2-2003

# An All-Digital Clock-Smoothing Technique—Counting-Prognostication

Xiaoyi Qin
*Tsinghua University*

Hansheng Wang
*Tsinghua University*

Lieguang Zeng
*Tsinghua University*

Fuqin Xiong
*Cleveland State University*, f.xiong@csuohio.edu



## Original Citation

Xiaoyi Qin; Hansheng Wang; Lieguang Zeng; Fuqin Xiong; , "An all-digital clock-smoothing technique - counting-prognostication," Communications, IEEE Transactions on , vol.51, no.2, pp. 166- 169, Feb 2003

# AN ALL-DIGITAL CLOCK-SMOOTHING TECHNIQUE—COUNTING-PROGNOSTICATION

Xiaoyi Qin, Hansheng Wang, Lieguang Zeng, and Fuqin Xiong

## ABSTRACT

**This letter presents a novel universal all-digital clock-smoothing technique—counting-prognostication. Operation principles, performance analysis, and comparisons are given. Analysis and measurement results show that this technique can efficiently smooth jitter and wander for a wide pull-in range and pull-out range, and jitter accumulation is small. A cycle-varying counting-prognostication method, which decreases pull-in time, is also suggested.**

***Index Terms*—Clock recovery, clock synchronization, digital multiplexing, jitter.**

## I. INTRODUCTION

PHASE-LOCKED LOOPS (PLLs) are usually used for smoothing the clock. In order to filter jitter and wander with very low frequencies and very large amplitudes, the bandwidths need to be very narrow, such as several Hertz or even smaller. Since the pull-in ranges and the pull-out ranges of narrow-bandwidth PLLs are generally small (some PLLs can get wider pull-in ranges by nonlinear processing [1]–[3], but their pull-out ranges still need be narrow to reduce low-frequency jitter and wander), instantaneous large frequency or phase differences can easily make PLLs unlocked. Therefore, a tradeoff must be made between pull-out ranges and small jitters and wanders. Moreover, if the frequencies are far from the PLLs center frequencies, the pull-in time will be too long to bear. So it is unfit for the same PLL design to smooth clocks with different frequencies. PLLs use the phase differences of the two clocks to adjust voltage-controlled oscillators' (VCOs') frequencies. For large jitter of the input clock, the phase differences of the two clocks cannot reflect the input clock's frequency accurately. Thus, jitter of the input clock will be easily transferred to the output clock, and the jitter accumulation increases. Another clock-smoothing technique, the bit-leaking or adaptive bit-leaking method, proposed in [4] for reducing large jitter induced by pointer adjustments, is only applicable in synchronous digital hierarchy (SDH), and it still must be followed by an ordinary PLL. To overcome some disadvantages of PLL-based techniques, a novel universal all-digital clock-smoothing technique, the counting-prognostication method, is presented. The method is efficient for smoothing low-frequency large-amplitude jitters and wanders, and it can be applied in many areas.

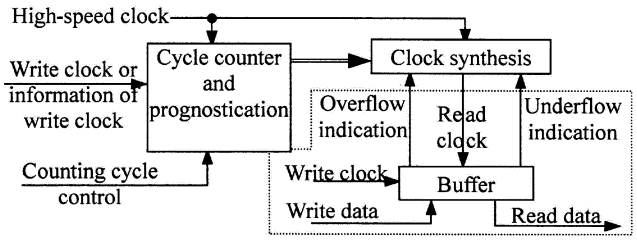


Fig. 1. Block diagram of the counting-prognostication method.

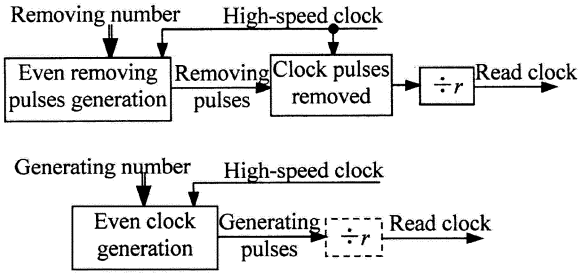


Fig. 2. Two implements of clock synthesis module.

