Sign-based Zero-Forcing Adaptive Equalizer Control for High-Speed I/O

Yasuo Hidaka
Fujitsu Laboratories of America, Inc.
November 18, 2013

IEEE Circuits and Systems Society
Santa Clara Valley Chapter
Outline

- Background
- State-of-the-art Equalizer for High-Speed I/O
- Conventional Adaptive Equalizer Control for High-Speed I/O
- Sign-based Zero-Forcing Adaptive Equalizer Control
- Implementation and Evaluation Results
- Summary
Outline

- Background
  - Applications of High-Speed I/O
  - Frequency-Dependent Channel Loss
  - Inter-Symbol Interference (ISI)
- State-of-the-art Equalizer for High-Speed I/O
- Conventional Adaptive Equalizer Control for High-Speed I/O
- Sign-based Zero-Forcing Adaptive Equalizer Control
- Implementation and Evaluation Results
- Summary
Applications of High-Speed I/O

- **Application Channels**
  - On-board: Chip to chip with no connector
  - Module I/F: Chip to module via 1 connector
  - Backplane: Board to board via 2 connectors
  - Cable: Rack to rack via 2 connectors

- **Examples**
  - Standard High-Speed Interfaces
    - OIF CEI, IEEE 802.3 Ethernet, PCI Express, etc
  - Proprietary High-Speed Interfaces
    - CPU I/F, Bridge chip I/F, Switch chip I/F, etc

- **State-of-the-art Performance of HSIO**
  - Data rate: 25~32 Gbps per lane
  - Channel loss: 35~40 dB at Nyquist frequency
  - Channel length: 0.3~1m PCB, 3~7m cable
  - HSIO density: 20~100+ lanes per chip
Frequency-Dependent Channel Loss\cite{1,2}

- **Dielectric Loss**
  
  Rotation of dipoles in dielectric loses energy per every Hertz.

  \[ \text{loss (dB)} \propto \text{freq} \]

- **Skin Effect**
  
  AC current flows only in metal surface with increased resistance.

  \[ \text{loss (dB)} \propto \sqrt{\text{freq}} \]
Channel Loss Example (Linear Frequency Axis)

\[ \text{loss (dB)} \propto k_1 \sqrt{f} + k_2 f \]

- **Skin effect**
  - Curved with concave up
  - Primary cause at low freq

- **Dielectric loss**
  - Straight line
  - Primary cause at high freq

*We often overlook or neglect low-frequency loss*

- Loss is small
- Degenerated at DC and hardly recognized
Channel Loss Example (Log Frequency Axis)

\[ \text{loss (dB)} \propto k_1 \sqrt{f} + k_2 f \]

- **Always exponential roll-off**
  - Regardless of skin effect or dielectric loss

- **Low-frequency loss**
  - Start as low as 10MHz
    - Skin depth 20um@10MHz
    - PCB trace thickness 35um
  - Gentle slope
    - < 3dB/dec
## Inter-Symbol Interference (ISI)

<table>
<thead>
<tr>
<th>Tx</th>
<th>Channel</th>
<th>Rx</th>
</tr>
</thead>
<tbody>
<tr>
<td><img src="image" alt="1010" /></td>
<td><img src="image" alt="1V ~0.01V" /></td>
<td><img src="image" alt="~0.01V" /></td>
</tr>
<tr>
<td><img src="image" alt="11001100" /></td>
<td><img src="image" alt="1V ~0.08V" /></td>
<td><img src="image" alt="~0.08V" /></td>
</tr>
<tr>
<td><img src="image" alt="11110000" /></td>
<td><img src="image" alt="1V ~0.25V" /></td>
<td><img src="image" alt="~0.25V" /></td>
</tr>
<tr>
<td><img src="image" alt="1-bit lone pulse" /></td>
<td><img src="image" alt="1UI ~0.12V" /></td>
<td><img src="image" alt="~0.12V" /></td>
</tr>
</tbody>
</table>

