# MEASURING, MONITORING, AND MAINTAINING TIMING AT LARGE AND SMALL SCALES

A Dissertation

Presented to the Faculty of the Graduate School

of Cornell University

In Partial Fulfillment of the Requirements for the Degree of

Doctor of Philosophy

by

Enkhbayasgalan Gantsog

December 2017

MEASURING, MONITORING, AND MAINTAINING TIMING AT LARGE AND SMALL SCALES


Enkhbayasgalan Gantsog, Ph.D.

Cornell University 2017



This thesis explores techniques for measuring, monitoring and maintaining timing at small and large scales. At small scales, timing non-idealities of clock signals is of interest. As clock speeds become higher and higher in modern circuits, non-idealities such as clock duty-cycle, clock skew and jitter become proportionally large. Therefore, on-chip characterization of the clock using low power is important. A stochastic technique for on-chip measurement of such non-idealities is introduced. The technique uses a simple noisy oscillator to perform random sampling, allows easy integration in a CMOS process and is a promising alternative to direct measurement. Theoretical analysis proving the accuracy and robustness of the technique is presented. An implementation in CMOS 65nm process, occupying an active area of 0.015 $mm^2$ and consuming 0.89 mW, achieves a root mean square error of 0.1 ps and 0.31 ps in externally referenced and self-referenced jitter measurements respectively. To the best of our knowledge, the stochastic technique is the only fully on-chip jitter measurement technique that does not require post processing to obtain the jitter amplitude.

At large scales, this work explores a technique to achieve and maintain low-power synchronization of long-range peer-to-peer (P2P) RF system. Once synchronized, radio nodes can achieve significant power savings by turning off the RF front-end most of the time. Such aggressive duty-cycling allows battery operated radio to directly communicate over long range, enabling a variety of applications, such as IoT devices that do not strain the existing infrastructure and communication in natural disaster scenarios where infrastructure is unavailable. Existing synchronization techniques for narrowband radio are not scalable to large number of nodes and are often asymmetric (e.g. they require one central node that consumes high power). To solve this problem of scalable, long-range, P2P narrowband radio synchronization, a low-power signal-processor utilizing the pulse coupled oscillator (PCO) scheme for low-latency detection of syncword for aggressive duty-cycling is presented. The signal processor is insensitive to phase and frequency mismatch and compatible with commercial RF front ends. It consumes 5.1 uW in 0.01% duty-cycled mode while detecting a 63-bit syncword at 1.25 Mbps with BER=$10^{-3}$ at SNR=18.4dB.

## BIOGRAPHICAL SKETCH

Enkhbayasgalan Gantsog received the B.S. degree (highest honors) in electrical engineering with a minor in business from Lehigh University, Bethlehem, PA, in 2011. He joined Dr. Alyssa Apsel's group at Cornell University, Ithaca, NY, in 2012 to pursue the Ph.D. degree in electrical engineering.

His research interests include analog and mixed-signal circuit design in high-precision timing measurements and synchronized RF mesh-network systems.

To My Late Parents

## ACKNOWLEDGMENTS

Like many things in life, hard work alone is not enough to conduct good research and obtain a PhD degree. It also takes guidance from mentors, help from colleagues, support from friends and family, and a bit of luck. During my time at Cornell, I was fortunate enough to meet many people who directly and indirectly helped me reach this final point of my PhD career.

First and foremost, I would like to thank my advisor, Dr. Alyssa Apsel. She has taught me how to think creatively, solve open-ended problems, and learn the ins and outs of conducting research. She has also been a very supportive and understanding advisor. Knowing that my advisor would always back me when things outside of my control occur is a great feeling to have.

I would also like to extend my gratitude to my committee members, Dr. Alyosha Molnar and Dr. Edwin Chihchuan Kan. Their insights and comments on my research have been very valuable. I only wish I had sought their expertise at the earlier stage of my second research project. I am also grateful to Dr. Ehsan Afshari for teaching me the basics of analog circuit design and microwave theory and taking the time to sit on my Q exam.

I extend thanks to the past and current members of Apsel group and the other circuits groups for providing a fertile environment for doing research. Special thanks go to Ivan Bukreyev, who did a tremendous amount of work for the PCO project. Without his help, I would not have been able to finish this project. I also thank Dr. Xiao Wang for taking so much time out of his busy life to provide interesting and stimulating discussions.

I was very fortunate that my second project was in collaboration with and funded by Qualcomm. I thank Frank Lane from Qualcomm (now at Mixcomm, Inc.) for his valuable comments and discussion about the research project as well as his mentorship.

I have many friends in Ithaca and all around the world that I would like to thank for making my time at Cornell a positive experience, but it would be impossible to list all of their names here. Special thanks to Ana, Ivan Stoev, Oat and Michael (in alphabetical order) for being there during the toughest periods of my life.

Last but not least, I would like to thank my family for their love. The memories of my late father's work ethic and courage inspired me to push myself when I hit a wall. I thank my late mother for her unconditional love and support. I am also grateful to my sister for her love and encouragement. She is my angel. I thank my brother for stepping up to take the responsibility for family obligations back home. Knowing that he was there and making the best decisions, I was able to focus on my studies at Cornell.

