LETTER Special Section on Analog Circuits and Related SoC Integration Technologies

A Study on Fully Digital Clock Data Recovery Utilizing Time to Digital Converter

Philipus Chandra OH[†], *Nonmember*, Akira MATSUZAWA[†a)], *Member*, *and* Win CHAIVIPAS[†], *Nonmember*

SUMMARY Conventional clock and data recovery (CDR) using a phase locked loop (PLL) suffers from problems such as long lock time, low frequency acquisition and harmonic locking. Consequently, a CDR system using a time to digital converter (TDC) is proposed. The CDR consists of simple arithmetic calculation and a TDC, allowing a fully digital realization. In addition, utilizing a TDC also allows the CDR to have a very wide frequency acquisition range. However, deterministic jitter is caused with each sample, because the system's sampling time period is changing slightly at each data edge. The proposed system does not minimize jitter, but it tolerates small jitter. Therefore, the system offers a faster lock time and a smaller sampling error. This proposed system has been verified on system level in a Verilog-A environment. The proposed method achieves faster locking within just a few data bits. The peak to peak jitter of the recovered clock is 60 ps and the RMS jitter of the recovered clock is 30 ps, assuming that the TDC resolution is 10 ps. In applications where a small jitter error can be tolerated, the proposed CDR offers the advantage of fast locking time and a small sampling error.

key words: *clock data recovery, time to digital converter, phase locked loop*

1. Introduction

The CDR system is an essential part in the receiver of digital communication instruments. A conventional CDR system contains a PLL [1]–[4], and it therefore also takes on the drawbacks of the PLL such as a long lock time, the need for many preamble data, the possibility of harmonic locking and the acquisition rate being at a low frequency (the VCO center-frequency needs to be within 25% of the data rate) [5].

A CDR system utilizing a TDC was presented in [6]. The information obtained from the TDC is fed to a triple loop analog feedback system consisting of a center frequency-tuning loop, a frequency detection loop, and a phase detection loop. This structure consumes power and requires time to lock.

However, by using a high-resolution TDC, the feedback system consists of just a few basic arithmetic calculation blocks to generate the sampling clock. In addition, the CDR has a very wide frequency acquisition range.

2. Proposed CDR Algorithm

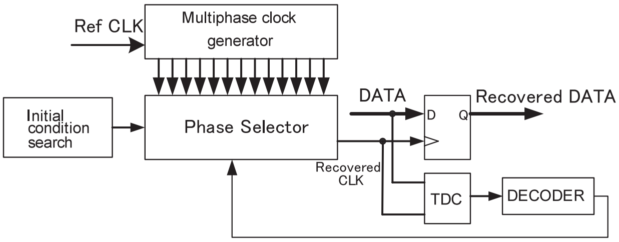

Fig. 1 Proposed CDR.

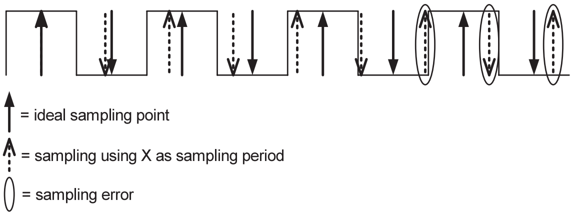

Fig. 2 Sampling directly using result from TDC.

Figure 1 shows the proposed CDR concept. Its operation can be split up in two stages: first, the initialization stage determines the sampling period within the resolution of the TDC. The information obtained from the first stage is then used to generate the recovered clock for sampling the incoming data. The proposed CDR system consists of several logic circuits, the TDC, a digital feedback, and a multiphase clock generator.

The reference clock that is used to generate the multiphase clock is independent of the data period.

2.1 Initialization Stage

The acquisition of the information regarding the sampling period is determined by using the TDC, logic circuits, and the preamble data in the initialization stage. The information, i.e., the clock for sampling the incoming data, is recovered by selecting the phase clock from the multiphase clock generator. The selection is made by using the TDC and some simple logic in the feedback system.

First, the preamble data period, X, is measured using a TDC. Sampling the data directly with the measured data-period X leads to sampling-errors, as is illustrated in Fig. 2.

