation functions like sigmoid or tanh provide a smooth transition between excitation and inhibition, which improves the neural response [14]. The authors in [15] proposed the direct computation of a single sigmoid activation function with CORDIC architecture. For negative values, they implemented it with $2's$ complement arithmetic computation. Similarly, the authors employed approximation of log-sigmoid transfer function with CORDIC algorithm in [17] for e^{-x} for sigmoid function realization. However, these techniques require multiple stages for AF realization which lead to high area overheads and longer critical paths. An analog-design-based approach also proposed using a near-threshold CORDIC design technique for low-power applications. An internal sub-block logarithmic shifter and adder of the CORDIC core are designed using dynamic logic [16]. However, none of these works targeted semi-customs ASICs in general.

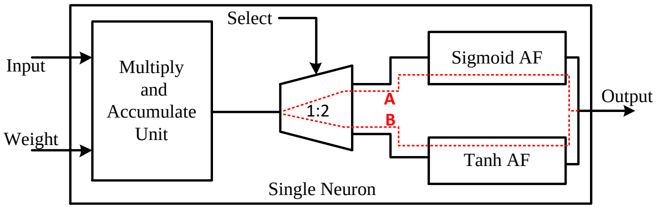


Fig. 2. Typical ASIC-based design architecture of single neuron with configurability in use of multiple activation function.

To overcome the above limitations, we have designed a configurable activation function using CORDIC-based architecture that realizes $sigmoid$ as well as $tanh$ functions. We have proposed an area and power efficient architecture with reduced critical path delay and employed power gating technique for reducing the static power dissipation in configurable activation function design.

III. Recursive Signed CORDIC Architecture

The **CO**ordinate **R**otation **DI**gital **C**omputer (CORDIC) algorithm realizes various mathematical functions by rotating a vector. The underlying principle allows solving the trigonometric relationships involved in plane coordinate rotation and conversion from rectangular to polar coordinates [18].

The CORDIC architecture is configured to operate in three rotation modes – circular, linear or hyperbolic rotation. In this connection, a unified algorithm for linear and hyperbolic CORDIC is an extension of the basic CORDIC algorithm for a circular trajectory as explained in [7]. The equations for hyperbolic coordinate computations includes a rotation matrix and is given in the equation below.

$$\begin{bmatrix} X_{(i+1)} \\ Y_{(i+1)} \end{bmatrix} = \begin{bmatrix} cosh\ \alpha_i & sinh\ \alpha_i \\ sinh\ \alpha_i & cosh\ \alpha_i \end{bmatrix} \begin{bmatrix} X_{(i)} \\ Y_{(i)} \end{bmatrix} \tag{1}$$

Here, (X_i,Y_i) are set of coordinate components representing an i^{th} state. In terms of polar coordinates, an angle α is used where the new coordinates (X_{i+1},Y_{i+1}) can be easily reached. The above equations describes a rotation with scaling factor of the plane vector v_i to v_{i+1} at each iteration. In above equations if we take $cosh\ \alpha_i$ term as common (called as K_i scaling factor) in CORDIC calculation, the equations for all mode of trajectories are thus formulated in unified CORDIC algorithm as:

$$\begin{bmatrix} X_{(i+1)} \\ Y_{(i+1)} \end{bmatrix} = K_i \cdot \begin{bmatrix} 1 & -m\cdot d_i\cdot 2^{-i} \\ d_i\cdot 2^{-i} & 1 \end{bmatrix} \begin{bmatrix} X_{(i)} \\ Y_{(i)} \end{bmatrix} \tag{2}$$

where $\alpha_i = tan^{-1}(2^{-i})$ or $tanh^{-1}(2^{-i})$

Here mode $m \in \{1, 0, -1\}$ indicates a circular, linear, and hyperbolic coordinate system, respectively. The d_i shows the rotation direction for i^{th} iteration. The α_i rotation angle in radians for the each iteration. K_i is scaling factor and it is different for different mode of operations (linear, circular or