## II. PRINCIPLES OF OPERATION

Fig. 1 is the block diagram of the counting-prognostication method. The write clock has gaps and large jitters, and is to be smoothed. The read clock is the smoothed clock with small jitters. The counting cycle control can be coherent to the write clock or the high-speed clock. The cycle counter and prognostication module examines the relationship between the write clock (or the information of the write clock) and the high-speed clock in a counting cycle, and prognosticate the number of the high-speed clock pulses to be removed or the number of the clock pulses (not necessarily the high-speed clock pulses) to be generated in the next counting cycle. In the following, we will call the number of the high-speed clock pulses to be removed the *removing number* and that to be generated the *generating number*. Commonly, in order to simplify the circuit realizations, the high-speed clock's frequency is selected slightly more than $r$ times of the read clock, to avoid adding the high-speed clock pulses. The cycle counter and prognostication module can be implemented in many ways. Generate a reference clock from the high-speed clock, which has the same nominal frequency as that of the write clock, and count the difference between the numbers of write-clock periods and the reference-clock periods in a counting cycle, then prognosticate the removing or generating number; or directly count the number of the write-clock pulses in a cycle, and the generating number in the next cycle is $r$ times the counts (since there is a $\div r$ module in the clock synthesizer, see Fig. 2), and so on. When the counting cycle is long enough, it is reasonable to assume that the clock difference in a counting cycle is approximately the same as the one in the previous counting cycle, and the previous cycle's counting results can be used to give the next cycle's prognostication numbers.

Depending on the output numbers (removing number or generating number) of the cycle counter and prognostication module, the clock synthesis module has two operation modes: removing and generating modes (Fig. 2). There are usually two strategies to assure the clock's jitter and wander performances. Evenly distributing the removing and generating pulses to the greatest extent can reduce the jitter to only one high-speed clock cycle. Supposing that the frequency of the high-speed clock is about $m$ times that of the generating pulses, then the jitter of the generating pulses is about $1/m$ UI (unit interval). Also, adding the divided-by-$r$ module after the removing and generating operation can reduce the jitter to $1/r$.

In the buffer (Fig. 1), depending on phase difference between the read clock and the write clock, an overflow indication or underflow indication is given out to advance or delay the read clock's phase. These indications are for adjusting the initial relative location between the read data and the write data in the buffer, and are generated only in initial or anomalous status [5]. Therefore, these operations have no influences on analysis of jitter and wander performances.

When the cycle counter and prognostication module adopts the first scheme, the counting-prognostication method is similar to the PLL. However, there are essential differences, described as follows. The PLL changes the read clock's frequency according to the phase differences between the write and the read clock, thus, the read clock's phase jitter can easily increase with the write clock's large momentary variations. The counting-prognostication method generates the read clock according to the frequency differences between the write clock and the reference clock. Thus, it reflects deviations of the write clock more accurately; hence, the jitter accumulation is smaller.

When the write-clock frequency is rather high, the high-speed clock frequency may be too high to be realized in application-specified integrated circuits (ASICs). This limits the maximum frequency of the read clock in this method to $f_{h\max}/k$, where $f_{h\max}$ depends on the ASIC process, and $k$ is $r$ for the removing number and $mr$ for the generating number. If $f_{h\max}$ is about 1 GHz and $k$ is 10, the maximum frequency is around 100 MHz. In order to increase the frequency limit of this method, the high-speed clock can be replaced with multiphase low-speed clocks [6]. The frequency of $N$ multiphase clocks $f_0 \sim f_{N-1}$ is near the frequency of the smoothed clock, and their phases are $1/N$ UI lagged in turn. These clocks can be generated though delay, DLL, and so on. Phases' advance and delay operations are implemented by switching the multiphase clocks, which corresponds to adding and removing operations of the high-speed clock. When switching $f_i$ into $f_{(i-1)\bmod N}$ $(0 \leqslant i \leqslant N-1)$, which is called "leading," the phases are advanced by $1/N$ UI; when switching $f_i$ into $f_{(i+1)\bmod N}$, which is called "lagging," the phases are delayed by $1/N$ UI. Increasing the times of "leading" or "lagging" can increase or decrease the frequency of the smoothed clock. Therefore, by using the multiphase clocks, this method can be used to smooth clocks of high frequency, even up to 1 GHz.