By switching between two almost identical sampling frequencies, sampling errors can be avoided; this is shown in Fig. 3. For example, if the data period is 1001 ps and the TDC resolution is 10 ps, X is either 1000 ps or 1010 ps. Sampling with the fixed period X leads to many sampling

Manuscript received November 20, 2006.
Manuscript revised January 15, 2007.

[†]The authors are with the Department of Physical Electronics, Tokyo Institute of Technology, Tokyo, 152-8552 Japan.
a) E-mail: matsu@ssc.pe.titech.ac.jp

DOI: 10.1093/ietele/e90–c.6.1311

errors, as was indicated in Fig. 2. However, if the sampling is done with two sampling-periods, e.g., 1000 ps (upper boundary) and 1010 ps (lower boundary) and by using an algorithm to determine the sampling-period for each sample, sampling errors should theoretically not occur.

Figure 4 explains the function of the initial condition search block to obtain the upper and lower boundary.

After X has been determined, each of the next two preamble data windows is sampled five times (A, B, C, D, and E in Fig. 4). The functions are defined as follows:

$$A = X + 2 \cdot TDC_{resolution} \tag{1}$$

$$B = X + TDC_{resolution} \tag{2}$$

$$C = X \tag{3}$$

$$D = X - TDC_{resolution} \tag{4}$$

$$E = X - 2 \cdot TDC_{resolution} \tag{5}$$

where A is the distance between A_n and A_{n+1}, and so on. If the sampling period is smaller than the incoming data period, the distance to the data edge decreases with every sampling step. The upper and lower boundaries are chosen from these distances after the distance between the sampling edge and data edge has been measured.

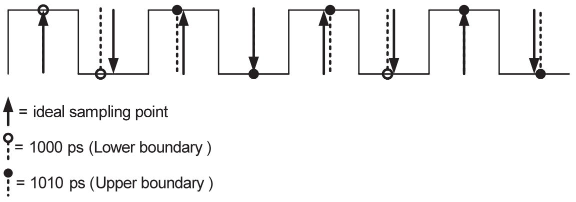

Fig. 3 Sampling with two almost identical sampling rates.

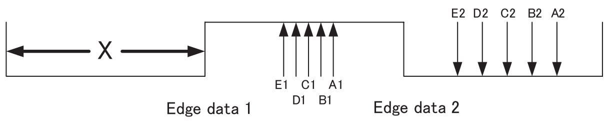

Fig. 4 Obtaining upper and lower boundary.

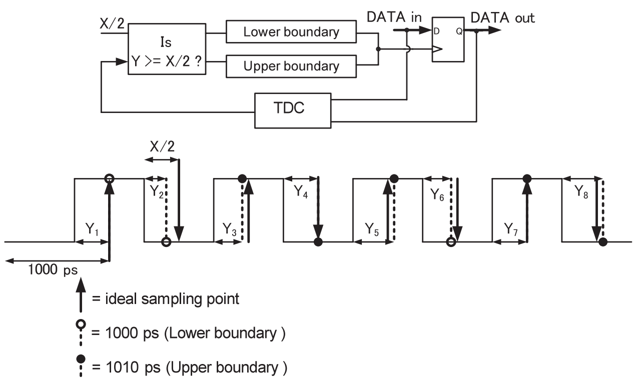

Fig. 5 Digital feedback flowchart and corresponding waveform.

2.2 Digital Feedback

The distance between the sampling edge and the data edge is measured, recorded, and compared to obtain the upper and lower boundaries. This time difference, called Y and illustrated in Fig. 5, is determined with the TDC. Y is then compared with $X/2$: if Y is greater than $X/2$, the lower boundary is used for sampling, otherwise the upper boundary is used. This process is shown in Fig. 5.

3. Simulation Results

The proposed system is verified using Verilog-A. The simulation, whose result is presented in Fig. 6, is performed with the following conditions:

TDC resolution = 20 ps,

Tdata = 987 ps.

Data type: Pseudo Random Binary Sequence (PRBS) $2^7 - 1$. The simulation result shows that the system is able to maintain Y to be around half of the bit period. The acquisition of the proposed CDR system is a few data bits (7 data bits), while the CDR using PLL [3] needs 40 data bits.

3.1 Jitter Analysis

Deterministic jitter is generated with each data sampling. Therefore, the jitter accumulates due to its digital nature when there is no data edge as shown in Fig. 7.

