

FPGA IMPLEMENTATION OF DIGITAL PREDISTORTION LINEARIZERS FOR WIDEBAND POWER AMPLIFIERS

Navid Lashkarian, Signal Processing Division, Xilinx Inc., San Jose, USA,
navid.lashkarian@xilinx.com,

Chris Dick, Signal Processing Division, Xilinx Inc., San Jose, USA,
chris.dick@xilinx.com

ABSTRACT

This paper reports on the FPGA implementation of a Volterra series PA pre-distorter. The implementation of the pre-distorter and the indirect learning architecture for initializing the system is described. We supply insight into the implementation of the adaptive process itself and how the pre-distorter can exploit new generation heterogeneous FPGAs that provide a massively parallel compute fabric for demanding real-time tasks and an embedded processor for processes that have softer schedules. A recent generation visual programming design flow has been used for the implementation. The paper comments on the design productivity and efficiency aspects of the final FPGA implementation using this development environment.

1. INTRODUCTION

Bandwidth efficiency and transmission power efficiency are often conflicting criteria in digital communication systems. One usually has to be traded-off with the other according to the system requirements. In wireless applications, the cost of bandwidth accounts for a considerable portion of overall cost, and it is therefore important to accommodate as many users in the system within the link frequency budget. This requirement imposes a heavy constraint on the power efficiency of the amplifier, contributing to nonlinear behavior in this part of the transmitter [1]. Nonlinear radio frequency (RF) power amplifiers (PA) generate intermodulation (IM) distortion as adjacent channel interference for many modulation formats. Therefore, the design of linearizers has become a key technology in wideband mobile communication transceivers.

One solution is the linearization of the amplifier by means of predistorter as shown in Figure 1. The digital predistortion (DPD) linearizer creates a version of the desired modulation making use of the feedback measurements of the actual amplifier output. The resulting signal, when passed through the nonlinear power amplifier creates a signal in which the power spectral density has significantly lower spectral leakage com-

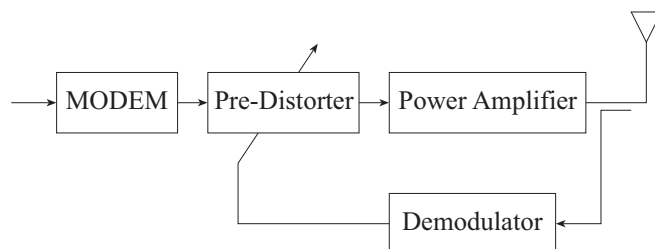


Figure 1: Adaptive digital predistortion.

pared to an uncompensated transmit signal.

Traditionally, digital predistortion was implemented using a lookup table (LUT) approach. The LUT employed in this approach is representative of the inverse of characteristic of the amplifier [2]. While this approach has widespread application in narrowband power amplifier (memoryless nonlinear systems) linearization, its effectiveness is hampered by the memory effects in wideband power amplifiers, such as those used in multi-carrier Universal Mobile Telecommunications System (UMTS) and CDMA2000 systems.

In this paper, we address the design of a linearizer based on an adaptive truncated Volterra series (TVS) approach. TVS systems have become a very popular tool in adaptive nonlinear signal processing [3]. However, their real-time implementation has been restricted by the computational complexity associated with the filtering and adaptive mechanisms. Field programmable gate arrays (FPGAs) are an attractive option for realization of these highly complex signal processing functions for reasons of performance, power consumption and configurability. We propose an efficient and robust architecture for the linearizer based on truncated Volterra filters and provide a simulation model of the system within the Xilinx System Generator for DSPTM [4] design flow. The implementation achieves up to 50 dB spectral suppression in neighboring frequency bands.

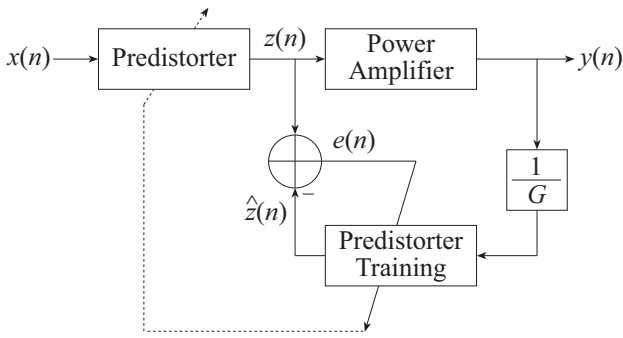


Figure 2: Baseband equivalent model of the DPD circuit. G represents the gain of the power amplifier.