**TABLE OF CONTENTS**

LIST OF FIGURES

LIST OF TABLES

Chapter 1

# INTRODUCTION

Precise timing is crucial in many applications. It enables high performance computing, fast data rate, efficient time division duplexing, and extended battery life through duty-cycling of the power delivered to circuitry. This thesis explores techniques for measuring, monitoring and maintaining timing at small and large scales.

At small scales, timing non-idealities of clock signal such as clock duty-cycle, clock skew and jitter are investigated and a stochastic technique for on-chip measurement of such non-idealities is introduced. At large scales, this work presents a technique to achieve and maintain low-power synchronization of long-range peer-to-peer (P2P) RF system using pulse coupled oscillators (PCO).

## *1.1 On-chip measurement of clock non-idealities*

As clock speeds increase in modern high-performance circuits, timing non-idealities, such as jitter, clock skew and unbalanced duty-cycle, become major bottlenecks of performance. On-chip measurement of these non-idealities has become increasingly important for debugging, diagnostics, calibration, and monitoring of device failure and aging. For example, delay measurement circuits [1]–[9] can be used for characterizing phase errors in track-and-hold circuits and phase interpolators. Jitter measurement circuits [10]–[16] can be used for characterizing the jitter in all-digital phase locked loops and clock-and-data recovery circuits.

Since the timing errors are small in absolute magnitude but proportionally large in high speed clocks, deterministic measurement of such errors often requires large area, high power consumption or heavy off-chip computation. Stochastic measurement using a noisy sampler is an attractive solution in this application because a simple low-precision circuit of a noisy oscillator, comparators and counters can perform high-accuracy measurement by utilizing the inherent averaging effect of the random process.

In the first part of this thesis, we present a stochastic technique for on-chip measurement of three different timing non-idealities: clock jitter, phase delay and duty cycle. This work provides theoretical analysis proving the validity and robustness of the technique in addition to demonstrations and measurement results for all three measurement types.

The next two chapters are organized as follows. Section 2.1 reviews the existing schemes for on-chip measurement of delay and jitter. Section 2.2 introduces the proposed technique first in the application of delay measurement and then in the application jitter measurement. Section 2.3 presents the theoretical analysis on the validity of the technique. Chapter 3 provides details for the circuit implementation of the technique (Section 3.1) and describes the measurements and results of the fabricated chip (Section 3.2).

### *1.2 Low-power and scalable synchronization of long-range P2P RF system*

Events such as Hurricane Sandy occasionally remind us of the fragility of our communication infrastructure. Ad-hoc discovery of peers and communication would reduce dependence on infrastructure and benefit emergency first-responders as peer-to-peer (P2P) communication is inherently less dependent on existent structures. It would also benefit other commercial applications by enabling "Internet of Things" (IoT) devices to avoid straining the existing infrastructure and utilize localized communication without sending geolocation information to the cloud, improving privacy.

Many of the capabilities of P2P networks have been demonstrated and the link management problems solved [17]. Yet long-range P2P communication is difficult to realize on battery-operated platforms (e.g. mobile). This is due in part to the high power consumption required to achieve P2P communication on such platforms without aggressive duty cycling. Duty-cycling, or turning off the power-hungry RF front-end most of the time and turning it on briefly at the right time to transmit/receive data for significant power savings, requires global synchronization, and in the case of an ad-hoc network, an "emergent" synchronous network capable of providing precise system timing in order to minimize the duration that the front end is on for.

Unlike in cellular networks, peer-to-peer nodes cannot rely on strong master-to-slave asymmetries. Since all peers have limited energy resources, none can be subjected to higher power demands of a master node. In addition, asymmetric P2P systems are typically not scalable due to the complexity that the master node must handle with increasing number of slave nodes. Conventional P2P protocols such as Bluetooth and ZigBee are examples.

One method that has been demonstrated for scalable synchronization of P2P networks is the idea of utilizing Pulse Coupled Oscillator (PCO) networks [18], [19]. Synchronization of oscillator networks has been rigorously proven to always occur in groups of any number of ideal oscillators, and requires only simple connectivity. This network was studied in less ideal conditions and shown to be useful in scalable wireless ultra-wideband (UWB) networks with pulse rates of 150 kHz [20].

In various communication schemes, it is a common practice to use a syncword to identify the start of a frame and synchronize a channel. This practice reduces interference and enables devices to identify other in-network devices without false triggers.

To achieve long-range synchronization, the correlator must be compatible with narrowband transceivers. At the same time, the entire system needs to be duty-cycled to conserve energy. Addressing both of these challenges requires consideration of several additional factors. First, the latency of detecting a syncword should be minimized in order to allow aggressive duty-cycling below 1%. To accomplish this, the radio must be able to switch on and off quickly and should not require a preamble be sent with every sync word. Therefore, the receiver must be insensitive to phase and frequency offsets by design. Second, the radio must be able to identify a syncword in an asynchronous fashion, without cross-correlation errors. Third, as the range of the network may be large, the receiver must tolerate a wide dynamic range of inputs without time misalignment or false positives. Finally, all of the above must be done with a very low power budget to enable energy harvesting or long battery lifetimes.

