

CHALMERS
UNIVERSITY OF TECHNOLOGY

High-Throughput Low-Power Digital Predistorter Based on Pruned Volterra Series

Downloaded from: https://research.chalmers.se, 2026-11-22 12:29 UTC

Citation for the original published paper (version of record):
Mubanda, T., Muñoz Bautista, J., Eriksson, T. et al (2026). High-Throughput Low-Power Digital Predistorter Based on Pruned Volterra Series. IEEE Nordic Circuits and Systems Conference (NorCAS)

N.B. When citing this work, cite the original published paper.

© 2026 IEEE. Personal use of this material is permitted. Permission from IEEE must be obtained for all other uses, in any current or future media, including reprinting/republishing this material for advertising or promotional purposes, or reuse of any copyrighted component of this work in other works.

(article starts on next page)

High-Throughput Low-Power Digital Predistorter Based on Pruned Volterra Series

Mubanda Talemwa*, Jorge Muñoz Bautista†, Thomas Eriksson†, and Per Larsson-Edefors*
*Dept. of Microtechnology and Nanoscience, Chalmers University of Technology, Gothenburg, Sweden
†Dept. of Electrical Engineering, Chalmers University of Technology, Gothenburg, Sweden
mubtxs@chalmers.se

***Abstract*—Reduced-complexity pruned Volterra series have been shown to deliver good linearization performance for digital predistortion in RF power amplifiers to adequately meet low-power hardware requirements. This paper explores DSP hardware implementation tradeoffs of high-throughput digital predistorters based on pruned Volterra series from which basis functions are selected to jointly optimize linearization performance and DSP power consumption. The proposed HA-BAPS algorithm generates basis functions that lead to a reduction in DSP power consumption. By concurrently processing multiple input samples, hardware parallelism is used to boost throughput. Through hardware reuse, low-power basis functions curb the rise in power consumption while maintaining high linearization performance. Total bitwidth reduction and relaxation of the IEEE-754 floating-point standard further decreases area and power consumption. Four basis sets configurations are explored for floating-point bitwidths of 16, 14, and 12 bits, with both strict and relaxed IEEE-754 standard. We design parallel hardware prototypes that process four input/output samples. The prototype with the best linearization performance vs. power efficiency vs. throughput tradeoff is compared to a baseline prototype with only linearization performance-aware basis functions. A 14-bit floating-point design which abandons IEEE-754 support achieves an improvement in throughput up to 2.2 GS/s. Compared to a parallel baseline design, this consumes 39% less power with a performance loss of only 0.32 dB in NMSE.**

***Index Terms*—Linearization Techniques, Digital Predistortion, Power Amplifiers, Energy Efficiency, Digital Signal Processing**

I. Introduction

Modern telecommunication systems operate at high data rates. In wideband multiple-input multiple-output (MIMO) systems, digital predistortion (DPD) power consumption grows significantly because it is performed per RF branch. Already at more than 16 antennas, the DPD power consumption may exceed that of the power amplifier (PA) [1]. In [2], a cost assessment of 5G massive MIMO systems shows that for a large number of users, data processing costs dominate other costs; moreover, infrastructure costs remain constant while power consumption and computational complexity scale with the number of users or antennas. The increased deployment of 5G/6G networks means that data processing and bandwidth requirements will continue to rise. Such developments highlight the challenge of designing efficient digital signal processing (DSP) hardware for high data traffic environments under strict power consumption restrictions.

High data rate in combination with high PA output power requirements leads to highly nonlinear PA behavior. Pruned Volterra series based DPD algorithms, such as BAsis Propagating Selection (BAPS), are well suited to counter PA nonlinear effects due to their high linearization performance with relatively low computational complexity [3]. However, to support high data rates, real-time DPD implementations require not only high accuracy and low power consumption, but also *high throughput*. Beyond simply increasing clock frequency, hardware platforms such as ASICs and FPGAs can achieve higher sample rates by increased levels of *parallelism*; but this comes with increased power consumption caused by the rising hardware complexity. Using algorithm and hardware co-design, we need to ensure that an algorithm generates parameters not only for high accuracy, but also for energy-efficient parallel hardware implementation.