2. PREDISTORTER ARCHITECTURE

Our approach to nonlinear predistortion is based on the method proposed by Eun & Powers [5]. In this approach, two identical truncated Volterra systems are used for training and predistortion. Figure 2 depicts the block diagram of the equivalent baseband model of the digital predistortion network. The objective of the linearizer is to find a transformation of the signal ($\tilde{z}(n) = V(x(n))$) that in combination with the nonlinear amplifier (responsible for the distortion) will result in an identity system that produces the signal of interest without distortion at the output of the power amplifier ($y(n) \simeq x(n)$). The main challenge of this approach is to track and identify the time varying characteristics of the amplifier. To address this task a stochastic gradient adaptation mechanism is employed. The adaptation of the truncated Volterra system is a two stage process. During initialization, the input and output signals of the power amplifier are probed and the Volterra filter coefficients are adapted off-line using Recursive Least Squares (RLS or Kalman Filtering) estimation. This process is also known as *initialization through indirect learning*. Once the adaptive filter is initialized at an optimum stationary point, a stochastic adaptive mechanism is used to track the time-varying characteristics of the nonlinear amplifier.

2.1. Memory Polynomial Predistorter

We use the memory polynomial model (Eq. 1) for the predistorter block as described in [6]

$$z[n] = \sum_{k=1}^K \sum_{q=0}^Q a_{kq} x(n-q) * |x(n-q)|^{k-1} \quad (1)$$

k even

where K is the nonlinearity order and Q represents the memory length of the power amplifier. In order to reduce the implementation complexity of the predistorter while maintaining acceptable performance, only the odd-order terms in the nonlinearity are included in the model. This compromise reduces the complexity of the predistorter by approximately 40% at the expense of 3 to 5 dB spectral regrowth. A detailed investigation of the benefits of even-order terms in the baseband model is presented in [7].

2.2. Indirect Learning

Initialization of the DPD linearizer is performed using optimum filtering, which is done as an off-line computation in our DPD implementation. Adaptive filter coefficient estimation can be considered a linear optimization task. Any of the common estimation methods - Least Square Estimation [8], minimum mean squared error (MMSE) [8] or Wiener Filtering, Kalman [8] or recursive least squares (RLS) filtering [8] - can be used.

We note that while all of the above methods try to solve one optimization problem, that is the linear parameter estimation, the stationary point obtained from using these methods might be quite different. This is mainly due to the fact that the error criterion for the approaches are different, causing a different profile for the error surface.

2.3. Tracking and Direct Learning

The inverse of the nonlinear amplifier is adaptively tracked using a stochastic gradient method. Least mean squares (LMS) adaptive filters are known to have a slow convergence rate. However, since the power amplifier characteristics vary slowly as a function of time, the LMS approach is a reasonable choice for performing parameter tracking.

At each iteration of the stochastic gradient algorithm, an update for the unknown vector is obtained from

$$W_{n+1} = W_n + \mu \times e_n \times \mathbf{X}_n^* \quad (2)$$

where the error vector is defined as

$$e_n = z(n) - W_n \times \mathbf{X}_n \quad (3)$$

X is the vector containing all the necessary nonlinear

products of the input sample and can be expressed as

$$\mathbf{X}_n \triangleq \begin{bmatrix} y(n) \\ y(n) * |y(n)|^2 \\ y(n) * |y(n)|^4 \\ y(n-1) \\ y(n-1) * |y(n-1)|^2 \\ y(n-1) * |y(n-1)|^4 \\ y(n-2) \\ y(n-2) * |y(n-2)|^2 \\ y(n-2) * |y(n-2)|^4 \end{bmatrix} \quad (4)$$

A nonlinear combiner is used to form all the necessary powers of the training signal as needed by the memory polynomial filter.

