Sub-Picosecond-Jitter Clock Generation for Interleaved ADC with Delay-Locked-Loop in 28nm CMOS

Jianping Gong, Sulin Li, John A. McNeill
Electrical and Computer Engineering Department
Worcester Polytechnic Institute, Worcester, MA 01609
Telephone: (508) 831-5567 e-mail: mcneill@ece.wpi.edu

***Abstract*—This paper presents a Delay-Locked-Loop (DLL) using a low-jitter design technique to generate sub-picosecond-jitter interleaved ADC sampling clock phases. To mitigate the effects of jitter accumulation, a low jitter delay line with digital control circuit is proposed. After 10 delay stages, the proposed DLL output can achieve <0.1psrms jitter clock. The DLL can operate with input clock frequency from 2GHz to 10GHz, enabling interleaved ADC sampling with a low-jitter sample clock over a 20GHz to 100GHz frequency range.**

I. Introduction

Recently, low-cost high-performance analog-to-digital converters (ADCs) have been widely used in wireless communication systems which require high speed and moderate resolution data conversion. Time-interleaved ADCs [1] are an attractive candidate due to better power efficiency and low frequency sample clock, and can achieve a high sampling rate through multiple sub-ADCs operating at a lower sampling rate [2]. However, time-interleaved ADCs require a precise sample clock to achieve acceptable effective-number-of-bits (ENOB), which can be degraded by jitter in the sample clock. Although several ADCs [3], [4] published to date can achieve acceptable performance with calibration of systematic clock phase mismatch [5], these work do not address the challenge of providing a low jitter sample clock for interleaved ADCs. This paper proposes a delay-locked-loop (DLL)-based method of generating a high frequency, low-jitter interleaved ADC sample clock from a low-cost, low frequency reference source.

ADC performance can be characterized by effective number of bits (ENOB), determined by the observed signal-to-noise ratio (SNR) in the frequency domain (FFT) ADC output

$$ENOB = \frac{SNR_{dB} - 1.76dB}{6.02dB} \tag{1}$$

The SNR impact of a sample clock jitter σ_J with a full-scale sinusoidal input at frequency f_{SIG} is given by

$$SNR = 20\log\left(\frac{1}{2\pi f_{SIG}\sigma_J}\right) \tag{2}$$

Equations (1) and (2) determine the required sampling clock uniformity necessary to achieve a desired ENOB. Note that ENOB can be degraded by either systematic clock errors (leading to spurs in the FFT) or random clock jitter (leading to increased noise floor in the FFT).

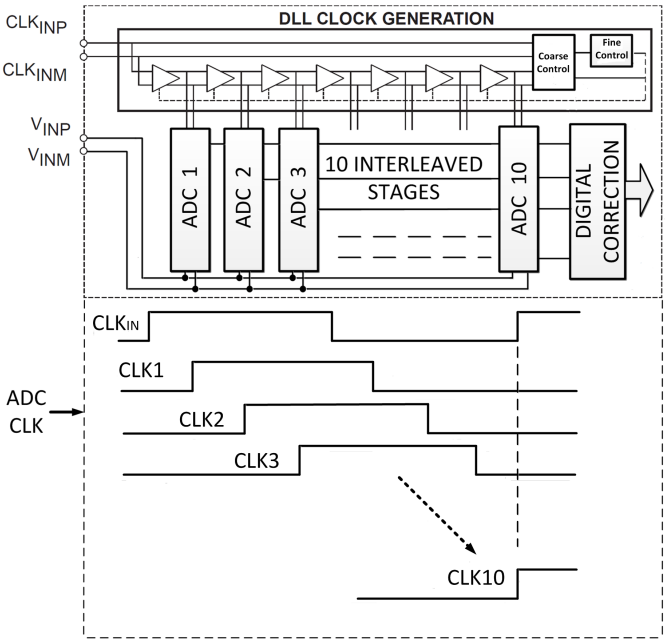

Fig. 1. Interleaved ADC system block diagram.

