An Ultra-Low-Power Serial Implementation for Sigmoid and Tanh Using CORDIC Algorithm

Yaoxing Chang*†, Petar Jokic*, Stephane Emery* and Luca Benini†
*CSEM SA, Switzerland; †ETH Zurich, Switzerland; Email: yaoxing.chang@csem.ch

***Abstract*—Activation functions (AFs) such as sigmoid and tanh play an important role in neural networks (NNs). Their efficient implementation is critical for always-on edge devices. In this work, we propose a serial-arithmetic architecture for AFs in edge audio applications using the CORDIC algorithm. The design enables to dynamically trade-off throughput/latency and accuracy, and possesses higher area and power efficiency compared to conventional methods such as look-up table (LUT) and piece-wise linear (PWL)-based methods. Considering the throughput difference among the designs, we evaluate average power consumption taking into account active and idle working cycles for same applications. Synthesis results in a 22nm process show that our CORDIC-based design has an area of 545.77 µm^2 and an average power of 0.69 µW for a keyword spotting task, achieving a reduction of 36.92% and 71.72% in average power consumption compared to LUT and PWL-based implementations, respectively.**

***Index Terms*—CORDIC, edgeML, sigmoid, tanh, serial architecture**

I. Introduction

Neural networks (NNs) are becoming ubiquitous among edge devices, where area and power efficiency are the key design targets. While spatial NNs often use simple activation functions (AFs), recurrent NNs require more complex sigmoid and tanh functions in audio applications. However, implementations of such non-linear functions are generally hardware-intensive. Various methods have been investigated for implementing AFs, such as look-up table (LUT)-based methods [1], piece-wise linear (PWL) and non-linear approximations [2], and CORDIC-based approaches [3]. LUT AFs use a set of registers to approximate the target function with finite values (entries). Higher accuracy can be attained with more entries at the cost of a dramatic growth of hardware resources (and power consumption). Instead, one can store only the coefficients of a piece-wise function and use arithmetic units to calculate proximal results. Less storage is needed in PWL AF for comparable accuracy, while additional computing units such as multipliers and adders will add extra costs. CORDIC AFs use only hardware-friendly shift and add operations [4]. Fewer registers and arithmetic units are required, at the cost of lower throughput and higher latency due to their iterative approximation.

Various works have proposed solutions to the major drawbacks of the CORDIC methods, *i.e.* latency and throughput. In [3], a high-speed pipeline implementation is proposed where extra LUTs are applied to reduce the iterations. In [5], a high-throughput pipeline implementation with adjustable precision is presented, and in [6], extra adders and dividers are deployed, where the CORDIC block only computes hyperbolic sine and cosine functions. However, the throughput and latency of the AF blocks are generally not the bottlenecks in audio applications due to their low signal bandwidth. The inference time is dominated by the multiply-accumulate (MAC) operations in the weighted sum that precedes AFs in the computation of neurons' activations. Instead, power consumption is more critical, especially for always-on blocks on edge devices. Yet these works do not cover approaches for low-power design and fully exploit its area and power efficiency.

In this work, we propose an ultra-low-power serial-arithmetic architecture for sigmoid and tanh using the CORDIC algorithm. We compare the performance with implementations of a LUT-based method and a PWL-based approximation and evaluate the arithmetic error under different quantization and iteration settings. We further investigate the inference accuracy on LSTM models in a keyword spotting (KWS) task.

This work was supported in part by the Electronic Components and Systems for European Leadership Joint Undertaking ANDANTE under Grant 876925.

