Design and Implementation of Self-Calibration for Digital Predistortion of Power Amplifiers

Dua Idris[2], Yannick Le Moullec[1,2], Patrick Eggers[2]
[1]Center for Software Defined Radio
[2]Department of Electronic Systems
Aalborg University
Niels Jernes Vej 12, DK-9220 Aalborg East, Denmark
dmi@es.aau.dk, ylm@es.aau.dk, pe@es.aau.dk

Abstract: The linearization of Power Amplifiers (PAs) on mobile handsets is a critical problem for stringent systems such as WiMAX. In this paper we present the results of our investigation on the feasibility of self-calibration from a theoretical point of view as well as on a FPGA-based hardware platform for the predistortion of PAs. We propose a system design in three phases i) self-calibration, ii) inverse function calculation and iii) predistortion. MATLAB simulation results show that the functionality of the self-calibration, inverse function calculation and predistortion can linearize the PA characteristics and that the FPGA implementation results of the self-calibration are promising, paving the way for an efficient self-calibrating digital predistorter.

Key–Words: DPD, self-calibration, PA

1 Introduction

The distortion caused by power amplifiers can be classified as linear and non-linear [1]. The most significant impairment is caused due to the non-linear behavior of the amplifier. In standards such as WiMAX that are based on OFDM, another major problem of concern is the Peak to Average Power Ratio (PAPR) [2]. Superposition of the subcarriers of an OFDM signal results in strong variations of the instantaneous signal power i.e. high signal peaks which are ultimately distorted non-linearly by the power amplifier. Non-linear distortion results in the formation of intermodulation products. Inter Carrier Interference (ICI) is caused by intermodulation products that disturb the transmitted signal whereas Out-Of-Band (OOB) radiation is due to intermodulation products located outside the transmission band which disturb signals transmitted on adjacent frequency bands. There are several techniques used for reducing the PAPR of the signal such as clipping and filtering, coding etc. Another problem with WiMAX standard is the use of non-constant envelope modulation schemes that places a strict requirement on the power amplifiers to operate linearly.

The remainder of the paper is organized as follows: Section 2 introduces the concept of predistortion and section 3 explains the characterisation of the PA. Section 4 is devoted to the description of the algorithm, system design and verification of the algorithm through MATLAB simulations. The FPGA implementation is discussed in detail in section 5. In section 6, we present the system testing and results which is followed by section 7 which concludes the paper.

2 Predistortion

In order to achieve a good spectral efficiency, higher order modulation schemes are used such as QAM which require well designed pulse shaping to reduce the out-of-band emissions. This kind of filtering places a requirement of a linear transmit amplifier due to the creation of a non-constant envelope signal that can provide fidelity to the varying signal envelope. Class A amplifiers can help to achieve linear fidelity but with certain backoff but this reduces the power efficiency[4]. Then there are other classes of amplifiers such as AB, B and C which have a good power efficiency but they have a non-linear response which is not desired since it creates broadening of the transmitted signal spectrum. Predistortion can be used to compensate for such a spectral spill over where in a signal is predistorted before power amplification in such a way that after amplifier distortion the resulting signal approximates the ideal signal [3].The concept involves the use a predistortion function which compensates for the nonlinear gain of the amplifier so that the cascaded functions produce a constant gain as shown in figure 1.

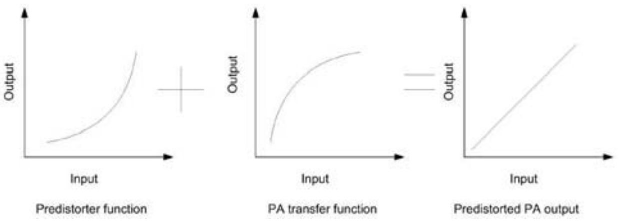


Figure 1: Predistortion concept.

In this work, we propose a self-calibration approach of which the concept is that in a mobile device, it is beneficial to be able to extract the changing characteristics of the PA in order to i) avoid the traditional measurements done during the production phase and ii) to adapt to changing conditions.

3 PA Characterization

The characteristics of the PA are needed to be able to calculate the inverse characteristics used in the predistorter. Hence the performance of the predistorter depends on the accuracy with which the characteristics are obtained. AM/AM characteristics give the non linear relationship between the input power and output power whereas the AM/PM characteristics give show the effect of the input signal on the phase of the output signal [5].

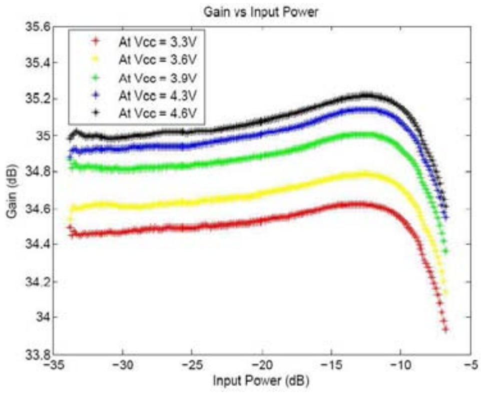


Figure 2: Gain Characteristic of the PA for different supply voltages.

The AM/AM and AM/PM characteristics of the PA[1] were obtained using a single tone Continuous Wave (CW) with power sweep from -27 to 0 dBm but including the attenuation it amounts to -33dBm to -6dBm.

The measurements were done for supply voltages ranging from 3.3 to 5V to be able to observe the variations in characteristics due to this factor and these are shown in figure 2 and in figure 3 for gain and phase shift characteristics respectively. It was observed that as the supply voltage drops, the gain decreases and the phase shift increases.