In this work, we study the PCO based synchronization scheme in narrowband P2P system and demonstrate a low-power signal processor for low-latency detection of a programmable syncword in wireless nodes. It is compatible with commercial RF front ends, and can enable aggressive duty-cycling for power savings in such systems

Chapter 4 presents the system level analysis of applying the PCO based synchronization in a narrowband radio. Different sequences are evaluated for the syncword. MATLAB simulation of the P2P network synchronization that takes into account the syncword processing delay and digital PCO is performed. Chapter 5 provides details for the circuit implementation of the baseband signal-processor/synchronizer block while discussing the factors that degrade the synchronization quality, such as carrier phase and frequency mismatch, varying signal power and multipath. Finally, the measurements and results of the fabricated chip and demonstration of the synchronization of 3 node system are presented.

Chapter 2

# STOCHASTIC MEASUREMENT OF CLOCK NON-IDEALITIES

## *2.1 Background*

Since measuring path delay is increasingly important in today's high-speed low-power digital circuits, there has been significant interest in high accuracy delay measurement circuits. Previous on chip delay measurement circuits can be classified into four main groups.

In ring-oscillator based circuits [1], [2], the delay path is used as part of a ring oscillator chain and oscillation frequency is measured with and without the delay path. However, the resolution of the measurement is lower. Vernier delay line based circuits [3], [4] achieve good timing resolution using small delay cells to create Vernier scale. However, measuring a long delay path requires a large circuit. Time-to-digital converter based measurements [5], [6] typically first convert the time delay to a voltage value and then use an ADC or similar circuit to obtain a digital value. For high resolution, these designs also have large area overhead.

Delay measurement circuits that use random sampling have been proposed [7]–[9], however all of these designs utilize a sampling oscillator whose frequency is modulated by a pseudo-random number generator (PRNG). Although the sampling signal should have uniform distribution with respect to the input signal transitions for linear measurement, it is not clear if PRNG based oscillator satisfies this requirement.

The technique presented here instead uses a simpler noisy VCO that generates a true random jitter and we provide analysis showing the sampling edges are drawn from uniform distribution.

Several different techniques for on-chip measurement of clock jitter have also been reported in literature. Similar to delay measurement, Vernier delay line and time-to-digital converter based circuits are also used for measuring jitter [10], [11]. A phase detector based circuit is proposed in [12]. Liang et al. measure the jitter of random data by correlating the phase detector outputs from two CDR paths [13]. Since the CDR is an integral part of the jitter measurement, this technique is not suitable for non-CDR circuits. Counter based circuits [14], [15] sweep an external clock (or the input clock itself) at several delay points to map the cumulative distribution function of the jitter. These and all of the previously reported jitter measurement techniques require off-chip post processing to obtain the jitter magnitude. The advantage of the technique proposed here is that the circuit directly outputs a digital value proportional to the jitter magnitude without the need for off-chip processing. This allows the opportunity to monitor aging/environmental effects on the clock jitter, automatically adjust control parameters for the clocking circuitry by placing the jitter measurement output in a feedback loop, and efficiently diagnose chips in volume manufacturing.

### *2.2 Proposed stochastic measurement technique*

We present the basic idea of using a noisy stochastic sampler constructed from a free running oscillator to measure delay and jitter between two clock signals on-chip.

In Section 2.3, we analyze the system and show that the error in the measurement is small. (E.g. the standard deviation of the variability in counter output for a delay equal to 0.1% of the input clock period is theoretically 0.005% of the period.)

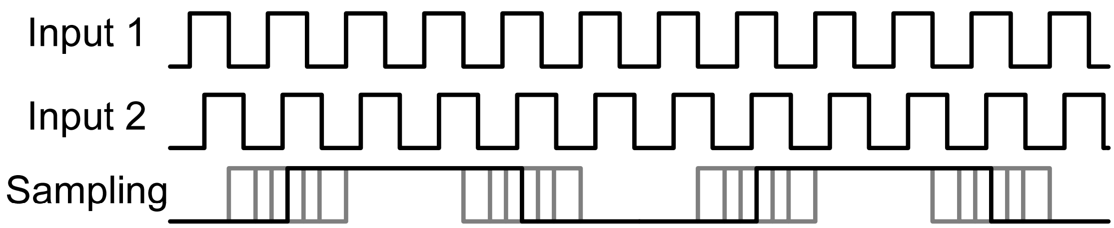



**Figure 2.1. Two input signals and an internal signal are used in the proposed technique. The internally generated sampling signal is used to check the state of the input signals at its rising edges, which fall at random locations due to high jitter (represented in gray).**

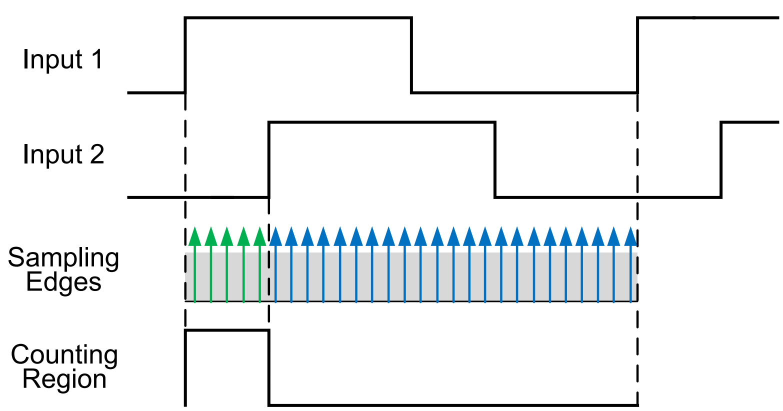



**Figure 2.2. The rising edges of the sampling signal may fall at random phases of the *unit period* of the input signals. Over many cycles of the sampling signal, its**