Huang et al. [4] present a fixed-point 16-bit parallel DPD FPGA design using a reduced complexity model similar to generalized memory polynomial (GMP). However, the throughput of this design is limited due to the relatively low clock rate of the FPGA at 300 MHz and the relatively complex model. A parallel DPD engine based on a piecewise linear (PWL) model is presented in [5]. However, PWL models have higher system complexity, due to the extra control hardware and memory needed for smooth switching between partitions, leading to poor scalability [6], [7]. In [8], the authors contrast the performance of the GMP model implementation: on a 16-bit fixed-point FPGA, against a 16-bit floating-point GPU. Although the GPU provides superior accuracy at high throughput due to its extensive parallelism, the FPGA has lower latency and consumes much less power [8], [9]. In [10], a 32-bit floating-point serial predistorter prototype is used in an exploration of low-power basis functions with high linearization performance and low complexity of only 13 coefficients. However, such a serial design has low throughput and does not support parallel sample rate processing.

We address the aforementioned challenges by introducing a parallel predistorter architecture demonstrating energy-efficient PA linearization for real-time high-bandwidth environments. A biased BAPS algorithm, referred to as hardware-aware BAPS (HA-BAPS), identifies hardware-efficient model parameters. We also explore energy-efficiency benefits from the customization of different floating-point formats, via implementation with shorter floating-point formats of 16-, 14- and 12-bit bitwidths with 5-bit exponent in both strict and relaxed support of IEEE-754 [11].

II. High-Throughput Predistortion

A DPD model involves multiply and accumulate (MAC) operations, as shown in (1) where ϕ_r denotes the basis function and θ_r the corresponding coefficient. Similarly, finite impulse response (FIR) filters are realized as MAC operations with the basis functions having a regular structure, as shown in (2) where x denotes the input and θ_i the coefficient.

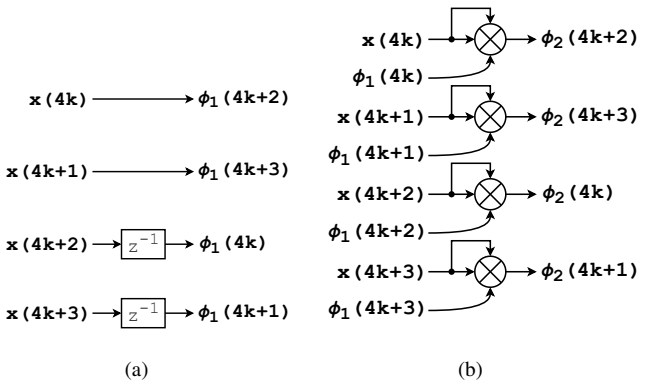
$$y(n) = \sum_{r=1}^{R} \theta_r \phi_r(n), \quad R = \text{number of coefficients} \tag{1}$$

$$y(n) = \sum_{i=0}^{N} \theta_i x(n-i), \quad N = \text{number of taps} \tag{2}$$

A. Hardware Parallelism

HA-BAPS generates the DPD model bases by iteratively constructing basis functions from the previous ones using two types of operations: type I and type II. A type I operation entails a delay of a previous basis function, while type II operation involves multiplication of three previous basis functions [3]. Type I operations involve memory shifts and consume less power compared to type II operations. An FIR filter is constructed using only type I operations, and has a regular structure that favors parallel data-streaming DSP implementation.

To support increased parallelism while maintaining a low rise in power consumption, HA-BAPS aims to generate basis functions with more FIR-like regular structure while maintaining high linearization performance. For example, given an input $x(n)$, an FIR structure with memory depth of 2 can generate two basis functions by delay: $x(n-1), x(n-2)$. In contrast, it would take five multiplications to generate non-FIR-like bases: $x^2(n)x(n-2), \overline{x}(n)x(n-1)$. If all the terms are part of the same basis set, then the former two terms are already present for the generation of the latter two terms. Even when concurrently processing samples of $x(n)$, the number of memory elements does not exceed the memory depth (see Fig. 1a), however, the number of multipliers will increase with the number of parallel samples (see Fig. 1b).



