ABSTRACT

In the context of high-speed serial links, data-dependent jitter compensation techniques performed using phase predistortion. Broadly contemplated is an expansion of the notion of pre-emphasis beyond conventional amplitude compensation of ISI, whereby phase pre-emphasis for compensating data-dependent jitter (DDJ) is introduced. DDJ can be addressed by exploiting the relationship between the data sequence and the timing deviation. Phase pre-emphasis improves the signal integrity with little additional power consumption in the transmitter and with no cross-talk penalty.
Figure 1

Programmable delay

Transition decoder

Tap weights

Transmitted data

DATA-DEPENDENT JITTER PRE-EMPHASIS FOR HIGH-SPEED SERIAL LINK TRANSMITTERS

FIELD OF THE INVENTION

[0001] The present invention relates generally to data equalization techniques for high speed serial links.

BACKGROUND OF THE INVENTION

[0002] High bandwidth chip-to-chip interconnection is a crucial part of many systems today. High speed inputs/outputs (I/Os) are extensively used in server processors, memory-central processing unit (CPU) interfaces, multiprocessor systems, and gaming applications. With increasing speed of on-chip data processing, there is an increasing demand for higher data rates. These high speed I/Os must also be capable of supporting low cost package and board technologies which introduce large signal degradation through bandwidth loss, reflection, and crosstalk.

[0003] Equalization of high-speed serial links has evolved to compensate intersymbol interference (ISI) caused by frequency-dependent attenuation found in interconnects. Pre-emphasis-based equalization in the transmitter and decision feedback equalization in the receiver figure prominently in overcoming signal degradation and improving Bit Error Rate (BER). Currently, one challenge of equalization is minimizing power consumption while still improving signal integrity in the presence of attenuation and reflections.

[0004] Transmitters are a significant portion of the serial link power budget. The transmitter is required to drive enough power over lossy interconnects to meet minimum receiver sensitivity requirements. The use of amplitude pre-emphasis techniques, such as feed forward equalization, increases the power consumption and chip area and places additional demands on the dynamic range of the transmitter. Also, in a transmission system with high cross-talk between the channels, especially the near end cross talk where a transmitter is leaking into a neighboring receiver, amplitude pre-emphasis will enhance the high frequency cross-talk of the victim receiver which implies that its signal to noise ratio will be degraded.

[0005] In view of the foregoing, a growing need has been recognized in connection with improving upon the shortcomings and disadvantages presented by conventional arrangements.

SUMMARY OF THE INVENTION

[0006] In accordance with at least one presently preferred embodiment of the invention, there is broadly contemplated an expansion of the notion of pre-emphasis beyond conventional amplitude compensation of ISI, whereby phase pre-emphasis for compensating data-dependent jitter (DDJ) is introduced. DDJ can be addressed by exploiting the relationship between the data sequence and the timing deviation. Phase pre-emphasis improves the signal integrity with little additional power consumption in the transmitter and with no cross-talk penalty.

[0007] In summary, one aspect of the invention provides an apparatus for providing data dependent jitter in data transmission, the apparatus comprising: an input arrangement for accepting data input; an arrangement for detecting and decoding transitions in the data input; the arrangement for detecting and decoding being adapted to ascertain data transition history in the data input; and an arrangement for applying at least one delay to at least a portion of the data input, based on the ascertained data transition history.

[0008] Another aspect of the invention provides a method of providing data dependent jitter in data transmission, the method comprising the steps of: accepting data input; detecting and decoding transitions in the data input; the step of detecting and decoding comprising ascertaining data transition history in the data input; and applying at least one delay to at least a portion of the data input, based on the ascertained data transition history.

[0009] For a better understanding of the present invention, together with other and further features and advantages thereof, reference is made to the following description, taken in conjunction with the accompanying drawings, and the scope of the invention will be pointed out in the appended claims.

BRIEF DESCRIPTION OF THE DRAWINGS

[0010] FIG. 1 is a block diagram of a first embodiment.

[0011] FIG. 2 is a schematic illustration of channel effect on data-dependent jitter.

[0012] FIG. 3 is a block diagram of a second embodiment.

DESCRIPTION OF THE PREFERRED EMBODIMENTS

[0013] Generally speaking, in a link where losses due to signal attenuation are dominating other channel impairments such as reflection due to discontinuities, the amount of DDJ of a random bit sequence at the end of a link, which corresponds to timing deviation from ideal transition time, depends on the previous bit sequences only. Thus, the amount of DDJ at a given transition time corresponding to a given channel can be predicted by simply keeping track of the previous transitions. Also, it can be shown that the most recent transitions in previous data bits contribute to DDJ with larger amount. Thus, most of the DDJ can be predicted and compensated for by only detecting some of the most recent transitions. For typical links, one needs to detect down to the three previous data transitions to predict most of the DDJ occurring at the current data transition.

