International Journal of Electrical, Electronic and Communication Sciences
ISSN: 2517-9438
Vol:4, No:10, 2010

Digital Predistorter with Pipelined Architecture Using CORDIC Processors

Kyunghoon Kim, Sungjoon Shim, Jun Tae Kim, And Jong Tae Kim

***Abstract*—**In a wireless communication system, a predistorter(PD) is often employed to alleviate nonlinear distortions due to operating a power amplifier near saturation, thereby improving the system performance and reducing the interference to adjacent channels. This paper presents a new adaptive polynomial digital predistorter(DPD). The proposed DPD uses Coordinate Rotation Digital Computing(CORDIC) processors and PD process by pipelined architecture. It is simpler and faster than conventional adaptive polynomial DPD. The performance of the proposed DPD is proved by MATLAB simulation.

***Keywords*—**DPD, CORDIC.

I. Introduction

CONVENTIONAL radio-frequency power amplifiers(PA) operating with wideband signals, such as wideband code division multiple access (WCDMA), are required to have high linearity. To ensure the high linearity of the signals, CDMA-based communication standards have to use linear PAs which are too expensive and have complex architecture. But it is difficult to give up to use distinguished linear PAs.

To get high linearity by using more inexpensive PAs, their performance must be improved and their non-linearity also must be compensated. One of the linearization method to compensate is using a predistorter(PD)[1]. The PD knows previously information about non-linearity of the PA's outputs. And it is located in front of the PA and predistorts the PA's input signal inversely by the information of the outputs before inputs go into PA. Finally, the PA's output characteristic become linear by using the predistorted input signal. Thus PA can get good performance.

For predistorting, it is needed to make a mathematical model for the PA's linear characteristic. It can be expressed by the relation about input's and output's amplitude(AM) and phase(PM). Predistortion is implemented by adapting inversion of related functions at the input signal.

A PD can be implemented as analog or digital circuits. The analog PD costs lower than the digital because it uses passive elements[2][3]. But communication transmitter's characteristics can be changed by flow of time, and be degraded by its environment such as temperature. But it is difficult for analog PD to change the predistorting characteristics. On the other hand, though the digital PD(DPD) is more expensive, it can cope flexibly with the change of environment[4][5]. Thus it is more advantageous for the transmitter which is very expensive and it can be used for a long time.

The typical methods of DPD are using look-up table(LUT) and using adaptive polynomial. The first one is that the DPD stores the predistorted output data for the DPD inputs in its memory[6]. On operation, the DPD choose the output by referencing the table for the input signal. Since DPD must have all the data corresponded to all inputs, its memory has to be very large and the precalculated data are too a lot. Another one calculates the output according to entered inputs[7][8]. Thus it needs comparatively large mathematics units for calculation about trigonometrical function and multiplication. Hence it makes the DPD more complex and larger and the DPD's processing speed is restricted. But in these days the improvement of the hardware technology frees it from those restrictions. So here, the second is focused on.

In this paper, a new pipelined architecture of DPD using CORDIC processor is proposed and its calculation performance is proved by MATLAB simulation.

II. The Proposed DPD

A. The Whole Architecture of the Proposed DPD

The proposed DPD locates in front of the PA and takes the signal which type is I/Q-phase such as shown Fig. 1. At first the I/Q-phase digital signal from analog-to-digital converter(ADC) goes into to input port of the DPD. And the data move in order as Pre-processor, Multiplexer(MUX), CORDIC processor, DEMUX, PD, MUX, CORDIC processor, Post-processor, digital-to-analog converter(DAC).

Pre-Processor changes the external data's format to the inner data's format used in CORDIC and PD. Contrariwise Post-Processor changes the output format from PD to previous format.

MUX classify whether data come from Pre-processor or PD. If from Pre-processor, MUX gives the data to CORDIC's vectoring mode and if not, to CORDIC's rotation mode. Contrariwise DEMUX gives the predistorted data to Post-processor or the non-predistorted data to PD block.

Kyunghoon Kim and Jun Tae Kim are with the Department of Mobile Systems Engineering, Sungkyunkwan University, Suwon, Gyeonggi-do 440-746, Republic of Korea (phone: +82-31-290-7173; fax: +82-31-299-4613; e-mail: {donno, kjtjjang}@ skku.edu).