Both Phase-Locked-Loops (PLLs) and DLLs have been used for clock generation and recovery in communication applications [6]. Although PLLs are preferred for applications such as frequency synthesis, the DLL is an attractive option for clock phase generation because of

- its better jitter performance, since the DLL avoids the jitter accumulation [7] characteristic of the PLL, and
- the unconditional stability [8] of the DLL feedback loop which has first order loop characteristics.

The main disadvantage of DLL for generating the ADCs sample clock is the nonuniformity of the sample clocks due to systematic mismatch in the DLL phase delays. The systematic mismatch can lead to spurs in the FFT, degrading SNR.

Previous work provides techniques to solve systematic error problem through either digital-domain [1] or analog-domain [9] techniques. In [1], the effects of deterministic phase misalignment are mitigated by a time domain calibration technique, making it possible to apply the low jitter DLL to

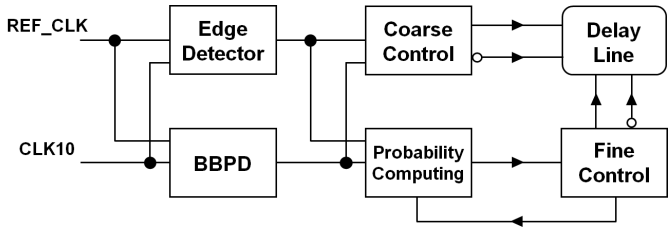

Fig. 2. Delay-Locked-Loop System Diagram.

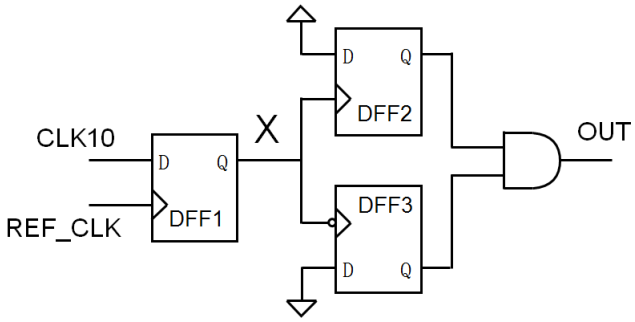

Fig. 3. Schematic of Edge Detector

generate the interleaved ADC sampling clocks.

Fig. 1 shows the system block diagram of a self-calibrating interleaved ADC using the proposed DLL output sample clock. In this work, ten DLL output clock phases connect to ten ADCs respectively; the ADCs use these clocks to sample the input analog signal. The speed of this interleaved ADC will be $10\times$ faster than each single ADC.

This paper is organized as follows: Section II provides an overview of the proposed work at a block diagram level. Section III describes in more detail the design techniques for low-jitter sampling clock generation. Section IV presents the results from the chip measurement in a 180nm CMOS. The conclusion of this paper is presented in Section V.

II. System Level Design

Figure 2 shows the system block diagram of the proposed DLL, which works to achieve phase alignment of input clock signals REF_CLK and CLK10. Coarse alignment is assisted with an edge detector, shown in Fig. 3. DFF1 is used to sample the state of CLK10 at each rising edge of the reference clock, then the sampled data will be stored at point X, which is used as the clock of DFF2 and DFF3. DFF2 is triggered by the rising edge of the voltage at X, and DFF3 is triggered by the falling edge of the voltage at X; the data input of both DFF2 and DFF3 are connected to ground. So whatever the initial state of CLK10 is, when detected CLK10 state change, either "1" to "0" or "0" to "1", one of DFF2 and DFF3 will be triggered and start working. The initial output of the AND gate is set to be "1", when DFF2 of DFF3 start working, it will sample the "0" to the input of AND gate, the output of edge detector will change to be "0".