4 Algorithm and System Design

4.1 Comparison of the LUT and Polynomial methods

The system operation can be classified into two main processes: i) Digital Predistortion (forward) and ii) Self-Calibration (feedback). The input signals are digitally predistorted before being amplified such that the overall characteristic is linear. This process is called the digital predistortion phase which is performed as a continuous process . The other process is called the self-calibration phase. Since these characteristics are time varying, the coefficients for predistortion need to be corrected. This update requires an adaptation algorithm, for instance the look up table (LUT) and polynomial methods [4].

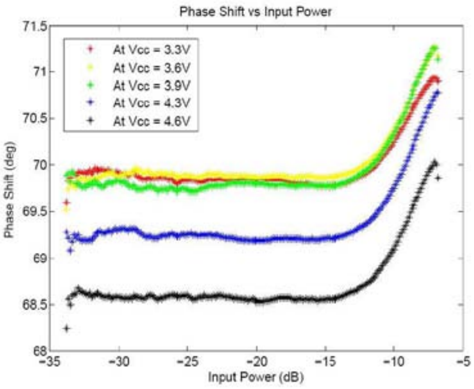


Figure 3: Phase Characteristic of the PA for different supply voltages.

In the LUT the correction values are stored in a table and indexing is used to select the required values. The only additional computation in this phase is calculation of index. In the case of polynomial method, the coefficients are calculated and stored in a table from where they are accessed by the polynomial function which calculates the inverse functions. This step of calculation is an added complexity for polynomial approach that is avoided in the LUT approach. Though less memory is required to store complex polynomial coefficients, it may not be preferred due to higher number of computations involved.

Once the amplifier has been characterized along its entire dynamic range to be able to obtain its nonlinear behavior, the obtained characteristics are stored in a table from which the inverse is calculated. For the LUT approach this would require scaling of gain, subtractions for phase error and memory to store the values. Then the inverse of these values is taken for using in the predistorter table. But for a polynomial approach, algorithm such as least squares(LS) [6] or matrix inversion is needed to calculate a polynomial fit to the data to extract the polynomial coefficients that are to be stored in a table. Then inverse of these coefficients is calculated to be stored in the predistorter LUT. The calculation of polynomial fitting curve is computationally complex compared to the LUT approach. Since the characteristics vary with time, the predistorter correction values need to be updated.

Table 1 compares the computational complexity of both the algorithms. Consider M to be the number of

[1] Confidential model reference

multiplications and A to be the number of additions. The value of N is usually 32, 64, 128 or 256 where N is the number of samples, and K is the order of the polynomial which could be 3,5,7 or even higher.

Table 1: Computational complexity comparison between LUT and polynomial. M:Multiplication, A:Addition, CM:Complex Multiplication, N:Number of samples, K:order of the polynomial.

Operation	LUT	Polynomial
DPD	(2M + 1A) and 1 CM	2(K-1)A, 2(2K)M + 1 CM
Self-calibration	N*M + N*A	(2N + 1)A and (2N + 1)M

Table 2 compares the memory requirements for both the algorithms, with N the number of samples and C the number of coefficients.

Table 2: Memory requirements comparison for LUT and Polynomial. m:memory units, N:Number of samples, C:number of Coefficients.

Operation	LUT	Polynomial
DPD	2Nm	2Cm
Self-calibration	3Nm	(2N + 2C)m

Considering the above two criteria, the LUT approach has been chosen over the polynomial approach for implementing predistortion.

4.2 System Architecture

For including a calibration set-up within the mobile device, the architecture shown in figure 4 could be used with the receiver chain. During calibration switches S1 and S2 are open and switch S3 is closed. Switch S3 is open when the device is transmitting or receiving. I1 and Q1 are the transmitted inphase and quadrature phase signals whereas I2 and Q2 are the received inphase and quadrature phase signals respectively. In principle, a time division duplexing (TDD) architecture can be used as it uses the same local oscillator for up and down conversion. But for a frequency division duplexing (FDD), a switch will be needed to select the same local oscillator for up and down conversion during the calibration phase which is done by switch S4.

In the next subsection, the system design required to implement the look up table algorithm is discussed.

4.3 System Design

This is divided into three main procedures: self-calibration (measurement of PA characteristic), Inverse Function Calculation and Predistortion, as shown in figure 5 and figure 6 respectively.

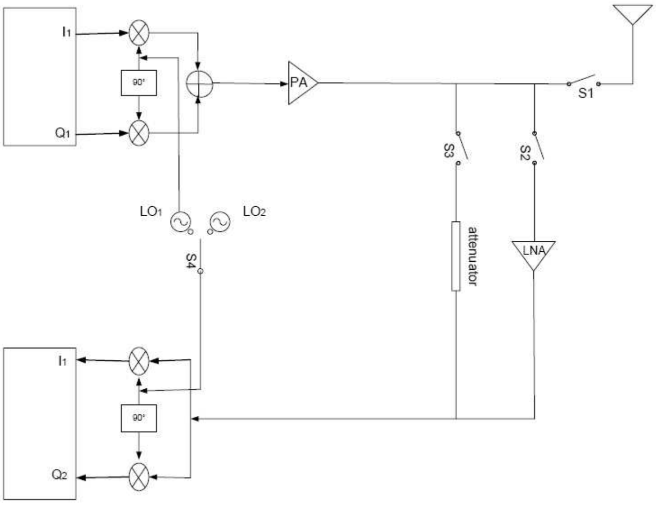


Figure 4: Architecture for including calibration setup within mobile device. Filters have been excluded for simplicity.