**edges will cover the whole unit period with fine resolution. Those edges that fall within the counting region are counted. The number of counted edges is proportional to the delay between the input signals.**

### 2.2.1 Delay and duty-cycle measurement

We begin by assuming a general delay measurement setup shown in Figure 2.1. The two input signals are assumed to be *identical* clock signals with a constant delay relative to each other. A third signal is generated internally by a free-running on-chip oscillator. We assume that this signal has extremely high jitter and runs at much slower frequency than the input clock signals. Its jitter is expected to be large enough that the transitions may fall at random phases relative to the input signals. Over large $N$ periods of the sampling signal, the transitions are expected to cover the full phase range of the "*unit period*" of the input signal with fine resolution (Figure 2.2). The transitions are then used to sample the state of the input signals at these phases. We will not call the sampling signal a clock since it is generated by a free-running oscillator and the input signals are actual clock signals.

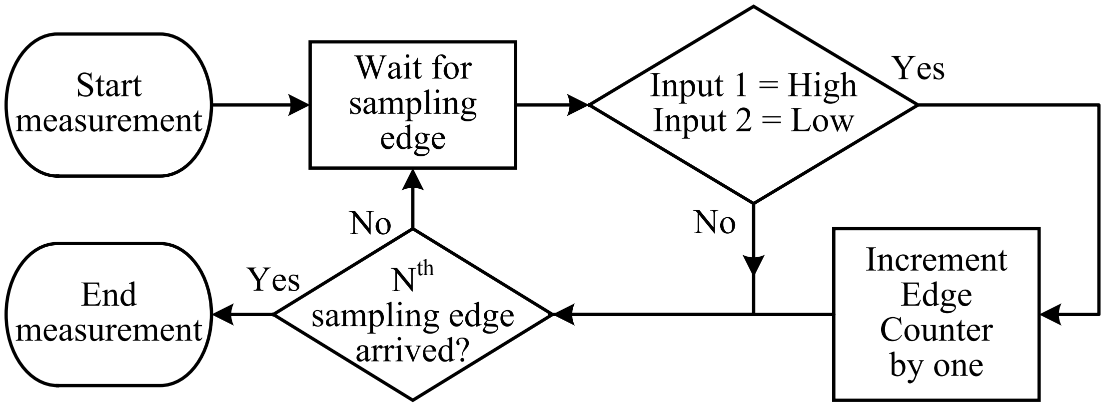



**Figure 2.3. The measurement runs for N cycles of the sampling signal. A sampling edge is counted if the first input is high while the other is low when the edge arrives. This condition defines the counting region.**

The stochastic sampling process can be described in the flow chart shown in Figure 2.3. At the rising edge of each sampling signal, the detection circuit checks if *Input 1* is high and *Input 2* is low. If that is the case, then the edge is counted, giving a measure of the delay between the input clock signals. After *N* sampling edges that sample across the full phase range of the unit period of the input signal, the number of counted edges is proportional to the delay between the input signals. When the two inputs are phase matched, the count is ideally 0 since the two inputs assume the same values at all phases. If either the first or the second input is delayed, the count is non-zero. Assuming the input clocks have 50% duty-cycle, this count can go up to half *N*, which corresponds to a delay of half a period ($T_{CLK}/2$). Increasing the delay beyond $T_{CLK}/2$ results in overlap with the previous period, reducing the counting region and thereby causing the count to decrease. The relationship between counter output and the delay is expected to be as shown in Figure 2.4.

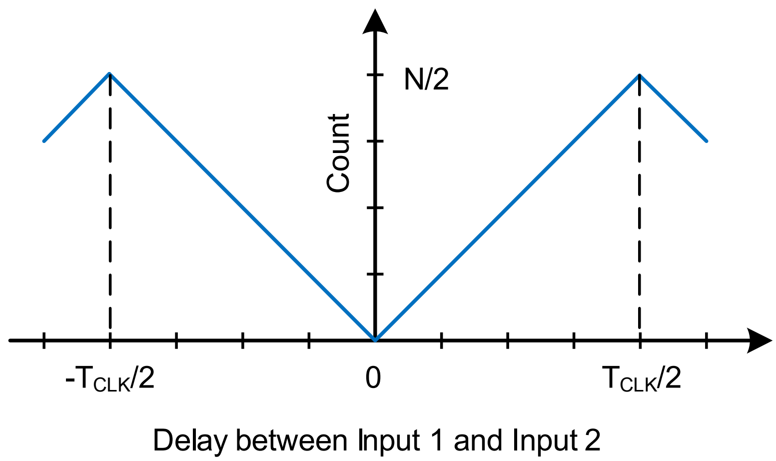



**Figure 2.4. The expected value of the measurement output has linear relationship with the delay. Assuming 50% duty-cycle in the clock signals, the count goes up to $N/2$, corresponding to a delay of half a clock period.**

