ASP-DAC 2025

30th Asia and South Pacific Design Automation Conference

HyPPO: Hybrid Piece-wise Polynomial Approximation and Optimization for Hardware Efficient Designs

LAKSHMI SAI NIHARIKA VULCHI, VALIPIREDDY PRANATHI, MAHATI BASAVARAJU AND MADHAV RAO

IIIT Bangalore, India

TABLE OF CONTENTS

ABSTRACT

- Hybrid Piece-wise Polynomial Approximation (PW-Hybrid) improves hardware implementation of non-linear functions by combining linear (PWL) and quadratic (PWQ) methods to reduce approximation errors.

- Particle Swarm Optimization (PSO) fine-tunes quantized bit-widths for polynomial coefficients, improving hardware efficiency. The PSO optimized hardware design is evolved for different non-linear functions.

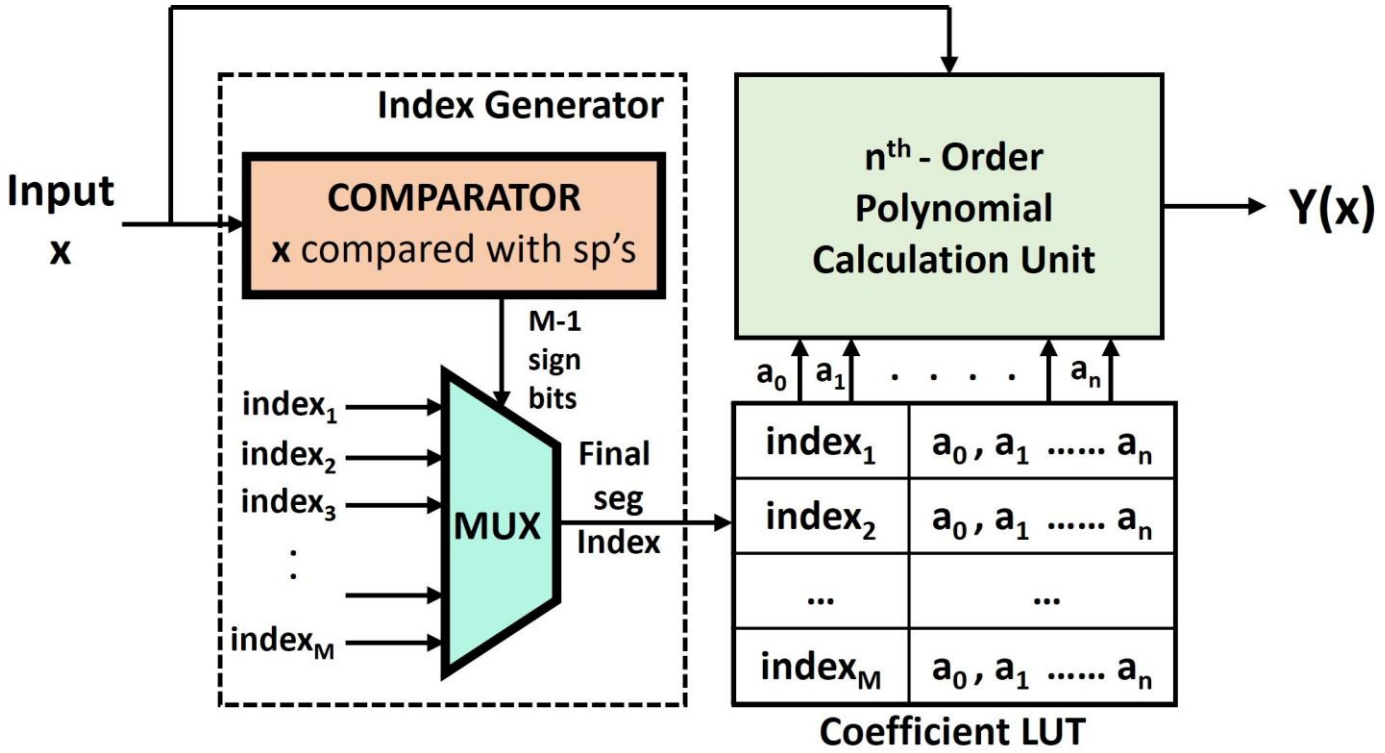
- Significant reductions in hardware resource usage and critical path delay achieved, evaluated using Cadence 45nm gpdk library.

