AN ASYNCHRONOUS DATA RECOVERY / RETRANSMISSION TECHNIQUE WITH FOREGROUND DLL CALIBRATION

S. Nagavarapu, J. Yan, E. K. F. Lee, R. L. Geiger

Department of Electrical and Computer Engineering,
Iowa State University,
Ames, IA 50011

ABSTRACT

A new technique for asynchronous data recovery based upon using a delay line in the incoming data path is introduced. The proposed data recovery system is well suited for tight tolerance channels and coding systems supporting standards that limit the maximum number of consecutive 0's and 1's in a data stream. The proposed system, which samples incoming data in the delay line on transitions of the data itself, has a reduced sensitivity to jitter on incoming data when compared to standard PLL-based data recovery systems. Further, the proposed technique actually recovers all data in the channel for a fixed interval prior to the first data transition. A system clock driven DLL is used to generate precise delays. This technique recovers the incoming data without recovering a clock signal. When used in a retransmitter, retimed data with a very low jitter that is essentially independent of the jitter on the incoming data stream is retransmitted.

1. INTRODUCTION

Data recovery systems generally have to contend with the conflicting requirements of fast acquisition time and immunity to internal noise. Especially, in the case of PLL based recovery systems, the Loop Filter bandwidth has to be selected as a compromise between the two requirements. DLL-based clock recovery systems have better immunity to noise as the error due to accumulated noise is flushed out at the end of the delay line and not recirculated as with PLL's. Many architectures have been proposed using DLL's. These structures either use a DLL to recover clock information that has been transmitted along with the data, or align a local clock with the incoming data [1],[3].

Recently, low speed applications of DLL's in generating precise delays for oversampling [4],[5] have been proposed. In their approach, the data and a local clock are both passed through delay lines, and the data is sampled based on the difference in delays between the clock and the data delay lines.

In most data coding systems, a certain number of transitions are guaranteed in every time interval for clock synchronization. In the case of 8B/10B encoding, the coding system used on the Fibre channel, DC balance is maintained by ensuring that the number of 0's and 1's in every consecutive two symbols is equal. With the symbol set used in the Fibre channel, this results in a maximum of 5 consecutive 0's or 1's in the data stream. Knowledge of the maximum non-changing bit sequence is exploited in the present system which uses the data transitions directly to latch incoming data that is temporarily stored in a delay line, thus eliminating the need for a recovered or sampled clock.

2. PRINCIPLE OF OPERATION

Consider incoming data with a nominal bit period T_d passing through a delay line as illustrated in Figure 1. If delay stages having an accurate delay of $T=T_d$ are available and the cumulative effects of input jitter at the output of the delay line is small compared to T, then the data in the delay line will be read accurately when it is sampled T/2 sec after an input transition. The issue of accurately controlling the delay line which will be achieved by using a Voltage Controlled Delay Line (VCDL) will be discussed later.

If the number of delay stages is greater than or equal to the maximum number of successive zeros or ones in the incoming data stream and if all outputs of the delay line are sampled T/2 after every data transition, then all incoming data can be recovered from the samples of the outputs of the delay line. Data recovery in the proposed system is based upon sampling outputs of a delay line driven by the input data stream. Sampling takes place T/2 sec

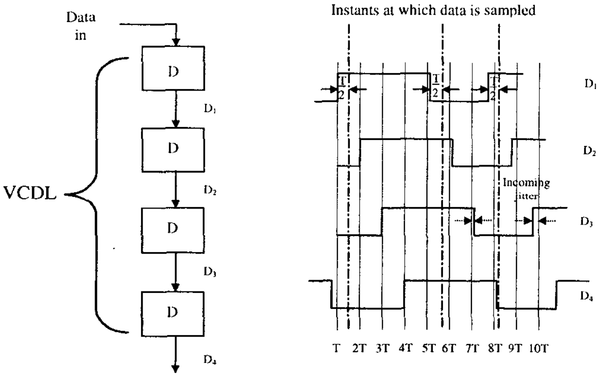

Figure 1: Reading data using data transitions

0-7803-5471-0/99/$10.00©1999 IEEE

after each input transition. When incoming data transitions are closer together than the delay associated with the total length of the delay line, some data points may be sampled more than once on successive data transitions. Simple logic circuitry purges any redundant samples and sequentially places the recovered data into memory. A synchronizer retimes the data for retransmission. Thus, this asynchronous data recovery system does not recover a clock from the incoming data and uses the incoming data transitions themselves to initiate the sampling operation and the transfer of sampled data to a temporary memory.

This system offers several distinct advantages over PLL-based data recovery systems. System jitter introduced by noise in conventional PLL's, as well as the cumulative effects of incoming jitter is non-existent. There is no data loss associated with acquiring lock as is inherent in any PLL-based system. Actually, successive samples of the input prior to the first input transition equal in number to the number of delay stages are available. Finally, the jitter in the retimed data can also be very low since the retiming clock is no longer dependent upon jitter in a recovered clock.