4.3.1 Self-Calibration

In order to characterize the PA, the calibration signal is sent through the PA and the output is downconverted, measured and stored. This process is done once during boot up or during idle time. Switch S1 and S2 are open and switch S3 is closed in figure 4. The characterisation is done over the dynamic range of the PA. The steps can be chosen to be uniform or non-uniform. In [7] it was shown that simple amplitude based uniform spacing is the most practical and near optimum choice to linearize the PA. Hence, equal step size is chosen and is calculated as per [8]: $s = (Vmax - Vmin)/N$ where s is the step size, $Vmax$ is the maximum input voltage, $Vmin$ is the minimum input voltage and N is the total number of table entries.

The PA has been characterized for 200 points with the calibration procedure shown in figure 5 [10].

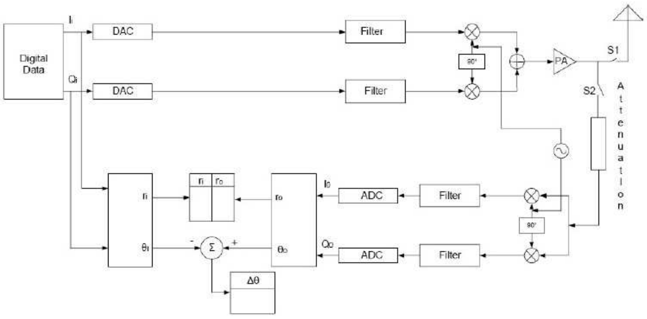


Figure 5: Self calibration phase of the power amplifier.

4.3.2 Inverse Function Calculation

When the PA has been characterized, all the measured input and output magnitudes and the phase difference per sample are stored in the look up table. The next step is to the calculate the inverse characteristic for the PA. This means that the gain and phase correction factors for the input samples have to be calculated. The gain correction factor compensates for the gain compression and the phase correction factor compensates for the phase shift per input sample. A simple algorithm is used for the inverse estimation and is explained in [8].

The basic concept of this step is to map the estimated table onto an inverse table [10].

4.3.3 Predistortion

The incoming sample is predistorted when it undergoes a complex multiplication with the corresponding values in the inverse look up table to obtain the inverse function so that the output of the PA can be linear. Figure 6 shows the implementation of digital predistortion. The table estimated from the calibration process is discarded and the inverse look up table is used to perform predistortion which can be done online.

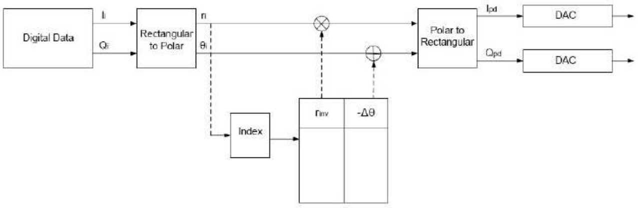


Figure 6: Digital Predistortion in Polar Co-ordinate.

4.4 MATLAB simulation results

The predistortion functionality is implemented in MATLAB to test the performance of the inverse function algorithm. The most relevant results are shown in the following figures.

Figure 7 shows the gain of the PA versus the input power. The compression of the gain at higher input power can be observed.

Figure 8 shows the inverse gain function which causes the gain expansion in order to compensate for the gain compression, such that the overall gain of the PA is linear.

Figure 9 compares the output power vs the input power for the case with and without predistortion and it is observed that after predistortion, the PA behaves linearly as expected.

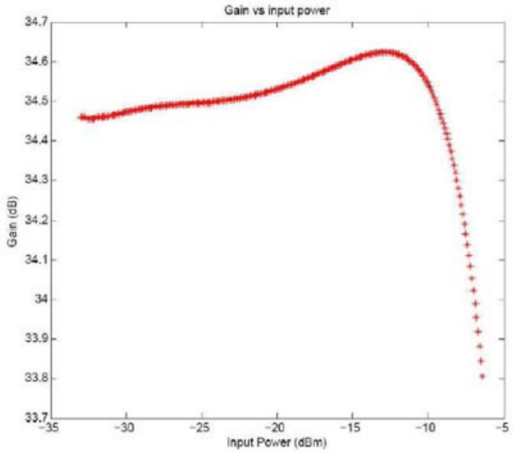


Figure 7: Gain Vs Input Power for the PA without predistortion.

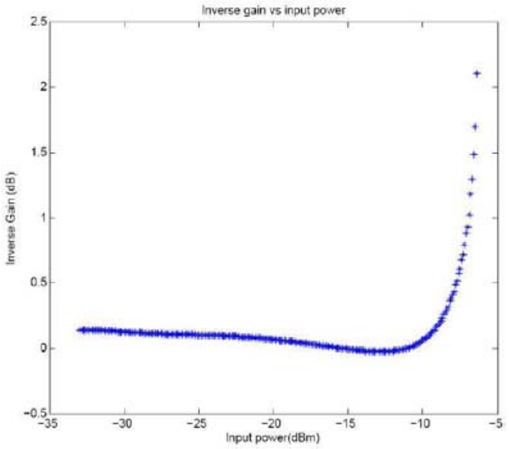


Figure 8: Gain expansion vs Input Power.