BACKGROUND INFORMATION

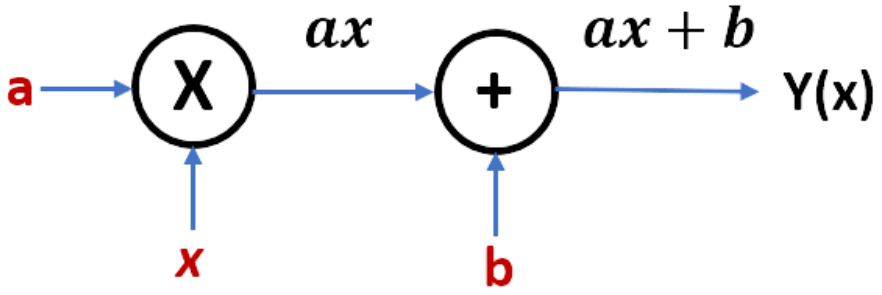
In Piece-wise Polynomial (PW-Poly) approximation, the domain of a function is divided into segments, and a polynomial is employed to define the function in each segment

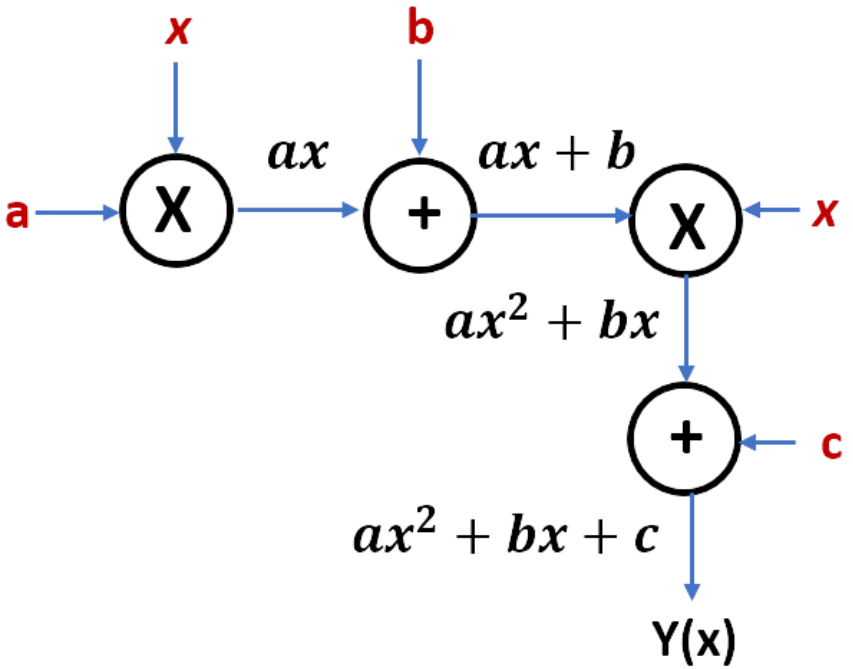
Approximation of non-linear functions

- Piece wise linear – Segment defined by a linear function, greater number of segments, requires more memory to store the coefficients
- Piece wise quadratic – Segment defined by a quadratic polynomial, computational hardware is high