**UI**: Unit Interval

**fb**: baud-rate frequency

**Loss**: 40.2dB @ fb/2, 23.6dB @ fb/4
Outline

- Background
- State-of-the-art Equalizer for High-Speed I/O
  - Conventional High-Frequency Equalizers
    - Overview
    - CTLE (Continuous-Time Linear Equalizer)
    - FFE (Feed-Forward Equalizer)
    - DFE (Decision-Feedback Equalizer)
    - Speculative DFE
  - Low-Frequency Equalizer
- Conventional Adaptive Equalizer Control for High-Speed I/O
- Sign-based Zero-Forcing Adaptive Equalizer Control
- Implementation and Evaluation Results
- Summary
State-of-the-art Equalizer for High-Speed I/O

FFE: Feed-Forward Equalizer
CTLE: Continuous-Time Linear Equalizer
LFE: Low-Frequency Equalizer
DFE: Decision-Feedback Equalizer
**CTLE (Continuous-Time Linear Equalizer)**[3-7]

- Continuous-time high-pass filter
- Example
  - 1st order CTLE

![CTLE Diagram]

**Transfer Function in s domain:**

\[
G(s) = A_{DC} \frac{1 - \frac{s}{2\pi f_z}}{(1 - \frac{s}{2\pi f_{p1}})(1 - \frac{s}{2\pi f_{p2}})}
\]

- 1 zero
- 2 poles

![Frequency Response]

\[|f_{p1}| \text{ is } \frac{1}{4} \sim \frac{1}{2} \text{ of baud-rate frequency.}\]
FFE (Feed-Forward Equalizer)[8-10]

- Discrete-time high-pass filter
- Example
  - 3-tap FFE

Transfer Function in z domain:

\[ G(z) = C_{-1} + C_0 z^{-1} + C_1 z^{-2} = C_{-1} \frac{(z - z_1)(z - z_2)}{z^2} \]

\[ z = \exp(s/f_b) \]
\[ z_1 = \exp(2\pi f_{z1}/f_b) \]
\[ z_2 = \exp(2\pi f_{z2}/f_b) \]

<table>
<thead>
<tr>
<th></th>
<th>(dB)</th>
</tr>
</thead>
<tbody>
<tr>
<td>A_{DC}</td>
<td>20dB/dec</td>
</tr>
</tbody>
</table>

\[ |G| |f_{z1}| |f_{z2}|^\dagger f_b/2 f_{(log)} \]

\[ ^\dagger : \text{An FIR filter cannot have a pole. } f_b/2 \text{ is not a pole, but the max effective frequency.} \]
\[ ^\dagger : f_{z1} (<0) \text{ on left half of s plane boosts gain. } f_{z2} (>0) \text{ on right half of s plane adjusts phase.} \]
DFE (Decision-Feedback Equalizer)\cite{11,12}

- Emulate ISI from previous decisions, and subtract it from input
- Example
  - N-tap DFE

\begin{align*}
\text{Input} & \quad \sum \quad \text{Output} \\
\text{Emulated ISI} & \quad 1\text{UI per tap} \\
\text{ISI} & \quad \text{No ISI} \\
\text{N-tap FIR Filter} & \quad C_1, C_N
\end{align*}

ISI: Inter-Symbol Interference
Speculative DFE\textsuperscript{[13,14]}

- Defer DFE feedback loop using speculative decisions

- Trade-off area/power for timing critical path
Outline

- Background
- State-of-the-art Equalizer for High-Speed I/O
  - Conventional High-Frequency Equalizers
  - Low-Frequency Equalizer
    - Performance Limit of Conventional High-Frequency Equalizers
    - Architecture of Low-Frequency Equalizer
    - Effect of Low-Frequency Equalizer
- Conventional Adaptive Equalizer Control for High-Speed I/O
- Sign-based Zero-Forcing Adaptive Equalizer Control
- Implementation and Evaluation Results
- Summary
Performance Limit of Conventional Equalizers

- EQ parameters are the best values
- Why is this eye so bad?