(a) (b)

Fig. 1: Generation of parallel basis functions. (1a) parallel type I basis function, (1b) parallel type II basis function

B. Basis Set Generation

HA-BAPS is initially configured for linearization performance alone, just as in conventional approaches. This is done to obtain a *baseline* basis set for later comparison of performance and power efficiency. HA-BAPS is then configured to incorporate power efficiency as a cost function for selecting basis sets within a given linearization performance margin. Table I shows four basis sets, including the baseline, from system identification of 12 new basis functions from $r = 13$ iterations. The input sample $x(n)$ is the first basis function. The iteration count r is also denoted as the basis number. The basis sets are denoted: `baseline`, `CFG_1`, `CFG_2`, and `CFG_3`.

III. DPD Design

Our predistorter is a highly parallel and pipelined design, with a data-streaming datapath for high throughput. High throughput is achieved by processing multiple input samples with a parallelization factor $F_{prll} = 4$, i.e., four data samples are input and output every clock cycle to obtain a sufficiently high output sample rate. Several hardware variants are designed to analyze the different parameters comprising different basis functions and floating-point customizations. The predistorter designs are synthesized and validated using Cadence design tools in the predictive ASAP7 standard cell library [12]. Data preparation and calculation is done on PC using Matlab, and RF measurements are performed on a GaN PA in the RF Weblab interface [13]. The baseband IQ data trace generated in Matlab has a bandwidth of 200 MHz.

A. Custom Floating-Point Formats

In RF wireless systems, a DPD sub-system drives a digital-to-analog converter (DAC) and receives feedback input from an analog-to-digital converter (ADC). For a more comprehensive analysis, the predistorter design is optimized to support total bitwidths of 12, 14, and 16. Fixed-point formats with these same bitwidths are supported by a wide range of data converters [14], [15]. Compared to fixed point with the same total bitwidth, floating-point representations offer higher dynamic range with a manageable rise in power consumption [16], [17]. To curb the rise in power consumption, the strict IEEE-754 floating-point standard [11] can be relaxed, which removes the need for special hardware for denormals and exception handling, which can substantially decrease power consumption [18].

B. Design Variants

Three main hardware parameters are varied to implement different designs: *configuration*, *floating-point format*, and *floating-point standard*. The predistorter design is optimized for data streaming; therefore, floating-point customizations directly affect the design size, but retain the configuration architecture. Each floating-point format has two design variants of the floating-point standard (IEEE or not), thus each configuration has six design variants since each of the three floating-point bitwidths are implemented twice. This results

TABLE I: BASIS SETS GENERATED BY HA-BAPS

Basis no.	baseline	CFG_1	CFG_2	CFG_3
1	$x(n)$	$x(n)$	$x(n)$	$x(n)$
2	$x(n)\|x(n)\|^2$	$x(n)\|x(n)\|^2$	$x(n)\|x(n)\|^2$	$x(n)\|x(n)\|^2$
3	$x(n-4)$	$x(n)\|x(n)\|^4$	$x(n)\|x(n)\|^4$	$x(n)\|x(n)\|^4$
4	$x(n-2)$	$x(n)\|x(n)\|^{14}$	$x(n)\|x(n)\|^{14}$	$x(n)\|x(n)\|^{14}$
5	$x(n)\|x(n)\|^4$	$x(n)\|x(n)\|^6$	$x(n-4)$	$x(n)\|x(n)\|^6$
6	$x(n-1)\|x(n-1)\|^2$	$x(n-4)$	$x(n-2)$	$x(n)\|x(n)\|^8$
7	$x(n-2)\|x(n-2)\|^2$	$x(n-2)$	$x(n-1)\|x(n-1)\|^2$	$x(n-4)$
8	$x(n-4)\|x(n)\|^2$	$x(n-1)\|x(n-1)\|^2$	$x(n-2)\|x(n-2)\|^2$	$x(n-2)$
9	$x(n-1)$	$x(n-2)\|x(n-2)\|^2$	$x(n-1)$	$x(n-1)\|x(n-1)\|^2$
10	$x^2(n-1)\overline{x}(n-2)$	$x(n-1)$	$x(n-1)\|x(n-1)\|^4$	$x(n-2)\|x(n-2)\|^2$
11	$\overline{x}(n)x^2(n-1)\|x(n-1)\|^2$	$x(n-1)\|x(n-1)\|^4$	$x(n-3)$	$x(n-1)$
12	$x(n)\|x(n)\|^{14}$	$x(n-3)$	$x(n-8)$	$x(n-1)\|x(n-1)\|^4$
13	$x(n)\|x(n)\|^6$	$x(n-3)\|x(n-3)\|^2$	$x(n-3)\|x(n-3)\|^2$	$x(n-3)$

in a total of twenty-four design variants. To maintain high throughput, memory terms are stored in registers integrated directly with floating-point operators.