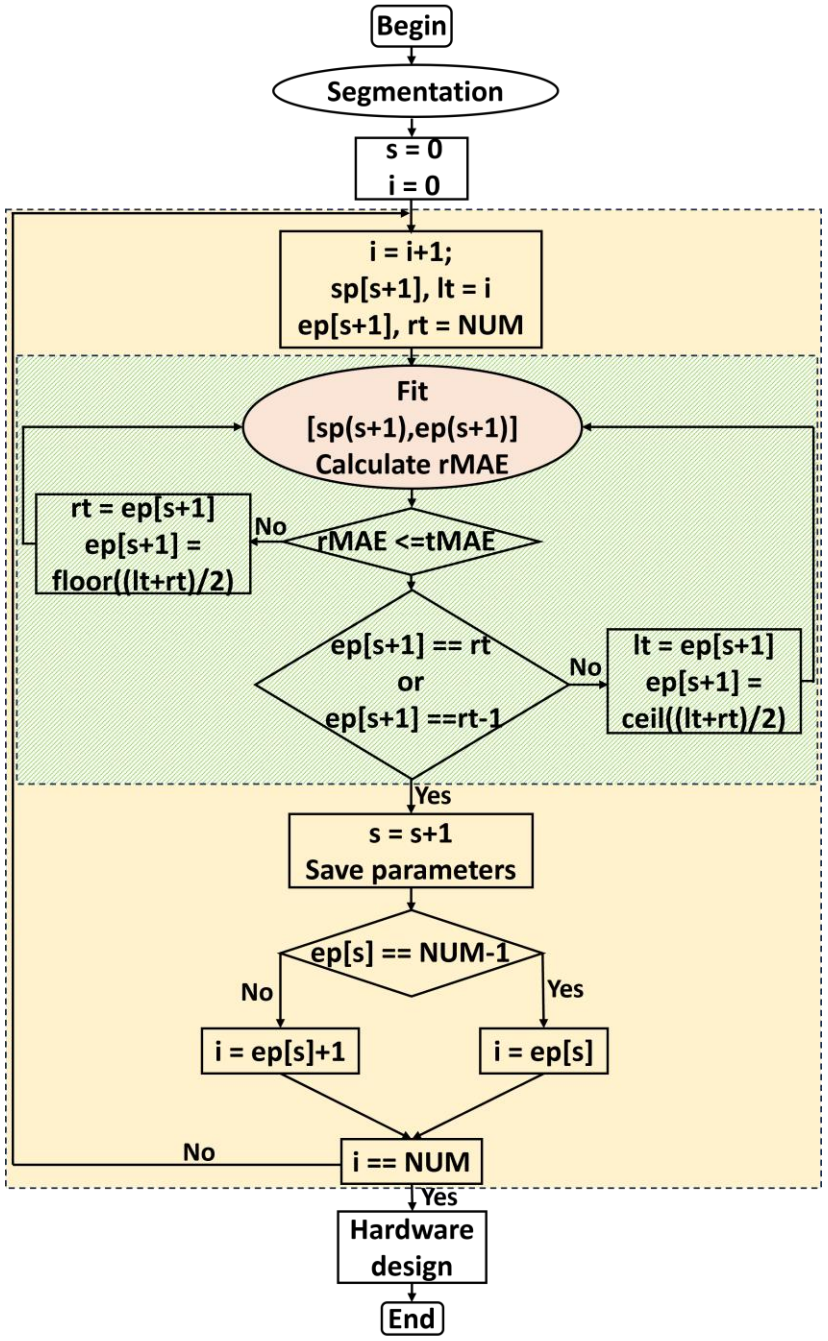


The hardware identifies the segment containing input 'x', retrieves the corresponding polynomial coefficients from a LUT, and uses them for approximation.