32Gbps, 4-tap FFE, 1st-order CTLE, 1-tap DFE

- Frequency-domain response
- Time-domain pulse response
- Eye diagram (PRBS31)
Uncompensated Low-Frequency Loss

- Conventional EQs cannot compensate for low-frequency loss
  - CTLE and FFE have too steep slopes (20dB/dec)
    - If zero is moved to lower frequency, too much gain at high frequency
  - DFE can compensate for only short-term ISI (i.e. high-frequency loss)

32Gbps, 4-tap FFE, 1st-order CTLE, 1-tap DFE
Low-Frequency Equalizer (LFE)\cite{15,16}

- Amplify low-frequency by a small amount (compared to DC)
  - Closely spaced a pair of pole and zero in low frequency
  - A variant of CTLE for low frequency
  - Easy to implement in analog, but maybe difficult in digital

**Example**

- Feedback with LPF

Transfer Function in s domain:

\[ G(s) = A_{DC} \frac{1 - \frac{s}{2\pi f_z}}{1 - \frac{s}{2\pi f_p}} \]

<table>
<thead>
<tr>
<th>( s = j\omega = j2\pi f )</th>
</tr>
</thead>
<tbody>
<tr>
<td>1 zero</td>
</tr>
<tr>
<td>1 pole</td>
</tr>
</tbody>
</table>

\[ f_p = (1 + gh)f_z \]

\[ A_{HF} = g \]

\[ A_{DC} = \frac{g}{1 + gh} \]

\[ A_{HF} \]

\[ A_{DC} = 1 + gh \]
Effect of LFE in Frequency Domain

**Only Conventional**

![Graph showing gain vs. frequency for conventional method](image1)

**With LFE**

![Graph showing gain vs. frequency for LFE method](image2)

- **FFE+CTLE+LFE**
- Channel loss 42.5dB@16GHz
- 4-tap FFE, 1st-order CTLE, 1-tap DFE

### Parameters:

- Zero: 0.212
- Pole: 0.311
- Gain: 1.44

**Linear**

![Graph showing gain vs. frequency in linear scale](image3)

**(Linear)**

32Gbps, Chan. loss 42.5dB@16GHz
4-tap FFE, 1st-order CTLE, 1-tap DFE
Effect of LFE in Time Domain

Only Conventional

 Integral of residual ISI magnitude w/o LFE

Channel

Channel+FFE+CTLE+DFE

With LFE

 Integral of residual ISI magnitude w/ LFE

Channel+FFE+CTLE+DFE+LFE

32Gbps, Chan. loss 42.5dB@16GHz
4-tap FFE, 1st-order CTLE, 1-tap DFE
Effect of LFE on Eye Diagram (PRBS31)

Only Conventional

With LFE

DDJ=0.42UI

DDJ=0.21UI

32Gbps, Chan. loss 42.5dB@16GHz
4-tap FFE, 1st-order CTLE, 1-tap DFE
Outline

- Background
- State-of-the-art Equalizer for High-Speed I/O
- Conventional Adaptive Equalizer Control for High-Speed I/O
  - Least Mean Square (LMS and SS-LMS)
  - Zero Forcing (ZF)
  - Max Eye Opening
  - Spectrum Matching
- Sign-based Zero-Forcing Adaptive Equalizer Control
- Implementation and Evaluation Results
- Summary
Least Mean Square† (LMS\textsuperscript{[17]}, SS-LMS\textsuperscript{[18]})

