

This application note describes the decision feedback equalization (DFE) feature found in the Stratix<sup>®</sup> IV device equalizer. Use the DFE feature to improve the high frequency signal-to-noise ratio by compensating for inter-symbol interference (ISI). The DFE feature boosts the high frequency components of a signal without noise amplification.

-  Use the associated reference design at [AN612\\_RefDesign.zip](#) to evaluate and implement the decision feedback equalization feature. The Stratix IV DFE Reference Design User Guide, in the zip folder, describes the step-by-step procedure to use the reference design.

Signals transmitted across a backplane or transmission medium are subjected to filtering and attenuation effects. These effects are a potential source of error for the receiver, primarily in the form of reduced signal-to-noise ratio and ISI. To compensate, Stratix IV transceivers have a programmable equalizer block that you can use to negate or reduce these error sources, ensuring that the receiver can function without error.

The dynamic reconfiguration controller (ALTGX\_RECONFIG) provides an Avalon<sup>®</sup> memory mapped user interface to step through the DFE tap settings. You are responsible for determining the optimal settings by monitoring the BER of the received data at each setting. For any given data rate and data pattern, there may be multiple DFE settings which yield a BER of 0. You can use the EyeQ feature to find the width of the eye for all DFE settings with a BER equal to 0. Then choose the DFE setting which yields the widest eye.

## Implementing the DFE Feature

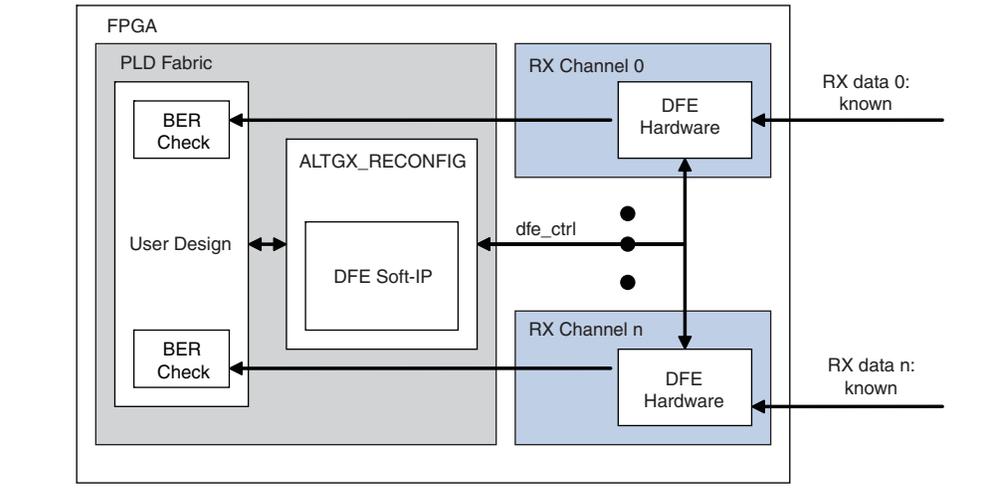
The DFE feature includes three partitions:

- DFE hardware—this is the hardened IP that provides the DFE functionality. An instance of the DFE hardware exists on each receive channel, not including the clock multiplier unit (CMU) channels.
- DFE soft IP—this block is part of the ALTGX\_RECONFIG megafunction which acts as a bridge between you and the DFE hardware. The IP provides an Avalon memory mapped interface that allows you to access a set of indirect registers. These registers allow you to access the DFE hardware (refer to “[Enabling the DFE Soft IP](#)” on page 2).
- DFE user design—this is the soft IP that provides an interface to the user design to control the DFE hardware. For minimal functionality, a user design would consist of a state machine to the control DFE settings and some method of generating and checking an input pattern for the BER.

 Full details of the DFE user design is beyond the scope of this application note.

Figure 1 shows a high-level diagram of a full DFE implementation in a Stratix IV device.