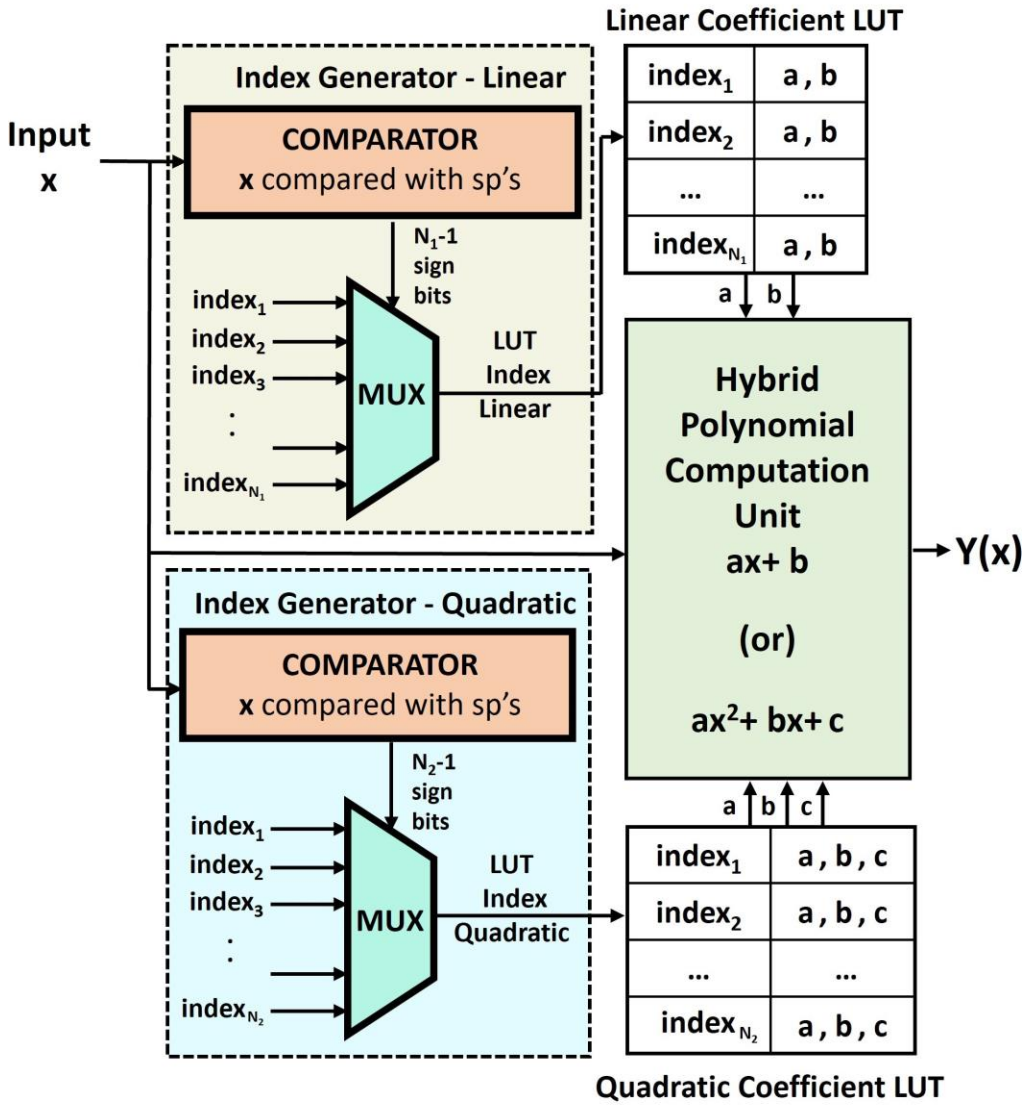


The coefficients are used in addition and multiplication units to compute the polynomial and generate the approximated output for the target function.



Polynomial coefficients are computed through two steps: **fitting**, which minimizes the maximum absolute error (MAE), and **segmentation,** which optimally divides the target interval into segments.

PROPOSED METHOD – Piecewise Hybrid



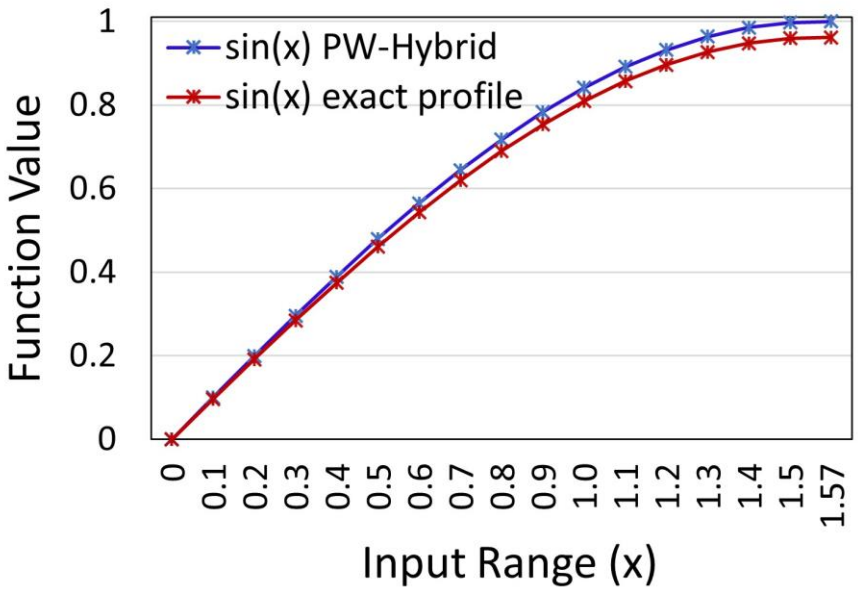
- PWL requires more segments than PWQ to meet error conditions but offers faster computation, while PWQ involves more operations.
- This work combines polynomials of different degrees (1 and 2) across segments for efficient non-linear function approximation.