- Automatically achieve *Minimum Mean-Square Error* (Wiener Filter)
- Widely used in Digital Signal Processing (DSP)
- Strong restriction on filter structure (filter must be in below form)
  - Applicable to DFE, but often NOT to CTLE or LFE
  - NOT applicable to Tx FFE

\[ e_k \]

\[ \int \int \int \cdots \]

\[ \text{Multi-tap filter banks} \]

\[ \text{Feedback for DFE} \]

\[ \text{Linear Combiner} \]

\[ \text{Decision} \]

\[ \text{Recovered data} \]

\[ \sum \]

\[ \text{Channel} \]

\[ C_1, C_2, \ldots, C_n \]

† A.k.a. Stochastic Steepest Descent Method
Zero Forcing (ZF)\(^{19,20}\)

- Force *weighted* sum of ISI towards zero
  - Also capable to achieve almost *Minimum Mean-Square Error* (Wiener Filter)
    - For the target (worst) channel, with proper weight vectors (e.g. Jacobian Matrix)
    - Not optimal for other channels, but it is usually acceptable for wide range of variation
    - Equivalent to LMS, if applied to DFE

- Flexible filter structure
  - Applicable to CTLE, LFE and Tx FFE

![Diagram of Zero Forcing (ZF)](image)

- Multi-tap filter banks
- Linear Combiner
- Decision
- Recovered data
- Error
- Shift Register
- This structure (order) may be transformed
- These nodes are not observed, and do not have to exist
Max Eye Opening\textsuperscript{[21]}

- Maximize eye opening by adjusting EQ parameters
- Flexible filter structure
- Slow or inaccurate
  - Similar eyes for slightly different parameters must be compared
  - Precision eye measurement takes quite a long time

![Diagram of multi-tap filter banks and equalizer](image-url)
Spectrum Matching\(^{[3,6]}\)

- Force imbalance of spectrum towards zero
- Options to measure imbalance of spectrum
  - Multiple filters with different bands (e.g. LPF and HPF)
  - Slicer may be optionally used to generate the reference
  - Edge slew rate or pulse width may be used instead of spectrum
- Difficult to control more than one parameter
- Only for CTLE

\[\text{CTLE Recovered data} \leq C_1 \text{ LPF HPF Rectifier} \]
Outline

- Background
- State-of-the-art Equalizer for High-Speed I/O
- Conventional Adaptive Equalizer Control for High-Speed I/O
- Sign-based Zero-Forcing Adaptive Equalizer Control
  - Problems of Conventional Adaptive Equalizer Control
  - Pattern Filtering
  - Convolution and De-convolution of ISI
  - Sign-based Zero-Forcing
  - Extension of S-ZF for Low-Frequency Equalizer
- Implementation and Evaluation Results
- Summary
Problems of Conventional Adaptive EQ Control

- Need random data
  - If data is not random, equalizer parameter can drift
  - Some wireline standards use 8B10B code
    - With 8B10B, data sequence can be periodic such as continuous 1010
    - Idle sequence in 8B10B has limited randomness

- Limited flexibility or accuracy (except ZF)
  - LMS/SS-LMS easily achieve MMSE, but are not flexible
  - Max Eye Opening is flexible, but inaccurate or slow
  - Spectrum Matching is not flexible

- ZF is attractive, because it is flexible and able to achieve almost MMSE
  - But not well studied in DSP context, because LMS is better than ZF for DSP
  - Conventional ZF for DSP requires ADC which we would like to avoid
Outline

- Background
- State-of-the-art Equalizer for High-Speed I/O
- Conventional Adaptive Equalizer Control for High-Speed I/O
- Sign-based Zero-Forcing Adaptive Equalizer Control
  - Problems of Conventional Adaptive Equalizer Control
  - Pattern Filtering
  - Convolution and De-convolution of ISI
  - Sign-based Zero-Forcing
  - Extension of S-ZF for Low-Frequency Equalizer
- Implementation and Evaluation Results
- Summary
Pattern Filtering$^{[22]}$ (1/3)

- Do not passively use all received error information
- Actively choose error information to use

START

Choose a Filter Pattern $FP_i$

Watch for $FP_i$ in received data

Received?