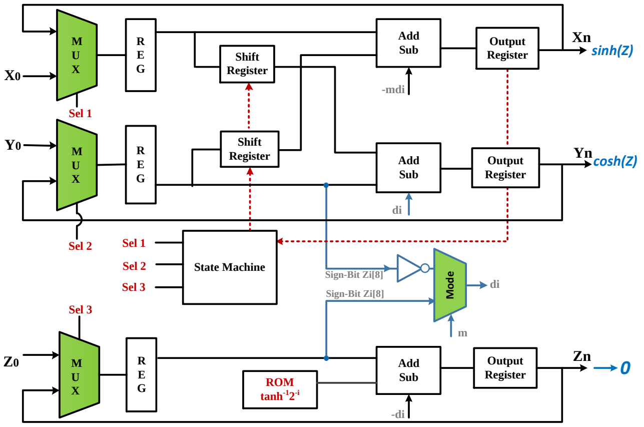


Fig. 3. The efficient recursive sign 8-bit precision CORDIC architecture

hyperbolic). For trigonometric calculation in circular rotation mode, the scalar constant K_i (= $cos\ \alpha_i$) and is 1.6467 whereas, in hyperbolic calculation K_i (= $cosh\ \alpha_i$) is 0.8281 as elaborated in [8]. The functionality of add/sub block used in Fig. 3 depends on the d_i direction. In this connection, add/sub block designed with an identical property of adder/subtractor for output calculation using the equations shown in TABLE I.

The CORDIC architecture for all modes of operation is shown in Fig. 3. The propagation delay of a conventional CORDIC is hence the sum of the delay of MUX, adder/subtractor, barrel-shifter, and feedback resistor which involves one CORDIC unit for each micro-rotation. The modified architecture is iterative in nature as it uses only one CORDIC unit for all micro-rotations.

The functional validation of designed architecture is done for both trigonometric and hyperbolic trigonometric. For easy understanding, we elaborate the working principle of CORDIC architecture for trigonometric calculations (same follows for hyperbolic trigonometric). Here, CORDIC operates in circular mode. In this case, the scale-factor, K converges. The constants used for circular trajectory in the above matrix calculation are K_i = 1.6467 and m = 1. The i^{th} iteration with each step to force the angle to converge to the desired final rotation, uses constants $tan^{-1}(2^{-i})$ for the trigonometric calculations as shown in TABLE II. Moreover, the constant $tanh^{-1}(2^{-i})$ similarly in Table II is first stored in the *ROM* for hyperbolic calculations *(shown in the red box in Fig. 3).*

The calculation is carried out for circular mode calculation of *sin(30)* and *cos(30)*. From TABLE III, inputs X_0[8:0] Y_0[8:0] are applied as $1/K_i$=0.6072 and 0 respectively in coordinate matrix calculation shown in Eq. 2. The input at Z_0[8:0] is the MAC output which has to undergo through activation function. For the coordinate matrix shown above (Eq. 2), the equations then converges as follows:

$$X_{i+1} = X_i - d_i{\cdot}\, Y_i{\cdot}\, 2^{-i}$$
$$Y_{i+1} = Y_i + d_i{\cdot}\, X_i{\cdot}\, 2^{-i}$$
$$Z_{i+1} = Z_i - d_i{\cdot}\, \alpha_i$$

Here, $di = -1$ if $z_i < 0$ and $m = 1$
Using these equation we have given the calculation for trigonometric calculations *sin(30)* and *cos(30)* and is calculated in 5 clock cycles as shown grayed out in TABLE III.

TABLE I
USED ADDER AND SUBTRACTOR FUNCTIONAL SIMILARITY IN CORDIC DESIGN ARCHITECTURE

Components	Sum / Difference	C_{out}/B_{out}
Adder	S = X ⊕ Y ⊕ C_{in}	C_{out} = X ⊕ Y · C_{in} + XY
Subtractor	D = X ⊕ Y ⊕ B_{in}	B_{out} = $\overline{X \oplus Y}$ · B_{in} + $\overline{X}$ Y

TABLE II
THE i^{th} ITERATION WITH EACH STEP TO FORCE THE ANGLE TO CONVERGE TO THE DESIRED FINAL ROTATION.

i α_i=tan^{-1}(2^{-i})	0	1	2	3	4
α_i **in Degrees**	45.00	26.57	14.04	7.13	3.58
α_i **in Radians**	0.7854	0.4636	0.2450	0.1244	0.0624