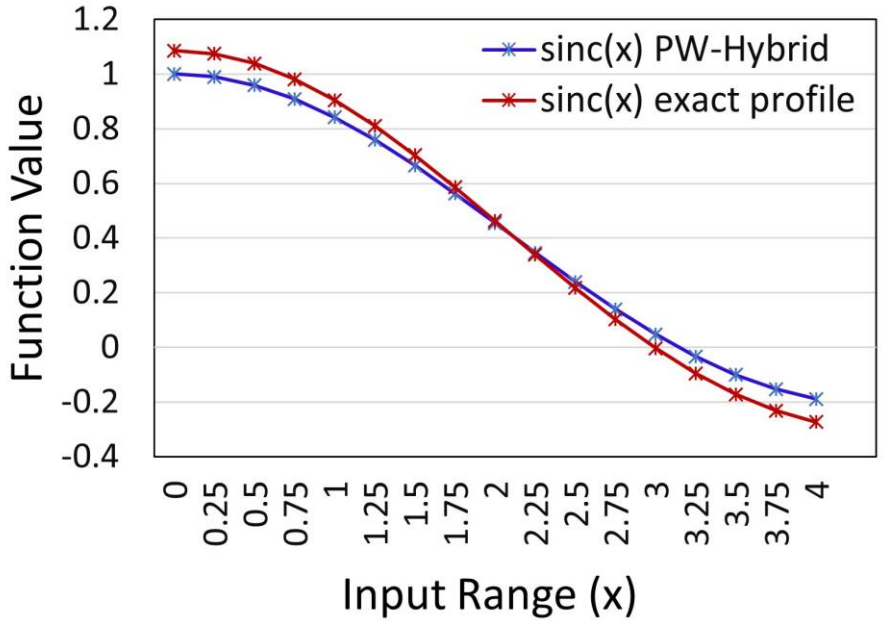
PSO Designed Coefficient Bit Widths

- Particle Swarm Optimization (PSO) optimizes bit widths to minimize area, power, and delay product while ensuring accuracy in hardware performance.
- The bit widths for the coefficients a, b, and c, which are determined after the segmentation and fitting steps, have been optimized.
- For each quadratic segment, count its linear segments; use Linear Approximation for segments with fewer linear mappings, keeping the rest as Quadratic Approximation.