## III. Performance Analysis

Jitter and wander are the main performance measures for clock signals. In the following, jitter amplitudes are all peak-to-peak values.

### A. Jitter and Wander Performances

Jitter of the smoothed clock in the counting-prognostication method is made up of high-frequency sawtooth waveforms and low-frequency undulations. The removing and the generating operations cause high-frequency jitter. From the principles of operation, we can see that the high-frequency jitter is approximately $n_i/T$ Hz in frequency, and $1/k$ UI in amplitude, where $T$ is the counting cycle and $n_i$ is the counting result (the removing number or the generating number) of the $i$th counting cycle. In every counting cycle $T$, $n_i$ pulses are evenly removed from the high-speed clock or generated, which is then divided by $r$ to produce the smoothed read clock. Because $n_i$ in every $T$ is different from that in another $T$, there is a low-frequency undulation superimposed on the high-frequency jitter. Supposing that $f_h$ is the frequency of the high-speed clock and $f_l$ is the average frequency of the read clock, the low-frequency undulation is actually a phase difference between the read-clock frequency and $f_l$. Supposing that $\Delta\Phi_i$ is the phase increase in the $i$th counting cycle, then for the removing operation the read-clock frequency in the $i$th counting cycle is $(f_hT - n_{i-1})/(rT)$, thus, $\Delta\Phi_i = (f_hT - n_{i-1})/r - f_lT$. For the generating operation, the read-clock frequency is $n_{i-1}/(rT)$, thus, $\Delta\Phi_i = n_{i-1}/r - f_lT$. Hence, the phase of the low-frequency undulation is

$$\Phi_i(t) = \Phi_{i0} + \Delta\Phi_i \cdot \frac{t-(i-1)T}{T} \tag{1}$$

where $(i-1)T \leq t \leq iT$, $\Phi_{i0} = \Phi_0 + \sum_{j=1}^{i-1}\Delta\Phi_j$, and $\Phi_0$ is the initial phase that can be assumed as zero. As the average frequency of the read clock in all counting cycles is $f_l$, $\lim_{I\to\infty} 1/I\sum_{i=1}^{I}(f_hT - n_{i-1})/(rT) = f_l$ for the removing operation, or $\lim_{I\to\infty} 1/I\sum_{i=1}^{I} n_{i-1}/(rT) = f_l$ for the generating operation. Set $\bar{n} = \lim_{I\to\infty} 1/I\sum_{i=0}^{I-1} n_i$, and $\Delta n_{i-1} = n - n_{i-1}$ (removing operation) or $\Delta n_{i-1} = n_{i-1} - n$ (generating operation), (1) can be rewritten as

$$\Phi_i(t) = \sum_{j=1}^{i-1}\frac{\Delta n_{j-1}}{r} + \frac{\Delta n_{i-1}}{rT}[t-(i-1)T] \tag{2}$$

where $(i-1)T \leq t \leq iT$. From (2), we can deduce the spectrum expression of the low-frequency undulation

$$\Phi(f) = \sum_{i=1}^{\infty}\left[\frac{\Phi_{i0}\sin(\pi fT)}{\pi f}e^{-j2\pi f(i-1/2)T} + \frac{\Delta n_{i-1}}{4\pi^2 f^2 rT}(1 + j2\pi fT - e^{j2\pi fT})e^{-j2\pi f\cdot iT}\right]. \tag{3}$$