Similarly, in order to evaluate the *sinh(Z)* and *cosh(Z)*, we choose m = -1 and K_i=0.8281 as the scaling factors in pseudo rotation which is compensated by applying (i) $1/K_i$=1.2075 at X_0[8:0], (ii) Y_0[8:0] = 0 and (iii) MAC output as input at Z_0[8:0]. The direction, d_i is chosen in the range ∈ {-1,1} based on the sign of the previous output i.e current input in the each iteration. After 5 clock cycles (implying 5 iterations), the hyperbolic calculation uses the constants to give the following equations:

$$X_n = K_i{\cdot}\,(X\ coshZ + Y\ sinhZ) \tag{3}$$

$$Y_n = K_i{\cdot}\,(Y\ coshZ + X\ sinhZ) \tag{4}$$

$$Z_n = Input\ at\ Z_0 \longrightarrow 0 \tag{5}$$

After 5 iterations, for next evaluation on the 6th clock cycle, Y is again zero. This gives X_n and Y_n as *cosh* and *sinh* functions respectively of the previous evaluation. The constant K_i is compensated as X is taken as $1/K_i$. The *sinh* and cosh functions are further used to calculate *tanh* and *sigmoid* function as activation function calculation (explained in the following section). Z_n gives the output of activation function applied to MAC output which is the final resultant.

IV. Proposed Architecture for Hybrid Activation Function

Earlier works have proposed the activation functions for ANN using the CORDIC architecture as discussed in section-II. In order to understand the proposed architecture, we have to revisit the works done in [17] and [12]. The authors in [17] used Eq. 6 for realizing sigmoid function by inverting the input to calculate e^{-z}. Moreover, tanh function can not be realized

TABLE III
THE i^{th} ITERATION (CLOCK) STEPS TO FORCE THE ANGLE TO CONVERGE TO THE DESIRED FINAL COS & SIN IN MODIFIED CORDIC ARCHITECTURE IN ROTATION MODE.

i	**d_i**	**$X_{i+1}\to$ cos**	**$Y_{i+1}\to$ sin**	**$Z_{i+1}\to$ 0**
Clock	Direction	$1/K_i$ = 0.6072	0.00	Input = 30.00
0	1	0.6072	0.6072	-15.00
1	-1	0.9108	0.3036	11.56
2	1	0.8349	0.5313	-2.47
3	-1	0.9013	0.4269	4.65
4	1	0.8747	0.4833	1.07

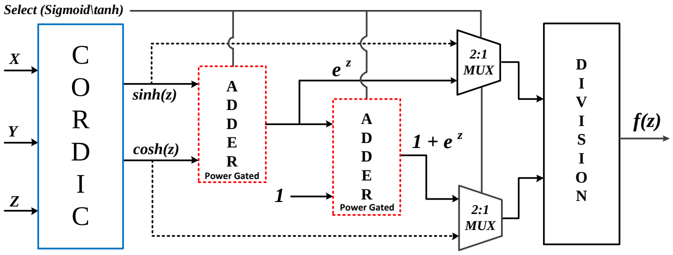


Fig. 4. Block level of proposed configurable CORDIC based activation function

on the same RTL design architecture. The configurable activation functions are realized in [12]. It scales the sigmoid input by two for tanh activation function realization using the Eq. 6 and 7.

$$sigmoid(z) = \frac{1}{1+e^{-z}} = \frac{1+tanh\,(z/2)}{2} \tag{6}$$

$$tanh(z) = 1 - 2\ sigmoid(-2z) = 2\ sigmoid(2z) - 1 \tag{7}$$

To overcome the double multipliers overheads with state-of-the-art [12], we have systematically investigated the configurable architecture for both *sigmoid* and *tanh* AF design. The proposed architecture for configurable activation function block design is shown in Fig. 4. The CORDIC module is used in hyperbolic rotation mode for the generation of *sinh* and *cosh* functions. The signed 8-bit precision CORDIC architecture is designed. The most significant bit (MSB) of 1-bit inputs, $Y[8]$ and $Z[8]$ is used for generating the 'di' signal. The signal 'di' is fed to the adder/subtractor block which decides whether addition or subtraction has to be done as shown in Fig. 3.