Sungjoon Shim is with the Department of Electrical and Computer Engineering, Sungkyunkwan University (e-mail: joonii63@ skku.edu).

Jong Tae Kim is with the Department of Mobile Systems Engineering and the Department of Electrical and Computer Engineering, Sungkyunkwan University (e-mail: jtkim@ skku.edu).

The CORDIC processor changes the I/Q-phase input data to AM/PM signal and the predistorted data to I/Q-phase. The CORDIC algorithm and its pipelined architecture are explained at phase *D*.

PD block is the core part of predistortion. It calculates the preditortion by inversion of the PA's characteristics using the AM/PM data from CORDIC processor. It will be also explained in detail at phase *E*.

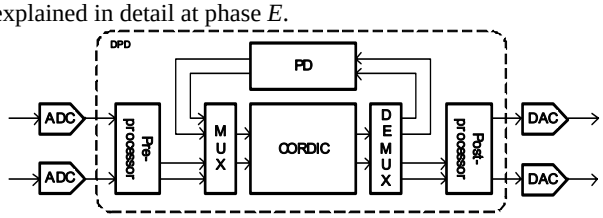

Fig. 1 Block diagram of DPD

B. Flow and Format of Data in DPD

Fig. 2 shows the data flow in DPD. At first, PA's input signal on analog I/Q-phase form is changed to digital data by ADC, and goes into DPD. Then it is used to Pre-processor's input, and it is changed to 16-bit data which are used on the inside of DPD. That format is formed of the 1 sign bit, 2 integer bits and the 13 fraction bits sequentially from MSB to LSB. It is marked as {1, 2, 13}. On the inside of DPD, the calculation for the real number is implemented by *fixed point number*. *Fixed point* will be explained later. The data passed through Pre-processor is changed to the AM/PM data after CORDIC vectoring mode process and become the predistorted data by PD process. Then the predistorted data is changed again to I/Q-phase form by the CORDIC rotatioin mode process, and changed to the DAC input format by Post-processor. And the PA's input signal are generated by DAC and go into PA's input.

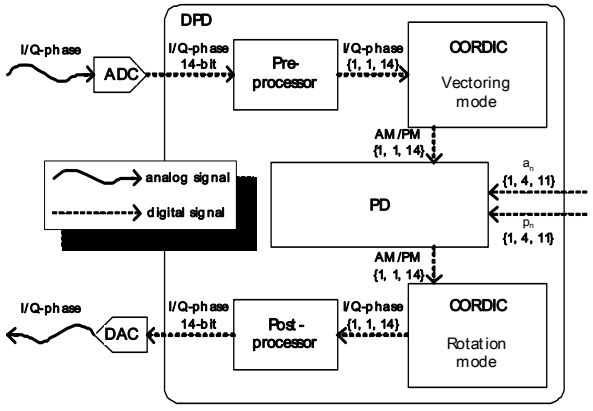

Fig. 2 Data flow and format of DPD

C. Binary Scaling

Bianry Scaling is a computer programming technique used mainly by embedded C, DSP and assembler programmers to perform a pseudo floating point using integer arithmetic. It is both faster and more accurate than using floating point instructions because it can have more the significant figures in case of handling not that small number such as 3.12345, not 1.23235×10^{-20}.

In the Fig. 3, there is an arbitrary binary number on 16-bit word. If the virtual binary point is at B0 position of Fig. 3, that binary 16-bit number can get the value from -1.0 to 0.999999. Similarly B1 gets the range of -2.0 to 1.999999 and B2 gets a range of -4.0 to 3.999999.

In this paper, the binary scaling number is represented such as next. The data after Pre-process is {1, 2, 13} and the coefficients of polynomial of PD are used to {1, 4, 11}. The scaling factor has the form of {1, 1, 14}. The scaling factor will be mentioned later.

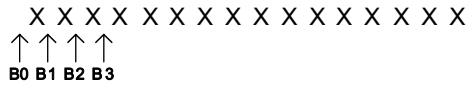

Fig. 3 Location of virtual binary pointer for binary scaling

D.CORDIC Algorithm and Pipeline Architecture

CORDIC is a simple and efficient algorithm to calculate hyperbolic and trigonometric functions[9]. It is commonly used when no hardware multiplier is available and it requires only addition, subtraction, bitwise shifting and table lookup.