3. EFFECTS OF JITTER ON INCOMING DATA

In serial Communication systems, jitter refers to the mis-positioning of significant edges in a sequence of data bits from their ideal positions. Sources of jitter include thermal noise, crosstalk, interaction between neighboring clocks, EMI, and bandwidth limitations of the system. Jitter can be broadly classified into two categories – Deterministic Jitter (DJ) and Random Jitter (RJ). Deterministic Jitter is bounded in amplitude and includes low frequency harmonic modulation, Data Dependent Jitter (DDJ) and Duty Cycle Distortion (DCD). Random Jitter is unbounded and probabilistic in nature and is dominantly Gaussian. It is also referred to as phase noise and is caused by thermal noise, shot noise, crosstalk etc.

Jitter is characterized in the time domain, but it has the effect of frequency modulation on the data. For example, low frequency drifts appear as sidebands in the frequency spectrum, while phase noise creates 'noise skirts' around the clock frequency.

Component	Jitter at Receiver (p-p in UI)
Sinusoidal swept frequency (637 kHz to 5 MHz)	0.10
DJ (637 kHz - 531 MHz)	0.38
RJ (637 kHz - 531 MHz)	0.22
TOTAL	0.70

Table 1. 1.0625 Gigabaud Jitter Tolerance Allocation at Receiver

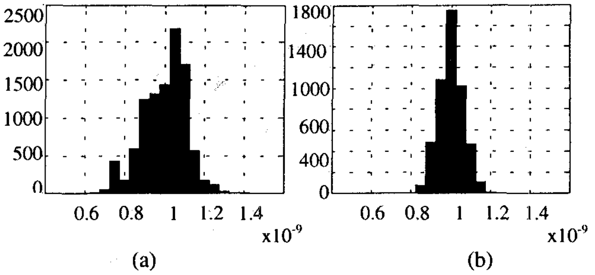

Figure 2. Distribution of normalized pulse widths (a) Single (X) (b) Sum of consecutive pulses (Y)

Jitter is harmful and undesirable in any data recovery system and is the major source of bit errors. In the asynchronous system proposed, it is especially harmful if the incoming jitter results in data transitions being significantly displaced. The effect of increased jitter is an increase in Bit Error Rates.

Jitter tolerance standards have been established for the 1.0625 Gbaud Fibre Channel resulting in the jitter budgets specified in Table 1 [8] using modeling information from various sources. [9], [10]

In an asynchronous system, which is 'clockless' in nature, jitter is defined in relative as opposed to absolute terms, i.e. $J_1 - J_2$ where J_1 and J_2 represent instantaneous jitter on 2 different data transitions. To relate the jitter data to the proposed scheme, a new variable $X_n = J_{n+1} - J_n$ is defined, where n and n+1 refer to two successive transitions. A histogram representing the distribution of X_n for a sample set of 50000 data points is shown in Figure 2(a) based upon a jitter model meeting the tolerance allocation specifications of Table 1.

A study of the random process $X(N) = [X_1\ X_2 \ldots X_N]$ where $X_1, X_2, \ldots X_N$ have the same distribution as $X_n = J_{n+1} - J_n$, shows an autocorrelation sequence with a high degree of periodicity and a significant component at π in the resulting Power Spectral Density

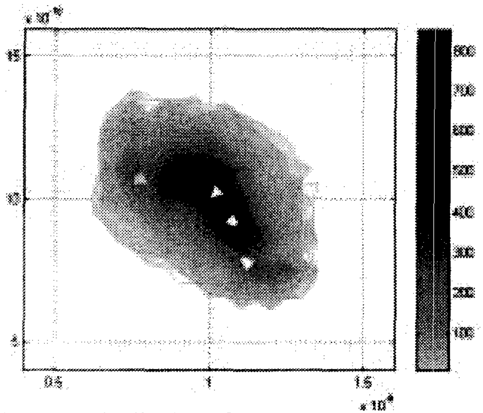

Figure 3. Distribution of consecutive pulses (normalized)