**Figure 1. Using the DFE Feature in a Stratix IV Device**

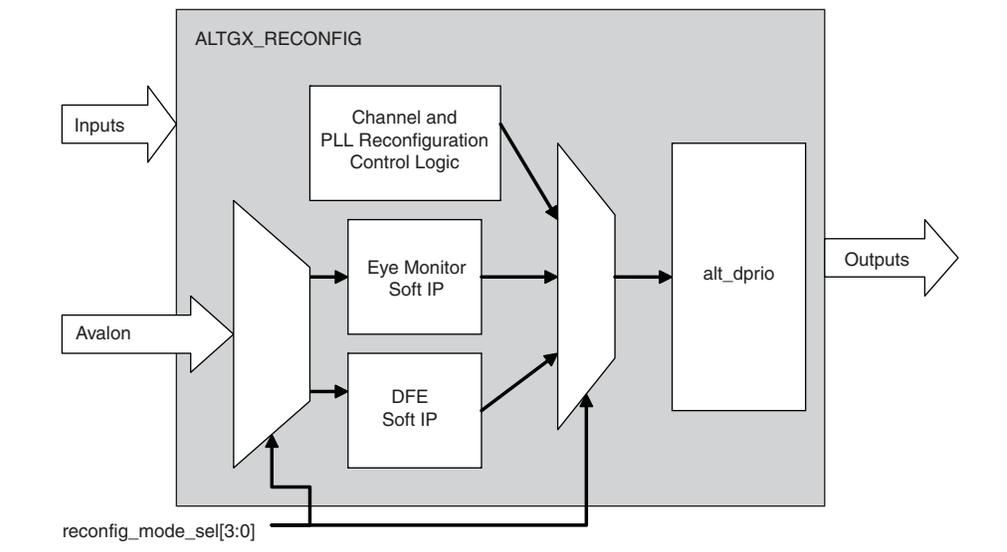


## Enabling the DFE Soft IP

The DFE soft IP is implemented inside the ALTGX\_RECONFIG megafunction.

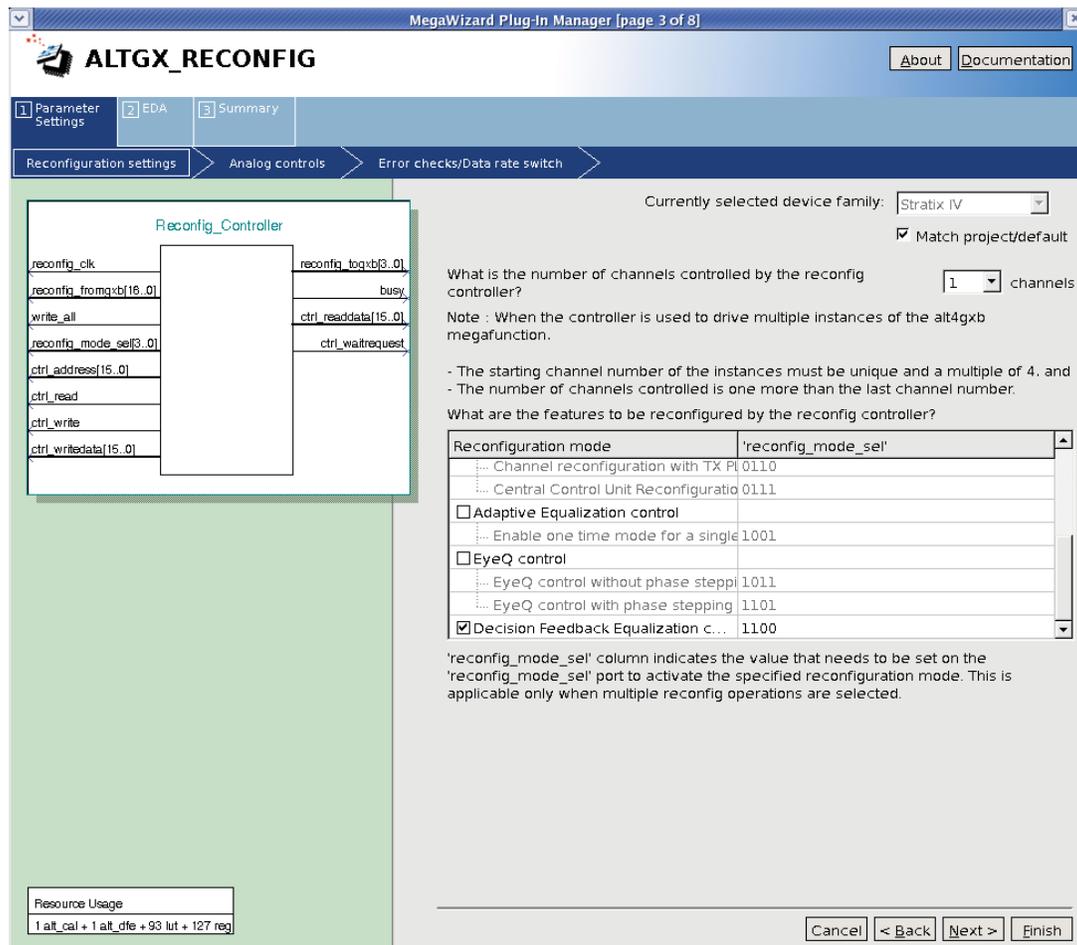
Figure 2 shows a block diagram of the ALTGX\_RECONFIG hierarchy.