Generalized CORDIC algorithm is expressed as:

$$\begin{aligned} x^{(i+i)} &= x^{(i)} - \mu d_i y^{(i)} 2^{-i} \\ y^{(i+i)} &= y^{(i)} + d_i x^{(i)} 2^{-i} \\ z^{(i+i)} &= z^{(i)} - d_i e^{(i)} \end{aligned} \tag{1}$$

At the (1), i means the number of iteration and d_i is the variable number which can be changed for the CORDIC operation mode. It can get the value -1 or 1. The CORDIC mode can be changed for what value user want to get and according to the mode, y or z is closed to 0 by iteration of the equation. And user can get the values they want to get by other variables.

In the vectoring mode, as y is closed to 0, each x and z is closed to $K\sqrt{x^2+y^2}$ and $z+\tan^{-1}(y/x)$. This K is some constant value. Thus it can be eliminated by multiplying the inverse of K after CORDIC calculation. In the proposed DPD, I-phase for x and Q-phase for y and 0 for z are substituted. Then the AM/PM data can be gotten. In the rotation mode, AM for x and 0 for y and PM for z, the I-phase on x and Q-phase on y can be gotten.

$e^{(i)}$ is calculated and stored at LUT previously. Its equation for each mode is as :

$$\begin{aligned} \mu = 1 \quad & Circular \quad & e^{(i)} &= \tan^{-1} 2^{-i} \\ \mu = 0 \quad & Linear \quad & e^{(i)} &= 2^{-i} \\ \mu = -1 \quad & Hyperbolic \quad & e^{(i)} &= \tanh^{-1} 2^{-i} \end{aligned} \tag{2}$$

Proposed DPD uses only circular rotation mode and circular vectoring mode for calculating trigonometric functions.

2^{-i} can be accomplishment by bitwise shifting operation. Thus the computation of $x^{(i+1)}$ or $y^{(i+1)}$ requires an i-bit right shifting and an addition/subtraction. Thus each CORDIC computation part involves two times shifting, a table lookup, and three additions.

The result of CORDIC operation is more accurate and close to the value to find by iteration. To improve the speed of computation, some same blocks operate as in (1) for pipelined architecture.

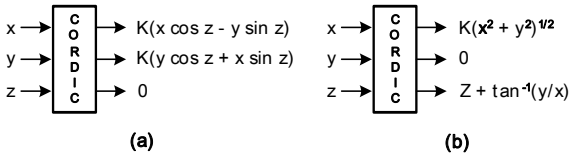

Fig. 4 The CORDIC operation of circular rotation mode (a) and vectoring mode (b)

E. Pipelined Architecture of PD Block

Fig. 5 shows PA's non-linearity by AM/PM, AM/AM graphs. The inverse equation of this non-linearity can be expressed to m th-order polynomial as :

$$A_{DPD} = \sum_{n=1}^{m} a_n \times A_D{}^n$$
$$P_{DPD} = \sum_{n=1}^{m} p_n \times A_D{}^n + P_D \quad (3)$$

In (3), A_D and P_D mean the AM and PM of the original signal which are amplified by PA, A_{DPD} and P_{DPD} mean the predistorted signal for compensation of the linearity. a_n and p_n represent the coefficient of m th term. Equation (3) can be represented as :

$$\begin{aligned} A_1 &= a_m \times A_D + a_{m-1} \\ A_2 &= A_1 \times A_D + a_{m-2} \\ &\cdots \\ A_n &= A_{n-1} \times A_D + a_{m-n} \\ &\cdots \\ A_{m-1} &= A_{m-2} \times A_D + a_1 \\ A_m &= A_{m-1} \times A_D + 0 = A_{DPD} \end{aligned} \quad (4)$$

and

$$\begin{aligned} P_1 &= p_m \times A_D + p_{m-1} \\ P_2 &= P_1 \times A_D + p_{m-2} \\ &\cdots \\ P_n &= P_{n-1} \times A_D + p_{m-n} \\ &\cdots \\ P_{m-1} &= P_{m-2} \times A_D + p_1 \\ P_m &= P_{m-1} \times A_D + P_D = P_{DPD} \end{aligned} \quad (5)$$

the general form of A_n and P_n are same in both (4) and (5). At the result of PD, A_m and P_m can be represented to the iteration of the general form for A_n and P_n. Thus PD also can be implemented by pipelined architecture such as CORDIC processor. The block diagram of pipelined architecture can be shown in Fig. 6.