With the increase of the counting cycle $T$, transient phase variation of the write clock has less influence on the counting result $n_i$, and $\Delta n_i/(rT)$ gradually diminishes, thus, the spectrum amplitude of the low-frequency undulation diminishes as well. From (3), we can also conclude that the amplitude of the high-order harmonics that cause jitter decreases with the increase of the order. At the same time, the frequency of the low-frequency undulation is decreasing with $T$'s increase, so that the spectrum moves to an extremely low frequency range. Fig. 3 shows the spectrum of the undulation derived from (3) with various $T$, where $|\Delta n_i| = 9r$, $\Delta n_i = -\Delta n_{i-1}$, and $r = 25$. Therefore, to

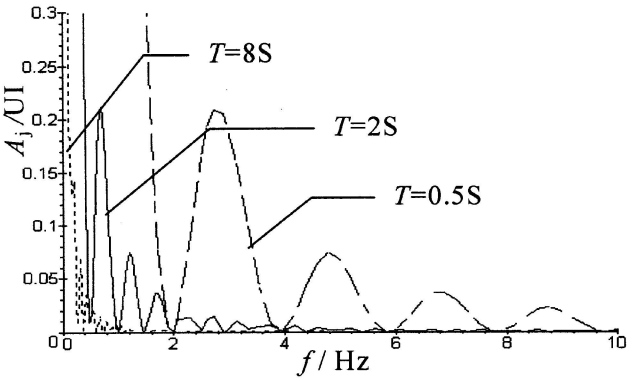


Fig. 3. Frequency spectrum of low-frequency undulation.

jitter and wander with any amplitude and frequency, we can always find $T_X$, when $T > T_X$, jitter and wander induced by the undulation are negligible. Hence, the low-frequency undulation exists in the form of wander, which has little influence on jitter. The read-clock jitter is about $1/k$ UI, which is mainly caused by the high-frequency sawtooth waveform. Comparing to jitter, wander is a phase variation whose frequency is below 10 Hz, and is expressed with maximum time-interval error (MTIE). Theoretically, by increasing $T$ infinitely, wander frequency decreases infinitely, and wander can be completely filtered, ultimately.

The counting cycle $T$ can be estimated by the wander requirement. The general requirement of wander is expressed as in (4), where $t_n$ is the wander measurement time

$$\text{MTIE} = \begin{cases} f_1(t), t \in [0, t_1) \\ f_2(t), t \in [t_1, t_2) \\ \vdots \\ f_n(t), t \in [t_{n-1}, t_n). \end{cases} \tag{4}$$

First, estimate the range of $\Delta n_i$ according to analysis of services and networks' practical situations, and get the corresponding maximum phase variation $P_{\max}$ of the read clock. Then $T$ can be selected as the value satisfying the condition $tP_{\max}/T \leqslant \text{MTIE}$. In addition, jitter and wander induced by regular and periodic leaps of the write clock can be eliminated when $T$ is selected as a multiple of the leaps' cycle.

### *B. Comparison to PLLS*

Unlike PLLs where two processes, frequency pull-in and phase lock-in, are involved in their operation, there is only the frequency pull-in process involved in the counting-prognostication method. Since the read clock can become stable after one counting cycle, the pull-in time can be as low as a counting cycle $T$, and be constant for arbitrary initial frequency error. In addition, since the counting-prognostication method uses no center frequency and prognosticates the read-clock frequency directly through counting the write-clock frequency or frequency difference, the pull-in range and the hold-in range are the same and quite wide, and are only limited by the cycle counter's width. Therefore, the same design using the counting-prognostication method can smooth clocks with a wide frequency range.