**Figure 2. DFE Soft IP Inside the ALTGX\_RECONFIG Megafunction**



In order to use the DFE soft IP, you must use the MegaWizard™ Plug-In Manager to generate the ALTGX\_RECONFIG megafunction. In the MegaWizard Plug-In Manager, select the **Decision Feedback Equalization control** option (Figure 3). This enables the Avalon interface and the reconfig\_mode\_sel [3:0] port. When accessing the DFE feature, set the reconfig\_mode\_sel bits to 4'b1100.

Figure 3. ALTGX\_RECONFIG MegaWizard Plug-In Manager



The user design accesses the DFE soft IP within the ALT\_RECONFIG block by the control signals provided.

Table 1 lists the input port control signals.

Table 1. Input Port Control Signals (Part 1 of 2)

Input Port	Description
ctrl_writedata[15:0]	Write data bus as part of the Avalon memory map interface for the DFE feature or eye monitor IP.
ctrl_read	Start of a read transaction as part of the Avalon memory map interface for the DFE feature or eye monitor IP.
ctrl_write	Start of a write transaction as part of the Avalon memory map interface for the DFE feature or eye monitor IP.

**Table 1. Input Port Control Signals (Part 2 of 2)**

Input Port	Description
ctrl_address[15:0]	Address port as part of the Avalon memory map interface for the DFE feature or eye monitor IP.
Reconfig_mode_sel[3:0]	Use to select the reconfiguration mode for the ALTGX_RECONFIG megafunction. The mode and the functionality are as follows: <ul style="list-style-type: none"> <li>1011 = Reconfiguration for eye monitor</li> <li>1100 = Reconfiguration for the DFE feature</li> </ul>

Table 2 lists the output port control signals.