A configuration refers to the design implementation of the set of basis functions, and affects the overall architecture of a design variant. There are four hardware configurations shown in Table I. Floating-point format refers to the bitwidth for a design's input/output data. Each configuration is implemented for a total bitwidth of 16, 14, and 12 bits including a 5-bit exponent. Floating-point standard refers to whether the design supports denormals and NaNs according to IEEE-754 floating-point standard or not: When support for IEEE-754 standard is disabled, denormal inputs are flushed to sign preserved zero, NaNs are flushed to sign preserved infinity, oppositely signed zero is represented according to rounding mode [19]. All floating-point hardware is always configured to the IEEE-754 default rounding mode.

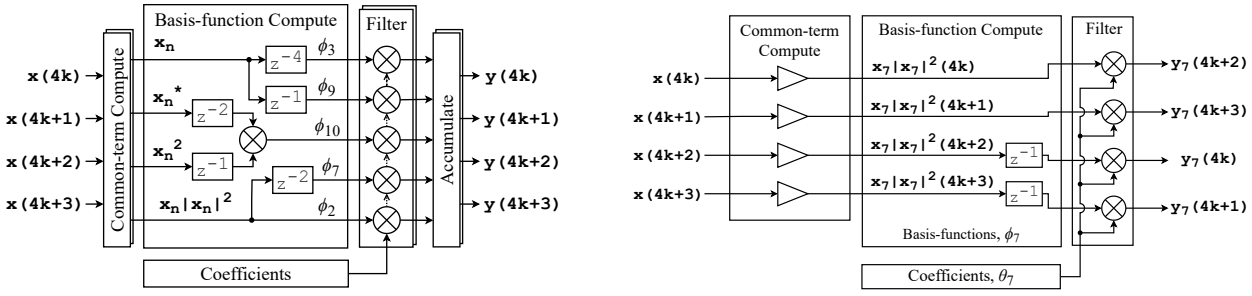
C. General Architecture

The predistorter is divided into four main blocks: common-term computation, basis-function computation, filtering, and accumulation as shown in Fig. 2. The double edges for the common-term, filter, and accumulate blocks indicate that these blocks are replicated by the level of parallelism $F_{prll} = 4$. The basis function block is parallelized through reuse.

The common-term computation block calculates all non-delayed polynomial terms in the design. First, successive even-order power terms are computed from the complex valued input sample: $|x(n)|^2 = (\Re(x(n))^2 + \Im(x(n))^2)$, $|x(n)|^4 = |x(n)|^2 \cdot |x(n)|^2$, ... Then odd-order terms are generated by multiplying the power terms with the complex valued input samples: $x(n)|x(n)|^2$, $x(n)|x(n)|^4$, ... In the basis-function computation block, basis functions are calculated by; 1) delaying the common-terms, 2) multiplying common-terms, or 3) multiplying common-terms with delayed versions of the common-terms. The filtering block carries out multiplication of the computed basis functions with coefficients, and the accumulation block adds up the filtered result to output the final result. Coefficients are stored in registers and are kept constant until they get updated. The blocks are separated by pipeline stages.

D. Custom Floating-Point Operators

The predistorter design uses complex-valued adder and multiplier operators. The complex-valued multipliers are directly implemented, i.e., using four real multiplications and



(a) Design overview with selected operations

(b) Detailed design of operation ($\theta_7 \cdot \phi_7$)

Fig. 2: Design and operation of Common-term, Basis-function, Filter, and Accumulate blocks. The squares in the basis-function block represent delay registers numbered to indicate memory depth. (2a) basis-function compute block shows implementation of a select number of baseline basis functions from Table I. (2b) Parallel implementation of `baseline` basis function 7. Note that the value for the 3^{rd} and 4^{th} output sample are simply forwarded, i.e., for a given basis function $x(n-2)$, then $x(4k+2) = x(4k)$, where k denotes the sample number.