3. DPD FPGA IMPLEMENTATION

The DPD linearizer implemented in this case study is defined in Eq. 5. This is a slight modification to the procedure in Eq. 1 and includes signal phase information.

$$z[n] = \sum_{k=1, k \text{ odd}}^K \sum_{q=0}^Q a_{kq} x(n-q)^k \quad (5)$$

In the DPD System Generator reference design the linearity order was selected as $K = 5$ with a PA memory duration $Q = 2$. Fully expanded, the linearized signal $z(n)$ is expressed as

$$z(n) = \sum_{\substack{k=1 \\ k \text{ odd}}}^5 \sum_{q=0}^2 a_{kq} x(n-q)^k \quad (6)$$

$$= a_{10}x(n) + a_{11}x(n-1) + a_{12}x(n-2) \quad (7)$$

$$+ a_{30}x^3(n) + a_{31}x^3(n-1) + a_{32}x^3(n-2) \quad (8)$$

$$+ a_{50}x^5(n) + a_{51}x^5(n-1) + a_{52}x^5(n-2) \quad (9)$$

Only odd terms are included in the model. Eq. 7 is recognized as a standard inner-product, with Eq. 8 and 9 contributing a weighted combination of third- and fifth-order non-linearities respectively.

The truncated Volterra adaptive filter is implemented using the minimum-multiplier *direct-form* realization shown in Figure 3.

The simulation setup for the DPD system is shown in Figure 4. A pseudo-random data sequence is first pulse-shaped and then processed by the predistorter before being presented to the power amplifier model. The coefficients in the DPD linearizer are updated using an LMS-based adaptive process. The adaptation

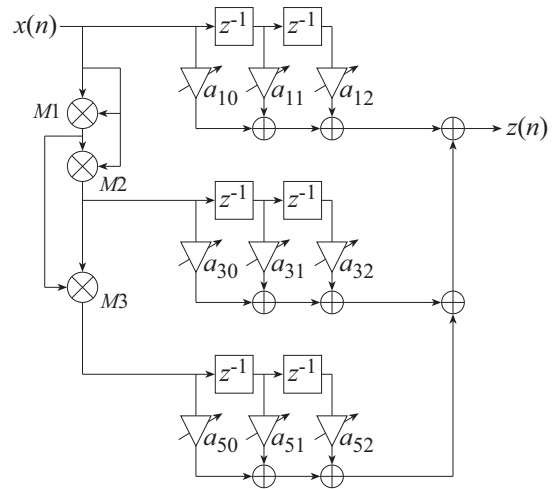


Figure 3: Direct form realization of the truncated Volterra series linearizer. This implementation provides minimum multiplicative complexity.

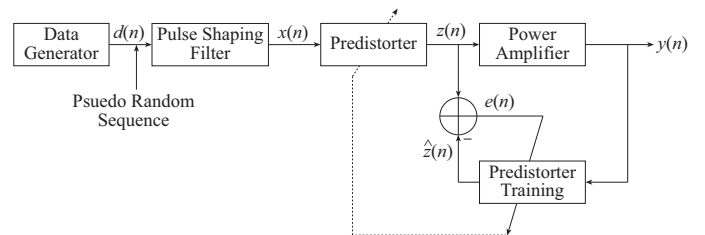


Figure 4: DPD simulation model comprising data generation, pulse shaping filter, the DPD circuit itself, PA model, and adaptive learning sub-system. The predistorter training is based on the LMS algorithm.

rate is a function of the system performance requirements. These requirements would typically include considerations that account for variations in the PA characteristics that are functions of time and temperature, in addition to electro-thermal effects that influence the PA effective memory. In our first implementation the Volterra filter coefficients are updated at each simulation time-step. Given the relatively long time-constant associated with changes in the PA characteristics this high rate of adaptation is probably too rapid. However, the flexible nature of the FPGA provides for any amount of hardware resource sharing to achieve a target performance/cost objective.

To produce a DPD simulation, a simulation model for the PA is required. The Wiener model [3] shown in Figure 5 will be employed in the simulation. This system consists of a linear time invariant (LTI) sub-

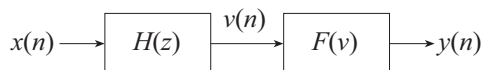


Figure 5: Wiener system PA model employed in the DPD simulation.