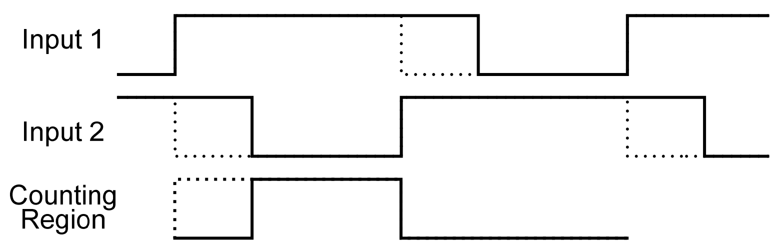



**Figure 2.5. When the duty-cycle of the input signals is not 50%, the counting region is reduced. The dotted lines represent the input signals with 50% duty-cycle and the corresponding counting region per input signal period.**

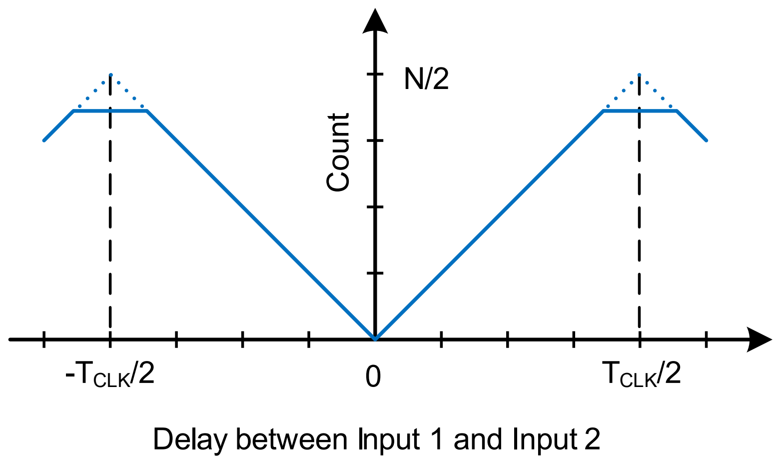



**Figure 2.6. When the duty-cycle of the input signals is not 50%, the maximum count is limited and ambiguity between the count and delay exists for large delays. The dotted lines represent the case where input signals have 50% duty-cycle.**

When the duty cycle is not 50%, the counting region is reduced (Figure 2.5). Therefore, the maximum value of the counter is clipped below $N/2$ and a corresponding ambiguity arises in the count-delay relationship (Figure 2.6). In order to eliminate this ambiguity and maximize the range of measurable delay, the circuit may incorporate a calibration knob to vary the clock duty cycle. Duty-cycle measurement of a clock is achieved by setting *Input 2* low so that the counting region is equal to the duty-cycle.

For a perfectly linear relationship between the counter output and the delay, the phases of the sampling edges should ideally be evenly distributed across the unit period of the input clock. This linear relationship could be accomplished with either an extremely fast sampling signal so that a large number of samples can be taken within a single period of the input signal, or a slower sampling signal that is perfectly locked to the input signals with a constant offset in frequency (i.e. with negligible phase noise) so that different phases of the unit period can be sampled each sampling cycle. However, the circuit implementations of both of these non-stochastic methods are prohibitively costly in terms of power and area. The technique presented in this paper uses a sampling signal where the phase of the sampling edges on the unit period of the clock signal is randomly drawn from uniform distribution (Section 2.3.2). After $N$ cycles, the sampling edges finely and nearly evenly cover the full unit period of the clock signal.

The main importance of the proposed technique is that utilizing a simple sub-sampling oscillator with large jitter allows the measurement of the delay or jitter of clock signals with high accuracy. We present the theoretical foundation of the technique in Section 2.3. Since the sampling edges fall at random phases of the unit period of the input signal, each measurement results in a slightly different counter output. Therefore, the expected value and the variance of the counter output and their relationship with the delay between the input signals are of interest. We provide this analysis in Section 2.3.3, preceded by the analysis of the probability distribution of the sampling edges in Section 2.3.2.

The on-chip oscillator that generates the sampling signal can be any kind of oscillator as long as its cycle jitter is large and Gaussian as discussed in Section 2.3.1.

This work uses a single-ended ring VCO that is designed to be extremely noisy for this purpose.

### 2.2.2 Jitter measurement

The delay measurement technique may also be used to measure the jitter of a clock signal. We assume that the jittery clock (JIT CLK), whose jitter is to be measured, is supplied to *Input 1*. A reference signal (REF CLK), which can be either an externally provided clock or an internally generated copy of JIT CLK, is supplied to *Input 2*. If REF CLK is externally supplied, it should have the same frequency as JIT CLK and negligible jitter.

The jittery clock signal and the reference signal are assumed to be phase matched and to have the same duty cycle. The circuit may incorporate a delay block for calibrating the phase match in addition to the duty-cycle calibration circuitry. If JIT CLK and REF CLK are jitter free, then the count will be 0 since they are phase matched. However, the clock jitter results in a random timing difference between the edges of the two signals every period (Figure 2.7). When the sampling edges occur during such timing differences, they are counted and the jitter is recorded.