TABLE II: COMPARISON OF LINEARIZATION PERFORMANCE FOR DIFFERENT CONFIGURATIONS AND FLOATING-POINT STANDARDS

Floating-point Standard	Configuration	NMSE (dB)				ACPR (dB)			
		32-bit	16-bit	14-bit	12-bit	32-bit	16-bit	14-bit	12-bit
IEEE-754 Compliant	baseline	-38.76	-38.82	-38.71	-37.31	-44.86	-44.92	-44.75	-43.27
	CFG_1	-38.69	-38.63	-38.49	-36.88	-46.13	-46.12	-45.98	-44.12
	CFG_2	-38.34	-38.32	-38.23	-36.83	-46.39	-46.39	-46.27	-44.16
	CFG_3	-38.41	-38.32	-38.25	-36.74	-45.92	-45.90	-45.81	-44.01
Non IEEE-754 Compliant	baseline	-38.79	-38.85	-38.71	-37.32	-44.93	-44.92	-44.76	-43.28
	CFG_1	-38.64	-38.58	-38.50	-36.89	-46.16	-46.13	-45.99	-44.12
	CFG_2	-38.35	-38.33	-38.23	-36.86	-46.39	-46.43	-46.28	-44.15
	CFG_3	-38.36	-38.38	-38.25	-36.74	-45.90	-45.91	-45.78	-44.02

two real additions. In [20], it was shown that the direct implementation achieves the lowest power consumption for tight timing constraints in fixed-point complex multiplications. This is also true for floating-point representations, because floating-point complex multipliers in terms of area complexity are dominated by integer multipliers.

The design variants are highly pipelined, resulting in simple parallel operations every pipeline stage, i.e., either a complex-valued addition or a complex-valued multiplication. This level of pipelining means that all configurations architecturally have the same critical path, i.e., through a real multiplier and real adder that comprise the direct implementation complex-valued multiplier. Hence, the same timing constraint is applied to all configurations during logic synthesis reducing variability, i.e., the timing constraint depends on the floating-point customization used.

IV. RESULTS

All design variants are evaluated based on the metrics of: linearization performance in terms of normalized mean square error (NMSE) and adjacent channel power ratio (ACPR) measured in dB, power consumption measured in mW, and throughput given by the number of samples per second (GS/s). All these metrics are retrieved from the predistorter hardware implementation and its netlist after synthesis. The linearization performance, maximum throughput, and the subsequent power consumption at the maximum throughput define the limits for the designs.

In addition to the 24 design variants (see Section III-B), single-precision 32-bit floating-point-based design variants are also synthesized following a similar procedure described in Section III. Single-precision floating point is a well established format with higher accuracy that can provide more confident comparison for linearization performance.

Table II shows the linearization performance for all design variants. From the point of view of the configuration architecture, the three configurations generated by HA-BAPS in energy-efficiency mode, significantly consume less power than the `baseline` with a less than 2 dB performance penalty in NMSE. The application of 13 bases with 4 type II basis functions provides a good tradeoff. Reduction by one type II basis function, such as in `CFG_2`, degrades linearization performance and addition of one more basis function, such as in `CFG_3`, does not improve linearization performance. Therefore, `CFG_1` provides a more favorable linearization performance vs. power tradeoff.

For a fair comparison, our power simulations for all design variants are initially performed at the maximum clock rate of the slowest design, i.e., that of the 32-bit IEEE-754 compliant design. For all bitwidths, Table III shows the power consumption of the `baseline` and the average power consumption of the other three configurations, which is simply denoted `CFG`.

TABLE III: COMPARISON OF `BASELINE` AND `CFG` POWER CONSUMPTION FOR DIFFERENT BITWIDTHS

Floating-point Std.	Configuration	Power (mW) for bitwidth of			
		32 bits	16 bits	14 bits	12 bits
IEEE-754 Compliant	baseline	529.8	139.5	109.0	74.8
	CFG	440.0	114.6	89.7	59.9
Non IEEE-754	baseline	459.7	103.5	71.0	48.7
	CFG	369.6	84.3	58.7	39.3

In addition to the above table, Fig. 3 focuses on the power consumption of the designs using shorter floating-point formats. Unlike `baseline`, the other three configurations exhibit a small difference in power consumption for all floating-point customizations, therefore, the average `CFG` power metric used in Table III is sufficient to analyze the overall power consumption trend.