[0014] Phase pre-emphasis manipulates the data transitions to neutralize DDJ at the end of the transmission channel. The delay of the data transition to be transmitted is dynamically changed depending on the previous bit sequence detected. In general, the highest frequency components of the data sequence are delayed before being transmitted in such a way that the arrival time of the different frequency components of the data sequence closely coincide at the end of the link. This delay (or phase) pre-distortion will thus increase both the timing and voltage margins of the signal at the end of the link.

[0015] In accordance with one preferred embodiment of the present invention, a full-rate clock transmitter architecture approach shows how a serial type phase pre-distortion is implemented. In accordance with a second preferred embodiment of the present invention, a quarter-clock transmitter architecture shows how a parallel type phase pre-distortion is implemented.
FIG. 1 illustrates the aforementioned first embodiment, corresponding to the front-end of a transmitter which performs phase pre-distortion on the serial data that is transmitted over the transmission medium possibly using a line driver. As shown, there are preferably provided single data bit time delay elements 102, 103 and 104. XOR gates 105, 106 and 107, a transition decoder 108 connected to outputs of XOR gates 105/106/107, programmable tap weights 109 and a programmable delay line 110.

Preferably, a serial data stream 101 is successively delayed by the single data bit delay elements 102, 103 and 104. Though only three delay elements are shown, the actual number of these delay elements can be as high as needed depending on how many previous transitions need to be detected in order to compute the phase pre-distortion delay to be applied to the current bit. XOR gate 105 compares signal 101 to the output of 102 to detect if a transition is occurring on the current bit to be transmitted. If a transition is present the output of XOR gate 105 will be high. XOR gates 106, 107 function similarly to gate 105 but, respectively, compare signal outputs from delay elements 102/103 and from 103/104. As with gate 105, if a transition is detected then the output of gate 106 or 107 will be high, as appropriate. Then, depending on the data transition history that is determined by the transition decoder 108, a phase pre-distortion delay is added or subtracted to the current data transition time using the programmable delay line 110.

The amount of pre-distortion delay is programmed through the programmable tap weights 109 and depends on the transmission medium that needs to be equalized. Decoder 108 can be embodied and implemented in essentially any suitable manner. For instance, decoder 108 can employ simple Boolean logic to carry out its function as described above. At the same time, the amount of delay to be applied by delay line 110, once the data transition history is known, can depend on the channel at hand. As such, the delay could be pre-set or continuously optimized (if the transmitter can receive information about the channel from the receiving end of the link).

For a typical transmission medium such as coaxial cable where the losses are dominated by frequency dependent attenuation, the high frequency components of a random data sequence arrive first at the end of the transmission medium, while the lowest ones arrive later, as illustrated in the left-hand portion of FIG. 2. Thus, in the illustrative and non-restrictive example shown, the higher-frequency component 250 arrives prior to component 252; the combined graph of these is indicated at 254. The pre-echo, if employed in accordance with the arrangement of FIG. 1, is shown in the right-hand portion of FIG. 2 upon exit from channel 256. Generally, the longest pre-distortion delay is added to the transmitted bit when the transition decoder 108 detects a transition on the previous bit, while the applied delay will be comparatively shorter to the extent that the previous transition detected by the transition decoder 108 occurs with “earlier” bits. As shown, with “input” components 250 and 252 now transformed into “output” components 258 and 260, respectively, the former (258) has a computationally greater delay applied, the result of which can be appreciated in the combined graph at 262 as well as the combined diagram at 264. The marked “DDJ” distance corresponds to this difference in delay. The “eye opening” is the region where data transitions are not present (when considering only DDJ and not jitter due to random noise); as such, data will be detected with no errors if sampled in this region.

FIG. 3 illustrates a second preferred embodiment of the present invention, corresponding to a quarter-rate transmitter which performs phase pre-distortion on the serial data that is transmitted over a transmission medium (e.g., a line driver), while using the quarter-rate data to detect the transition history. Included are data latches 301, 302, 303 and 304, a transition detector and decoder 305, programmable tap weights 306, programmable delay elements 307 and a quarter-rate 4:1 data multiplexer 308. Corresponding transmission graphs are illustrated to the right of the drawing. It should be clearly understood that while an illustrative and non-restrictive example of a quarter-rate implementation is shown in FIG. 3, the present invention, in accordance with at least one presently preferred embodiment, of course broadly embraces the implementation of essentially any workable arrangement involving parallel partial-rate data streams. Thus, for instance, the use of two half-rate data streams, as opposed to four quarter-rate streams, is conceivable in employing features discussed herein in connection with FIG. 3.