The generated trigonometric hyperbolic functions is used for producing exponential function as shown below. The 8-bit CORDIC output is applied to the adder for producing exponential output.

$$sinh(z) + cosh(z) = e^z \tag{8}$$

The configuration between sigmoid or tanh activation function realization is based on the *select* line. The *tanh* function is realized using the Eq. 9. If *select = 1* the CORDIC output directly transfers to the divider through the *MUX*. Similarly, *sigmoid* function is realized using the Eq. 10. For *select = 0*, all logic blocks are active and sequentially data are executed by the design architecture. In the proposed architecture compared with the previous work, we have reduced the area of logic design and overall delay of the function. The realization of tanh function in the proposed architecture can be represented in terms of sigmoid function as shown by below equations.

$$f_1(z) = tanh(z) = \frac{e^z - e^{-z}}{e^z + e^{-z}} = \frac{sinh(z)}{cosh(z)} \tag{9}$$

$$f_2(z) = sigmoid(z) = \frac{1}{1+e^{-z}} = \frac{e^z}{1+e^z} \tag{10}$$

Additionally, in the proposed architecture, the execution of *tanh* function does not require the additional adder block as compared to [12]. However, this unused block can dissipate static power. Addressing this issue and considering the trade-off between leakage current and speed of operation, the *Power Gating* (PG) technique is used to minimize the leakage power and to improve the performance. We have implemented adders with PG technique used in the proposed architecture as shown in Fig. 4. The *select* line is used for power supply connection to the adder logic block. We used logic gate with wide-gate sized transistors for so that the performance is not compromised. The coarse-grain technique is used in the design for better efficiency, less circuit complexity and moderate switching time.

From Eq. 10, the output of the adder is fed to the division block. The subtraction and the shift operation are the two basic operation which are used within the divider circuit [24]. The proposed work further explores the relationship between the tanh and sigmoid using Eq. 9 and 10, and the design architecture allows to implement both *tanh* and *sigmoid* using the same RTL design based on the select line. The input from the MAC unit will be directly applied to the proposed design for the realization of the activation function. This is the key difference between the previous approaches and the proposed architecture, leading to significant performance enhancement.

V. Experiment

To evaluate the proposed design architecture, the configurable activation function design is represented in HDL using Verilog hardware description language. The synthesis results are produced by *Design Compiler*-Synopsys [20]. The netlist file extracted from *Encounter*-Cadence and generated .cdl used for RTL digital design extraction into CMOS design using the *v2lvs*-Mentor Graphics [19]. The extracted design is simulated in *Virtuoso*-Cadence [21] for performance parameter validation at different process corner, temperature and mismatch.

In order to evaluate the impact of process variation and mismatch (which increases significantly in $65nm$ and lower CMOS technology), the circuit is also simulated at all PVT variations. Further, we carry out Monte-Carlo simulation to model the probability of different outcomes of dynamic power. It helps to calculate random variations in dynamic power dissipation due to device-mismatch in the characteristics of identically designed devices, occurring during the manufacturing of ICs.

VI. Results and Discussion

The synthesis results for different precision is shown in Table IV and VI for $180nm$ and $45nm$ technology node

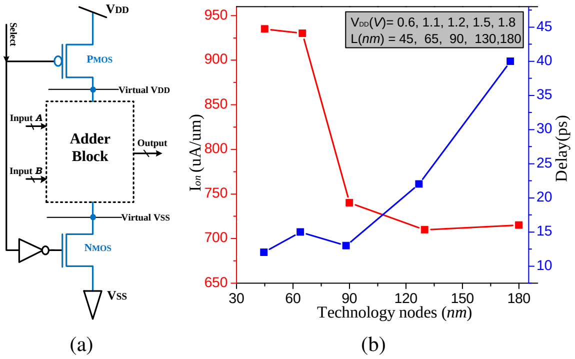
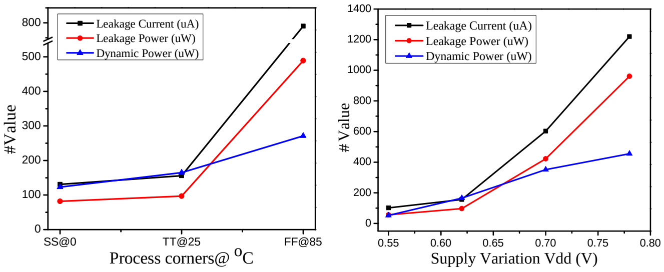


Fig. 5. (a) Basic cells circuit design using power gating technique and *select* pin is use to isolate the circuits from power supply when not in used. (b) Inverter Circuit ON current and Delay variations with respect to technology scaling.