EXPERIMENTAL RESULTS

PW-Hybrid vs Exact

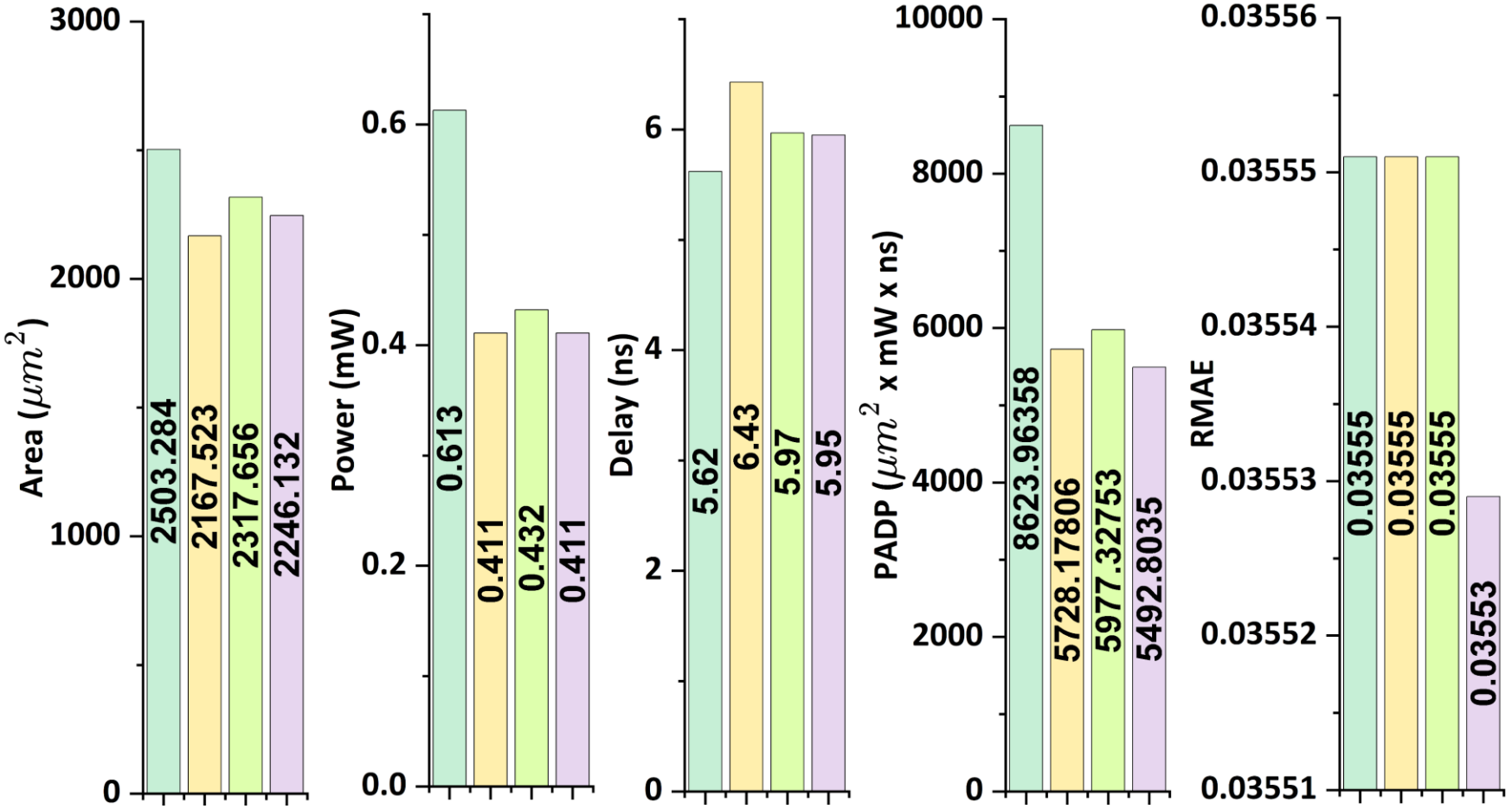




ASIC Results

$Function : \mathbf{sinc(x)}$
$IW : 16$
$Interval : (0, 4)$

Method	Coefficient Bitwidths	Segments
PWL	{16,16}	97
PWQ	{16,16,16}	17
PW-Hybrid	{16,16}, {16,16,16}	6L, 11Q
HyPPO	{13,11}, {15,13,10}	6L, 11Q