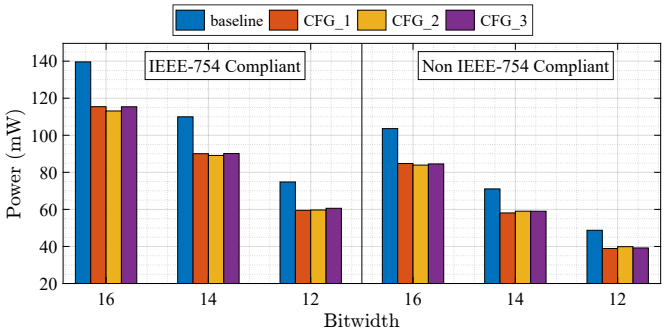


Fig. 3: Power consumption for design variants of all configurations with 16-, 14-, and 12-bit formats

A. Floating-Point Format Performance

From Table II, it can be observed that the linearization performance of the 16- and 14-bit design variants is much closer to that of the 32-bit designs. For example, the 14-bit designs have a minor drop of 0.08 dB in NMSE without IEEE-754 support. In contrast, Table III shows that both 16- and 14-bit designs consume significantly less power than the 32-bit designs. For example, in the non-IEEE-754 case, the 16-bit designs consume approximately 73% less power for both `baseline` and `CFG` respectively. Therefore, with such excessive power consumption, we can deduce that higher precision floating-point representation beyond 16 bits (half precision) is unnecessary for good linearization performance.

B. Floating-Point Customization

Comparing 16- and 14-bit designs, there is a minor reduction in linearization performance when IEEE-754 standard is dropped. For example, when comparing 14-bit `baseline` to 14-bit `CFG`, relaxing the IEEE-754 floating-point standard barely affects the linearization performance, but exhibits a significant drop in power consumption of at least 38 mW and 30 mW respectively. We demonstrate that support for strict IEEE-754 standard is also not necessary for good linearization performance. This further indicates that denormal and NaN exceptions rarely occur in predistortion calculations and not handling them barely affects accuracy.

As shown in Table III, 12-bit designs achieve the greatest power savings compared to their 16- and 14-bit counterparts but at a cost of worsening linearization performance. Table II shows that the 12-bit variants exhibit a significant 1.61 dB drop in NMSE. This degradation in linearization performance, due to insufficient total bitwidth, is further illustrated by the `CFG_1` spectrum plot in Fig. 4. Signal degradation is caused by quantization errors and manifests as a rise in the noise floor of the 12-bit compared to 16- and 14-bit designs.

The 16-, 14- and 12-bit formats all have 5-bit exponent, meaning that reduction in bitwidth only affects the mantissa, resulting in reduced precision at same available dynamic range. The significant drop in NMSE from 14 to 12 bits indicates that the 14-bit format offers the minimum precision necessary for good linearization performance.

C. Throughput

The throughput of a design variant is determined by the timing of its datapath. All designs variants for a given configuration have a similar architecture, meaning the delay for that configuration is predominantly governed by the bitwidth. Figure 5 shows the power consumption for 16-, 14-, and 12-bit `CFG_1` variants when run at their maximum possible throughput. The design variants exhibit a general trend of increasing throughput from 16- to 12-bit bitwidth with the non-IEEE-754 designs being faster for a given bitwidth. For the non-IEEE-754 compliant case, 14-bit designs exhibit a throughput 8% higher than the 16-bit case revealing that although the 16- and 14-bit design variants have similar linearization performance, the 14-bit variant is not only more energy-efficient but faster as well. On further comparison, we observe that the non-IEEE-754 compliant 12-bit variant also exhibits a 11% speedup over the 14-bit variant.