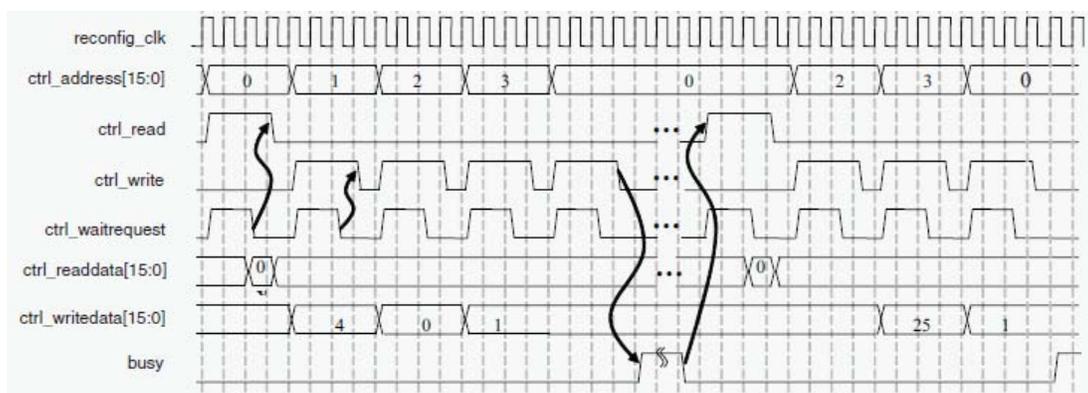
**Table 2. Output Port Control Signals**

Output Port	Description
ctrl_readdata[15:0]	Read data bus as part of the Avalon memory map interface for the DFE feature or eye monitor IP.
ctrl_waitrequest	Backpressure signal as part of the Avalon memory map interface for the DFE feature or eye monitor IP. When this signal is asserted, the values on the read, write, readdata, and writedata buses must remain constant.
Busy	Indicates that the function is completing a control operation. All asserted control signals are ignored when busy is high.

## Describing the DFE Soft IP Interface

The user logic accesses the DFE soft IP by setting the reconfig\_mode\_sel[3:0] control bits to 4'b1100. The ctrl\_\* and busy signals are used to access the registers within the DFE soft IP using the Avalon memory mapped interface protocol. Figure 4 shows a read and write transaction with the ctrl\_\* and busy signal.

 For more information about the Avalon memory mapped interface, refer to the [Avalon Interface Specification](#).

**Figure 4. Avalon Memory Mapped Read and Write Transaction**

## DFE Register Memory Map

The Avalon memory map interface allows you access to a set of 16-bit wide registers to access the DFE hardware bits. The DFE soft IP translates the Avalon memory map requests into `alt_dprio` requests. [Table 3](#) lists the memory map of the DFE registers.

 Bits not listed in [Table 3](#) are reserved and considered read-only.

**Table 3. DFE Register Map on the Avalon Memory Map Interface**

Avalon Memory Map Address	Description
0x0	Control and status register. <ul style="list-style-type: none"> <li>Bit 0: “start” command issues the programmed command to <code>alt_dprio</code>. Writing this bit automatically clears any error bits.</li> <li>Bit 1: “read/~write” selects whether the command is to be a read or a write operation.</li> <li>Bit 13: error status. This bit is set to <b>1'b1</b> if the programmed channel address is invalid. Writing a 1'b1 to this bit clears the error.</li> <li>Bit 14: error status. This bit is set to <b>1'b1</b> if the programmed word address is invalid. Writing a 1'b1 to this bit clears the error.</li> <li>Bit 15: busy status. This value can be polled to determine if the <code>dprio</code> request has completed. <b>Note that when this active-high bit is asserted, all registers become read only until this bit is de-asserted.</b></li> </ul>
0x1	Channel address [15:0]. This specifies the channel to be written to. This must match the <code>logical_channel_address</code> input port.
0x2	DFE settings register address [15:0]. This specifies the DFE control register to be read or written. For more information about this memory map, refer to “ <a href="#">DFE Settings Register Mapping</a> ”.
0x3	Data [15:0]. For writes, you must write this data. For reads, the data contained in the register is only valid when the busy status is low. An indirect register read operation overwrites the current contents of this register.

## DFE Settings Register Mapping

[Table 4](#) lists the DFE settings register map. You can indirectly access these registers using the Avalon memory map interface by accessing the DFE register 0x2.

 Bits not listed in [Table 4](#) are reserved and considered read only. These bits must be written to 0 on a write.