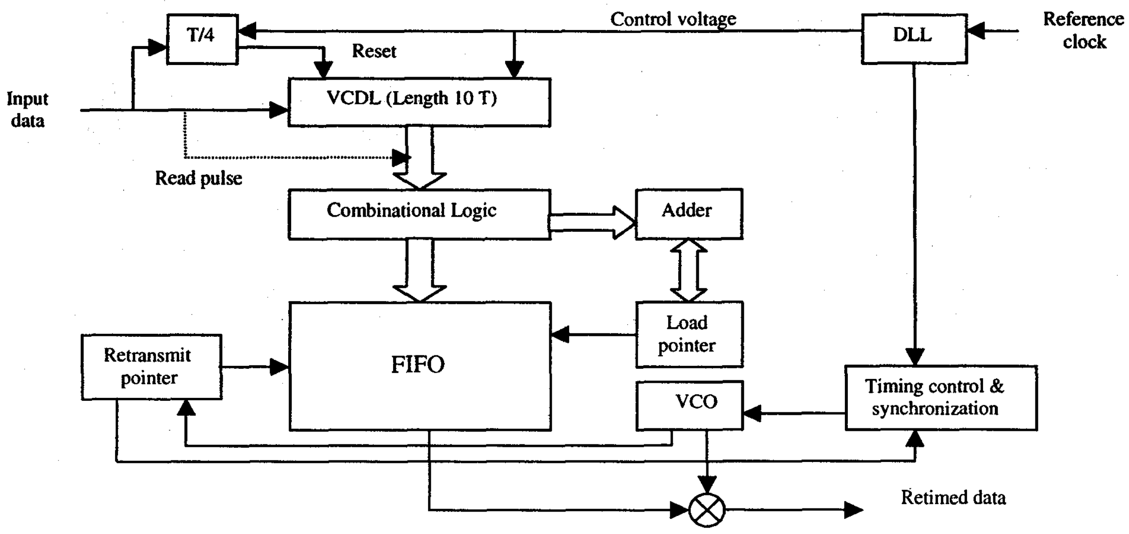

Figure 4. Block diagram of proposed implementation

(Deterministic jitter component). Since the autocorrelation sequence at n=1 $R_{XX}(1)$ is of significant negative amplitude, a much tighter distribution can be obtained with the variable $Y = X_n + X_{n+1}$ as shown in Figure 2(b).

A receiver with a reduction in BER can be designed using consecutive pulse information. A detector that can accomodate the sum of consecutive pulse widths can be achieved by doubling the length of the VCDL. The widths of consecutive pulses are thus correlated. The correlation of consecutive pulses is demonstrated by the 2-dimensional scatter plot in Figure 3 generated by a psuedorandom 8B10B pulse train.

4. ARCHITECTURE AND IMPLEMENTATION ISSUES

The block diagram of the proposed implementation scheme is shown in Figure 4. The Voltage Controlled Delay Line is 10 T long with a resolution of T/4 where T is 941 ps (Fibre Channel Specifications). Data is rippled through the delay line, which is read on every positive going transition of the data and reset T/4 later. The control voltage for the delay cells is generated by a DLL synchronized with the system clock.

The latched data is sent to a combinational logic that outputs the actual transmitted data and stores it in a FIFO style RAM, with a load pointer containing the address of the last data bit loaded. This pointer is updated using an encoder and adder every time data is loaded into the FIFO.

Once a sufficient number of bits accumulate in the FIFO, they are clocked out using the retransmit clock. Another pointer, the retransmit pointer, stores the location of the data bit being clocked out. Using the information from the load and retransmit pointer, the frequency of the retransmit clock is adapted to achieve elasticity, i.e. to adjust for the frequency difference between the transmitting and local clock.

Two key performance characteristics must be satisfied for successful operation of this system. First, the delay T of the delay line must be very close to T_d, where T_d is the reciprocal of the reference clock frequency. Secondly, the delays in each stage must be equal and independent of the input data sequence.

The first characteristic is readily achievable in tight tolerance channels, such as the Fibre Channel with a ±100 ppm tolerance window for the transmit clock frequency. Specifically, a DLL can be used to lock the delay of a programmable delay cell to a crystal reference signal that is generally available in tight tolerance channels. If matching of the delay cells in the DLL to those in the signal path delay stage is achieved, the delay in the signal path delay stages will also be locked to the crystal reference. In a tight tolerance channel, this delay can be maintained to be very close to the bit period of the incoming data T_d, with only a small accumulated time error due to propagation through the signal delay line, provided the number of delay stages is not too large. The referenced DLL in the control loop inherently has small jitter and since it is not in the signal path, it can be put in a sleep mode once the correct delay has been established. It need only be reactivated at power up or when temperature changes become sufficiently large so as to cause a problematic change in delay.

The second concern is that of data dependency which could lead to increased jitter in the delayed stream resulting in increased bit error rates, or even loss of data bits. Delay cells with minimal data dependency can be designed to address the second concern. A delay cell that satisfies this requirement is shown in Figure 5 along with some simulation results for a 0.35um CMOS process available through MOSIS. The delay can be programmed over a wide range with V_{bn} as evidenced by the plot of Figure 6.