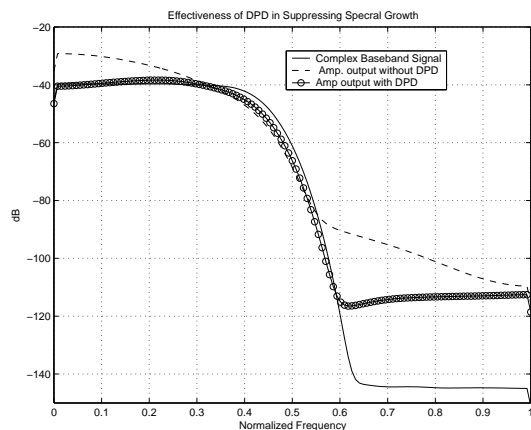


Figure 6: Effectiveness of DPD in suppressing spectral regrowth. The figure shows an overlay of the baseband signal spectrum, the PA output without linearization and with linearization.

system, $H(z)$, in cascade with a memoryless nonlinearity $F(v)$. Ding [6] provides expressions for

$$H(z) = \frac{1 + 0.5z^{-2}}{1 - 0.2z^{-1}} \quad (10)$$

and

$$y(n) = \sum_{k=1, k \text{ odd}}^K b_k v(n) |v(n)|^{k-1} \quad (11)$$

where $v(n)$ and $y(n)$ are the input and output of the memoryless non-linearity $F(v)$. Ding [6] provides values for the coefficients b_k based on measurements from a class AB PA as

$$b_1 = 1.0108 + j0.0858 \quad (12)$$

$$b_3 = 0.0879 - j0.1583 \quad (13)$$

$$b_5 = -1.0992 - j0.8891 \quad (14)$$

Figure 6 shows the effectiveness of the baseband predistortion in suppressing spectral regrowth. As shown in the figure, DPD can effectively reduce spectral regrowth by 40 dB.

3.1. Simulation Model

When the System Generator simulation is opened in the Simulink environment a *pre-load function* is called that computes an initial estimate of the system coefficients using RLS estimation. The optimum coefficients resulting from the estimate are

$$\begin{aligned} &0.0003 - j0.0066 \\ &0.0005 + j0.0120 \\ &-0.0036 + j0.0005 \\ &1.1632 - j0.0936 \\ &0.0890 + j0.3610 \\ &-0.0554 + j0.0254 \\ &-0.6712 + j0.0543 \\ &-0.0525 - j0.2041 \\ &0.0295 - j0.0144 \end{aligned}$$

In order to demonstrate adaptive tracking one of the coefficients is deliberately perturbed - the fourth coefficient ($1.1632 - j0.0936$) is scaled by a factor of 3. The modified coefficient vector is used as the initial condition for the adaptive processor.

The least mean squares processor in the system adaptively updates the coefficients, iteratively forcing the perturbed coefficient back to its optimum value.

Figure 7 shows the trajectory of the real component of the modified element (fourth entry of the vector) as a function of the LMS update iteration number. The figure provides an overlay of the Matlab double-precision floating-point simulation and the fixed-point arithmetic FPGA implementation. The floating-point and fixed-point simulations are in close agreement. The residual mean-squared error (MSE) of the FPGA based LMS filtering is plotted in Figure 8.

The System Generator predistorter reference implementation employs a fully parallel adaptive processor for the adaptive learning sub-system. This means that the 9 complex coefficients in Eq. 6 are all updated at the output sample rate. This is a very high-frequency update rate and may be too rapid for many applications. It is straightforward to modify the adaptive processor to employ a decimated update. In this scenario the coefficients would be updated at a lower frequency than the Volterra filter processing rate. Using a decimated update permits functional unit folding in the adaptive processor so that the FPGA footprint can be reduced, i.e., both the number of logic slices and embedded multiplier can be minimized.

When the simulation completes a post simulation *stop function* is executed that plots the linearizer input function overlaid with the predistorter output, generating a plot similar to Figure 6.