II. Methods and Implementation

To implement CORDIC AF, we derive the sigmoid function as (1), where $\cosh(x)-\sinh(x)$ can be calculated by CORDIC in hyperbolic rotation mode, and $\frac{1}{x}$ can be obtained in linear vectoring mode. As shown in Fig. 1a), we reuse one generic CORDIC block for both modes using a time-serial strategy. The number of iterations can be changed at run-time to trade-off accuracy and latency/throughput.

$$sigmoid(x) = \frac{1}{1+e^{-x}} = \frac{1}{1+\cosh(x)-\sinh(x)} \quad (1)$$

For LUT AF, we implement a small LUT with 16 16-bit entries. A search tree is used to select corresponding entries based on input values, as shown in Fig. 1b). For PWL AF, we use first-order Taylor expansion where a function f(x) can be presented as $f(x) = f(a) + xf'(a) - af'(a) = f'(a)x + (f(a) - af'(a)) = Ax + B$. Coefficients A and B are pre-calculated and stored in two 16-bit register sets. Two entry distributions are investigated in LUT and PWL approaches, as shown in Fig. 2, where entry points are evenly distributed along axis x(Dx) and y(Dy), respectively. For all three approaches, we implement tanh as $tanh(x) = 2sigmoid(2x)-1$ using shift and add operations based on the sigmoid computation flow.

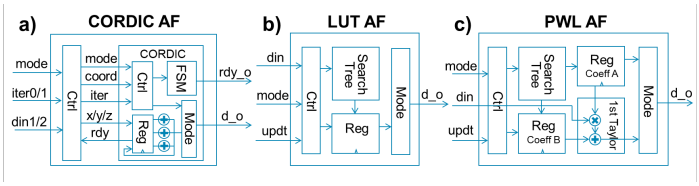

Fig. 1. Block diagrams of CORDIC AF, LUT AF and PWL AF.

978-3-9819263-7-8/DATE23/© 2023 EDAA

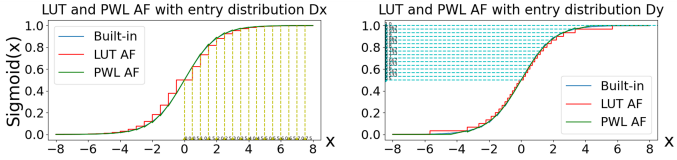

Fig. 2. Two entry distributions in LUT AF and PWL AF, namely Dx and Dy.

III. Results

Fig. 3 shows the mean squared error (MSE) compared with float64 results for Gaussian random inputs. PWL AFs achieve 1 to 2 orders of magnitude lower MSE compared to LUT AFs with the same number of entries. For CORDIC AFs, fixed-point width (quantization) determines the upper limit of the accuracy, while the configurable iterations provide the ability to trade-off latency/throughput and accuracy at run time.

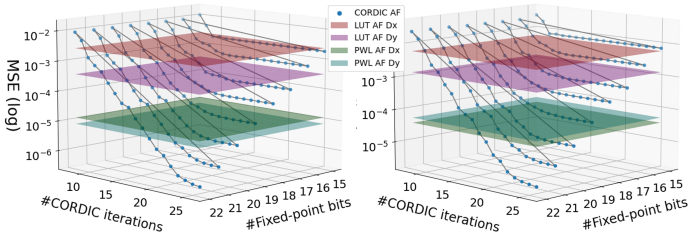

Fig. 3. MSE sweep under different quantization and iteration settings, for sigmoid (left) and tanh (right) functions.

To evaluate the AF performance on NN applications, a 10-command KWS task is used on an LSTM model with a unit size of 118 and a feature size of 10. AFs are directly replaced from the pre-trained model without re-training. Fig. 4 shows the KWS accuracy with and without post-training quantization.

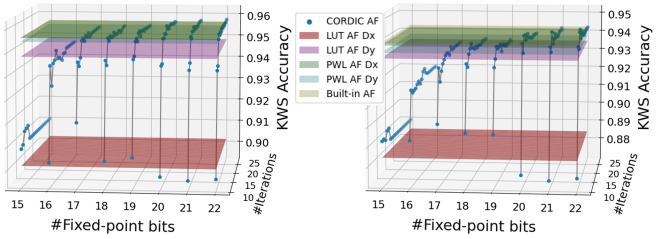

Fig. 4. KWS accuracy of LUT, PWL and CORDIC-based design without model quantization (left) and with post-training model quantization (right).