The coarse control block is used to fast lock the reference clock and the CLK10; this method saves more time and power than the traditional direct control. In coarse control mode when the loop is converging toward lock, the output of the coarse control is the DAC control code for the delay line; this control code will change according to the output of the BBPD. Initially, the output of coarse control code is set to midscale of the binary DAC code. When the output of the BBPD is "0", it means the reference clock is faster than the CLK10, then the delay line needs to decrease the delay, the coarse control code will increase the output number to increase the current in the delay line, and the delay will decrease. Conversely, when the output of the BBPD is "1", it means the reference clock is slower than the CLK10, then the delay line needs to increase the delay, the coarse control code will decrease the output number to decrease the current in the delay line, and the delay will increase.

When the edge detector detects the CLK10 edge at rising edge of reference clock, it will send an "enable" signal to the fine control system, which contains probability computing and fine control blocks in Fig. 2. Once fine control is enabled, coarse control is disabled until the system is reset. Exact phase alignment of the REF_CLK and CLK10 signals would correspond to an equal distribution of "early" and "late" 0 and 1 signals; however exact equality is unlikely in the presence of jitter even when the DLL is locked. Rather than enforce exact equality between 0 and 1 at the BBPD output, a probabilistic lock condition for the probability of 1 at the DLL output P$\{1\}$ is defined to prevent "hunting" of the DLL loop and the associated increased jitter. Assuming a Gaussian distribution, a range of "reasonable" probability can be determined Each time fine control block changes the control code, it will collect a large number of samples of the BBPD output to estimate $\hat{P}\{1\}$. If $\hat{P}\{1\}$ falls in the reasonable region, then the fine control block decides that the reference clock and CLK10 effectively locked. After each control code change, the fine control block will collect the data from BBPD again and analyze the data to for next time fine control code change.

With a delay step resolution of proposed delay line of 0.2ps, Fig. 4 shows results of a MATLAB simulation of $\hat{P}\{1\}$ for a sample size of 64. The reference clock is chosen randomly, the horizontal axis shows the steps of changing the control number, and the vertical axis shows $\hat{P}\{1\}$ at the edge of CLK10. From Fig. 4 it shows that, with 0.2ps step resolution, a range of 15% to 85% range can be defined so CLK10 will be locked with 0.1psrms jitter.

To improve immunity to supply-induced jitter, the proposed DLL uses a differential delay line, so the input of the delay line is actually REF_CLK and $\overline{\text{REF\_CLK\_BAR}}$; the delay line has 10 delay stages, each stage with two delay cells. Details of the delay cell will be presented in section III-C.

III. Design Techniques

The schematic of each delay cell is shown in Fig. 5. The drain current of the inverter PMOS/NMOS is controlled by a current source DAC. Transistor sizing for the DAC is designed to provide a ≈ 0.2ps step resolution for the delay line. The design requirement is based on the jitter theory in [10], which

