ABSTRACT

Traditional methods for channel equalization often involve cascading, a pipelined A/D converter and a FIR filter. Given the nature of pipelined converters, a new method is proposed that performs partial equalization within the pipelined A/D thus reducing the number of taps required for the FIR filter. Simulations have shown it is possible to save 54 flip-flops at the cost of -32 dB of NSR error for a typical Ethernet channel equalizer. Further hardware can be saved if more error can be tolerated in an application. The system level design and equations describing the hardware requirements have been provided.

1. INTRODUCTION

Pipelined A/D converters possess several features that make them suitable for integrated digital communication receivers. Up to a few hundred megahertz, they are amenable to CMOS implementation, consume less power than flash converters and can have greater resolution for less hardware [1, 2, 3]. They have been applied to many applications including gigabit Ethernet [4], magnetic storage [5], cable modems [6], and DSL [7]. In all of these applications, the converter is generally followed by a digital equalizer to eliminate intersymbol interference. The equalizers are usually finite impulse response (FIR) filters with programmable tap weights.

At high clock rates, the equalizers can have high power consumption, especially when long word lengths necessitate many registers in the data path [8]. Furthermore, the latency associated with a pipelined converter and linear equalizer can be problematic when enclosed within a timing recovery loop.

Pipelined converters, by their nature, simultaneously process several time-delayed samples of the system input. This fact can be exploited by combining the functions of the FIR equalizer with the digital error correction logic included in all practical converters resulting in a system with less hardware and latency and little or no noticeable difference in performance.

In section 2 of this paper, the architecture of a standard 1.5-bit per stage pipelined A/D converter is described, along with the modifications required to accommodate additional outputs. The additional outputs are utilized to perform partial equalization in section 3. The resulting hardware savings are estimated. In section 4, simulations are performed using gigabit Ethernet as a candidate application for the proposed technique to quantify the impact on system performance.

Figure 1: Left: Typical M-Bit Pipelined A/D converter, Right: Stage design. (HA: Half Adder, FA: Full Adder)
2. MODIFIED PIPELINE ARCHITECTURE

In general, Pipelined A/D converters have an overall system architecture illustrated in Figure 1. A 1.5 bits per stage architecture is shown with 0.5 bits of overlap per stage to allow for error correction in the decoding logic. For the details of each stage, the reader is referred to (for instance) [9].

For an N-bit A/D converter N-1 stages are necessary. This implies after N-1 clock cycles an N-bit digital representation of the input will be available. Upon closer inspection, it can be noted that due to the sample and hold units between each stage and the flip-flops before the adders, future digital samples exist within the pipeline stream. For instance, if a present N-Bit sample, X [k], is available, by tracing back one flip-flop from each adder, one can access the (N - 1) most significant bits of a future sample that will arrive on the next clock cycle, i.e. X [k + 1]. This process can be repeated to access the X [k + 2] sample with (N - 2) Bits of accuracy. Based on this observation, by adding a few more half adders to the design of the pipelined A/D converter, we can extract X[k], X[k+1]...X[k+B] for all B < N - 1 simultaneously; the only tradeoff in terms of performance would be the drop of one bit of resolution for each additional output.

The first \( M = (N - B) \) Bits of the pipeline are designed typically and the remaining B-Bits are designed as illustrated in Figure 2. The pipeline stages are all unchanged; changes are required only to the decoding logic. (The modified decoder requires more half adders to provide the X[k], X[k+1]...X[k+B] samples simultaneously.) The last stage produces X [k] with N = (M + B) Bits of data while the previous stages produce X[k+1]...X[k+B], each with one bit less resolution than the following stage.

3. PARTIAL EQUALIZATION IN THE A/D

In the last section, a modification to the decoding logic in a traditional pipelined A/D converter was described which provides multiple samples of the input waveform simultaneously (albeit with reduced resolution). Figure 3 illustrates how having X[k], X[k+1]...X[k+B] available simultaneously can be used to create a FIR filter that uses no delay elements, i.e. no registers or flip flops. Naturally, this is favorable due the hardware reduction.