**Table 4. DFE Setting Registers**

DFE Setting Register Address	Description
0x0	<ul style="list-style-type: none"> <li>Bit [0]: DFE second tap polarity (1'b0 = positive; 1'b1 = negative)</li> <li>Bit [1]: DFE third tap polarity</li> </ul>
0x1	<ul style="list-style-type: none"> <li>Bit [0]: DFE Enable</li> <li>Bit [3:1]: DFE third tap setting</li> </ul>
0x2	<ul style="list-style-type: none"> <li>Bit [2:0]: DFE first tap setting</li> <li>Bit [5:3]: DFE second tap setting</li> </ul>

## Accessing the DFE Setting Register Bits

To access the DFE settings register bits through indirect register mapping, complete the following:

1. Issue an Avalon memory map read to the DFE register 0x0 (control and status register) to check the busy status. A clear status bit indicates an idle status.
2. Issue an Avalon memory map write to the DFE register 0x1 (channel address register) to select the desired channel.
3. Issue an Avalon memory map write to the DFE register 0x2 (DFE settings register address) to select the desired DFE settings register.
4. Issue an Avalon memory map write to the DFE register 0x3 (data register) to provide the data to be written to the target DFE settings register.
5. Issue an Avalon memory map write to the DFE register 0x0 (control and status register) to specify read/~write and to issue the start command.
6. Poll the DFE register 0x0 (control and status register) and wait for the busy status to be de-asserted. After the status is no longer busy:
  - For writes, the data is considered to be successfully written.
  - For reads, the contents of the data register have been updated and can be read out.



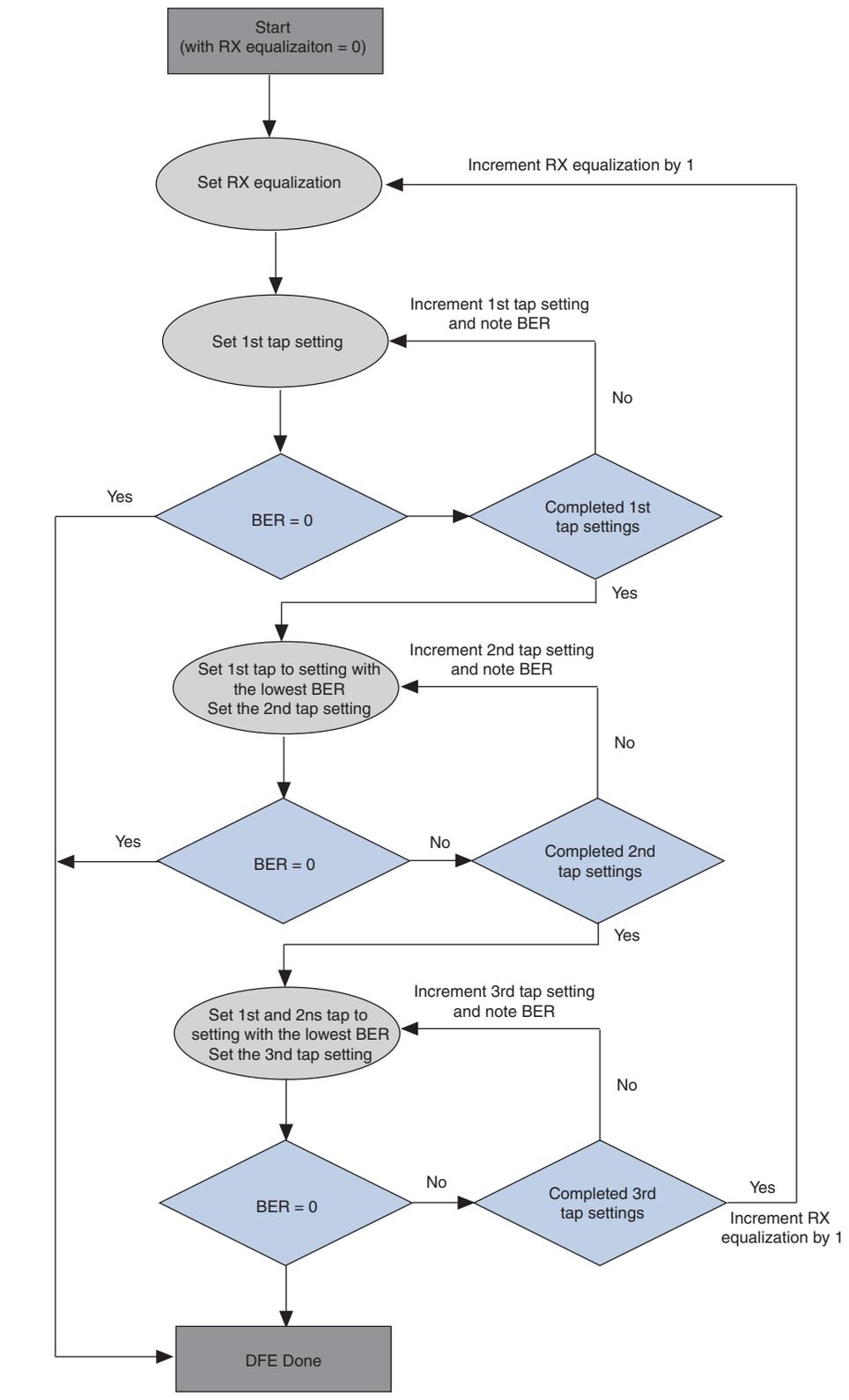
All writes which occur when the busy status is asserted are ignored; all registers become read only.