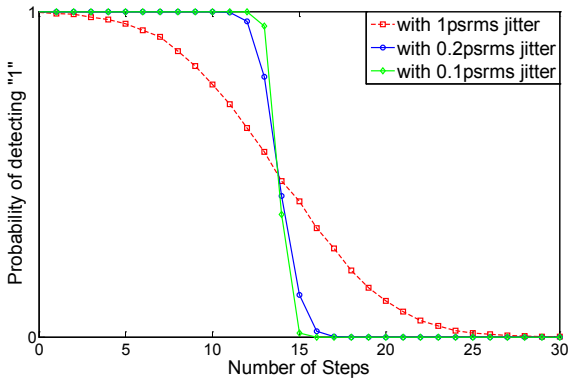

Fig. 4. Probability of detecting "1" with different values of jitter

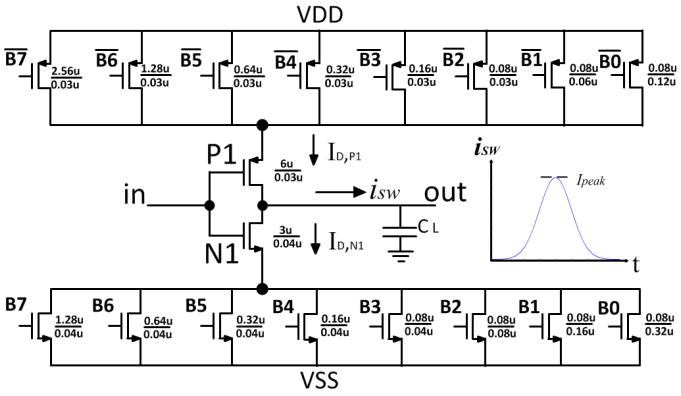

Fig. 5. Schematic of Single Delay Cell.

is covered in more detail in section III-A. These values for I_{MIN} and I_{MAX} result in the DAC resolution of 7 bits.

A. Jitter in Delay Line

According to the theory in [10], the jitter accumulates in the delay line of the DLL, and jitter is proportional to the square root of propagation delay ΔT and can be characterized by a figure of merit κ:

$$\sigma_{\Delta T} = \kappa\sqrt{\Delta T} \tag{3}$$

given independent jitter errors in each stage. For an individual delay stage with delay $\Delta T = t_{PD}$, (3) gives for the stage output jitter σ_{stage}

$$\sigma_{stage} = \kappa\sqrt{t_{PD}} \tag{4}$$

The proposed DLL has 10 delay stages, and different jitter will be observed at each output because the total delay will increase as the number of delay stages increases. Since jitters from each delay state are independent, the jitter of the nth output will add in rms fashion:

$$\sigma_{out(n)} = \sqrt{\sigma^2_{input} + \sigma^2_{stage1} + ... + \sigma^2_{stage(n)}} \tag{5}$$

Because the delay stages in the delay line are all the same, we assume the magnitude of the jitter contributed by each to be the same, that is $\sigma_{stage(i)} = \sigma_{stage}$. Thus the jitter expected at each stage output can be determined by:

$$\sigma_{out(n)} = \sqrt{\sigma^2_{input} + n\sigma^2_{stage}} \tag{6}$$

The jitter allowed for each individual delay of the $n = 10$ stage delay line, in terms of the input and output clock jitter, is given by rearranging (6) with $n = 10$:

$$\sigma_{stage} = \sqrt{\frac{1}{10}\left(\sigma^2_{out10} - \sigma^2_{input}\right)} \tag{7}$$

For a total added jitter < 0.1psec from σ_{input} to σ_{out10}, (7) implies $\sigma_{stage} < 0.03$ psec rms, giving from (4) a required $\kappa = 1E-8\sqrt{\text{sec}}$ with $t_{PD} = 0.01$nsec of each delay stage.

B. Figure of Merit κ

A major advantage of parameter κ is to quickly link system level jitter performance to circuit level considerations such as power dissipation. From equation (3), we can calculate the κ from the clock jitter and propagation delay. According to the theory in [10], κ is related to the power dissipation P_{DISS} by

$$\kappa \approx \sqrt{2}\sqrt{\frac{kT}{P_{DISS}}} \approx \sqrt{2}\sqrt{\frac{kT}{i_{SW}V_{DD}}} \tag{8}$$

where k is Boltzmann's constant and T is absolute temperature. To realize $\kappa = 1E-8\sqrt{\text{sec}}$ at T=300K with $V_{DD} = 1$V, (8) implies $i_{SW} \approx 83\mu$A. This determines the maximum required current I_{MAX} for the drain curent in delay cell in Fig. 5