If a copy of JIT CLK is delayed by one period and used as REF CLK, the circuit operates in self-referenced mode. In this mode, period jitter is measured as opposed to the absolute jitter in externally referenced mode.

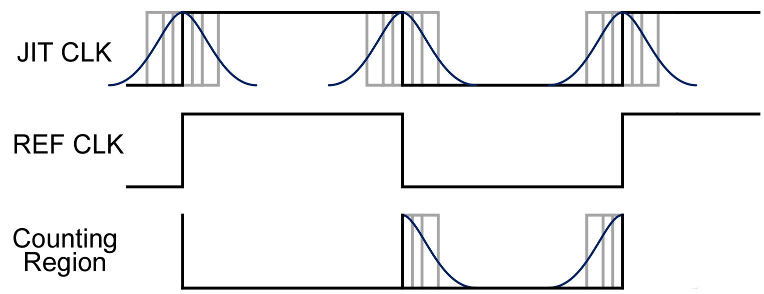



**Figure 2.7. The counting region is empty if JIT CLK and REF CLK are jitter free. Any jitter event in JIT CLK (represented in gray) causes the counting region to appear. If any sampling edge falls within the region, then it is counted. REF CLK blocks half of the jitter distribution in JIT CLK as shown.**

## *2.3 Theory*

### 2.3.1 Engineering the oscillator cycle jitter to be Gaussian

The proposed technique requires the cycle jitter (as defined in 2.3.4) of the oscillator to be large and Gaussian to guarantee a uniform distribution of the sampling edges across the unit period of the clock signal. This is achieved with a voltage controlled oscillator (VCO) that is extremely noisy. During the design of the VCO, we aimed to maximize the white Gaussian noise and minimize flicker noise because flicker noise may not be Gaussian. Therefore, thermal and/or shot noise must be maximized to be the dominant sources of the sampling signal jitter. The output signal of the VCO will accumulate noise from the supply, ground, and substrate. Hence, good substrate isolation and supply decoupling are assumed to minimize noise that may not be white Gaussian.

As the VCO accumulates random device noise over time to generate the cycle jitter, the sum of thermal noise perturbations is Gaussian distributed since thermal noise is white and Gaussian, hence independent over time. Similarly, the jitter portion due to the accumulation of shot noise is Gaussian. The sum of these two portions of different noise sources is also Gaussian as the thermal and shot noises are independent. Once the VCO edge transition is used to sample the input signals, the VCO starts accumulating device noise again, which is independent of the device noise perturbations that contributed to the jitter of the previous edge. Therefore, the cycle jitter of the VCO can be assumed to be Gaussian distributed.

### 2.3.2 From Gaussian jitter to uniform sampling

As noted in previous sections, one of the advantages of the proposed technique lies in that the VCO is designed to operate at a much lower frequency than the clock signal, but with rms jitter on the order of the clock period, $T_{CLK}$ (Figure 2.8), giving the effect of random sampling. Here we analyze the distribution of sampling edges and show that over many cycles this produces an approximation to uniform sampling.

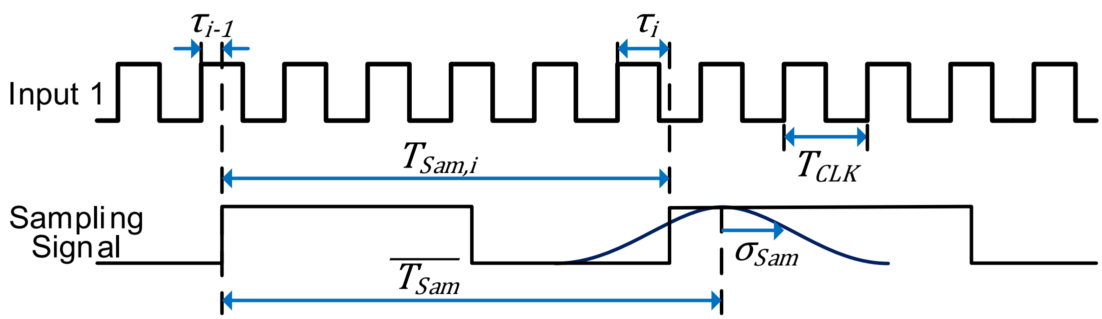



**Figure 2.8. The sampling edge delay referred to the clock unit period is the phase difference between the sampling edge and the last rising edge of the clock.**

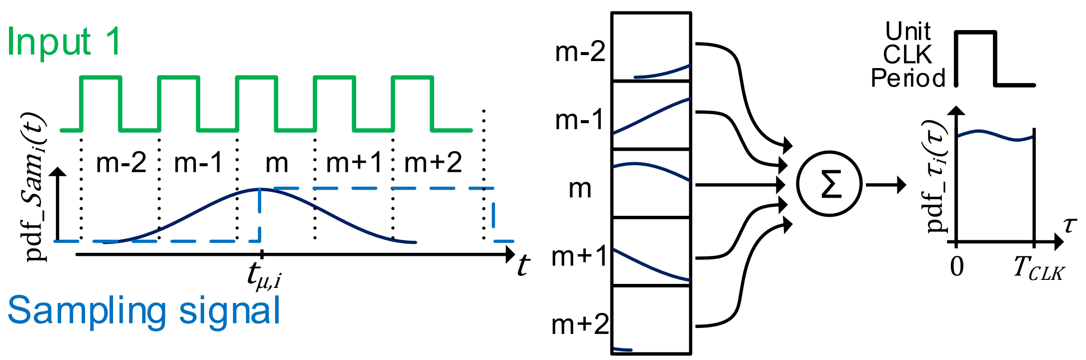



**Figure 2.9 The pdf of the sampling edge on the clock unit period is the sum of the pdf tails of the VCO cycle jitter distribution. The clock period where the sampling signal is most likely to occur is denoted *m*.**

