
Charles E. Berndt, Student Member, IEEE and Tad Kwasniewski, Senior Member, IEEE

Department of Electronics,
Carleton University,
Ottawa ON, K1S 5B6
cberndt@doe.carleton.ca; tak@doe.carleton.ca

Abstract

This paper provides a review of the structures and algorithms used in receive end adaptive equalization. These structures are necessary in order to allow high speed signaling over several gigabits per second across a serial backplane channel. As the data rates continue to increase over these channels the causes and the techniques used mitigate interference become more and more important.

1. Introduction

The circuitry of interest in this paper is used to allow communication over the high speed serial backplanes commonly found in large computer systems and telecommunications equipment. It should be noted that the architectures and techniques to be reviewed here are not limited to this application. They were originally employed for data communication over telephone lines and continue to find new applications in developing areas such as optical fiber systems [1], [2]. Regardless of the applications, the primary driving forces of this equalization technology is the desire for greater data transfer rates in electronic equipment. Also of great importance are the manufacturing costs and technological limitations associated with adding extra chip-to-chip interconnections [3]. Therefore, there has been a significant amount of research on making serial connections faster rather than increasing the amount of parallel connections. These efforts have yielded a variety of circuits and algorithms which can adaptively change in order to allow for error free communications at high speed. A design using the techniques discussed here has recently been reported with a bit error rate less than 10\(^{-15}\) at 6.25Gb/s over a 30 inch, 4-layer fire-resistant (FR-4) backplane [13].

2. The Channel

This paper is primarily concerned with the methods used to transmit high-speed serial data across backplane channels found in large scale computer and telecommunications equipment. Therefore, it is of great importance to understand the channel composition as well as how data is altered across this channel.

2.1. Construction and Limitations

Fig. 1. The typical Backplane set-up. Both the receiver IC and the transmitter IC are mounted onto daughter cards which are connected to the Backplane via a connector. The dotted line indicates the path of the transmitted symbols.

As the transmitted signal is sent from one IC mounted onto a daughter card to another, it must pass through many different discontinuities along the
channel. These include the chip bonding, package, card, connector and most importantly the various lengths of copper trace. These elements act through various processes such as reflection and attenuation, which all affect the information symbols as they are sent across the channel. The net effect results in distortion of the information bits and is called Inter-Symbol Interference (ISI.)

Looking at the impulse response of a typical 30” FR-4 Backplane (seen below in Fig. 2) one can see how a transmitted bit is distorted (by the low-pass nature of the channel) into having a long ‘tail’. This long impulse response causes information bits sent along the channel to have a detrimental effect on the following bits. At higher data rates this effect is only enhanced and can cause a significant number of errors.

![Normalized Impulse Response of a 30” FR-4 Backplane Channel](image)

**Fig. 2.** The Normalized Impulse response model of a 30” FR-4 Backplane channel. It can clearly be seen that the lengthy roll off will directly interfere with any following data bits, causing an offset.

### 2.2. Crosstalk

Another major source of interference found in the backplane environment is crosstalk. This effect stems from the fact that there are many separate copper traces in close proximity running parallel along the backplane, through connectors and packages. The close proximity and length causes a great deal of capacitance through which information bits are coupled onto the adjacent traces. Crosstalk is further divided into the two separate types commonly seen in this environment. Fig. 3 clearly illustrates both of these effects and where they occur in the topology. The first is called Far End Crosstalk or FEXT. It is caused by a transmitted symbol being coupled from one channel onto a parallel channel that is connected to a separate receiver which is at the same end as the intended receiver. It occurs along the length of the channel and has therefore been attenuated. It appears as noise to the second receiver, and is not severe [5].

The second and much more pronounced crosstalk type effect is called near end crosstalk or NEXT. This occurs when a symbol is coupled onto the channel whose receiver is connected at the same end as the offending transmitter. Since the transmitter and receiver are in close proximity, the amplitude of the undesired symbol can be greater than the desired symbol. Therefore this effect is much more serious than FEXT, since incoming information could be completely eliminated [5]. Fortunately there are special types of circuitry which can counteract this type of coupling, similar to those discussed later in this paper.