C. Design of Delay Cell From The Jitter Theory

Figure 5 shows the schematic of each delay cell. From the jitter theory in sections III-A and III-B, the drain current of transistors P1/N1 should be large enough to realize a sub-picosecond jitter clock. The delay cell has 7 bits; the higher 4 bits are controlled by the coarse digital control block and the lower 3 bits are controlled by the fine digital control block. From Fig. 5 it can be seen that the size of LSB transistor is very small, ensuring that when fine control digital block is working, each resolution step is small enough for the DLL to reach phase lock. Simulation results show the DAC delay resolution during the fine control state is 0.2ps; according to Fig. 4, 0.2ps step resolution is small enough for 0.1psrms jitter clock to lock at the DLL output phases. The maximum value of the drain current of transistor P1/N1 is around 400 μA, which exceeds the requirement of 83 μA from (8), ensuring that even at smaller digital control codes, the drain current of the delay cell in Fig. 5 can still achieve the requirement of output clock jitter <0.1psrms. The proposed DLL can work in the frequency range from 2GHz to 10GHz, so the sampling frequency of interleaved ADC is $10\times$ faster than the DLL output clock, which means the interleaved ADC sampling frequency can be from 20GHz to 100GHz.

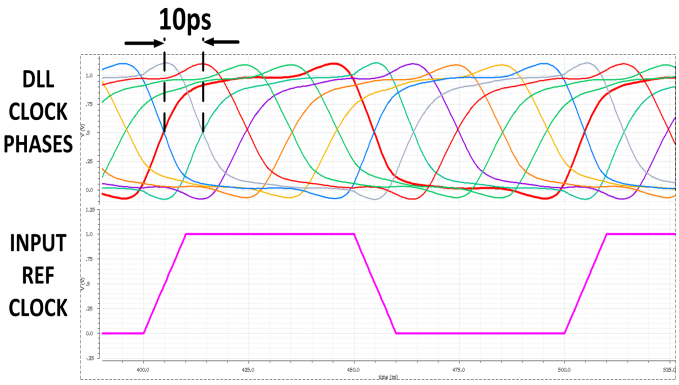

Fig. 6. Output of DLL in Locked State

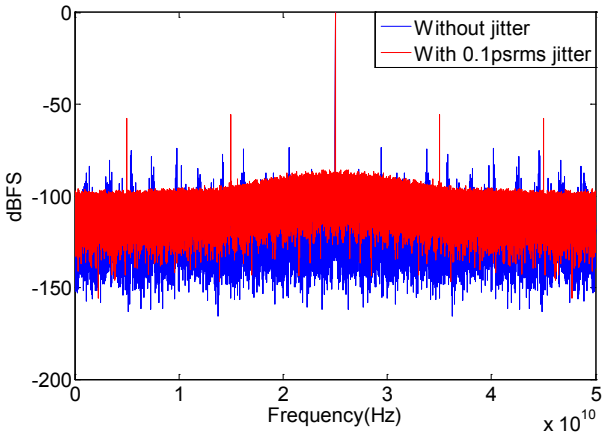

Fig. 7. FFT of ADC output with/without 0.1psrms jitter

IV. Simulation Results

The proposed Delay-Locked-Loop is designed and simulated in 28nm CMOS process. The deterministic non-uniformity of the clocks can be accommodated by the background calibration method of [1]. Simulations show that the proposed design in 28nm CMOS enables ADC speeds in the 20-100GSps range.

Fig. 6 shows the simulation result of the ADC outputs when the reference input clock is 10GHz. From Fig. 6 it can be seen that the propagation delay of two adjacent DLL outputs is 10ps, and ten DLL outputs divide one reference clock cycle on average. The 10th output clock (red waveform in Fig. 6) is locked by the reference clock.

Fig. 7 and Fig. 8 show behavioral simulation result using MATLAB. The sampling frequency of the ADC is 100GHz. Fig. 7 shows that, the jitter of sampling clock can degrade SNR of interleaved ADC. From Fig. 8 it can be seen that if the 10th stage output of DLL has 0.1psrms jitter, the ENOB will degrade from 9 to 6.4 when the input signal frequency increases.

V. Conclusion

The paper has presented a DLL-based low-jitter design technique for generation of ADC sampling clock phases from a low-cost low-frequency reference clock source. The proposed DLL can provide the interleaved ADC with 20GHz to 100GHz low jitter sampling clock. Jitter of <0.1psrms is enabled using DLL-based frequency multiplication method. The deterministic non-unifomity of the output clock are mitigated through digital background correction [1] which can be performed to calibrate interleaved ADC mismatch errors.