We define $\tau_i$ as the time delay of $i^{th}$ sampling edge with respect to the preceding rising-edge of the clock signal (Figure 2.8). $\tau_i$ takes a value between 0 and $T_{CLK}$, which denotes the nominal period of the input clock signal. The period of the sampling signal, $T_{Sam,i}$, is a Gaussian random variable with mean $\overline{T_{Sam}}$ and standard deviation $\sigma_{Sam}$ due to the large jitter in the sampling signal. Therefore, $\tau_i$ is also a random variable and it can be defined as

$$\tau_i = (\tau_{i-1} + T_{Sam,i}) mod\ T_{CLK}. \tag{2.1}$$

While the location of the $i^{th}$ sampling edge is randomly drawn from a Gaussian distribution, the time delay $\tau_i$ has a different probability distribution as it is defined across the clock signal period where the sampling edge falls. Since the Gaussian distribution of the sampling clock jitter is designed to be much wider than $T_{CLK}$, the sampling edge may fall in any of the several clock periods coinciding with the Gaussian distribution (Figure 2.9). Hence, the probability density of $\tau_i$ is the sum of the probability density of the sampling edge in each of those clock periods. In other words, the probability density function (pdf) of $\tau_i$ can be derived by dividing the Gaussian pdf of the sampling edge into sections of $T_{CLK}$ and summing these sections as shown in Figure 2.9. Mathematically, the Gaussian pdf of $i^{th}$ sampling edge is represented as

$$pdf\_Sam_i(t) = \frac{1}{\sqrt{2\pi\sigma_{Sam}^2}} exp\left(-\frac{\left(t - t_{\mu,i}\right)^2}{2\sigma_{Sam}^2}\right), \tag{2.2}$$

where $t_{\mu,i} = (\tau_{i-1} + \overline{T_{Sam}})\text{mod } T_{CLK}$. Hence, the pdf of $\tau_i$ may be represented as

$$pdf\_\tau_i(\tau) = \sum_{k=-\infty}^{\infty} [pdf\_Sam_i(\tau + kT_{CLK})]$$
$$= \sum_{k=-\infty}^{\infty} \left[ \frac{1}{\sqrt{2\pi\sigma_{Sam}^2}} exp\left( -\frac{1}{2\sigma_{Sam}^2} \left(\tau + kT_{CLK} - t_{\mu,i}\right)^2 \right) \right], \tag{2.3}$$

where $\tau$ is the time variable defined over the unit period of the clock signal on the interval $(0, T_{CLK})$.

Although the theoretical calculation of the series in (2.3) is complicated, a numerical analysis shows (Figure 2.10a) that the resulting pdf of $\tau_i$ is a flat line of $1/T_{CLK}$ with one cycle of small-amplitude sinusoidal variation. The following equation has an excellent fit with the numerical data (correlation coefficient > $1 - 10^{-15}$):

$$pdf\_\tau_i(\tau) = \frac{1}{T_{CLK}} + A\,cos\left(2\pi \frac{\tau - t_{\mu,i}}{T_{CLK}}\right), \tag{2.4}$$

where $A$ is the amplitude of the sinusoidal variation and a function of $T_{CLK}$ and $\sigma_{Sam}$. When $A$ is minimized, pdf_$\tau_i$ becomes uniform distribution and the location of the previous sampling edges (e.g. $\tau_{i-1}$) does not matter. Figure 2.10b shows that $A$ gets smaller with increasing $\sigma_{Sam}/T_{CLK}$. For example, when $\sigma_{Sam}/T_{CLK}$ is larger than 0.62, $A$ normalized to $1/T_{CLK}$ is smaller than $0.001$. Hence, $\tau_i$ can be assumed to have a uniform distribution when the rms jitter of the sampling signal is on the order of the clock period, minimizing the sinusoidal variation of the pdf.