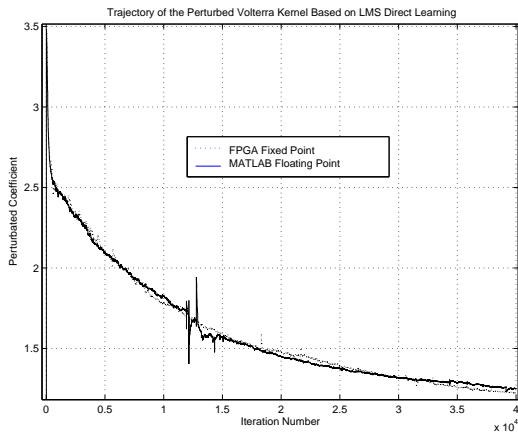


Figure 7: Volterra Kernel tracking based on LMS indirect learning. The figure shows the evolution of the fourth coefficient in the model. The floating-point and FPGA fixed-point simulation results are overlaid in the figure.

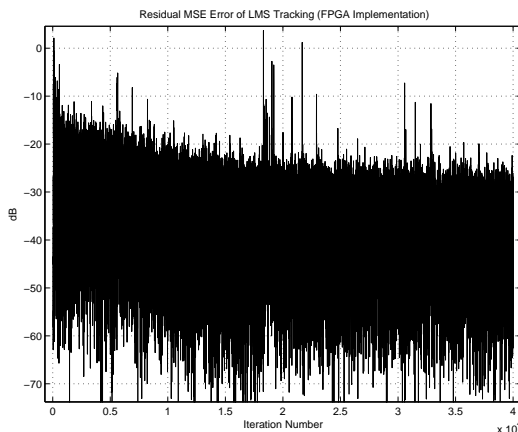


Figure 8: Residual MSE of the LMS coefficient update.

Table 1 provides the FPGA resource utilization for the complete design employing a fully parallel DPD coefficient update operating at the output sample rate.

In applications where the adaptive learning is not needed (due to the time invariant characteristics of the amplifier), the DPD can be implemented using only 2032 FPGA slices and 48 embedded multipliers.

The predistorter sub-system in the design operates at a clock frequency of 212 MHz in a Virtex-II Pro XC2VP50FF1152-7 FPGA (System Generator v 6.2, ISE 6.3.03i, Speedfile, Production 1.86 2004-05-01).

There is adequate resources available in this device to support the predistorter along with other system functions such as up-conversion and crest factor reduction.

Table 1: FPGA Resource Utilization for DPD and LMS Adaptive Learning. The Volterra filter coefficients are updated at the full output data rate using a fully parallel LMS processor. The design is easily modified to accommodate a decimated update using a reduced number of embedded multipliers.

	Volterra Filter	LMS	Total
Slices	2032	3483	5515
Block Memory	0	0	0
Multipliers	48	106	154

The computation rate of the predistorter alone is $212e^6 \times 48 = 10.176e^9$. This 10 Giga-op processing rate exceeds the compute capacity of other programmable DSP technologies. The FPGA implementation easily supports the processing requirements, while providing the system architect with a flexible solution that can be easily modified based on evolving specifications or future system requirements.

4. ADAPTIVE COEFFICIENT UPDATE USING EMBEDDED PROCESSING

In many typical applications the PA characteristics do not change rapidly with time. The PA characteristics vary as a function of temperature drift and component aging, parameters that have long time-constants.

The previous section described a predistorter design that employed a dedicated customized datapath constructed using the logic fabric and embedded multipliers, to implement the DPD coefficient update. Depending on system requirements, and in particular the required rate of coefficient adaption, an FPGA embedded processor could be employed to realize the update. In this approach a buffer of the samples $y(n)$ and $z(n)$ in Figure 2 are prepared and processed offline. This lowers the overall implementation requirements of the system. State-of-the-art FPGAs like Virtex-II Pro [9] and Virtex-4 [10] include embedded Power PC 405 (PPC405) processing cores. The adaptive algorithm can be coded in C and executed on the PPC405. When a new coefficient vector is available the PPC405 can transfer this data to the coefficient memory in the Volterra filter. The PPC could also be used for other tasks in the system, in addition to periodically servicing the DPD processor.

The Xilinx Microblaze™ soft processor core [11] could also be used for implementing the adaptive update. Microblaze is supported by the Virtex-II Pro and Virtex-4 platforms, in addition to architectures

like Virtex-II [12] and the low-cost Spartan-3 [13] family that do not include embedded PPC405 processors.