Peak to peak of the deterministic jitter depends upon the digital resolution and the data type. For PRBS with a sequence length of $2^N - 1$, the peak to peak jitter is bounded by

$$Jitter_{p-p} \leq 2 \cdot N \cdot TDC_{resolution}. \tag{6}$$

This is shown in Fig. 8 which is a compilation of jitter

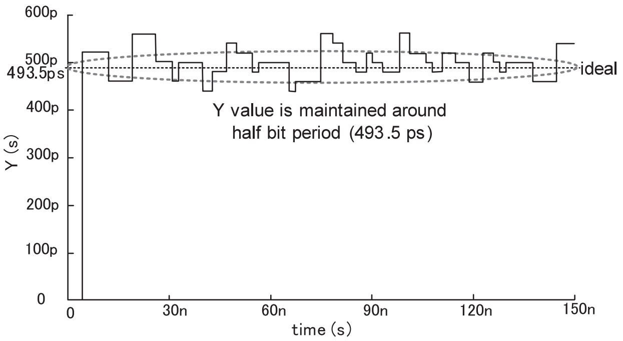

Fig. 6 Distance between sampling edge and data edge using proposed algorithm.

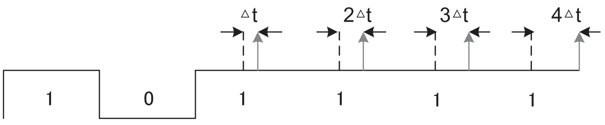

Fig. 7 Accumulating jitter.

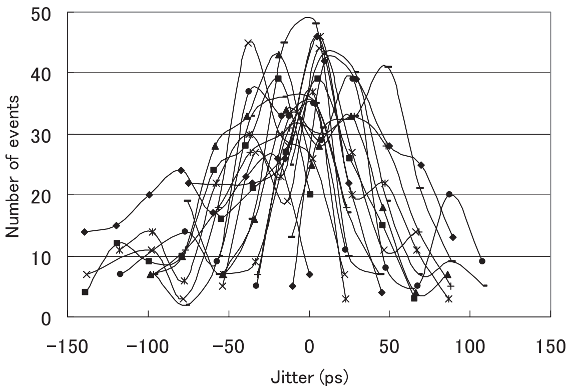

Fig. 8 Deterministic jitter histogram compilation for Tdata varying from 981 ps to 999 ps.

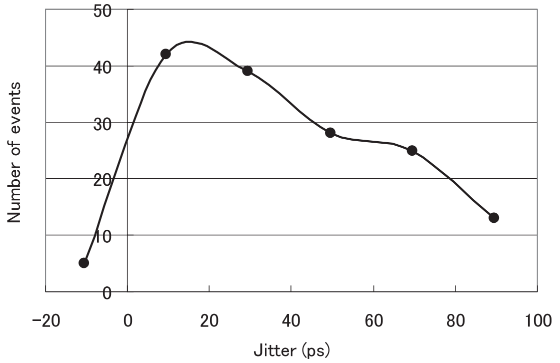

Fig. 9 Deterministic jitter histogram when Tdata = 981 ps.

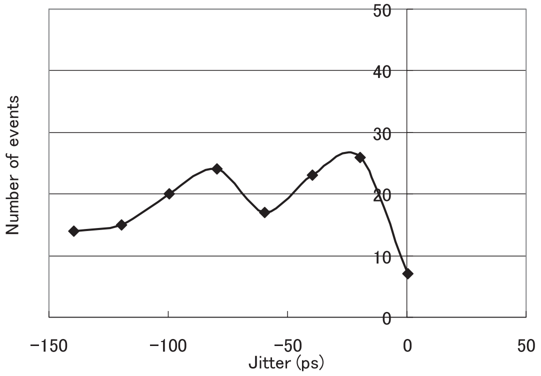

Fig. 10 Deterministic jitter histogram when Tdata = 999 ps.

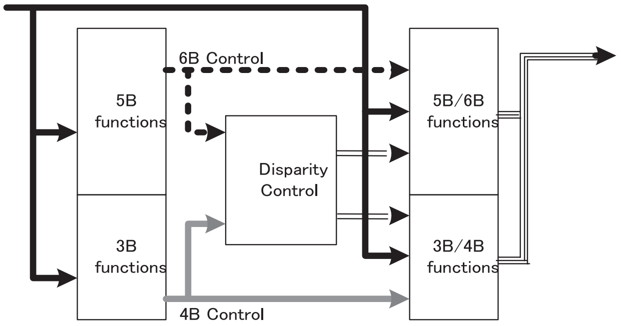

Fig. 11 Block diagram of 8B10B encoder.

