Implementation of Self Interference Cancellation for WCDMA Repeater

Pon Rattanawichai and Chaiyod Pirak
The Sirindhorn International Thai-German Graduate School of Engineering (TGGS)
King Mongkut’s University of Technology North Bangkok
1518 Pibulsongkram Road, Bangsue, Bangkok 10800, Thailand

Abstract. In this paper, the FPGA (Field Programmable Gate Array) implementation of self interference cancellation technique based on an adaptive LMS (Least Mean Square) algorithm is presented. By using the FPGA Virtex@6 HW module, the field data measured through a RF repeater is adopted to improve a signal quality and to reduce oscillation of the system due to the feedback interference signal coming from transmit antenna of a WCDMA radio repeater. As a result, the proposed module, which was simulated by Xilinx System Generator, a high efficiency and achieves the a similar performance in comparison with the implementation result. Because of FPGA implementation, the designed technique offers more flexibility than existing methods.

Keywords: Interference cancellation, FPGA.

1. Introduction

Adaptive filtering algorithms [1] are used widely in digital signal processing. Typical applications in communication systems are adaptive equalization, echo cancellation, speech and image encoding, and noise and interference reduction. An adaptive filter is a digital filter that performs digital signal processing, and it can adapt its performance based on the input signal. The structure of an adaptive algorithm consists of a temporal variant system where the transfer function directly [2] depends on the statistical characteristics of the input signal, an algorithm connecting the inputs, and a signal informing about the error obtained by the adaptive system. The adaptive filter is subdivision in Digital signal processing (DSP). The adaptive filtering algorithms can be implemented on many different platforms, such as an Application Specific Integrated Circuits (ASIC) custom chip to general purpose processors or DSP microprocessors. While DSP microprocessors provide the benefits of flexibility and programmability in implementation of a large set of algorithms, their design is suboptimal for specific applications due to the sequential execution that limits possibilities for parallel operation. Moreover, their power consumption is high for using in portable applications and internal hardware structure is not optimized. On the other hand, ASIC could provide the solution that meets all of the constraints, but they lack the flexibility that could be offered on a DSP processor in FPGA. Hence, the design and development process of an ASIC can be time consuming and expensive, being considered only when high volume manufacturing is foreseen or very strict applications require such design and interface. Historically, the signal processing hardware has been a limiting factor on DSP implementation. The computation time and physical size of the hardware necessary for complex DSP estimation has often precluded them from being deployed practically in modern mobile communication systems [3]. Recently, advances in the fabrication of DSP in communication devices[4], i.e. FPGA-based processing, has emerged as one of the most attractive technologies for complex DSP estimation due to the inherent flexibility of the hardware in addition to the ability to optimize the execution of the algorithm between hardware and software.
FPGA solutions alleviate the risks due to silicon development costs and design turnaround times. At the same time, the multiprocessor abstraction retains the advantage of software programmability and provides an easy way to deploy applications from an existing code base. FPGA also allows [6] the designer to customize the multiprocessor for a target application. The designers can iteratively explore other configurations or offload critical functions into co-processors on the fabric to improve performance. This application requires an implementation that meets these constraints with the shortest time to market.

In the traditional approach, a design tool takes the VHDL description of the hardware and generates a bit stream (configuration file for the FPGA device). Nowadays, the use of High-Level Synthesis (HLS) tools is increasing, enabling designers to enter designs at a much higher level of abstraction. Such an HLS tool would accelerate the design process taking the algorithmic description of the required hardware, introducing certain area and performance requirements according to the design needs and automatically generating the bit stream. In this paper, we develop HLS implementation of the interference cancellation system (ICS) based on adaptive filtering algorithms for a WCDMA repeater [5], providing some hints and comparisons for each design process. Xilinx® devices and software environment have been chosen. Even though other manufacturers offer similar devices and tools in this case, Xilinx devices are only considered in this paper.

This paper is organized as follows. In Section 2, a background on the LMS algorithm and the multimedia board and tools used for implementation are presented. In Section 3, the specifications of the system in the hardware implementations in described. In Section 4, the simulation results in terms of time and accuracy are demonstrated, and the paper is concluded in section 5.

2. The Interference Cancellation Radio Repeater System Model

![Figure 1. Interference cancellation system with adaptive filter.](image)

In this section, the interference cancellation algorithm based on the least mean–squared (LMS) algorithm [1] is presented for the WCDMA repeater. In this algorithm, the adaptive filter is trained to produce the desired output for a given radio signal. The implementation of this algorithm requires three steps: (1) input filtering, (2) adaptive processing, and (3) output filter playback;

The LMS algorithm is a widely used technique for adaptive filtering. It is based on the estimation of the gradient toward the optimal solution using the statistical properties of the input signal. A significant feature of the LMS algorithm is simplicity. In this algorithm, filter weights are updated with each new sample as required to meet the desired output. The computation required for weights update is illustrated by equation (4). Now, the input values \( u(n), u(n-1), u(n-2) \ldots u(n-N) \) form the tap input vector \( u(n) \), where \( N \) denotes the filter length, and the weights \( \hat{w}_0(n), \hat{w}_1(n), \ldots \hat{w}_{N-1}(n) \) form the tap weight vector \( \hat{w}(n) \) at iteration \( n \). It can be shown that the LMS algorithm is given by the following equations:
\[ y_E(n) = \tilde{w}^H(n)u(n) \]  

\[ \text{in}_{\text{sum}}(n) = d(n) + y_F(n) \]  

\[ e(n) = \text{in}_{\text{sum}}(n) - y_E(n) \]  

\[ \tilde{w}(n+1) = \tilde{w}^H(n) + \mu u(n)e(n) \]  

where \( y_E(n) \) denotes the output update, \( d(n) \) denotes the desired output, \( e(n) \) denotes the filter error, \( y_F(n) \) denotes the feedback channel which is used to update the weights, \( \mu \) denotes a step side, and \( \tilde{w}(n+1) \) denotes the new weight vector that will be used in the next iteration.