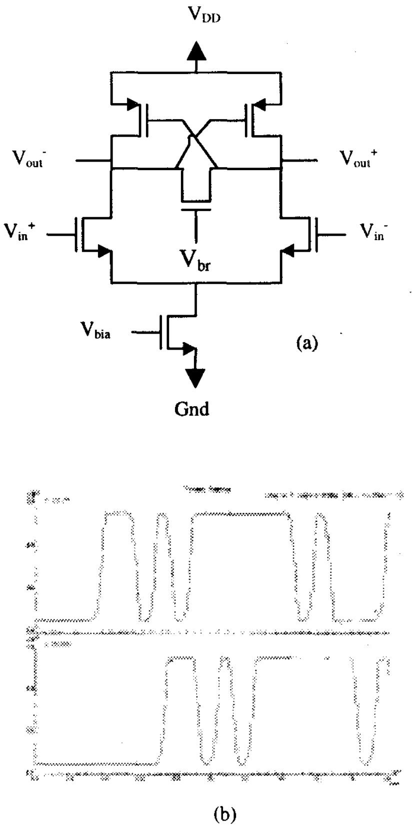

Figure 5. Delay cell (a) Schematic
(b) Simulation results

5. CONCLUSION

A new technique for data recovery in tight tolerance channels has been introduced. This technique is applicable for recovery of data in high-speed systems such as the 1.0625 Gb/s Fibre channel. The technique is based upon putting the incoming data stream into a delay line in which the delays are precisely referenced to a stable reference frequency and in which the incoming data transitions themselves define the sampling points. This system does not require clock recovery, suffers no loss of data during acquisition, has a reduced sensitivity to jitter in the incoming data and does not exhibit jitter enhancement associated with VCO tracking in a PLL. The system can be designed so that the retimed data has a significant reduction in jitter below what is often seen in clock recovery circuits.

6. ACKNOWLEDGEMENTS

This work was supported in part by Rocket Chips Inc. and Texas Instruments Inc.

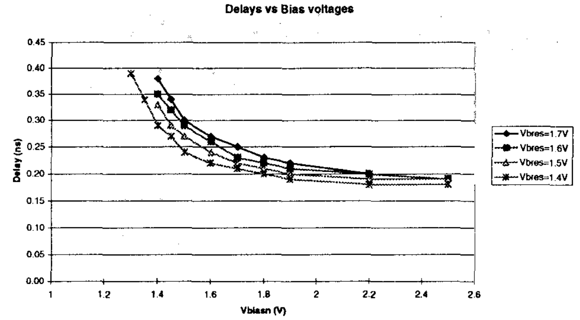

Figure 6. Performance of delay cell

7. REFERENCES

1. T.H. Lee, J.F. Bulzaccheli, 'A 155-MHz Clock Recovery Delay- and Phase-Locked Loop', IEEE J. Solid State Circuits, vol.27, No. 12, pp. 1736-1745, Dec 1992.
2. Behzad Razavi, 'Design of Monolithic Phase-Locked Loops and Clock Recovery Circuits - A Tutorial'
3. M.G. Johnson, E.L. Hudson, 'A Variable Delay Line PLL for CPU-Coprocessor Synchronization', IEEE J. Solid-State Circuits, vol. 23, pp. 1218-1223, Oct. 1988.
4. J. Kang, R.K. Cavin III, 'A CMOS High-Speed Data Recovery Circuit using the Matched Delay Sampling Technique', IEEE J. Solid-State Circuits, vol. 32, pp. 1588-1596, Oct. 1997.
5. C.T. Gray, W. Liu, W.A.A. Van Noije, T.A. Hughes, Jr., R.K. Cavin III, 'A Sampling Technique and its CMOS Implementation with 1 Gb/s Bandwidth and 25 ps Resolution', IEEE J. Solid-State Circuits, vol. 29, pp. 340-349, March 1994.
6. He Du, 'High Frequency Differential VCO with Common biased Clipper', United States Patent no. 5638030, June 10, 1997.
7. Wavecrest Corporation, 'Jitter Analysis "101" – A Foundation for Jitter Measurements', May 1998.
8. 'Fibre Channel – Methodologies for Jitter Specification', NCITS Working Draft proposed Technical Report, Rev 7 <http://www.schelto.com/t11_2/Jitter/jitter.htm> Dec 19, 1998.
9. O. Agazzi, 'QAM-Based 1000BASE-T˙ Transceiver', <http://grouper.ieee.org/groups/802/3/z/public/presentations/mar1997/>, March 1997.
10. G.H. Im, D.D. Harman, G. Huang, A.V. Mandzik, M.H. Nguyen, J.J. Werner, '51.84 Mb/s 16-Cap ATM LAN Standard', IEEE J. Selected areas in Comm., vol. 13, pp. 620-632, May 1993.
11. M. Heshami, B.A. Wooley, 'A 250-MHz Skewed-Clock Pipelined Data Buffer', vol. 31, pp. 370-383, March 1996.