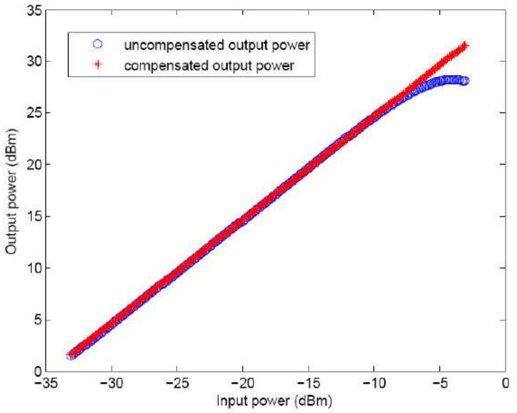


Figure 9: Output Power vs Input Power with and without predistortion.

5 FPGA Implementation of the self-calibration process

5.1 Setup and tools

In order to evaluate the feasibility of the proposed approach, the calibration phase (note that the FPGA implementation of the other phases, inversion and predistortion, are still under investigation) has been implemented on a XtremeDSP kit featuring a Virtex-4 FPGA [12] and tested with the PA connected through the ADCs and DACs of the kit. The tool used to program the FPGA is Xilinx's System Generator [13], a high level tool used inside Simulink/MATLAB, providing a convenient and rapid path to FPGA prototyping.

5.1.1 Hardware Cosimulation

Furthermore, we have used the co-simulation capabilities of System Generator to exchange data between Simulink and the FPGA. Hardware in the loop cosimulation allows to compile a Simulink design into hardware platform. The advantage of this feature is that the user can achieve hardware acceleration by running the design on hardware and also verify that the design is working in the hardware itself.

Figure 10 summarizes the process from the development of the algorithm till the stage when it is downloaded onto the FPGA [11].

5.2 Design for populating the calibration table: Description and test

In the design for populating the calibration table[10] the different amplitude levels are stored in the read only memory (ROM).Here, these values are set from 0.0145 to 0.3241 with a step size of 0.0049 to cover the entire dynamic range of the PA with a table size of 64 as per the reference measurements.

In order to make sure calibration loop is working correctly, in the initial test the PA is made to operate only in the linear region which means very low input levels are used ranging from 0.007077 to 0.22383 with step size of 0.0034. Only one sample is considered per amplitude level in this version of the design. In a completed version of the design, another counter will be included to count the number of samples per gain level. The DC value is sent through the DACs to the ADCs via the direct loopback and stored in the random access memory (RAM).

64 amplitude levels were stored in the ROM. There is a delay (or latency) of 1 time unit per read operation. The design is first simulated within Simulink. Simulation time of 67 time units is needed to write all the values to the RAM. A latency of 2 time units is observed as expected and as this was known due to the latency caused by the read operations, the value in the first RAM address is not considered. Another procedure to compensate for the latency would be to introduce some delay in writing to the RAM by inserting a delay block before the RAM.

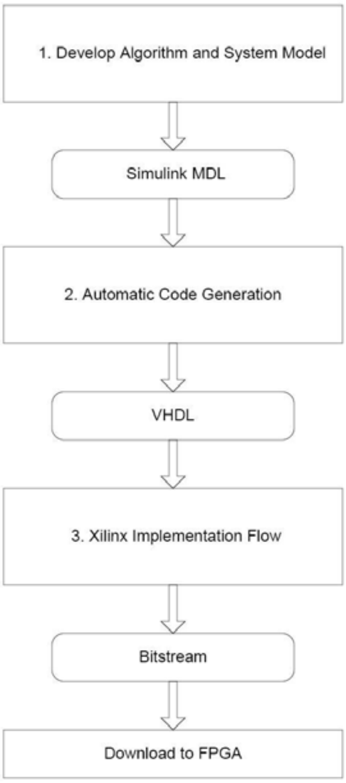

1. Develop Algorithm and System Model
Simulink MDL
2. Automatic Code Generation
VHDL
3. Xilinx Implementation Flow
Bitstream
Download to FPGA

Figure 10: System Generator Development Flow.

When the design was cosimulated, there are two modes available with JTAG for hardware cosimulation clocking called step mode and the free running mode which describes the way in which the hardware cosimulation block are synchronised with the associated FPGA hardware. In both the modes, the cable speed can be chosen to be 200KHz, 2.5MHz or 5MHz.

For hardware cosimulation using single step clock, the results displayed on the scopes are as shown as in figure 11 [10]. The values stored in the RAM are slightly attenuated by about 0.03V.

5.2.1 Implementation Results for the self-calibration module

The device utilization results obtained after synthesis and mapping using Integrated Software Environment (ISE) tool is summarized in table 3 and compared with the results obtained from the Xilinx resource estimator block that provides the estimate of FPGA resources re-

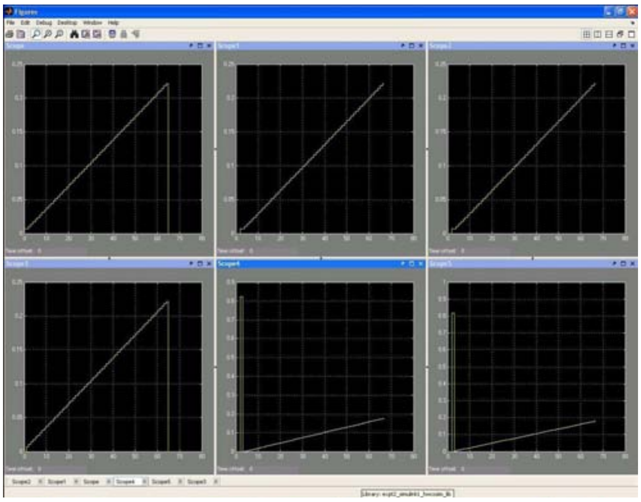


Figure 11: Display on scopes for populating the table using single step clock mode. Scope (top-left), scope1 (top-center) and scope2 (top-right) display the output of the ROM, RAM1 and RAM2 obtained by pure Simulink simulations whereas scope3 (bottom-left), scope4 (bottom-center) and scope5 (bottom-right) display the scope outputs for the same by performing hardware cosimulation.