A 20-bit CORDIC design is selected for area and power comparison. LUT and PWL AF follow the same 16-bit designs as in error/accuracy evaluation. GF22 process at room temperature and typical corner is used for all implementations. The minimal clock frequency is set as 25 kHz to fulfil the real-time system throughput requirement for KWS models at a standard sampling frequency of 16 kHz. Table I shows the hardware specifications compared with prior works.

Typically, the throughput of NNs is constrained by repetitive MAC operations computed in the parallel processing elements (PEs) dedicated to the linear part of the model. Non-linear operations are executed in pipeline where AFs should be calculated before the next MAC results are ready. The maximum iteration for CORDIC AFs in LSTMs can be given as $iter_{max} = (F + U)/\#PEs$, where F and U are the feature size and unit size of the model. To ensure fair comparisons between the AF designs, we perform an average power sweep

TABLE I
Hardware Specifications Compared with SOTA

	Tech Node	*#Bits*	*Area (μm^2)*	*f_{clk} (Hz)*	*$P_{leakage}$ (μW)*	*P_{total} (μW)*
CORDIC AF*	22nm	16	545.77	25k	0.675	0.703
LUT AF*		16	770.22		1.092	1.104
PWL AF*		20	1866.13		2.435	2.466
PWL SC [2]	90nm	8	3650	-	-	81.62
CORDIC [6]	45nm	8	2280	-	72.94	248.94
CORDIC [5]	40nm	⩾25	36513	1G	-	12350

*Proposed AF designs.

at different frequencies and CORDIC iteration settings in Fig. 5, considering active and idle working cycles in the same KWS applications. Average power consumption can be calculated as $P_{avg} = P_{leakage} + P_{dynamic} * (iter_{act}/iter_{max})$. We investigate two scenarios with different model and hardware sizes. For smaller models as in the left graph, NN systems can work on lower frequencies(*e.g.* 25kHz), where leakage power dominates. CORDIC AF has an average power consumption of 0.69 µW, showing a significant power benefit among the designs thanks to its high area efficiency. For larger LSTM models, the required throughput for AF blocks reduces due to the data-dependency (waiting for the MAC operations). The ratio of active AF cycles is lower, making the average power consumption of CORDIC AF the lowest even at higher clock frequencies (*e.g.* 1MHz).

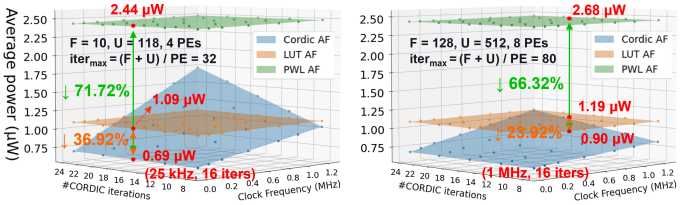

Fig. 5. Average power consumption considering active/idle cycles, for a model size of 16F118U with 4PEs (left), and 128F512U with 8PEs (right).

IV. Conclusion

We proposed an ultra-low-power CORDIC-based serial-arithmetic architecture for sigmoid and tanh computation in edgeML applications and compared its performance to traditional LUT and PWL-based approaches. In a 22nm process at 25kHz, our CORDIC AF design achieved a reduction of 29.14% in area and 36.92% in average power consumption compared to a lower-precision LUT AF, and a reduction of 70.75% in area and 71.72% in power compared to a PWL-based design with a similar level of accuracy.

References

[1] F. Piazza et al. Neural networks with digital LUT activation functions. In *IJCNN*, 1993.

[2] V.-T. Nguyen et al. An efficient hardware implementation of activation functions using stochastic computing for deep neural networks. In *MCSoC*, 2018.

[3] X. Chen et al. Efficient sigmoid function for neural networks based FPGA design. In *LNCS*. 2006.

[4] J. E. Volder. The CORDIC trigonometric computing technique. *IEEE Trans Comput.*, 1959.

[5] H. Chen et al. A CORDIC-based architecture with adjustable precision and flexible scalability to implement sigmoid and tanh functions. In *ISCAS*, 2020.

[6] G. Raut et al. A CORDIC based configurable activation function for ANN applications. In *ISVLSI*, 2020.