Y

Update equalizer parameters using error info associated with $FP_i$

N
Keep watching for the selected *Filter Pattern* \( F_{P_i} \) (no timeout)

- If the *Filter Pattern* \( F_{P_i} \) is missing in the sequence, just wait forever
  - The sequence will eventually change
  - Do not update EQ parameters, when spectrum may be inadequate
    - Not halting updates for non-random data may cause parameter drift

*Pattern Filtering* works with limited random data

- For mostly periodic patterns, adaptation works slowly but steadily by catching limited random data
  - E.g. Ethernet frame with short random header and long payload filled by periodic 8B10B data
Pattern Filtering (3/3)

- Two options to choose a *Filter Pattern* $FP_i$

  - Randomly (similar to conventional adaptive control)$^{[22]}$
    - Make conventional adaptive controls operable for non-random data

  - From a specific set of patterns, in some specific way$^{[23]}$
    - Enable De-convolution of ISI and S-ZF adaptation scheme
Outline

- Background
- State-of-the-art Equalizer for High-Speed I/O
- Conventional Adaptive Equalizer Control for High-Speed I/O
- Sign-based Zero-Forcing Adaptive Equalizer Control
  - Problems of Conventional Adaptive Equalizer Control
  - Pattern Filtering
  - Convolution and De-convolution of ISI
  - Sign-based Zero-Forcing
  - Extension of S-ZF for Low-Frequency Equalizer
- Implementation and Evaluation Results
- Summary
Convolution and De-convolution of ISI

Pulse Response (Cursor + ISI)

Data sequence \{+1,-1\}

Convolution

De-convolution

Received Waveform (Ref. Level + Error)

Ref. Level

\[ \approx \text{Cursor} \times \text{sign(Data)} \]

Time Domain

Convoluted:

\[ B(f) \]

\[ A(f) \times \rightarrow C(f) = A \times B \]

De-convoluted:

\[ B(f) \downarrow \]

\[ A(f) = C \div B \quad \text{Division} \]

Frequency Domain

Multiplication

\[ A(f) \times \rightarrow C(f) \]

\[ \text{Division} \]

\[ A(f) = C \div B \]

Ref. Level
Convolution: \( ISI \times Data \rightarrow Error \)

Pulse response
\[ = \text{Cursor} \left( h_0 \right) + ISI \]

Data Sequence:
\[
\begin{array}{cccccc}
D_0 & D_1 & D_2 & D_3 & D_4 & D_5 & D_6 \\
\end{array}
\]

Ref. Level + Error \( E_4 \)

Error \( E_4 = \text{sign}(D_6)h_{-2} + \text{sign}(D_5)h_{-1} + (\text{sign}(D_4)h_0 - \text{Ref. Level}) \\
+ \text{sign}(D_3)h_1 + \text{sign}(D_2)h_2 + \text{sign}(D_1)h_3 + \text{sign}(D_0)h_4 \]

\text{sign}(D_n) : \{+1, -1\}
De-convolution: ISI $\leftarrow$ Error $\div$ Data$^{[23]}$

**ISI** is de-convolution of *Data* out of *Error*

**FP0:** 0 0 1 1 0 1 0

**FP1:** 0 0 1 0 0 1 0

Residual ISI $h_1$ can be calculated as the difference between error $E_4$ values for FP0 and FP1 which differ only at $D_3$.