Table 3: Device Utilization Summary of the the design used to perform the self-calibration process.

	Slices	Flip Flops	LUTs	IOBs	BRAM
Quick Resource Estimation	212	134	347.25	129	0
Post-Map Resource Estimation	50	78	41	0	3
ISE synthesis and Impl. Results	36	50	41	130	3
Device Utilization Summary	1%	1%	1%	29%	1%

quired to implement a System generator model. The quick resource estimation sums all of the FPGA area fields on the blocks whereas the post-map area resource estimation invokes the Xilinx map tool on the entire subsystem and reads the results from the created map report file.

The minimum design period is 4.022ns i.e. maximum frequency is 248.633MHz. This shows that the time required is extremely small in the range of few nanoseconds and the device utilization summary shows that only few resources are required.

5.3 Design of the Sine-Cosine Generator

In the design for sine-cosine generator [10] the Xilinx counter implemented was set to free running type of an up counter whose output is given by the equation 1.

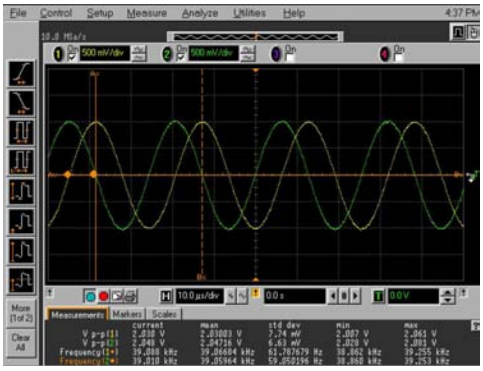
$$out(n) = \begin{cases} \text{Initial value if n} = 0 \\ (out(n-1) + \text{step})\text{MOD}2^N \text{ Otherwise} \end{cases} \tag{1}$$

where N is the number of bits in the counter. The output frequency is determined by the address increment. Multipliers can be used to multiply the sinusoids with the varying gain levels. In the completed system design, these gain levels will be stored in a ROM which will be controlled using a counter. The higher the value of N, the better the resolution. Full precision has been used for the multiplier blocks. For a test design as in this case, the amplitude level is provided as a constant manually via Simulink to test the functionality of the generator. The amplitude and frequency control functions are described next.

5.3.1 Amplitude control

In order to perform the calibration over the dynamic range of the PA, the amplitude has to be controlled accordingly. For varying the amplitude, the value of the constant is changed in Simulink manually. The design is first tested within Simulink and then cosimulated with the hardware in loop.

With the hardware in loop, observations are taken from the real oscilloscope and simulation time was set to 'inf'(infinite). The result is as shown in figure 12.



File Control Setup Measure Analyze Utilities Help
500 mV/div
500 mV/div
10.0 µs/div
0.0 s
0.0V
More (1of 2)
Clear All
Measurements Markers Scales

```
                current      mean          std dev        min          max
   V p-p(1)     2.038 V      2.03083 V     7.74 mV        2.007 V      2.061 V
   V p-p(2)     2.048 V      2.04716 V     6.63 mV        2.028 V      2.081 V
Frequency(1*)   39.088 kHz   39.06684 kHz  61.787679 Hz   38.862 kHz   39.255 kHz
Frequency(2*)   39.010 kHz   39.05964 kHz  59.058196 Hz   38.860 kHz   39.253 kHz
```

Figure 12: Observation made on oscilloscope when constant value = 1 using cosimulation. The yellow waveform is output from DAC1 and the green waveform is the output from DAC2. It is noted that V_{p-p} =2.00V.

The yellow waveform is the output of DAC1 which is connected to channel1 of the oscilloscope whereas the green waveform is the output of DAC2 connected to channel2. For channel1 and channel2, it was noted that V_{p-p} =2.06V which was as expected. This test was repeated for different values of step factors and it was noted that the amplitude was varying accordingly.

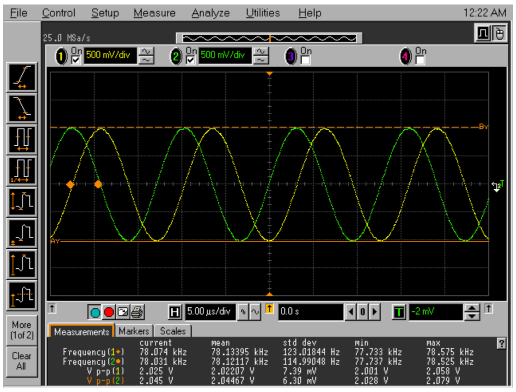
5.3.2 Frequency control

In order to perform the calibration of the PA over a range of frequencies, the frequency has to be controlled. To vary the frequency, the step size of the counter is varied. Similar to the amplitude control test, this function-

Table 4: Device Utilization Summary of the Sine-Cosine Generator.