F. Scaling Factor

In some cases, it can be found that the PA's output is smaller than the input by watching for the characteristic of the PA's

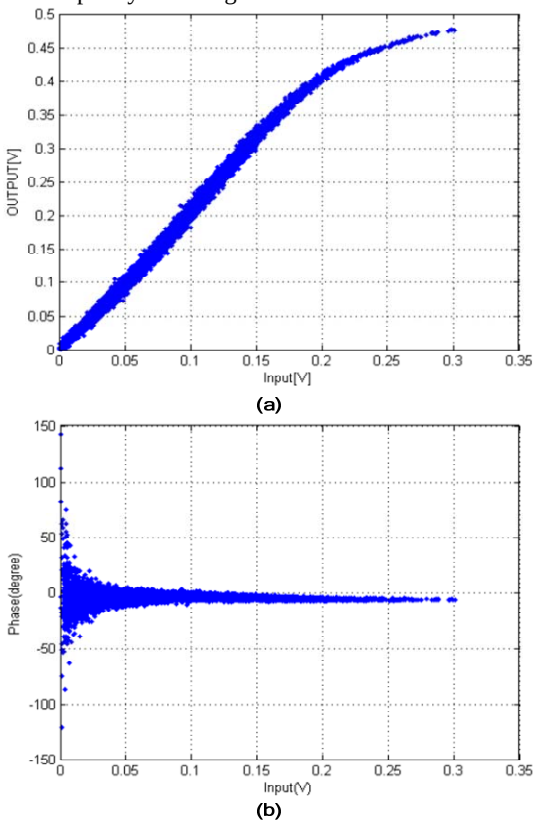

Fig. 5 PA's non-linearity for AM/AM (a) and AM/PM (b)

non-linearity. In this case, the predistored signal must have to small values as compared with the original value in order to turn up the output. In other words, form of inverse function applied to DPD would have bigger values than the original. When applying the DPD, ADC's output range uses the whole range of DPD's input for the high resolution and high accuracy. But the bigger inverse function makes the output range exceeding. Then it would make critical errors.The scaling factor is used to prevent errors over these ranges. The biggest value of the output is calculated according to the inverse functions and the scaling factor which makes that the output is in the range of DAC is found. The proposed DPD receives the precalculated scaling factor, adjusts its output to be in the range by multiplying it to the value after the PD operation.

III. Simulation

To verify the reliability of proposed DPD, the algorithm is simulated by MATLAB. Inverse function of the real PA's characteristic is represented by 5th-order polynomial and the data for the simulation also are taken by analyzing the real PA. Also it is assumed that the ADC and DAC have 20 MPSP speed and 14-bit input and output. In Fig. 7, (a) shows the power spectrum density of DPD output which is predistorted when the PA input is inserted in DPD and (b) shows the result of the PA output for the predistorted DPD output. The DPD+PA output doesn't have sidelobes.

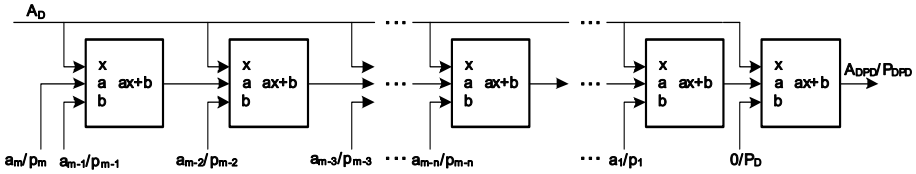

Fig. 6 Pipelined architecture of PD block

Thus it is confirmed that the proposed DPD can compensate the PA's linearity.

The table I shows the average error size for each output range. The result is taken by comparing the computation of the proposed DPD model with the general computer arithmetic. They are sorted for each 0.1 unit. The input range is assumed from 0 to 1. Since for I/Q-phase, each phase isn't over the 0.707. The errors are small enough to ignore.

IV. Conclusion

In this paper, the new polynomial DPD processor with pipelined architecture using the CORDIC processors is proposed. The proposed DPD can be changed the predistorded data according to varying environment and operate faster by pipelined architecture. Accuracy of its operation and compensation for PA's non-linearity is confirmed by simulation. The real DPD IC will operate as the algorithm and prove out. It will be recognized for communication standard by IP registration and will be used on the polar modulation transmitters.