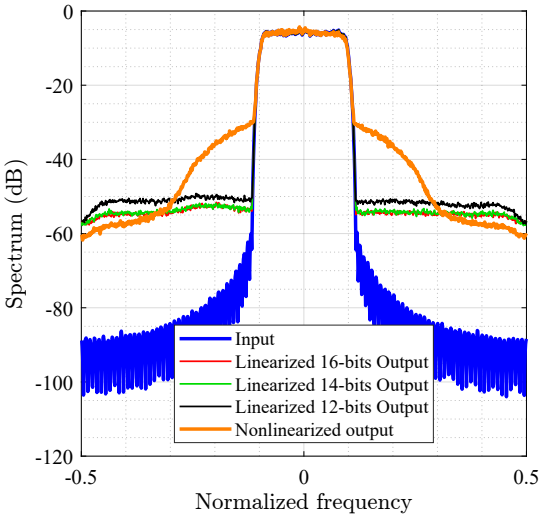


Fig. 4: Spectra of the linearized PA outputs for the non-IEEE compliant `CFG_1` at different bitwidths

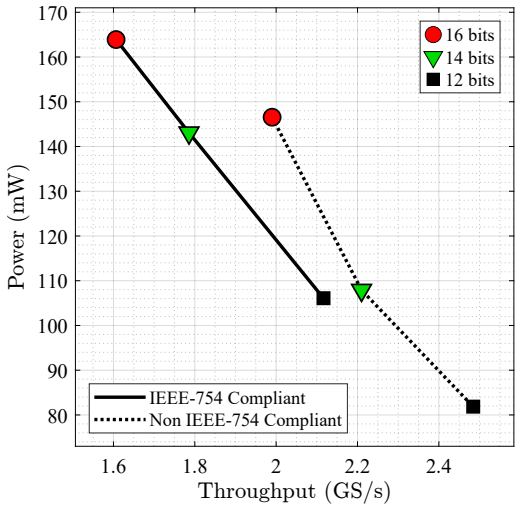


Fig. 5: DPD power consumption vs. maximum throughput of `CFG_1` for different bitwidths.

From the different analyzes, we observe that for the non-IEEE-754 compliant designs: 1) the 16-bit design variants have a minor comparative advantage over their 14-bit counterparts, and 2) the 12-bit designs sacrifice linearization performance for lowest power consumption and highest throughput. Therefore, we can conclude that the 14-bit `CFG_1` design variant without IEEE-754 support provides the most favorable trade-off of linearization performance vs. power consumption vs. throughput. In most cases, optimizing NMSE also results in good ACPR since the NMSE metric considers both in-band and out-of-band distortions. However, we observe that it might be necessary for HA-BAPS to jointly optimize the ACPR and NMSE.

V. Conclusion

The power consumed by the DPD subsystem in modern wireless transmitters has become prominent. We introduce HA-BAPS as an extension of the BAPS algorithm to select basis functions with high accuracy for energy-efficient ASIC hardware implementation. We present a hardware-algorithm co-design for implementation of a robust digital predistorter that provides a good tradeoff for lowering power consumption, improving linearization performance, and increasing throughput. HA-BAPS generates sets of basis functions to implement energy-efficient DPD hardware architectures. Additional floating-point customization, by reducing bitwidth, and by relaxing the strict IEEE-754 compliance, further reduce power consumption while maintaining high accuracy. Such energy-efficient architectures offer increased throughput by higher levels of hardware parallelism, enabling the predistorter to process even more data under the same power budget.

Acknowledgement

The authors thank the Swedish Foundation for Strategic Research (SSF) for funding through the classIC project.

References

[1] J. Zanen, E. Klumperink, and B. Nauta, "Power efficiency model for MIMO transmitters including memory polynomial digital predistortion," *IEEE Transactions on Circuits and Systems II: Express Briefs*, vol. 68, no. 4, pp. 1183–1187, 2021.

[2] A. L. P. Fernandes, D. D. Souza, C. Natalino, F. Tonini, A. M. Cavalcante, P. Monti, and J. C. W. A. Costa, "A cost assessment methodology for user-centric distributed massive MIMO architectures," *IEEE Open Journal of the Communications Society*, vol. 5, pp. 3517–3543, 2024.

[3] W. Cao, S. Wang, P. N. Landin, C. Fager, and T. Eriksson, "Complexity optimized digital predistortion model of RF power amplifiers," *IEEE Transactions on Microwave Theory and Techniques*, vol. 70, no. 3, pp. 1490–1499, 2022.