	Slices	Flip Flops	LUTs	IOBs
Quick Resource Estimation	592	688	1119	54
Post-Map Resource Estimation	598	622	1043	0
ISE synthesis and Implementation Results	598	594	1043	55
Device Utilization Summary	3%	1%	3%	12%

ality is first verified in Simulink and then using hardware in loop.



```
                 current      mean           std dev          min          max
Frequency(1)     78.074 kHz   78.13395 kHz   123.01844 Hz     77.733 kHz   78.575 kHz
Frequency(2)     78.031 kHz   78.12117 kHz   114.99048 Hz     77.737 kHz   78.525 kHz
      V p-p(1)   2.025 V      2.02207 V      7.39 mV          2.001 V      2.058 V
      V p-p(2)   2.045 V      2.04467 V      6.30 mV          2.028 V      2.079 V
```

Figure 13: Observation made on oscilloscope with counter step factor = 2 using cosimulation. The yellow waveform is output from DAC1 and the green waveform is the output from DAC2.

The result from cosimulation with the hardware in loop is as shown in figure 13 when counter step is set to 2 and the frequency measured is 78.575KHz which is as expected. This test was repeated for different values of counter step factors and it was observed that the frequency was varying accordingly.

Hence, the amplitude and frequency controls for the sine cosine generator design have been implemented and tested.

5.3.3 Implementation Results for the sine-cosine module

The device utilization results obtained after synthesis and mapping using the ISE tool is summarized in table 4 and compared with the results obtained from the Xilinx resource estimator block.

The minimum design period is 9.69ns i.e. maximum frequency is 103.199MHz. The time required for this design is almost twice the time required for the design used to populate the table.

5.4 Design for Rectangular to Polar Conversion

The inverse function is calculated by compensating for the gain expansion and phase shift. The inverse table stores the inverse gain that has to be applied to the input signal and the phase correction factor by which the phase of the input signal has to be shifted. It is easier to represent this in polar co-ordinates. Xilinx provides computer coordinate rotation digital computer (CORDIC) blocks that can be used for the rectangular to polar conversion which is implemented using a fully parallel CORDIC algorithm in Circular vectoring modeIf complex input: inphase (I) and quadrature phase (Q) are applied, then the output will be its magnitude ’m’ and angle ’a’ which is computed as given in equation 2 and equation 3 respectively.

$$m = k\sqrt{I^2 + Q^2} \tag{2}$$

$$a = arctan\frac{Q}{I} \tag{3}$$

The magnitude has to be further scaled by a factor K = 1.646760 as it is not compensated for by in the processor which is explained in [9].

Figure 14 shows the scope outputs for magnitude and phase when cosimulation is performed with hardware in loop.

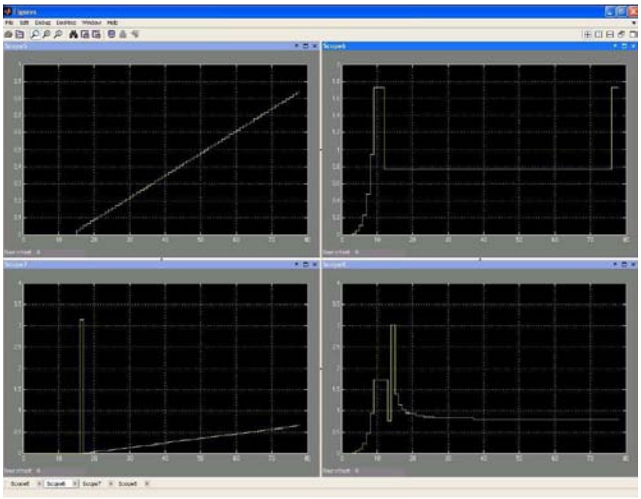


Figure 14: Cosimulation result with hardware in loop using the CORDIC block. Scope5 (top-left) and scope6 (top-right) show the magnitude and phase of the ROM content respectively whereas scope7 (bottom-left) and scope8 (bottom-right) show the magnitude and phase of the signal respectively, which is received via the direct loopback and saved in the RAM.

The CORDIC blocks are used for calculating the magnitude and phase. It has been tested as an individual module and the next step would be to integrate it with the other blocks. The artifacts observed in the figure for magnitude and especially phase calculation are due to the intermediate iterations and could be discarded.

Table 5: Device Utilization Summary of the rectangular to polar conversion design using CORDIC.

	Slices	Flip Flops	LUTs	IOBs	BRAM
Quick Resource Estimation	1568	1050	2903	244	0
Post-Map Resource Estimation	910	1624	1387	0	7
ISE synthesis and Impl. Results	896	1596	1387	245	7
Device Utilization Summary	5%	5%	4%	54%	3%

5.4.1 Implementation Results for the rectangular to polar conversion module

The device utilization results obtained after synthesis and mapping using ISE are summarized in table 5 and compared with the results obtained from the Xilinx resource estimator block. The minimum design period is 5.789ns i.e. maximum frequency is 172.741MHz.

The design for the population of calibration table and rectangular to polar conversion is as shown in figure 15.

6 Testing and Results

6.1 RF Setup

For testing the calibration process with PA in the loop, the RF setup made with off the shelf mini circuit components as shown in figure 16.

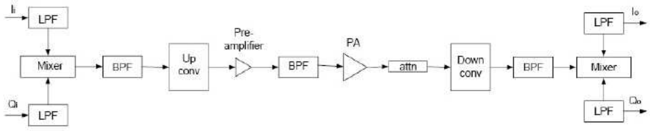


Figure 16: RF setup for testing the power amplifier.

The I and Q signals from the DACs are low pass filtered using filters with a bandwidth of 30MHz. The filtered signals are fed to the quadrature mixer. The output is filtered through a bandpass filter with bandwidth of 3MHz. The combined signal is upconverted to 2.43GHz. A preamplifier is added to boost the signal level by 10dB. This is followed by a bandpass filter to remove the spurious components due to amplification. Finally, the signal is amplified by the PA that has a gain of 34dB. In order to feedback the output of the amplifier back to baseband, the amplified signal has to be downconverted. But before this, it has to be attenuated so that the downconverter can tolerate or handle the signal. An attenuator of 30dB is used before downconversion. The downconverted signal is bandpass filtered before giving it to a quadrature mixer. This mixer produces I and Q signals which are bandpass filtered and then given to the ADCs.