If the next operation is to the same register and same channel, you do not need to repeat steps 2 and 3.

## Choosing the DFE Settings

When you are using the DFE feature, the minimum peak-to-peak voltage envelope value at the receiver (after the back plane) must be greater than 500 mV. The pattern you use for calibration must be similar to the actual data expected on the link. You can find the optimal DFE results by trying out the DFE settings according to the DFE setting search algorithm shown in [Figure 5](#). Altera recommends engaging the RX equalization along with the DFE feature. You must use the minimum possible RX equalization setting.

Figure 5. Flow Diagram for Choosing the DFE Setting



## DFE Tap Settings Time Estimates

If you automate the DFE tap settings search, the time estimated to complete one iteration of the DFE tap settings is shown below. The worst case is assumed where all three tap settings have to be traversed. The data rate used in this calculation is 6.5 Gbps. The BER is checked for  $3 \times 10^{12}$  bits to achieve a BER of  $1 \times 10^{-12}$  with 95% confidence level.

- Time to program one DFE tap setting = 32  $\mu$ s
- Time to program all DFE tap settings:
  - 1st tap =  $32 \times 8 = 256 \mu$ s
  - 2nd tap =  $32 \times 16 = 512 \mu$ s (eight settings with polarity inversion = 16)
  - 3rd tap =  $32 \times 16 = 512 \mu$ s (eight settings with polarity inversion = 16)
- Time to check BER for one tap setting =  $153 \text{ ps} \times 3 \times 10^{12} = 7.7$  minutes
- Time to check BER for 8 + 16 + 16 tap settings = 5.1 hours
- Time to go through one iteration of DFE settings and check BER is approximately 5.1 hours.

This cycle is repeated for each of the RX equalization settings. In the worst case, where all 16 of the RX equalization settings must be tried, it will take approximately 82 hours.

## Implementing DFE Block Access

In order to access the DFE block in a Stratix IV GX device, complete the following:

1. Enable the **Analog controls** option under the **Reconfiguration Settings** tab of the altgxb instance file. DFE soft IP requires Analog PMA reconfiguration to be enabled in ALTGXB.
2. Generate an alt\_reconfig instance with the **EyeQ** option selected.
3. In the ALT\_RECONFIG module, enable the **Decision Feedback Equalization control** option to add DFE soft IP.
4. Design a BERT in the FPGA core to test the DFE feature.
5. Design the user IP to access the DFE registers.

## Appendix A

The following sections further describe the DFE feature.

