A Full FPGA-based Implementation of an Adaptive Digital Predistorter

P.L. Gilabert, G. Montoro and E. Bertran

Dept. of Signal Theory and Communications, Universitat Politècnica de Catalunya (UPC)
Av/ Canal Olímpic s/n, 08860 – Barcelona - Email: plgilabert@tsc.upc.edu – Spain.

INTRODUCTION

This paper presents an adaptive Digital Predistorter (DPD) for Power Amplifier (PA) linearization whose implementation and real time adaptation are fully performed in a Field Programmable Gate Array (FPGA) device responsible for the co-processing tasks. We consider the PA as a long-term time-variant (due to aging, heating, load impedance variations) nonlinear system capable to operate in reconfigurable scenarios. Therefore, the PA behavior needs to be periodically monitored in order to eventually adapt the DPD to counteract possible changes in its behavior or in its mode of operation. Unlike in [1], there is no need to stop the transmission and switch into a training mode, because the DPD adaptation can be enabled/disabled by the user in a hot manner. It is thus possible to simultaneously transmit and adapt the DPD thanks to the real-time parallel processing performed by the FPGA. This architecture improves the DPD configuration in [2], where the predistortion function was already implemented in a FPGA by means of a set of basic predistortion cells (BPCs), but the update of the contents of these BPCs (complex gains stored in look-up tables) was performed in a host PC. Following the same principles explained in [3], the adaptation of the DPD is carried out using the Least Mean Square (LMS) algorithm to update all single complex gains that fill a BPC.

ARCHITECTURE OF THE ADAPTIVE DIGITAL PREDISTORTER

A block scheme of the adaptive LMS-based DPD is depicted in Fig. 1. First of all and in order to ensure a good functioning of the DPD+PA system it is necessary pre-equalize the return path (from the PA output at RF to baseband), that is, to perform an amplitude and phase correction of the received baseband signal (*Rx_Data* in Fig. 1) with the PA acting as a linear device. Additionally it also necessary to include an offset cancellation block and a time-alignment block, in order to remove I-Q offsets and to have the transmitted (*Tx_Data*), the predistorted (*Tx_DPD*) and the received (*Rx_Data*) signals synchronized.

The predistortion function is carried out as in [2] and [3] by means of BPCs. Here we have considered 1 BPC for testing the performance of a memoryless DPD. However, the addition of parallel BPCs allow to linearize PAs presenting memory effects, just by replicating the same DPD structure several times. In order to perform the real-time adaptation it is necessary to use Dual Port RAM (DPR) memory blocks. A DPR block has two independent sets of ports for simultaneous reading and writing. Independent address, data, and write enable ports allow shared access to a single memory space.

The predistortion of the signal to be transmitted (*Tx_Data*) is performed in BPC#1, obtaining the predistorted data (*Tx_DPD*) to be upconverted and fed to the PA. In order to perform the DPD adaptation, the combination of BPC#2 + LMS block is continuously monitoring *Tx_DPD* and *Rx_Data* to extract the PA complex gains (*NEW GAINS PA*) that will fill BPC#3. In BPC#3, the input signal is the data to be transmitted (*Tx_Data*) and the output is the estimated output signal of the PA (*Rx_MOD*). Finally, the BPC#4 + LMS block is responsible to continuously post-distort the estimated PA output (*Rx_MOD*) and thus to invert the PA model previously extracted. The complex gains (*NEW GAINS DPD*) estimated in BPC#4 + LMS are continuously being copied in BPC#1 to perform predistortion.

Using a more generic notation ($x(k)$ and $\hat{y}(k)$), the input-output relation in a BPC is defined in eq. (1) while the LMS algorithm is defined in eq.(2).

$$\hat{y}(k) = x(k) \cdot G_{LUT}^{*}(|x(k)|) \quad (1)$$

$$G_{LUT}^{(new)}(|x(k)|) = G_{LUT}^{(old)}(|x(k)|) + \mu \cdot x(k) \cdot e^{*}(k) \quad (2)$$