[4] H. Huang, J. Xia, and S. Boumaiza, "Novel parallel-processing-based hardware implementation of baseband digital predistorters for linearizing wideband 5G transmitters," *IEEE Transactions on Microwave Theory and Techniques*, vol. 68, no. 9, pp. 4066–4076, 2020.

[5] M. Almoneer, H. Barkhordar-pour, P. Mitran, and S. Boumaiza, "Hardware-efficient implementation of piece-wise digital predistorters for wideband 5G transmitters," in *2022 IEEE/MTT-S International Microwave Symposium*, 2022, pp. 514–517.

[6] Y. Li, X. Wang, and A. Zhu, "Reducing power consumption of digital predistortion for RF power amplifiers using real-time model switching," *IEEE Transactions on Microwave Theory and Techniques*, vol. 70, no. 3, pp. 1500–1508, 2022.

[7] C. Pedrosa, D.-K. G. Pham, P. Rashev, P. Almairac, J.-C. Nanan, and P. Desgreys, "Discontinuity characterization and low-complexity smoothing in RF-PA polynomial piecewise modeling," *Sensors*, vol. 25, no. 21, 2025.

[8] W. Li, G. Montoro, and P. L. Gilabert, "GPU versus FPGA implementation of a digital predistortion linearizer for wideband radiofrequency power amplifiers," *International Journal of Electronics and Communications*, vol. 174, p. 155040, 2024.

[9] M. Arucu and T. Iliev, "Performance evaluation of FPGA, GPU, and CPU in FIR filter implementation for semiconductor-based systems," *Journal of Low Power Electronics and Applications*, vol. 15, no. 3, 2025. [Online]. Available: https://www.mdpi.com/2079-9268/15/3/40

[10] M. Talemwa, J. M. Bautista, T. Eriksson, and P. Larsson-Edefors, "DSP hardware tradeoffs for digital predistortion based on pruned Volterra series," in *2025 IEEE Workshop on Signal Processing Systems (SiPS)*, 2025, pp. 1–5.

[11] *IEEE Standard for Floating-Point Arithmetic*, IEEE Std. 754-2019, 2019.

[12] V. Vashishtha, M. Vangala, and L. T. Clark, "ASAP7 predictive design kit development and cell design technology co-optimization," in *2017 IEEE/ACM International Conference on Computer-Aided Design (ICCAD)*, 2017, pp. 992–998.

[13] P. N. Landin, S. Gustafsson, C. Fager, and T. Eriksson, "Weblab: A web-based setup for PA digital predistortion and characterization [application notes]," *IEEE Microwave Magazine*, vol. 16, no. 1, pp. 138–140, 2015.

[14] X. Li and L. Zhou, "A survey of high-speed high-resolution current steering DACs," *Journal of Semiconductors*, vol. 41, no. 11, p. 111404, nov 2020. [Online]. Available: https://doi.org/10.1088/1674-4926/41/11/111404

[15] S. Bashir, S. Ali, S. Ahmed, and V. Kakkar, "Analog-to-digital converters: A comparative study and performance analysis," in *2016 International Conference on Computing, Communication and Automation (ICCCA)*, 2016, pp. 999–1001.

[16] Z. Yu, M. Talemwa, and P. Larsson-Edefors, "Exploration of short floating-point numbers for hardware-friendly digital predistortion," in *2025 IEEE Nordic Circuits and Systems Conference (NorCAS)*, 2025, pp. 1–5.

[17] B. Barrois and O. Sentieys, "Customizing fixed-point and floating-point arithmetic — a case study in k-means clustering," in *2017 IEEE International Workshop on Signal Processing Systems (SiPS)*, 2017, pp. 1–6.

[18] P. Larsson-Edefors, "Energy-efficient computation of TensorFloat32 numbers on an FP32 multiplier," in *IFIP/IEEE 33rd Int. Conf. on Very Large Scale Integration (VLSI-SoC)*, 2025.

[19] *Genus®, ChipWare IP Components Guide*, Cadence Design Systems, Inc., March 2026.

[20] P. Larsson-Edefors and E. Börjeson, "Implementation evaluation of fixed-point multipliers for complex numbers," in *IEEE 32nd Symposium on Computer Arithmetic (ARITH)*, 2025, pp. 81–84.