Such a filter can be used to perform the first few taps of equalization in a communications receiver. Often, the first few tap weights in an equalizer do not dominate the filter response, so the reduced resolution in these tap weights has minimal impact. For instance, if three taps are to be incorporated, then X[k], X[k+1]...X[k+3] must all be present concurrently. In order to accomplish this, we separated the design of the Pipelined A/D converter into two phases, i.e. Figure 2. The first phase is designed as a typical pipelined A/D converter, and the second phase, with some extra adders in the digital decoder, provides the multiple outputs necessary to create a register-less FIR filter.

Figure 2: Two stage design of a Pipelined A/D converter with concurrent access to X[k], X[k+1]...X[k+B]

Figure 3: Register-less FIR Filter
Figure 4 illustrates the traditional versus the new method for channel equalization. For example, if an application requires channel equalization with a 9-Bit A/D converter followed by a 10-Tap FIR filter; one can use the architecture shown in Figure 4B. Should we decide to incorporate 4-Taps of the filter into the A/D converter, we can design the 9-Bit pipelined A/D converter, as described in Section 2, to provide concurrent access to $X[k], X[k+1], \ldots X[k+4]$. As illustrated in Figure 3, samples $X[k+1], \ldots X[k+4]$ can be used to create a register-less filter. The following FIR filter can then be reduced to 6 taps and thus many flip-flops will be saved.

The number of flip-flops, half and full adders required using this approach are a function of the number of bits of the A/D converter, the total number of taps of the FIR Filter and the number of taps incorporated into the A/D converter, Given:

$$n = \text{Pipeline A/D Converter Resolution (Bits)}$$
$$b = \text{Number of Taps Incorporated into A/D}$$
$$t = \text{Number of Taps required for FIR Filter}$$

$$\# \text{Flip-Flops} = n \times (t - b - 1) + \sum_{i=0}^{n-b-3} ((n - b - 2 - i) \times 2) + \sum_{i=n-b}^{n-1} i$$

$$\# \text{Half Adders} = 1 + \sum_{i=n-b+1}^{n} i$$

$$\# \text{Full Adders} = n - b - 2 \quad \text{(for } b <= n - 2)$$

It is important to note that the second & third equation does not measure the over all number of half adders but rather the difference when compared to the traditional design of an A/D followed by a FIR filter. In any case where the lower limit of the sum is smaller than the upper limit (for instance, $b = 0$) that summing element evaluates to zero.

The number of half adders increases with the number of taps that are incorporated into the A/D; however the number of flip-flops is reduced with more backtracking. The tradeoff is generally favorable, as demonstrated by the results in the next section.

Due to the reduced resolution of samples $X[k+1], \ldots X[k + B]$, this method of channel equalization will introduce some error when compared to the original method. This error will increase as one decides to incorporate more and more taps into the A/D converter. For example, the sample $X[k + B]$ will only have $(N-B)$ Bits of resolution, and no matter how high the resolution of the filter coefficients are, error will be introduced. Therefore there will be a point where the hardware reduction won’t justify the error introduced. This idea is further explored in the simulation section of this paper.

4. ETHERNET SYSTEM SIMULATION

In order to illustrate the performance of such a system in a practical application we have chosen to simulate an Ethernet channel with an equalizer. Figure 5 shows our Matlab simulation setup. The ABCD parameters of 100 meters of Cat-5 twisted pair Ethernet cable have been calculated, from which the impulse response of the channel has been extracted, thus creating a model for the Ethernet channel. Using Matlab, 5 level PAM data at 125 M Symbols/Sec was generated and fed into the channel model. The output of the channel was equalized by a traditional (reference) system including a 9-bit pipelined A/D converter followed by a 10-tap FIR filter. The output of the channel was also fed into models of pipelined A/D converters with $B$ taps of FIR equalization incorporated followed by FIR filters with $(10 - B)$ taps. For instance, if the pipelined A/D converter has 4 taps incorporated into it, the FIR filter following this A/D converter is 6-taps deep rather than 10-taps. The output was subtracted from the reference design to measure the error introduced by the simplified hardware.
Thus this method is an effective way to reduce hardware requirements, while maintaining acceptable performance.

6. REFERENCES