![Crosstalk Diagram](image)

**Fig. 3.** An overhead view of a 30” FR-4 Backplane channel, daughter cards and ICs. Both FEXT and NEXT are shown by the dashed lines. Arrowheads indicate the direction of data flow and coupling along the copper traces.

### 3. Equalization types

In order to cope with the increase in speed and the associated ISI, special circuitry and systems were developed in order to recover the desired information. These circuits and techniques are called channel equalization, and can take several different structures and implementations. The ideal form would have a filter with a response which is the direct inverse of the channel response, and would therefore perfectly cancel the ISI. Unfortunately, this type of filter would be infinitely long and impossible to construct [1].

In reality, a designer must choose an appropriate architecture or architectures which will provide a
tradeoff between the amount of equalization and the complexity of the circuitry. Furthermore, the conditions of the channel such as temperature and power supply levels will change over time. Therefore the structures that will equalize the channel should not be fixed, but rather change with the variable conditions. This has led to the use of adaptive filters which allow a common architecture to be used on all of the backplane channels with each circuit adapting to the local conditions. The remainder of this paper illustrates the various choices that are available with emphasis on the structures and algorithms that are commonly used at the receive end of the channel.

3.1. Pre-Empahsis

Pre-Emphasis involves using an equalizer on the transmitter end of the channel. This equalizer uses the current and future information bits to emphasize the rising edges of the transmitted bits. The emphasis serves to enhance the high-frequency content of the outgoing symbols which will counteract the low-pass filtering of the channel. The net overall result will reduce the ISI seen at the receive end of the channel. Unfortunately, there are two severe limitations to this technique. The first limitation is the maximum allowable amplitude available from the pre-emphasis circuit. Without the required high power levels, this equalization technique will still be inadequate. The second and much more important limitation is a result of the increase in the high-frequency content and power. These factors enhance the NEXT coupling of the transmitted symbols onto the adjacent channels, and therefore degrade the performance of the system [6].

4. Receive Equalizer Structure

Another method to remove the ISI caused by the channel is to apply the equalization circuitry at the receiver. This will allow the designer to avoid enhancing the noise enhancement while still recovering the data. Unfortunately the techniques used to accomplish this are not as simple as those used in pre-emphasis [6].

There are two main structures for the receive-end equalizer. The first is the feed forward equalizer structure, which closely resembles a finite impulse response filter. And the second is the decision feedback equalizer, whose structure closely resembles an infinite impulse response type filter. The following section will discuss the two main structures and how they are implemented.

4.1. Feed Forward Equalizer

The first structure is called the feed forward equalizer or FFE. This structure pictured in Fig.4 below is also called a linear transversal filter, or a tapped-delay line filter [7].

![Fig. 4. The Feed Forward Equalizer. As data symbols enter the filter on the left, they pass through a delay-line where each successive symbol is multiplied by a Tap value then summed up to give the output symbol.](image)

The filter works by summing weighted versions of a finite number of past symbols. This makes it ideal for canceling the interference from future symbols on the present symbol of interest; also known as pre-cursor ISI. If the desired symbol is at position N of the delay line, a certain number of symbols in positions N-1 and N-2 could be subtracted via the summation node. Equation (1) describes the mathematical operation of the linear transversal type filter. Unfortunately this structure does have a significant limitation in the fact that it utilizes symbols as they are received into the filter. Therefore any noise added by the channel will be enhanced and applied to other symbols [1], [2].

\[
Y(t) = \sum_{n=0}^{N} Tap(n)U(t - nT)
\]  

4.2. Decision Feed Back Equalizer

The second architecture commonly utilized at the receiver is the decision feedback equalizer or DFE. Like the linear transversal equalizer, it contains a tapped delay line which sums weighted version of past symbols. However the structure is termed a nonlinear transversal filter due to the fact that a decision is made on the symbol before it is fed back into the loop. It is this decision that allows the filter to cancel out post-
cursor ISI [7] without enhancing the noise. In order to properly utilize the fed-back data, the tap weights must be chosen correctly to match the distortion of the previous symbols. Furthermore, this structure assumes that all of the past decisions are correct. This assumption can lead to error propagation, since each mistake which is fed back through the filter can create a greater chance of another error [1].