\[
E_4^{FP0} = -h_2 + h_1 - h_0 - \text{Ref. Level} + h_1 + h_2 - h_3 - h_4
\]

\[
E_4^{FP1} = -h_2 + h_1 - h_0 - \text{Ref. Level} - h_1 + h_2 - h_3 - h_4
\]

\[
E_4^{FP0} - E_4^{FP1} = 2 \times h_1
\]
Outline

- Background
- State-of-the-art Equalizer for High-Speed I/O
- Conventional Adaptive Equalizer Control for High-Speed I/O
- Sign-based Zero-Forcing Adaptive Equalizer Control
  - Problems of Conventional Adaptive Equalizer Control
  - Pattern Filtering
  - Convolution and De-convolution of ISI
  - Sign-based Zero-Forcing
  - Extension of S-ZF for Low-Frequency Equalizer
- Implementation and Evaluation Results
- Summary
Sign-based Zero-Forcing (S-ZF)\[23\]

- FPD statistically de-convolves Data out of Error to ResISI\textsubscript{n}
  - Average of \{+1,-1\} of ResISI\textsubscript{n} indicates quantity of residual ISI at n UI
- Force weighted sum of average ResISI\textsubscript{n} towards Zero
Sign-based ISI De-convolution by FPD

- FPD performs $E_4^{FP0} - E_4^{FP1}$ statistically in a long term using *only error signs* without error quantities.

**Truth Table in FPD**

<table>
<thead>
<tr>
<th>Filter Pattern</th>
<th>D₀</th>
<th>D₁</th>
<th>D₂</th>
<th>D₃</th>
<th>D₄</th>
<th>D₅</th>
<th>D₆</th>
<th>E₄</th>
<th>ResISI₁ (sign(h₁)=sign(E₄)*sign(D₃))</th>
</tr>
</thead>
<tbody>
<tr>
<td>FP₀</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>+1</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td>0</td>
<td>-1</td>
</tr>
<tr>
<td>FP₁</td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td>1</td>
<td>-1</td>
</tr>
<tr>
<td></td>
<td>0</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td>1</td>
<td>0</td>
<td></td>
<td>0</td>
<td>+1</td>
</tr>
</tbody>
</table>

- Average ResISI₁ shows difference of $E_4$ between FP0 and FP1
- ResISI₁ is correlation between $D_3$ and $E_4$ : same as conventional ZF which measures correlation between $D_3$ and $E_4$ for random data

- Data values other than $D_3$ are chosen a priori
  - Error value should NOT be always 1 or 0
  - Filter patterns should be always received during adaptation period
Key Operations in FPD and FPB

- Filter Pattern Decoder checks FP0/FP1 equally
  - To perform statistical subtraction correctly
  - Implemented by *alternately* watching for FP0 and FP1

- Filter Pattern Balancer checks multiple FPDs equally
  - To define adaptation characteristics only by weight const.
  - Implemented by watching for *FPDs randomly*

- FPD/B keep watching for the FP until it is received
  - *No timeout* to guarantee above statistics
  - Pattern tolerant
    - Will not drift for any non-random data sequence
Outline

- Background
- State-of-the-art Equalizer for High-Speed I/O
- Conventional Adaptive Equalizer Control for High-Speed I/O
- Sign-based Zero-Forcing Adaptive Equalizer Control
  - Problems of Conventional Adaptive Equalizer Control
  - Pattern Filtering
  - Convolution and De-convolution of ISI
  - Sign-based Zero-Forcing
  - Extension of S-ZF for Low-Frequency Equalizer
- Implementation and Evaluation Results
- Summary
A Problem of Sign-based Zero Forcing

- Not applicable to Low-Frequency Equalizer
  - Filter Pattern must be very long to detect long-term ISI for LFE
    - LFE cancels long-term ISI in the range of 20~100 UI

- Adaptation speed is too slow for Low-Frequency Equalizer
  - 50bit FP is received once every 31 hours for random data at 10Gb/s
  - 100bit FP is received once every $4 \times 10^{12}$ years for random data at 10Gb/s
Extension of S-ZF for Low-Freq Equalizer\cite{24}

- **Basic idea:**
  - Handle Long-term ISI collectively in an aggregate manner
    - ISI for each individual bit is similar and not important for long-term ISI
    - Low-frequency equalizer cancels long-term ISI collectively
Extended Filter Pattern (EFP)