Now, if we particularize the notation for the BPC#2, the variable *x(k)* is *Tx_DPD*, while $\hat{y}(k)$ is *Rx_MOD* and the LMS error is defined as: *e(k)= Rx_Data - Rx_MOD.* Analogously, particularizing for the BPC#4, the variable *x(k)* is *Rx_MOD*, while $\hat{y}(k)$ is *Tx_MOD* and the LMS error is defined as: *e(k)= Tx_Data - Tx_MOD.*

Some preliminary results are depicted in Fig. 2, Fig. 3 and Fig. 4, showing both linearized and unlinearized output power spectra, AM-AM characteristics and demodulated 16-QAM constellations respectively. Preliminary results were obtained considering a behavioral model for the RF subsystem (PA + converters) and implemented in a Xilinx Virtex-IV FPGA. Future work will be aimed at testing this architecture in a complete RF subsystem and at including additional BPCs in the predistortion function to compensate for PA memory effects.

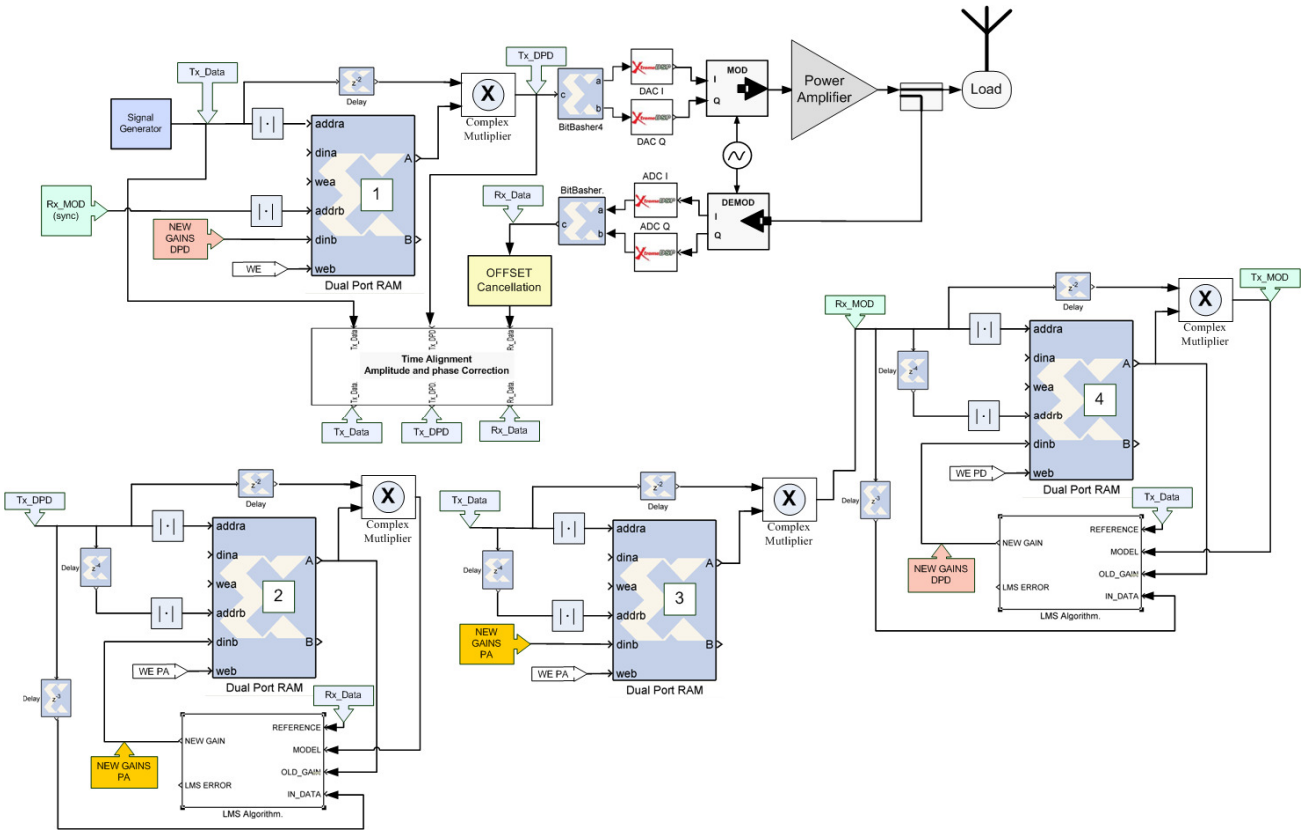

Fig. 1. Block scheme of the adaptive LMS-based digital predistorter

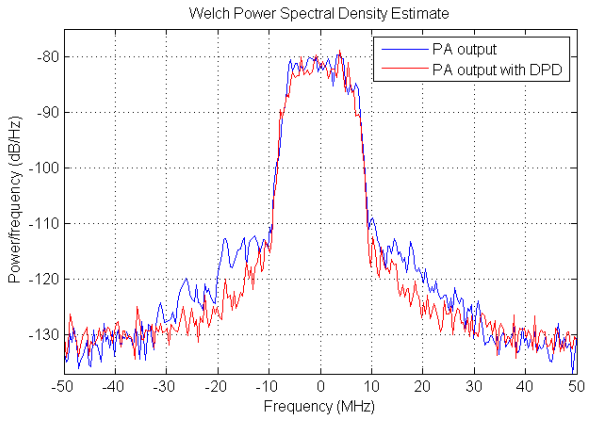