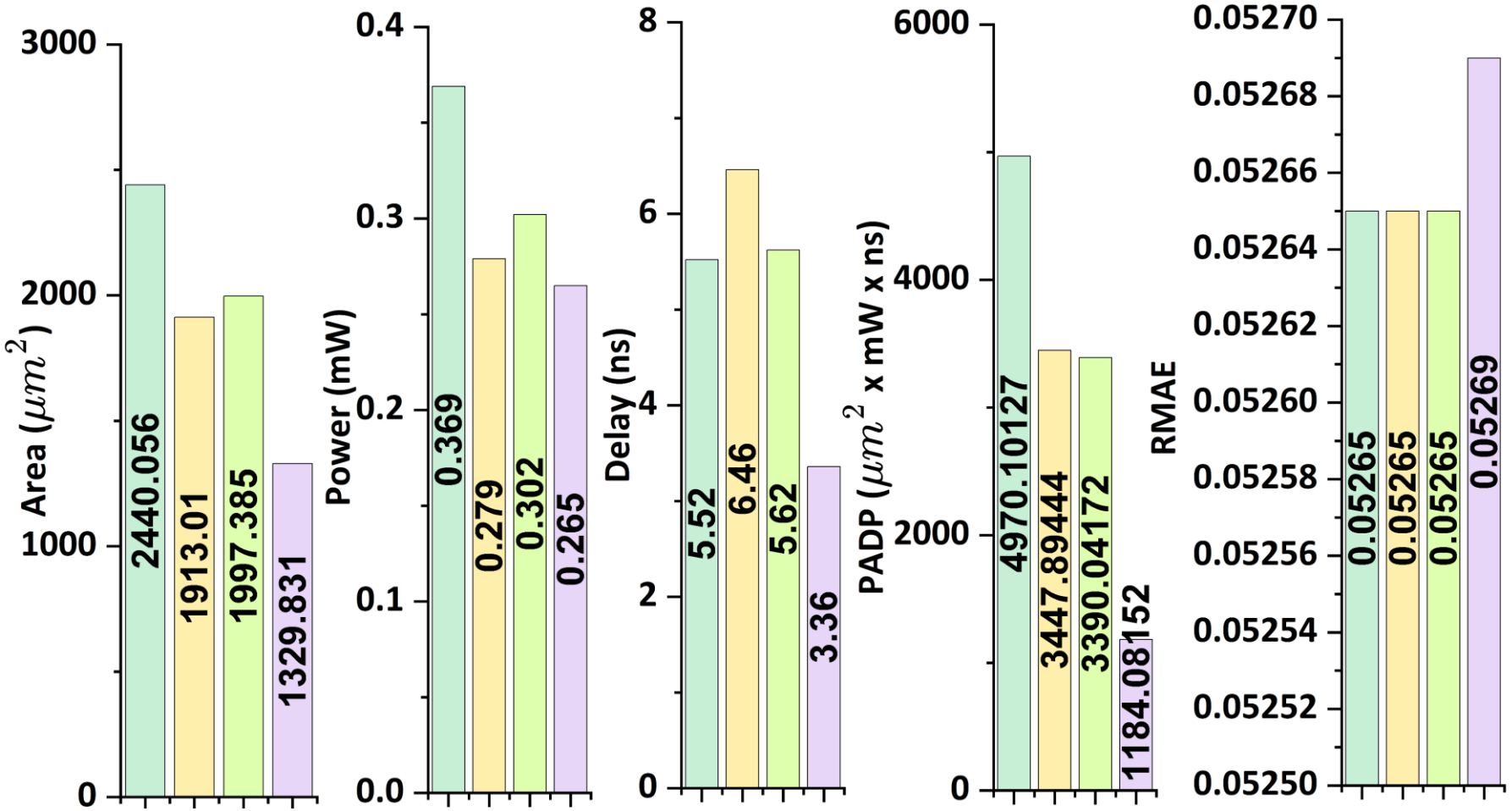


$Function : \mathbf{sin(x)}$

$IW : 16$

$Interval : (0, \pi/2)$

Method	Coefficient Bitwidths	Segments
PWL	{16,16}	75
PWQ	{16,16,16}	10
PW-Hybrid	{16,16}, {16,16,16}	3L, 7Q
HyPPO	{14,13}, {11,9,9}	3L, 7Q



ASIC Results – PWL vs PWQ comparison

Function	Method	Interval	IW	Segments	Area (μm^2)	Power (mW)	Delay (ns)	PADP ($\mu m^2 \times mW \times ns$)
$log_2(1+x)$	PWL	(0,1)	16	12	1700.6766	0.0224	10.274	39.141
	PWQ	(0,1)	16	3	2076.167	0.0323	11.735	78.695
tanh(x)	PWL	(0,1)	8	3	577.98	0.0069	4.34	17.500
	PWQ	(0,1)	8	3	615.78	0.0071	4.71	20.700
Softsign(x)	PWL	(-8,8)	8	15	839.062	0.018	6.648	10.110
	PWQ	(-8,8)	8	6	975.66	0.019	7.109	13.710
Sigmoid(x)	PWL	(0,8)	8	2	313.44	0.003	3.48	3.28 0
	PWQ	(0,8)	8	2	356.84	0.00369	4.94	6.5
exp(x)	PWL	(0,2)	8	20	1617.705	0.0343	4.94	273.987
	PWQ	(0,2)	8	4	1222.090	0.0243	5.52	163.926