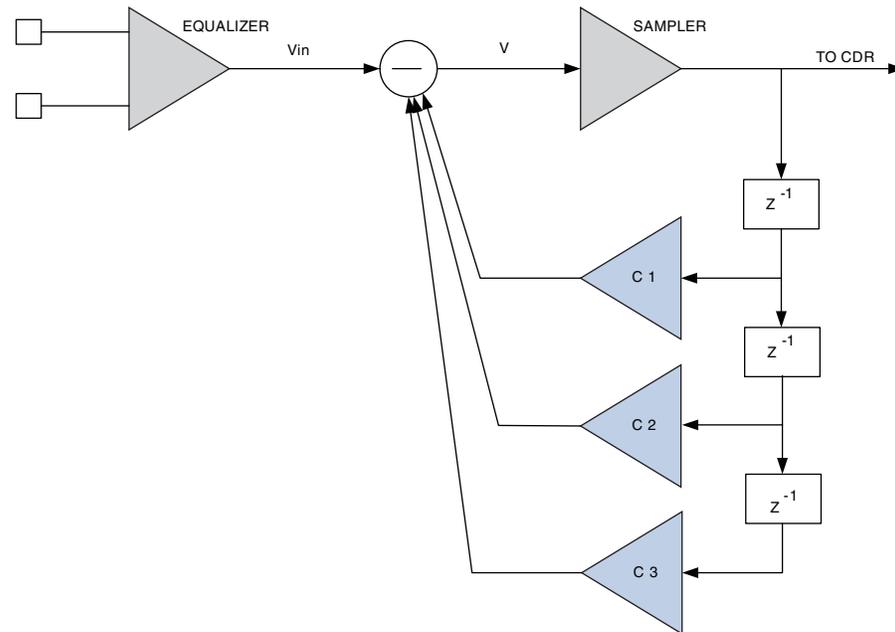
### DFE Principles

The purpose of the linear equalizer is to compensate for the backplane attenuation due to insufficient bandwidth. The DFE feature works by actively shifting the incoming signal based on the history of the received data.

The purpose of the DFE feature is to cancel out the post-cursor caused by ISI. The advantage of the DFE feature is to boost the power of the highest frequency component of the receive data without increasing noise power. You can use the DFE feature in conjunction with the receiver's linear equalization.

Figure 6 shows the theoretical DFE scheme.

**Figure 6. Theoretical DFE Scheme**



The following describes the information shown in Figure 6.

- Each Z-1 unit represents 1 UI of delay.
- The signal presented to the sampler at time point T0 is:  

$$V(T_0) = Vin(T_0) - C_1 \times D(T-1) - C_2 \times D(T-2) - C_3 \times D(T-3)$$

Where “Vin(T0)” is the output of the equalizer at time point T0 and “D(Tn)” is the quantified signal (data) at time point Tn.

The component “C1 × D(T-1)” compensates for the post-cursor of the signal at time point T-1 leaking to the signal at time point T0; the component “C2 × D(T-2)” compensates for the post-cursor of signal at time point T-2; and the component “C3 × D(T-3)” compensates for the post-cursor of signal at time point T-3.

Theoretically, you would need an infinite number of taps to compensate for all the post-cursor because post-cursor amplitude approaches zero only when n goes to infinity. However, due to the exponential decay nature of post cursor, the first three components contribute 95% of the effect. In this design, 3-tap architecture was chosen. C1, C2, and C3 are the three tap settings.

## DFE Gain for the Tap Settings

Each tap has seven settings that are independently controlled. The gain values are accurate for a voltage envelope greater than 500 mV. The inverse polarity for the second and third tap generates a negative gain.

## Document Revision History

Table 5 lists the revision history for this application note.

**Table 5. Document Revision History**

<b>Date</b>	<b>Version</b>	<b>Changes</b>
September 2015	1.2	Clarified steps 1 and 3 in the “Implementing DFE Block Access” section.
January 2011	1.1	Added a link to the reference design on page 1.
December 2010	1.0	Initial release.