3. **Hardware Implementation**

In this section, the ICS described in the previous section is implemented on the Xilinx Vertex-6 board that provides a convenient environment board for embedded processing applications. The board contains a Xilinx XCE6VLX240T FPGA with up to 360 user-I/O pins, 750-pin FPGA package and over 240,000 logic cells. The Vertex-6 board also provides the MicroBlaze 32-bit embedded RISC processor and the Xilinx System Generator [9].

In fact, it is possible to use floating-point on FPGA, but it is very intensive in terms of logic resources usage and does not take full advantage of the parallelization possibilities offered by an FPGA [11]. Therefore, the fixed point calculation are use instead, but this can introduce analog to digital conversion (ADC) [7], [8], [9] quantization error, coefficient quantization error, overflow error and round off error. All of these issues must be addressed and handled properly to reduce the errors committed by fixed point arithmetic. In this work, we use 32-bit normalization. This means that the scale constant is equal to 2,147,483,648 that is the size of all data inputs and outputs.

3.1. **Implementation Considerations**

The ICS hardware implementation process is composed of an input filter, an LMS-based adaptive filter, and an output filter, as shown in Figure 1. Parts of the output signal radiating from the transmit antenna go through the feedback channel between the transmit antenna and the receive antenna, as shown in Fig. 1, and enter the receive antenna. The feedback signal is combined at the receive antenna with the input signal that comes from a base station or mobile stations. The LMS algorithm implemented for the adaptive channel estimation calculates the coefficient vector \( w(n) \) for a transversal filter. The calculation is based on the error signal \( e(n) \), the delayed output signal vector \( x(n) \), and the previous coefficient vector \( w(n) \). In equation (1), \( \mu \) is the parameter that controls the convergence rate of the LMS algorithm. In order to cancel the actual feedback signal \( y_F(n) \) using the estimated feedback signal \( y_F(n) \), it is important to ensure that the delay estimation of the feedback signal is exact. The delay of the feedback signal is calculated as the sum of the propagation delay between the transmit and receive antennas of the repeater, and the system delay caused by various filters and signal processing blocks in the repeater. The delay may be obtained by correlating the desired output \( d(n) \) and output signal \( x(n) \) in the adaptive filter of the repeater illustrated in Fig. 1.

3.2. **FPGA Implementation Based on Simulink System Generator High Level Synthesis (HLS) Tool.**

The main parts of the proposed system are composed of:

1. “New Weight Vector” block: obtaining an internal value named “Input Values” to be used further in the coefficients’ update. Figure. 2 shows such block diagram.
2. “ICS Input” block: buffering the input received signal in FPGA after performing ADC conversion.
3. “Filter Error Calculation” block: canceling the interference feedback signal \( y_F \) using the estimated feedback signal \( y_F \).
4. “ICS Output” block: buffering the cleaned output signal before performing DAC conversion.
5. “Normalization” block: suppressing the amplitude signal to an effectively sufficient level in order to accelerate the convergence rate of the LMS algorithm.

![Figure 2. Top level block diagram for the HLS tool.](image)

### 4. Experimental Results

System Generator for HLS descriptions has been used to simulate approaches for hardware implementation. Regarding ICS output result, for HLS simulation results, Fig 3 (a1) shows the desired signal output $d(n)$, (a2) shows the filter output result $x(n)$ of the ICS system, and (a3) shows the error of the desired signal output $d(n)$ with the filter output result $x(n)$ is close to zero. Figure (a4) shows the filter output result $x(n)$ (Pink) compared with the simulated desired signal output $d(n)$ for simulation and (a5,a6) shows the feedback signal $y_F(n)$ and the filter output $y_E(n)$, respectively. In fact, we want to cancel the feedback signal $y_F(n)$ whenever it enters the repeater; hence the estimated feedback signal $y_E(n)$ is generated by an estimated channel filter with an impulse response $w(n)$. From the result of figure (a3), the error signal of desired signal output $d(n)$ and filter output $x(n)$ converges to zero. As a result that appeared in figure (a4), a filter output signal $x(n)$ converges to desired signal output $d(n)$ so that the filter output $y_E(n)$ and feedback signal $y_F(n)$ are nearly coincided.

$$
EVM = \frac{\sqrt{\sum_{k=0}^{L-1} |e_k|^2}}{\sqrt{\sum_{k=0}^{L-1} |R_k|^2}} \times 100\% \tag{5}
$$

In (5), $e_k(n)$ is the error vector between the output signal of the ICS system, and the real input signal $R_k(n)$ is the reference signal. In addition, $L = 1000$ is the number of samples needed to obtain a reliable EVM value. Overall, as shown in Figure 4, the simulation results show that the EVM performance of the proposed interference cancellation algorithm based on LMS adaptive algorithm is about 20%.

![Figure 3. (a1) Desired signal $d(n)$. (a2) the data output $x(n)$. (a3) Input-Output error, (a4) Input-Output compere, (a5) $y_F(n)$. and (a6) $y_E(n)$.](image)
5. Conclusion

In this paper, the self interference cancellation system for a WCDMA repeater has been implemented on the Xilinx System Generator simulation platform. It is worth noticing that the performance of the implement system, measured by an error vector magnitude (EVM), is similar to the reference module such that the EVM is about 20%. In addition, the designed system can be implemented to the real FPGA development board, which would be the future work.

6. References

[6] Marc Defossez, Connecting Virtex-6 FPGAs to ADCs with Serial LVDS Interfaces and DACs with Parallel LVDS Interfaces, XAPP1071 (v1.0), June 23, 2010.