4.3. Implementation

There are many different techniques used to implement these types of circuits. Traditionally large DSP cores were used to implement only the DFE, however the increased speeds make this infeasible [2]. Therefore the FFE and DFE are placed together on a single custom designed IC which implements both a high-speed analog filter circuitry along with the adaptation engine [9], [13].

![Diagram of Infinite Impulse Response Type Structure Used for Decision Feedback Equalizer](image)

**Fig. 5.** The infinite impulse response type structure used for the Decision Feed Back Equalizer. As data symbols enter the filter on the right, they pass through the summation node where weighted versions of past decisions are added to the current symbol.

5. Filter Adaptation

The most important factor in the above mentioned equalization structures, is the tap coefficient values and how they are selected. The following section will explore the two common methods and issues of selecting the tap coefficient values.

5.1. Fixed versus Adaptive Coefficients

One method used [9] is to analyze the performance of the channel and to program the coefficient values to an optimal value. This involves complicated and expensive test equipment and must be performed for each individual implementation [10]. Although this method is effective in reducing ISI over the channel, it does not account for the changing conditions of the environment such as temperature and power supply levels [8]. Another much more effective method is to make the tap coefficients continually adaptive. This not only allows the filter to adjust its response to accommodate each individual channel but also to the variable conditions.

5.2. Adaptation Algorithms

There are several different algorithms that are available to the designer of an adaptive equalizer. Each one has different convergence properties and different levels of complexity. The designer must decide between the associated trade offs between the performance and the cost of the design. Some common types used are the least-mean-squared algorithm or LMS and the recursive-least-squared algorithm or RLS [7]. Due to its relative simplicity of implementation, the LMS algorithm and its variants are the most commonly used in receive end equalizers [8], [11]. Equations 2 through 5 below, illustrate the operation of the LMS type algorithms typically implemented for this type of application [11].

\[
C(n + 1) = C(n) + 2\mu e(n)d(n) \quad (2)
\]

\[
C(n + 1) = C(n) + 2\mu \text{sgn}[e(n)]d(n) \quad (3)
\]

\[
C(n + 1) = C(n) + 2\mu e(n)\text{sgn}[d(n)] \quad (4)
\]

\[
C(n + 1) = C(n) + 2\mu \text{sgn}[e(n)]\text{sgn}[d(n)] \quad (5)
\]

During each iteration of the adaptation process, each tap coefficient \( C(n) \) is updated by applying a correction to the old value. In the full algorithm (2) this correction is based on the product of the received data \( d(n) \), \( e(n) \) which represents the mean-squared-error (MSE) between the received and transmitted data, and the convergence factor \( \mu \) [7]. This factor is must be selected carefully as it dictates the convergence properties of the filter. Variations of this algorithm include; Sign-Error (eq.3), Sign-Data (eq.4), and Sign-Sign (eq.5), which utilize the sign (sgn) function to make the algorithm more hardware efficient while retaining most of the convergence properties of the full version [11]. Another technique used to reduce the hardware constraints and to save power, is to run the algorithm at a slower rate than the incoming data by averaging the updates over several symbols [10], [12].
5.3. Coefficient Training

In order to converge the optimum values, the adaptive structure typically utilizes a training sequence [1]. This sequence is transmitted across the channel and is also stored locally at the receiver. With this local copy, the equalizer is able to accurately calculate the MSE and converge the filter coefficients within a reasonable amount of time (Convergence times of 25µs have been reported. [10]). Unfortunately training requires the transmitter and receiver to utilize the exact same bit sequence and occupies the channel preventing data from being transmitted. To eliminate this issue, the adaptive DFE structure can be operated in what is called decision-directed mode where the convergence algorithm assumes the decisions made on the received symbols to be correct, and are typically used in (2) or (5). This method however can make the convergence time much longer if not properly designed [6]. The figure below (Fig. 6) illustrates the convergence of a 4-Tap decision-directed DFE, utilized on a 5Gbps PAM-2 signal over a 30" FR-4 backplane. It can clearly be seen that it takes over 4µs to converge the tap values.

6. Conclusion

This paper has reviewed some of the issues and structures used to implement adaptive equalizers at the receive end of a backplane channel. Without the use of these adaptive equalizers, high speed communications would not be possible over serial backplane channels. The authors would like to acknowledge the generous financial support from the Government of Ontario, NSERC and Altera Corporation.

7. References