6.2 Results

6.2.1 PA operated in Linear Mode

Figure 17 shows the output magnitude vs input magnitude plot when the amplifier was driven in the linear region. The input power is restricted to be below -5dBm. Due to noise floor, the curves are almost flat in the lower range of input magnitude and begins to rise linearly after 0.05V but at the higher input magnitude range.

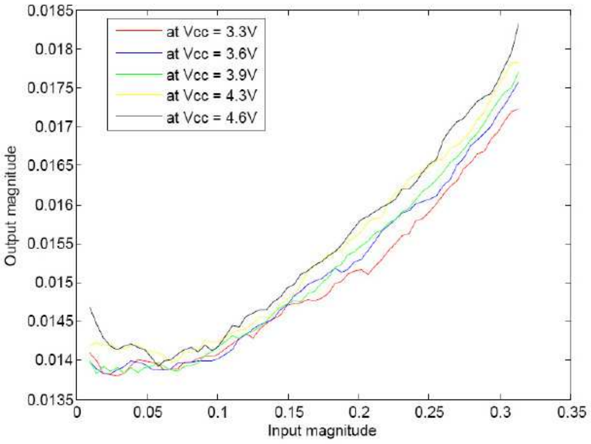


Figure 17: Output magnitude (in volts) vs input magnitude (in volts) for several supply voltages obtained from calibration procedure implemented on the FPGA when the PA is operating in the linear region.

6.2.2 PA operating under Compression

Figure 18 shows the output magnitude vs input magnitude plot when the amplifier was driven into the compression region. The input power exceeds -5dBm which is specified to be the maximum input power for the PA. It can be observed that the PA begins to saturate after it crosses the maximum input power allowed i.e. -5dBm.

7 Conclusion and Discussion

7.1 Conclusion

The need for dynamically linearising PAs on mobile handsets is a critical problem for stringent systems such as WiMAX. In this paper we have investigated the feasibility of self-calibration from a simulation point of view (MATLAB) as well as on a FPGA-based hardware platform. Firstly two approaches have been compared (LUT vs. polynomial). Our results shows that the LUT is cheaper than the polynomial method in terms of implementational complexity. Secondly, the system has been designed in three phases i) self-calibration, ii) inverse function calculation and iii) predistortion. Thirdly, MATLAB simulation results shows that the

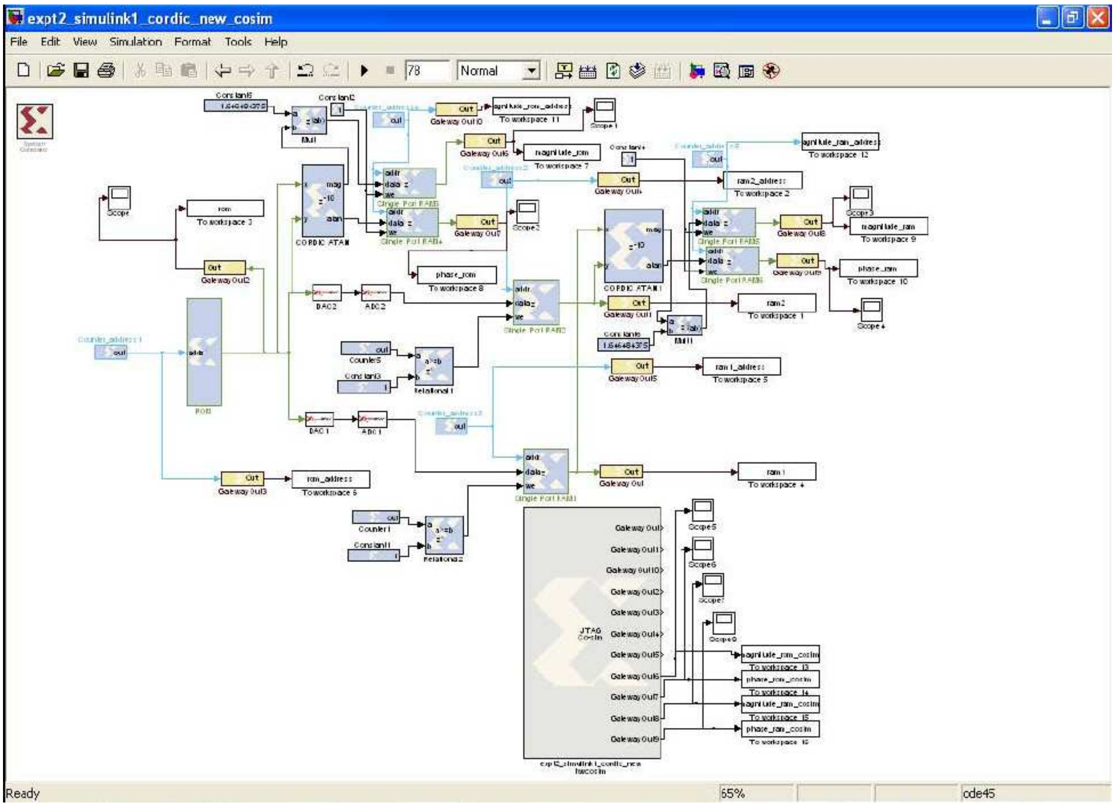


Figure 15: Design for population of table and rectangular to polar conversion using CORDIC.