Fig. 2. Output power spectra

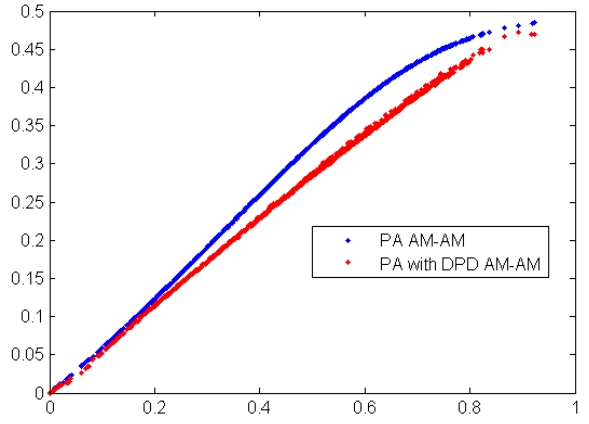

Fig. 3. AM-AM characteristic

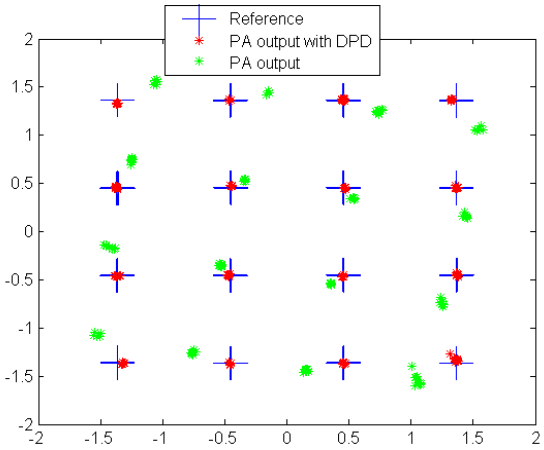

Fig. 4. 16-QAM constellation

REFERENCES

[1] A. Zhu, P. J. Draxler, J.J. Yan, T.J. Brazil, D.F. Kimball, P.M. Asbeck, "Open-Loop Digital Predistorter for RF Power Amplifiers Using Dynamic Deviation Reduction-Based Volterra Series," *IEEE Trans. on Microwave Theory and Techniques*, vol.56, no.7, pp.1524-1534, July 2008.

[2] P. L. Gilabert, A. Cesari, G. Montoro, E. Bertran and J. M. Dilhac "Multi Look-Up Table FPGA Implementation of an Adaptive Digital Predistorter for Linearizing RF Power Amplifiers with Memory Effects," *IEEE Trans. on Microwave Theory and Techniques*, vol. 56, nº 2, pp. 372 - 384 , Feb. 2008.

[3] G. Montoro, P. L. Gilabert, E. Bertran, A. Cesari and J. A. Garca, "An LMS-Based Adaptive Predistorter for Cancelling Nonlinear Memory Effects in RF Power Amplifiers," *Proc. Asia-Pacific Microwave Conference (APMC'07)*, Bangkok, Thailand, Dec. 2007.