(a)

(b)

Fig. 6. Performance parameter variation of proposed architecture for 45nm technology node (a) the different process corner and temperature at power supply = 0.62V (b) for TT corner with power supply variation.

respectively. At lower technology node, area and dynamic power savings are observed but there is an increases in static power dispassion. These savings are due to scaling in technology model which has an impact on physical parameters such as mobility (μ_0) and saturation velocity (V_{sat}).

At lower technology nodes, static power dissipation is the biggest concern. For the inverter with coarse-grain power gating circuit, *ON* current and delay variations with respect to technology scaling is shown in Fig.5 (b). Based on simulation results and merit, the power gate size should be large as compared to its normal size for handling the amount of switching time. We determine the power gate size for a larger slew rate with a lower response time and the same is used for circuit simulations. The proposed architecture using the coarse-grain technique is shown in Fig. 5 (a). In addition, the circuit design with lower power supply is beneficial as it minimizes power dissipation. However, it comes with an increased propagation delay. The increasing power supply dominates the large leakage current flows which means there is a trade-off between leakage current and speed of operation. The leakage power increases with the supply voltage and it exceeds the dynamic power for supply greater than $0.7V$ at 45nm technology. Moreover, the relation between the power supply and propagation delay in the CMOS circuit is given in Eq. 11.

$$T_d = \frac{C_L \cdot V_{dd}}{(V_{dd} - V_t)^{\alpha}} \tag{11}$$

The CMOS logic based circuit design is extracted in virtuoso cadence and post-layout circuit simulation of proposed configurable AF design is carried out for current and voltage variation at different process corners and supply variation. The parameters calculated at three different FF, TT and SS corners are shown in Fig. 6 (a). It is observed that the static power is more than a dynamic power in the fast-fast (FF) process corner with temperature 85^{o}C. The circuit simulated at different supply voltages with typical-typical (TT) process corner and observed variations in simulation results are shown in Fig. 6 (b). We design and use the adder with the power gating technique in adder block as shown in Fig. 5. We have designed and simulated the circuit designs from [12], [17] at 45nm process technology. Results and comparison of proposed architecture with and without power gating and state-of-the-art is shown in Table IV.

The proposed architecture is synthesized for different precision to have a fair comparison between CORDIC based proposed architecture and combinational logic [23] design for *sigmoid* function. The synthesis results at *180nm* and *45nm* technology node are shown in Table V and Table VI respectively, for 8-bit, 12-bit and 16-bit precision digital design. The results comparison shown in tables concludes that CORDIC based architecture is having linear increments in physical parameters such as area and power with respect to increment in bit precision. However, in combinational logic design, this increase is exponential. The proposed technique with CORDIC architecture offers a substantial saving of area over the combinational logic-based design proposed in [23]. It concludes that CORDIC based architecture is an admirable choice for higher precision AF implementation. However, the number of *clk* edges required is more as compared to combinational design. Hence, CORDIC based design is favorable for applications where the energy and area requirements are less. Hence, they offer an excellent choice for IoT devices where there is a tight area and power budget.

TABLE IV
PERFORMANCE PARAMETER METRICS OF ACTIVATION FUNCTION DESIGN USING CORDIC ARCHITECTURE FOR PROPOSED AND STATE-OF-THE-ART AT 45NM TT PROCESS CORNER

Sigmoid AF Design Arch. Parameters	**Rotation Scaling [12]**	**Inverted Input [17]**	**Proposed without PG**	**Proposed with PG**
Area (μm^2)	2983	2735	2145	2280
St. Power (μW)	176	121.7	96.7	72.94
Dy. Power (μW)	299.4	209.5	176	176
Delay (ns)	-	5.93	4.30	4.71
Precision $(bits)$	8	8	8	8

At lower technology, process and mismatch is also an important issue in terms of power dissipation, stability, and reliability. We have simulated the circuit at *45nm* technology node. We observed area saving as compared to the previous work as shown in Table IV. The Monte-Carlo simulation for dynamic power variation due to process and mismatch is carried out for 1000 samples for the proposed architecture and the state-of-the-art and shown in Fig. 7. The mean dynamic power and σ deviation of proposed architecture are $180.73\mu W$ and $51.7\mu W$ respectively. The mean dynamic power of proposed work is 79% compared to architecture proposed by [17] and 60% compared to [12]. The σ deviation is $51.7\mu W$ in power variation for proposed design which is less compared to the state-of-the-art with a similar approach as it is $66.15\mu W$ and $78\mu W$ in [17] and [12] respectively. It shows the proposed architecture is more reliable in terms of power variation due to process and mismatch.