5. CONCLUSION

In this paper we have provided an architecture study for the FPGA implementation of a wideband digital baseband predistortion processor. As communication infrastructure providers operating in the UMTS, CDMA2000 and military radio application spaces continue to increase transmission bandwidth and support multi-carrier systems, traditional look-up table approaches to power amplifier linearization are no longer appropriate, and alternative methods that support wideband signals are required. Linearization techniques based on non-linear signal processing techniques have been studied for some time, but their practical deployment has been restricted due to the limited processing capabilities of traditional configurable signal processors. While an application specific integrated circuit (ASIC) approach could meet the processing requirements, non-recurring engineering (NRE) costs, high mask-set costs, lengthy development schedules and lack of flexibility have limited the ASIC implementation of sophisticated PA linearizers.

The highly parallel nature of Xilinx FPGAs easily support the processing requirements of complex non-linear signal processing algorithms. The System Generator design described in this paper implements a baseband linearizer that includes a 5th order non-linearity and a 2nd order term that accounts for PA memory. These design parameters are easily modified to reflect the characteristics of any given power amplifier.

The LMS coefficient update procedure used in the implementation is a fully parallel design that updates all of the linearizer coefficients at the output sample rate. Depending on the system requirements, the adaptive processor could be modified to include functional unit time-sharing that would reduce the FPGA footprint in return for a decimated coefficient update rate. The coefficient update procedure could entirely, or partially, be relocated to embedded software running on either a Microblaze soft processor core or embedded PPC405 hard core in the Virtex-II Pro or Virtex-4 FPGA families.

References

- [1] C. Liang et. al., "Nonlinear amplifier effects in communications systems," *IEEE Transactions on Microwave Theory and Techniques*, Vol. 47, Issue 8, pp. 1461-1466, Aug. 1999.
- [2] J.K. Cavers, "Amplifier linearization using a digital predistorter with fast adaptation and low memory requirements," *IEEE Transactions on Vehicular Technology*, Vol. 39, Issue 4, pp. 374-382, Nov. 1990.
- [3] V. J. Mathews, G. Sicuranza, *Polynomial Signal Processing* John Wiley & Sons, 2000.
- [4] System Generator for DSP, Xilinx Inc., Xilinx Inc., http://www.xilinx.com/xlnx/xebiz/designResources/ip_product_details.jsp?key=dr-dt-system-generator
- [5] C. Eun, E. Powers, "A new Volterra Predistorter Based on the Indirect Learning Architecture," *IEEE trans. on Signal Processing*, Vol. 45, No. 1, January 1997.
- [6] L. Ding et. al., "A Robust Digital Baseband Predistorter Constructed Using Memory Polynomials," *IEEE trans. on comm.*, Vol. 52, No. 1, January 2004.
- [7] L. Ding et. al., "Effects of Even-Order Nonlinear Terms on Power Amplifier Modeling and Predistortion Linearization," *IEEE Transactions on Vehicular Technology*, Vol. 53, Issue 1, pp. 156-162, Jan. 2004.
- [8] S. Haykin, *Adaptive Filter Theory*, Prentice Hall, New Jersey, 1996.
- [9] Virtex-II Pro Datasheet, Xilinx Inc., http://www.xilinx.com/xlnx/xweb/xil_publications_display.jsp?category=Publications/FPGA+Device+Families/Virtex-II+Pro&iLanguageID=1
- [10] Xilinx Virtex-4 Revolutionizes Platform FPGAs, Xilinx Inc., http://www.xilinx.com/company/press/kits/v4_arch/v4_finalwhitepaper4.pdf
- [11] Microblaze Soft Processor Core, Xilinx Inc., http://www.xilinx.com/xlnx/xebiz/designResources/ip_product_details.jsp?sSecondaryNavPick=Design+Tools&key=micro_blaze
- [12] Virtex-II Datasheet, Xilinx Inc., http://www.xilinx.com/xlnx/xweb/xil_publications_display.jsp?category=/Data+Sheets/FPGA+Device+Families/Virtex-II&iLanguageID=1
- [13] Spartan-3 Datasheet, Xilinx Inc., http://www.xilinx.com/xlnx/xil_prodcats/landingpage.jsp?title=Spartan-3