- **FP (Filter Pattern)**: Fixed 0/1 sequence (balanced ISI)
  - Minimize $h_{FP}$
- **MP (Middle Pattern)**: Equal 0/1 count (balanced 0/1)
  - Minimize $h_{MP}$
- **TP (Tail Pattern)**: Non-equal 0/1 count (imbalanced 0/1)
  - Maximize $h_{TP}$

$h_{FP}$: aggregate ISI from FP

$h_{MP}$: aggregate ISI from MP

$h_{TP}$: aggregate ISI from TP

**Measure Error $E$**

$E_{EFP1} = +h_{FP} + h_{MP} + h_{TP}$

$E_{EFP0} = +h_{FP} + h_{MP} - h_{TP}$

$E_{EFP1} - E_{EFP0} = 2 \times h_{TP}$
Example of Extended Filter Pattern

<table>
<thead>
<tr>
<th>Earlier</th>
<th>Middle Pattern</th>
<th>Filter Pattern</th>
<th>Error</th>
<th>ResISI</th>
</tr>
</thead>
<tbody>
<tr>
<td>Tail Pattern</td>
<td>50 bits</td>
<td>20 bits</td>
<td>(D_0\sim D_6)</td>
<td>(E_4)</td>
</tr>
<tr>
<td>EFP1</td>
<td># of 1 &gt;= 34 bits</td>
<td># of 0 = 10 bits</td>
<td>0 1 0 1 0 1 0</td>
<td></td>
</tr>
<tr>
<td># of 0 &lt;= 16 bits</td>
<td>0</td>
<td></td>
<td></td>
<td></td>
</tr>
<tr>
<td>EFP0</td>
<td># of 0 &gt;= 34 bits</td>
<td># of 1 = 10 bits</td>
<td></td>
<td></td>
</tr>
<tr>
<td># of 1 &lt;= 16 bits</td>
<td>1</td>
<td>-1</td>
<td></td>
<td></td>
</tr>
</tbody>
</table>

Probability to receive above EFP0 or EFP1 in random sequence

\[
P = \sum_{k=0}^{16} \frac{C(50, k)}{2^{50}} \times \frac{C(20,10)}{2^{20}} \times \frac{1}{2^7} = 0.0077 \times 0.1762 \times 0.0078 = 1.06 \times 10^{-5}
\]

For example, if logic clock is 1GHz, we can detect > 10k EFPs per second.
Outline

- Background
- State-of-the-art Equalizer for High-Speed I/O
- Conventional Adaptive Equalizer Control for High-Speed I/O
- Sign-based Zero-Forcing Adaptive Equalizer Control
- Implementation and Evaluation Results
- Summary
Implementation Example

- Equalizer: CTLE + LFE + 2-tap Speculative DFE
- Process technology: 28nm CMOS
- Data rate: 25Gb/s
- Power supply: 1.0V
- Power consumption: 240mW (RX)
- Area: 0.33mm$^2$ (RX)
Convergence time < 5 msec in real-time operation.
Measured BER for PRBS31

Channel loss: 25dB at Nyquist Frequency
Summary

- LMS automatically achieves MMSE (Wiener Filter)
  - Because of this beauty of LMS, ZF is hardly used in DSP context

- However, ZF is much more attractive in mixed-signal HSIO context
  - ZF is more flexible than LMS regarding to mixed-signal circuit architecture
  - ZF is also capable to achieve almost MMSE regarding to observed ISI
    - For the target channel, with proper weight vectors

- S-ZF is enabled by accurate ISI detection using Filter Pattern
  - ISI is accurately detected by explicit de-convolution of Data out of Error
  - FP also solves the problem of parameter drift due to non-random data

- Low-frequency loss has significant effects and should be equalized

- S-ZF is extended for LFE by handling long-term ISI collectively
References


References


shaping tomorrow with you