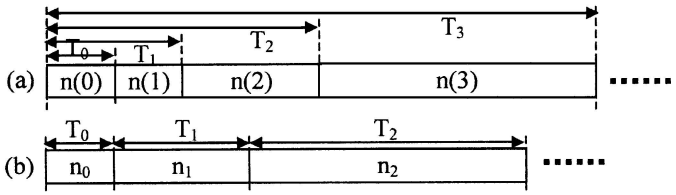


Fig. 4. Two ways of T's increase ($a = 1$).

A numeric comparison of the pull-in time to the ideal second-order PLL is given as follows. For this type of PLL, the pull-in time is $T_P = \Delta\omega^2/(2\zeta\omega_n^3)$, and the cutoff frequency of the loop filter is $\omega_c = \omega_n[2\zeta^2 + 1 + \sqrt{(2\zeta^2+1)^2+1}]^{1/2}$ [7], where $\omega_n$ and $\zeta$ are the natural frequency and damping ratio, respectively. In order to filter noise (jitter and wander) with very low frequency, such as 0.2 Hz, the cutoff frequency $f_c$ needs to be less than approximately 0.2 Hz. Selecting $\zeta$ as 0.6 that is close to producing the optimal values of $T_P$, and assuming $\omega_c = 2\pi f_c \approx 1.2$ rad/s, then $T_P \approx 3.85\ \Delta\omega^2$. For large initial frequency error, the $T_P$ will increase rapidly and intolerably. Even for a small initial frequency error of $\pm 5$ ppm to a clock of 2.048 MHz, $T_P$ is about 4.2 h, and is too large for any applications. While for the counting-prognostication method, the pull-in time does not increase for large initial frequency error. For the 9 UI jitter with a frequency of 0.2 Hz, that is, 9 UI phase leap every 5 s, and for the counting cycle $T = 8$ s, $n_i - n_{i-1}$ is $-9r$, 0, or $9r$, and $|\Delta n_i|/r$ is about 9/2. From Fig. 3, we can see that for $|\Delta n_i|/r = 9$, the jitter with frequency of 0.2 Hz is about 0.15 UI. Therefore, we can find out that $T_P = T = 8$ s is enough to reduce the 9 UI jitter to a satisfactory extent.

## IV. Varying Cycle Counting Prognostication

From the above analysis, we can see that when $\Delta n_i$ is rather large, it is necessary to enlarge the counting cycle $T$ in order to reduce the smoothed clock's jitter and wander to a sufficiently small value. Since the pull-in time is proportional to $T$, it is also quite large. For instance, when the write clock is lost and regained, too large a pull-in time may lead to the system's abnormal behavior (such as large jitter and even code errors) for a long time. The varying cycle counting-prognostication method can make the system not only get to work quickly, but also have satisfactory jitter and wander performances.

The procedure is as follows. At the beginning of frequency pull-in procedure, the counting cycle $T$ is set as $T_0$, which is the minimum time in which the read clock can be smoothed to within the maximum tolerable error. Then the counting cycle begins to enlarge. Usually for convenience of circuit realization, $T$ increases by a factor of $2^a$ for each iteration, that is, $T_i = 2^a T_{i-1}$ ($a$ is a positive integer). The increase stops until the designed maximum counting cycle, $T_{\text{Max}} = T_m = 2^{ma}\ T_0$, is reached. Fig. 4 gives out two ways $T$'s increase when $a = 1$. For Fig. 4(a), $n_0 = n(0)$, $n_1 = n(0) + n(1)$, and so on. In this method, the pull-in time $T_0$ can be very small so that the system can get to work quickly, while the time to obtain the minimum jitter and wander performances $T_M$ is enlarged. For Fig. 4(a), $T_M = 2^{ma}\ T_0$, and for Fig. 4(b), $T_M = \sum_{i=0}^{m} 2^{ia}\ T_0$. For a fixed $T_{\text{Max}}$, the larger the $a$ is, the smaller the $T_M$ is.