ASIC Results – PWL vs PSO optimized Comparison

Function	Method	Interval	Coefficient Bit-Widths	IW	Segments	Area (μm^2)	Power (mW)	Delay (ns)	PADP ($\mu m^2 \times mW \times ns$)	RMAE
$log_2(1+x)$	PWLO	(0,1)	{16,9}	16	15	1397.412	0.00158	9.77	21.7	0.00572
	[17]	(0,1)	{15,11}	16	15	1560.000	0.00193	9.50	28.6	0.00572
tanh(x)	PWLO	(0,1)	{10,4}	8	3	361.000	0.00573	4.04	8.36	0.004471
	[17]	(0,1)	{8,5}	8	4	395.000	0.004670	4.02	7.42	0.006305
Softsign(x)	PWLO	(-8,8)	{9,5}	8	15	639.198	0.013	5.61	4.68	0.0002140
	[17]	(-8,8)	{9,9}	8	19	679.896	0.0136	5.87	5.42	0.00021399
Sigmoid(x)	PWLO	(0,8)	{10,2}	8	4	167.000	0.00224	2.79	1.04	0.0120
	[17]	(0,8)	{8,5}	8	2	157.000	0.00245	2.94	1.13	0.0122
exp(x)	PWQO	(0,1)	{12,9,7}	8	4	977.760	0.0176	5.16	88.796	0.0016350
	[17]	(0,1)	{11,10,9}	8	4	989.230	0.0183	5.43	98.298	0.001649271

[17] Haoran Geng, Xiaoliang Chen, Ning Zhao, Yuan Du, and Li Du. Qpa: A quantization-aware piecewise polynomial approximation methodology for hardware-efficient implementations. IEEE Transactions on Very Large Scale Inte gration (VLSI) Systems, 31(7):931–944, 2023.

Percentage Improvement

Hardware Design	Area-delay-power product improvement
PWL over PWQ - log(1+x)	50.26%
PWL over PWQ - softsign(x)	49.53%
PWQ over PWL – exp(x)	40.17%
PWLO over PWL – log(1+x)	24.47%
PWLO over PWL – softsign(x)	14.76%
HyPPO over PW-Hybrid – sinc(x)	65.06%

CNN Accuracy for Activation Functions

- The study compares the ReLU activation function in AlexNet and VGG-11 architectures with Tanh, Sigmoid, and Softsign activations, applying them in over 40% of the layers. The performance is evaluated on the CIFAR-10 dataset using PyTorch.

- The comparison between PyTorch's in-built activation functions and the proposed Piecewise Linear Optimization (PWLO) design shows that PWLO achieves comparable accuracy, with an average accuracy drop of 3.3% and a minimum drop of 1%.

- In the VGG-11 architecture, Tanh and Softsign show a slight accuracy drop of over 3% compared to in-built methods. In AlexNet, Tanh and Sigmoid maintain comparable accuracy, while Softsign experiences a drop exceeding 6%.

Activation Function	Implementation Method	CNN Architecture	
		AlexNet	VGG-11
Tanh	Exact	83.06%	72.88%
	Proposed	82.42%	69.76%
Sigmoid	Exact	66.92%	67.16%
	Proposed	63.46%	67.76%
Softsign	Exact	83.60%	77.66%
	Proposed	77.17%	74.03%

CONCLUSION

- The PW-Hybrid method reduces the number of segments compared to SOTA PWL approximations, conserving silicon footprint and improving computational delay.
- PSO enabled coefficient bit-widths are found to make the designs more hardware efficient
- The HyPPO method, with PSO-optimized coefficient bit-widths, achieves a 65.06% PADP improvement over PW-Hybrid for the Sine function. The optimal bit-widths for hardware efficiency are (11,9,9) for quadratic segments and (14,13) for linear segments, with similar gains observed for other non-linear functions.

Questions are Welcome!

T H A N K Y O U

C O N T A C T U S

E-mail
- niharika.lakshmi@iiitb.ac.in
- valipireddy.pranathi@iiitb.ac.in
- mahati.basavaraju@iiitb.ac.in
- mr@iiitb.ac.in