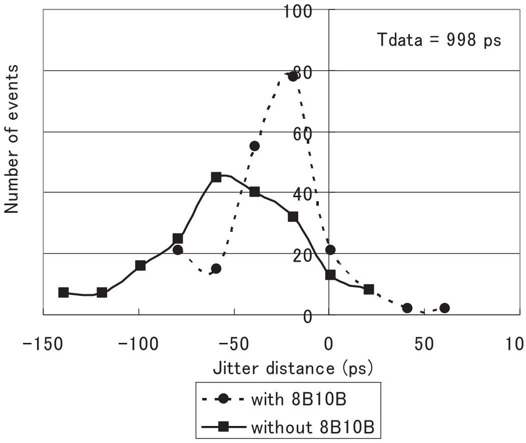

Fig. 12 Deterministic jitter histogram with and without 8B10B.

histograms with varied Tdata (980 ps to 1ns), with 1 ps increment, using the same TDC measured value (980 ps). It uses 128 clock phases. Figures 9 and 10 show the deterministic jitter histogram when Tdata is 981 ps and 999 ps, respectively.

The deterministic jitter histogram depends on the distance of the measured period with the actual data period. If Tdata is the same for the upper or the lower boundary, the deterministic jitter becomes static, as either only the lower or only the upper boundary is used constantly. This static jitter is the distance between the first sampling with the ideal location.

The deterministic jitter is proportional to the data's maximum run length. One of the ways to guarantee the data's run length is using an encoder, such as an 8B10B encoder [7]. The 8B10B encoder block diagram is shown in Fig. 11. It guarantees the maximum run length of 5 where as PRBS $2^7 - 1$ has a run length of 7.

Figure 12 shows the jitter histogram for the system with and without the 8B10B encoder. Figure 12 shows that the maximum jitter with the 8B10B encoder is smaller by 60 ps and the majority of the jitter lies closer to 0 ps by 30 ps.

4. Conclusion

The proper operation of a CDR system utilizing a TDC has been verified. The phase selection algorithm consists of simple arithmetic operations and a TDC. The system has been verified on system level in a Verilog-A environment.

The proposed system offers a faster lock time within just a few data bits (almost 7 data bits) by allowing small jitter. Additionally, by using an 8B10B encoder, the maximum jitter of the recovered clock decrease from 140 ps to 80 ps and RMS jitter of the recovered clock is 30 $\mathrm{ps_{rms}}$, assuming that the TDC resolution is 10 ps. In applications where a small jitter error can be tolerated, the proposed CDR offers the advantage of fast locking time and a small sampling error.

Acknowledgments

This work is supported by the VLSI Design and Education Center (VDEC) and the University of Tokyo in collaboration with Cadence Design Systems, Inc.

References

[1] C.R. Hogge, "A self correcting clock recovery circuit," IEEE Trans. Electron Devices, vol.3, pp.2704–2706, Dec. 1985.

[2] S.B. Anand and B. Razavi, "A 2.75 Gb/s CMOS clock recovery circuit with broad capture range," ISSCC Digest of Technical Papers, pp.214–215, Feb. 2001.

[3] T.H. Lee and J. Bulzachelli, "A 155-MHz clock recovery delay- and phase- locked loop," IEEE JSSCC, vol.27, no.12, pp.1736–1746, Dec. 1992.

[4] H.W. Jang, S.S. Lee, and J.K. Kang, "A clock recovery circuit using half-rate 4x-oversampling PD," IEEE ISCAS, vol.3, pp.2192–2195, May 2005.

[5] B. Razavi, Design of Analog CMOS Integrated Circuits, McGraw-Hill, 2001.

[6] J. Park and K. Wonchan, "An auto-ranging 50-210 Mb/s clock recovery circuit with a time-to-digital converter," ISSCC Digest of Technical Papers, pp.350–351, Feb. 1999.

[7] A.X. Widmer and P.A. Franaszek, "A DC-balanced, partitioned-block, 8B/10B transmission code," IBM J. Res. Dev., vol.27, pp.440–451, Sept. 1983.