TABLE I
COMBINED JITTER MEASUREMENTS

| Pointer Test Sequence | Maximum Output Jitter (UI-PP) | | | |
|---|---|---|---|---|
| | B1 ($f_1$-$f_4$) | | B2 ($f_3$-$f_4$) | |
| | Requirement≤0.4 | | Requirement≤0.075 | |
| | $T$=2S | $T$=128S | $T$=2S | $T$=128S |
| Sequence A | 0.090 | 0.030 | 0.020 | 0.020 |
| Sequence B | 0.145 | 0.030 | 0.020 | 0.020 |
| Sequence C | 0.150 | 0.030 | 0.020 | 0.020 |
| Sequence D | 0.150 | 0.030 | 0.020 | 0.020 |

A: Single pointers of opposite polarity
B: Regular pointers plus one double pointer
C: Regular pointers with one missing pointer
D: Double pointers of opposite polarity

## V. PERFORMANCE MEASUREMENT RESULTS OF APPLICATIONS IN THE SDH CHIP MXLO21E1–3

The SDH chip MXLO21E1–3 developed by Tsinghua University, Beijing, China, can arbitrarily add/drop 21 E1 (2.048 Mb/s) signals to/from the bidirectional VC-4 buses. It adopts the advantages of several existing similar ASICs and enlarges the scale of circuits, and implements 21 E1 tributaries' mapping in a single chip. The desynchronizer in the receiver adopts the counting-prognostication method. Table I shows the maximum combined jitter in the experimental system measured by the TEKTRONIX CTS850 SDH test set. B1 and B2 in Table I are measurement bandpass filters defined in ITU-T recommendation G.783 [8] where $f_1 = 20$ Hz, $f_3 = 18$ kHz, $f_4 = 100$ kHz, and rise or fall by 20 dB/decade. It is obvious that the counting-prognostication method can achieve much better jitter and wander performances than ITU-T specifications.

## VI. CONCLUSION

As a universal clock-smoothing technique, the counting-prognostication method can efficiently smooth the low-frequency large-amplitude jitter and wander as demonstrated by simulation and experiment results. It has significant advantages over the PLL-based techniques: jitter accumulation is smaller, the pull-in range and the pull-out range are still quite wide when jitter and wander are very small, and it can smooth clocks with different frequencies. In addition, the varying cycle counting-prognostication method can decrease the pull-in time to the desired extent. Although the measurement results given are the application in circuit switching networks, the counting-prognostication method can also be used in package networks.

## REFERENCES


[1] K. Ng and H. Ij, "Design of ADPLL for both large lock-in range and good tracking performance," *IEEE Trans. Circuits Syst. II*, vol. 46, pp. 1192–1204, Sept. 1999.
[2] T. Y. Hsu, B. J. Shieh, and C. Y. Lee, "An all-digital phase-locked loop (ADPLL)-based clock recovery circuit," *IEEE J. Solid-State Circuits*, vol. 34, pp. 1063–1073, Aug. 1999.
[3] K. Kishine, K. Takiguchi, and H. Ichino, "2.5 Gbit/s clock and data recovery circuit IC using novel duplicated PLL technique," *Electron. Lett.*, vol. 35, pp. 360–361, Mar. 1999.
[4] H. Sari and G. Karam, "Cancellation of pointer adjustment jitter in SDH networks," *IEEE Trans. Commun.*, vol. 42, pp. 3200–3207, Dec. 1994.
[5] H. Wang and L. Zeng, "The method and equipment of regenerating tributary clocks in receiving terminals by the double-self technique of frequency prognostication," Chinese Patent 96 1 09 748.5, 1996.
[6] H. Wang, "The transmission and processing of clock synchronization in digital multiplexing system," Ph.D. dissertation, Tsinghua Univ., Beijing, China, 1998.
[7] F. Gardner, *Phaselock Techniques*, 2nd ed. New York: Wiley, 1979.
[8] "Characteristics of SDH multiplexing equipment functional block,", ITU-T Rev. Rec. G.783, 1996.