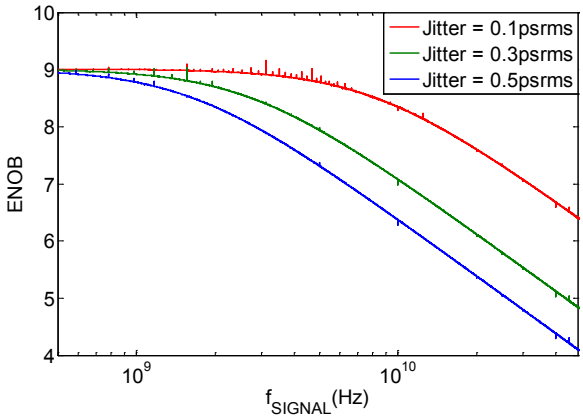

Fig. 8. ENOB vs Input Signal Frequency, f_s=100GHz

Acknowledgment

This work was supported by the National Science Foundation under grant ECCS-1232085, and by Analog Devices Semiconductor, Inc.

References

[1] J. McNeill, C. David, M. Coln, R. Croughwell, "Split ADC Calibration for All-Digital Correction of Time Interleaved ADC Errors," *IEEE Trans. Circ. Syst. II*, vol. 56 no. 5, pp. 344-348, May, 2009.

[2] I. Ku, Z. Xu, Y. Kuan, Y. Wang, M. Chang.,"A 40-mW 7-bit 2.2-GS/s Time-Interleaved Subranging CMOS ADC for Low-Power Gigabit Wireless Communications," *IEEE J. Solid-State Circuits*, vol. 47, no. 8, pp. 225-228, Aug, 2013.

[3] S. Lee, Z. Xu, A. Chandrakasan, J. Lee.,"A 1 GS/s 10b 18.9 mW Time-Interleaved SAR ADC With Background Timing Skew Calibration," *IEEE J. Solid-State Circuits*, vol. 49, no. 12, pp. 2846-2856, Dec, 2014.

[4] R. Sehgal, F. Goes, K. Bult.,"A 12 b 53 mW 195 MS/s Pipeline ADC with 82 dB SFDR Using Split-ADC Calibration," *IEEE J. Solid-State Circuits*, vol. 50, no. 7, pp. 1592-1603, July, 2015.

[5] M. Kijima, K. Ito, K. Kamei, S. Tsukamoto.,"A 6b 3GS/s Flash ADC with Background Calibration," *IEEE Custom Integrated Circuits Conf.*, pp. 283-286, Aug, 2009.

[6] R. Zhang, G. S. La Rue, "Fast Acquisition Clock and Data Recovery Circuit With Low Jitter" *IEEE J. Solid-State Circuits*, vol. 41, no. 5, pp. 10161024, May. 2006

[7] M.-J. E. Lee, W. J. Dally, T. Greer, H.-T. Ng, R. Farjad-Rad, J. Poulton, and R. Senthinathan,"Jitter transfer characteristics of delay-locked loops - theories and design techniques" *IEEE J. Solid-State Circuits*, vol. 38, no. 4, pp. 614621, Apr. 2003

[8] H.Chang, S.Liu, "A wide-range and fast-locking all-digital cycle-controlled delay-locked loop," *IEEE J. Solid-State Circuits*, vol. 40, no. 3, pp.661-670, March 2005

[9] H. Wei, P. Zhang, B. Datta Sahoo, B. Razavil, "An 8-Bit 4-GS/s 120-mW CMOS ADC," *IEEE Custom Integrated Circuits Conf.*, pp. 22-25, Sept, 2013.

[10] John A. McNeill and David Ricketts, "The Designer's Guide to Jitter in Ring Oscillators." Springer, 2009.