As shown, four parallel quarter-rate data streams D0, D1, D2 and D3 are received using sampling latches 301, 302, 303 and 304, respectively. These latches are not mandatory but are usually used to align the incoming parallel data to the local clock. As opposed to the first embodiment where the transition decoder was operating on serialized data, here the transition decoder 305 operates on the quarter-rate data stream to determine the data sequence to be transmitted through the channel. (It should be understood that this quarter rate example could be extended to even wider parallel data at the input that would be then time multiplexed before being transmitted.) Decoder 305 hereby generates a code that is then passed to the programmable delay elements 307 through the programmable tap weights 306 that will set the amount of phase pre-distortion on the data depending on the transmission medium used. To that end, the programmable delay elements 306 will pre-distort the phase of the four quarter-rate clocks C4_0, C4_90, C4_180 and C4_270 that are used to time multiplex the parallel quarter-rate data into a high speed serial data that is then transmitted through the transmission medium.

In recapitulation, there are broadly contemplated herein new equalization techniques involving phase pre-distortion as an alternative to conventional amplitude pre-emphasis. These techniques are in particular suitable for low power and low area serial link applications, especially when dealing with a lossy transmission medium that introduces excessive pattern dependent jitter. Phase pre-distortion is less power hungry than amplitude pre-emphasis and requires a smaller chip area. Also, phase pre-distortion does not degrade the signal to noise ratio due to the enhancement of channel cross talk at high frequency that is caused by amplitude pre-emphasis.

If not otherwise stated herein, it is to be assumed that all patents, patent applications, patent publications and other publications (including web-based publications) mentioned and cited herein are hereby fully incorporated by reference herein as if set forth in their entirety herein.

Although illustrative embodiments of the present invention have been described herein with reference to the
accompanying drawings, it is to be understood that the invention is not limited to those precise embodiments, and that various other changes and modifications may be affected therein by one skilled in the art without departing from the scope or spirit of the invention.

What is claimed is:

1. An apparatus for providing data dependent jitter in data transmission, said apparatus comprising:
   an input arrangement for accepting data input;
   an arrangement for detecting and decoding transitions in the data input;
   said arrangement for detecting and decoding being adapted to ascertain data transition history in the data input; and
   an arrangement for applying at least one delay to at least a portion of the data input, based on the ascertained data transition history.

2. The apparatus according to claim 1, wherein the input data comprises a serial data stream.

3. The apparatus according to claim 2, wherein said input arrangement comprises a plurality of single data bit delay elements for successively delaying the serial data stream of the input data.

4. The apparatus according to claim 3, wherein said detecting and decoding arrangement comprises a plurality of XOR gates for comparing portions of the serial data stream at different sampling positions among said plurality of single data bit delay elements.

5. The apparatus according to claim 2, wherein said arrangement for applying at least one delay is adapted to add or subtract a pre-distortion delay to the serial data stream of the input data.

6. The apparatus according to claim 1, wherein the data input comprises a plurality of parallel partial-rate data streams.

7. The apparatus according to claim 6, wherein said input arrangement comprises a plurality of receiving elements for aligning the partial-rate data streams to a local clock.

8. The apparatus according to claim 6, wherein:
   said detecting and decoding arrangement is adapted to influence the partial-rate data streams to determine a data sequence to be subsequently transmitted in an output serial data stream; and
   said detecting and decoding arrangement is adapted to generate a code informing an amount of pre-phase distortion to be applied to the input data.

9. The apparatus according to claim 6, wherein said arrangement for applying at least one delay is adapted to pre-distort the phase of a clock used in transitioning the partial-rate data streams into an output serial data stream.

10. The apparatus according to claim 6, wherein the data input comprises a plurality of parallel quarter-rate data streams.

11. A method of providing data dependent jitter in data transmission, said method comprising the steps of:
   accepting data input;
   detecting and decoding transitions in the data input;
   said step of detecting and decoding comprising ascertaining data transition history in the data input; and
   applying at least one delay to at least a portion of the data input, based on the ascertained data transition history.

12. The method according to claim 11, wherein the data input comprises a serial data stream.

13. The method according to claim 12, wherein said accepting step comprises accepting a plurality of single data bit delay elements for successively delaying the serial data stream of the input data.

14. The method according to claim 13, wherein said detecting and decoding step comprises comparing portions of the serial data stream at different sampling positions among the plurality of single data bit delay elements.

15. The method according to claim 12, wherein said step of applying at least one delay comprises adding or subtracting a pre-distortion delay to the serial data stream of the input data.

16. The method according to claim 11, wherein the data input comprises a plurality of parallel partial-rate data streams.

17. The method according to claim 16, wherein said input step comprises aligning the partial-rate data streams to a local clock.

18. The method according to claim 16, wherein:
   said detecting and decoding step comprises influencing the partial-rate data streams to determine a data sequence to be subsequently transmitted in an output serial data stream; and
   said detecting and decoding step comprises generating a code informing an amount of pre-phase distortion to be applied to the input data.

19. The method according to claim 16, wherein step of applying at least one delay comprises pre-distorting the phase of a clock used in transitioning the partial-rate data streams into an output serial data stream.

20. The method according to claim 16, wherein the data input comprises a plurality of parallel quarter-rate data streams.