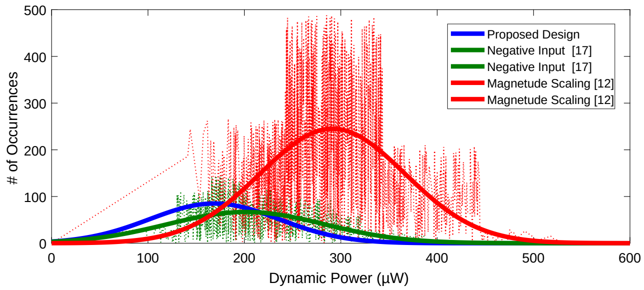


Fig. 7. 1000 Monte Carlo simulation with process variation and mismatch for mean dynamic power variation of signed 8-bit activation function.

TABLE V
parameter metrics @180nm TT Process Corner for activation function design architecture using Combinational Logic [23] and CORDIC based architecture

Sigmoid Function	**Combinational**	**Dynamic**	**Leakage**	**Critical Path**	**Required**
Bit Precision	**Area** (μm^2)	**Power** (μW)	**Power** (μW)	**Delay** (ns)	**No. of Clocks**
8_bit Combinational	1637.50	77.50	18.70	7.04	1
8_bit CORDIC	2145.00	510.00	72.10	6.11	5
12_bit Combinational	17 735.00	804.70	190.00	8.80	1
12_bit CORDIC	2377.50	635.00	85.00	7.63	5
16_bit Combinational	83 796.50	35 708.50	665.30	15.86	1
16_bit CORDIC	2825.00	895.80	119.40	8.87	5

TABLE VI
parameter metrics @45nm TT Process Corner for activation function design architecture using Combinational Logic [23] and CORDIC architecture

Sigmoid Function	**Combinational**	**Dynamic**	**Leakage**	**Critical Path**	**Required**
Bit Precision	**Area** (μm^2)	**Power** (μW)	**Power** (μW)	**Delay** (ns)	**No. of Clocks**
8_bit Combinational	631.70	33.73	19.12	4.76	1
8_bit CORDIC	307.40	176.60	96.77	4.30	5
12_bit Combinational	4899.50	198.30	110.90	5.10	1
12_bit CORDIC	658.40	252.04	130.30	4.92	5
16_bit Combinational	23 408.20	8542.70	3960.00	9.53	1
16_bit CORDIC	782.30	257.40	141.50	5.12	5

VII. Conclusion

The semi-custom ASIC approach is investigated for CORDIC architecture. The contribution of the work is two-fold. First, the proposed configurable activation function is able to realize both tanh and sigmoid AF, using the same VLSI architecture. The simulated and synthesized results prove that the proposed CORDIC based AF model is the desirable choice in ASIC based high precision artificial neural network. To draw the quantitative comparison, with different precision performance parameters were calculated. The semi-ASIC/SoC design model is synthesized at the 45nm technology node and validated using ModelSim. Secondly, the power gating technique is used for saving static power dissipation. This model can be used for the different types of activation function by changing the select signal value, producing multi activation function in neural networks: tanh, sigmoid, etc.

Acknowledgement

The authors would like to thank the HiPEAC Collaboration Grant, European FP7 ICT Cooperation program for providing financial support, and University Grant Commission (UGC), Government of India and Special Manpower Development Program Chip to System Design, Department of Electronics and Information Technology (DeitY) under the Ministry of Communication and Information Technology, Government of India for providing necessary research facilities. This work is supported in part by the German Research Foundation (DFG) within the ”Center for Advancing Electronics Dresden” (CfAED) at the Technische Universitat Dresden

References

[1] Du, Ke-Lin, and M. N. S. Swamy. “Neural Network Circuits and Parallel Implementations.” In Neural Networks and Statistical Learning, Springer, 2019.