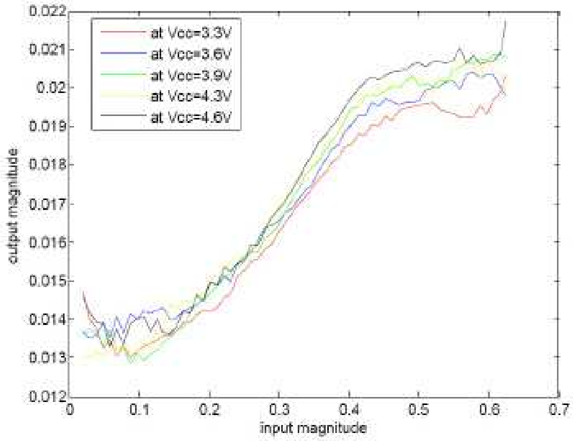


Figure 18: Output magnitude (in volts) vs input magnitude (in volts) for different supply voltages obtained from calibration procedure implemented on the FPGA when the PA is operating under compression.

functionality of the self-calibration, inverse function calculation and predistortion can linearize the PA characteristics. Finally, we designed, implemented and tested the self-calibration process on a FPGA and tested it using the PA.

7.2 Discussion

Some considerations related to the implementation of the system have been made which require further investigation. These considerations are explained using figure 19 which illustrates the plot of amplitude vs time for the transmitted against received signal.

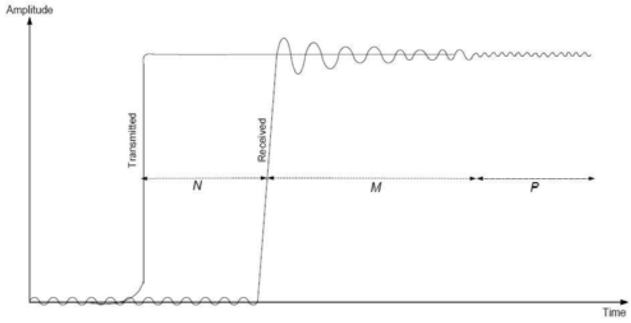


Figure 19: Amplitude vs Time plot for Transmitted and Received signal

The transmitted signal is the calibration signal in this case and the received signal is the signal amplified by the PA and fedback to the baseband for estimating the characteristics of the PA. Three important points can be noted from this figure which are as follows:

- N is the number of samples by which the received signal is delayed. This could be calculated using the correlation function.
- M is the number of samples it take for the received signal to settle down. It is important to wait at a particular gain level for a time period equivalent to the settling time.
- P is the number of samples considered for noise averaging.

In this work, DC signal is used for testing hence delay does not need to be considered. The system is assumed to settle in negligible time and is also assumed to be noiseless.

Acknowledgement

The authors would like to acknowledge the support of MOTOROLA A/S in Aalborg and explicitly thank Per Dahlgaard, Jorge Martires and Bjarne Moller-Jensen for their help in carrying out the PA characteristic measurements. And from Aalborg University, they would like to thank Jan Mikkelsen for the valuable discussions and also Kim Olesen for providing technical support.

References:

[1] National Instruments, Sources of Error in I/Q Based RF Signal Generator.

[2] A. Saul, Ed., Analysis of Peak Reduction in OFDM Systems Based on Recursive Clipping, Vol. 1.Proc in Int. OFDM Workshop, September 2003.

[3] N.Ceylan, J.E.Mueller and R.Weigal, Optimization of EDGE Terminal Power Amplifiers Using Memoryless Digital Predistortion, IEEE Transactions on Microwave Theory and Techniques, Vol.53, No.2, February 2005.

[4] K.C. Lee and P.Gardner, Comparison of Different Adaptation Algorithms for Adaptive Digital Predistortion based on EDGE standard, IEEE Transactions on Microwave Theory and Techniques, Vol.53, No.2, February 2005.

[5] O.Hammi, S.Boumaiza and F.M.Ghannouchi, On the Robustness of Digital Predistortion Function Synthesis and Average Power Tracking for Highly Nonlinear Power Amplifiers , IEEE Transactions on Microwave Theory and Techniques, Vol.55, No.6, June 2007.

[6] T. Lynge Kjeldsen et al., Adaptive Estimation of a Wireless Channel, Aalborg University, Applied Signal Processing and Implementation, 8th Semester Project Report, 2006.

[7] J. K. Cavers, Optimum Table Spacing in Predistorting Amplifier Linearizers, in IEEE trans. on Vehicular Technology, vol. 48, no. 5, September 1999.

[8] D. S. Han and T. Hwang, An Adaptive Predistorter for the Compensation of HPA Nonlinearity, in IEEE transacions on broadcasting, vol. 46, no. 2, June 2000.

[9] J. E. Volder, The cordic trigonometric computing technique, in IRE transacions on Electronic Computers, vol. 8,1959.

[10] Dua Idris, Yannick Le Moullec and Patrick Eggers,Design and Implementation of Self-Calibration for digital predistortion of Power Amplifiers, DNCOCO 2007, November 5-7, Trinidad and Tobago Islands.

[11] Steve Zack and Suhel Dhanani,DSP Co-Processing in FPGAs: Embedding High-Performance, Low-Cost DSP Functions, White paper, WP212(V1.0), March 18, 2004.

[12] XtremeDSP Development Kit-IV User Guide, NT107-0272 Issue 1, March 9, 2005.

[13] www.xilinx.com