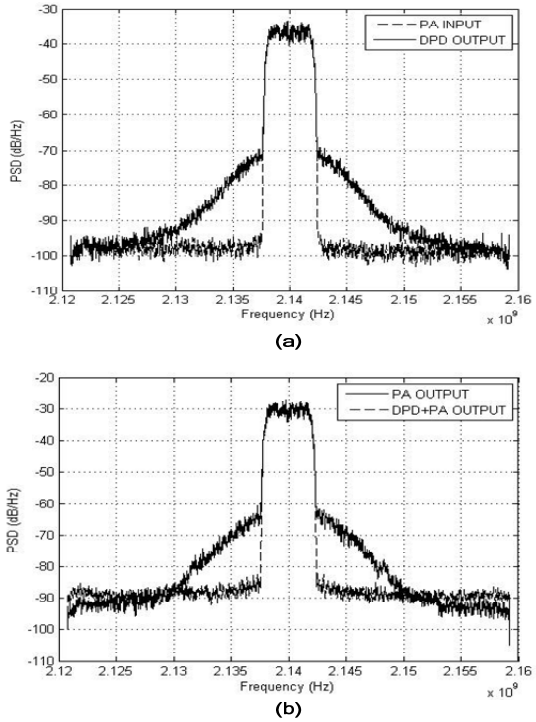

Fig. 7 PSD of the PA input and output. (a) PA input and its DPD output and (b) PA output for using that two inputs

TABLE I
The Amplitude of Average Errors of Proposed DPD

Output Range	Average Errors on Q-phase($\times 10^{-4}$)	Average Errors on I-Phase($\times 10^{-4}$)
0.0 ~ 0.1	3.62	3.63
0.1 ~ 0.2	4.12	4.14
0.2 ~ 0.3	4.74	4.70
0.3 ~ 0.4	5.19	5.26
0.4 ~ 0.5	5.93	6.26
0.5 ~ 0.6	7.00	6.93
0.6 ~ 0.7	8.21	7.93
0.7 ~	11.89	5.17

Acknowledgment

This work was supported by the Korea Science and Engineering Foundation (KOSEF) grant funded by the Korean government (MOST) (No. R01-2007-000-20377-0).

References

[1] J. K. Cavers, "Amplifier Linearization Using a Digital Predistorter with Fast Adaptation and Low Memory Requirements," *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*, VOL. 39. NO. 4 pp. 374-382, NOVEMBER 1990.

[2] J. Namiki, "An automatically controlled predistorter for multilevel quadrature amplitude modulation," *IEEE Trans Commun*., vol. COM-3 1, pp 707-712, May 1983.

[3] D. Hilbom, S. Stapleton, and J. Cavers, "An adaptive direct conversion transmitter," in Proc. *IEEE Vehicular Technol. Conf*, Boulder, CO, May 1992.

[4] Y. Nagata, "Linear amplification technique for digital mobile communications,"in Proc. *IEEE Vehicular Technol. Conf*, May 1989, pp 159-164.

[5] J. K. Cavers "Amplifier linearization using a digital predistorter with fast adaptation and low memory requirements," *IEEE Trans*. Vehicular Technol., vol. 39, pp 374-382, Nov. 1990.

[6] K. J. Muhonenm, K. Kavehrad, "Look-Up Table Techniques for Adaptive Digital Predistortion: A Development and Comparison," *IEEE TRANSACTIONS ON VEHICULAR TECHNOLOGY*, VOL. 49, NO. 5, pp. 1995-2002, SEPTEMBER 2000

[7] M. Ghaderi, S. Kumar, D. E. Dodds, "Fast adaptive polynomial I and Q predistorter with globaloptimization," Communications, *IEE Proceedings*, VOL. 143, No. 2, pp. 78-86, April 1996.

[8] E. Westesson, L. Sundström, "Low-Power Complex Polynomial Predistorter Circuit in CMOS for RF Power Amplifier Linearization," *Solid-State conference*, 2001. *ESSCIRC* 201. Proceddings of the 27th European, pp. 486-489, September 2001.

[9] B. Parhami, "Computer Arithmetic : Algorithms and Hardware Designs," pp. 361-377, Oxford University Press, Inc. 2000.