[2] Umuroglu, Yaman, et al. “FINN: A framework for fast, scalable binarized neural network inference.” ISFPGA, 2017.

[3] Wong, Henry, Vaughn Betz, and Jonathan Rose. “Comparing FPGA vs. custom CMOS and the impact on processor microarchitecture.” ISFPGA, 2011.

[4] Basterretxea, Koldo, et al. “An experimental study on nonlinear function computation for neural/fuzzy hardware design.” IEEE trans. neural networks 2007.

[5] Meher, Pramod K and Valls, Javier and Juang, Tso-Bing and Sridharan, K and Maharatna, Koushik. “50 years of CORDIC: Algorithms, architectures, and applications.” In TCAS-I, 2009.

[6] Vachhani, Leena, et al. ”Efficient CORDIC algorithms and architectures for low area and high throughput implementation.” TCAS-II, 2009.

[7] Lang, Tomas, and Elisardo Antelo. “CORDIC-based computation of ArcCos and ArcSin.” In ASAP, 1997.

[8] Meher, Pramod K., et al. ”50 years of CORDIC: Algorithms, architectures, and applications.” IEEE Transactions on Circuits and Systems I: Regular Papers 56.9 (2009): 1893-1907.

[9] Qian, Meng. “Application of CORDIC algorithm to neural networks VLSI design.” Computational Engineering in Systems Applications”. Vol. 1. IEEE, 2006.

[10] Kuon, Ian, and Jonathan Rose. “Measuring the gap between FPGAs and ASICs.” TCAD, 2007.

[11] Arthurs, Aaron, Justin Roark, and Jia Di. “Ultra-low voltage digital circuit design: A comparative study.” In 2012 IEEE Faible Tension Faible Consommation, pp. 1-4. IEEE, 2012.

[12] Tiwari, Vipin, and Nilay Khare. “Hardware implementation of neural network with Sigmoidal activation functions using CORDIC.” Microprocessors and Microsystems-2015.

[13] Raut, Gopal, et al. “Efficient Low-Precision CORDIC Algorithm for Hardware Implementation of Artificial Neural Network.” International Symposium on VLSI Design and Test. Springer, Singapore, 2019.

[14] Wedlake, Martine, and Harry L. Kwok. “A CORDIC implementation of a digital artificial neuron.” 1997 IEEE Pacific Rim Conference on Communications, Computers and Signal Processing, PACRIM.

[15] Pottathuparambil, Robin, and Ron Sass. “An FPGA-based neural network for computer vision applications.” Workshop on Computer Vision on Low-Power Reconfigurable Architectures, Chania, Crete, Greece. 2011.

[16] Chou, Pei-Yuan, et al. “Near-Threshold CORDIC Design with Dynamic Circuitry for Long-Standby IoT Applications.” 31st IEEE International System-on-Chip Conference (SOCC). IEEE, 2018.

[17] Alçın, Murat, et al. “Hardware design and implementation of a novel ANN-based chaotic generator in FPGA.” Optik 127, 2016.

[18] J. E. Volder, ”The CORDIC Trigonometric Computing Technique,” in IRE Transactions on Electronic Computers, 1959.

[19] https://communities.mentor.com/docs/DOC-3114

[20] https://www.synopsys.com/implementation-and-signoff/rtl-synthesis-test/design-compiler-graphical.html

[21] https://www.cadence.com/en_US/home/tools/custom-ic-analog-rf-design/circuit-design/virtuoso-schematic-editor.html

[22] Thamaraimanalan, T., and P. Sampath. “Leakage Power Reduction in Deep Submicron VLSI Circuits Using Delay-Based Power Gating.” National Academy Science Letters (2019).

[23] Rani, SP Joy Vasantha, and P. Kanagasabapathy. “Multilayer perceptron neural network architecture using vhdl with combinational logic sigmoid function.” Inter. Conference on Signal Processing, Communications and Networking, IEEE, 2007.

[24] Sorokin, Nikolai. “Implementation of high-speed fixed-point dividers on FPGA.” Journal of Computer Science Technology 6 (2006).