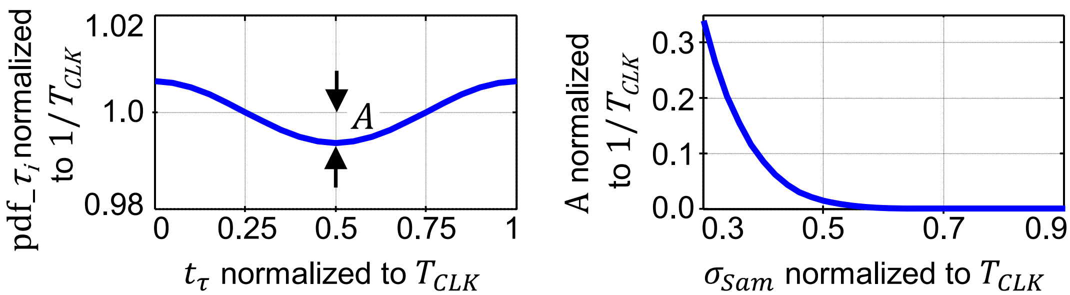



**Figure 2.10 (a) The probability density function of the location of the sampling edge with respect to the clock unit period. Both plots are obtained by numerically solving (2.3) with $\boldsymbol{\sigma_{Sam}/T_{CLK} = 0.54}$ and $\boldsymbol{t_{\mu,i} = 0}$. $\boldsymbol{A}$ is the amplitude of the sinusoidal variation on the uniform distribution. (b) The sinusoidal variation of pdf_$\boldsymbol{\tau_i}$ becomes negligible as $\boldsymbol{\sigma_{Sam}}$ grows. Hence, when the rms jitter of the sampling oscillator is on the order $\boldsymbol{T_{CLK}}$, $\boldsymbol{\tau_i}$ may be assumed to have a uniform distribution.**

The design challenge of generating large enough sampling jitter when the clock is too slow is the main limitation of the proposed technique. For applications where the clock is very slow, the random edge generator should be designed to have more jitter, such as utilizing a noisy differential ring oscillator or injecting external white Gaussian noise.

### 2.3.3 Error in uniform sampling and the output error in delay measurement mode

The previous section demonstrated that the location of a sampling edge on the unit period of the clock signal is randomly drawn from a uniform distribution. Over many cycles of the sampling signal, the locations of the sampling edges superimposed onto the unit period of the clock signal are randomly spread out, but they are not perfectly evenly distributed. If we imagine $\Delta t$ to be the time distance between the adjacent sampling edges on the unit $T_{CLK}$ period (i.e. not the edges of consecutive sampling cycles), the $\Delta t$ partitions are not all the same size and they are different for each measurement. Therefore, the number of sampling edges within the counting region is slightly different for each delay/jitter measurement. This means that, for a given delay between input signals, the counter will output slightly different numbers. Conversely, a counter output from a single measurement can only estimate the delay to be a range of values. Intuitively, the more sampling edges the measurement uses, the more accurate the measurement becomes since $\Delta t$ is reduced. In this section, we estimate the ambiguity in the counter output and calculate how many sampling cycles are required for the measurement.

One way to characterize the error due to irregular spacing of the sampling edges is to define a counting region (Figure 2.2) with width $\Delta T$ and quantify how many sampling edges fall within that partition. If we define the random variable $Y$ to describe the event where a sampling edge falls within the region $\Delta T$, then $Y$ has a

Bernoulli distribution with success probability of $p = \Delta T / T_{CLK}$ since the sampling edge has uniform distribution over $T_{CLK}$. After $N$ cycles of the sampling signal, the number of sampling edges that fall within $\Delta T$ (i.e. the value that the counter outputs) is a random variable $S$ that is defined by

$$S = Y_1 + Y_2 + \cdots + Y_N. \tag{2.5}$$

Since the sum of independent and identically distributed (i.i.d) Bernoulli distributions has Binomial distribution, $S$ has Binomial distribution with mean

$$\mu_S = Np = N \frac{\Delta T}{T_{CLK}} \tag{2.6}$$

and variance

$$\sigma_S^2 = Np(1-p) = N \frac{\Delta T}{T_{CLK}} \left(1 - \frac{\Delta T}{T_{CLK}}\right). \tag{2.7}$$

Equation (2.6) proves that the expected value of the counter output has linear relationship with the delay between the input signals as the counting region $\Delta T$ can be used to represent the delay. Equation (2.7) shows that the delay measurement error is the largest when the delay is half the clock period ($\Delta T = T_{CLK}/2$) and the error reduces as the two input signals are more aligned (e.g. as $\Delta T$ approaches 0 or $T_{CLK}$).

Since Binomial distribution with large $N$ can be approximated by Gaussian distribution, 99.7% of the time $S$ will obtain a value within $\pm 3\sigma_S$ of $\mu_S$, and the measured delay will be within $\pm 3 \frac{\sigma_S}{N} T_{CLK}$ of the actual delay 99.7% of the time. In order to choose a value for $N$, we state that the delay measurement error should be less than $T_{CLK}/200$ for the delay with the worst error (i.e. $\Delta T = T_{CLK}/2$):

$$6\frac{\sigma_S}{N}T_{CLK} \leq \frac{T_{CLK}}{200}. \tag{2.8}$$

Using (2.7), (2.8) becomes

$$6\frac{T_{CLK}}{N}\sqrt{N\frac{T_{CLK}/2}{T_{CLK}}\left(1-\frac{T_{CLK}/2}{T_{CLK}}\right)} \leq \frac{T_{CLK}}{200} \tag{2.9}$$

and after simplification

$$N \geq 360{,}000. \tag{2.10}$$

We chose $N = 360{,}000$. Note that the measurement error is smaller at other delay values. For example, for a delay equaling 1% of the clock period, the measurement error is smaller than $T_{CLK}/1005$, 99.7% of the time.

### 2.3.4 Theoretical description of the jitter measurement process

Jitter can be described as absolute jitter, period jitter and cycle jitter among many others. Since there are conflicting definitions of the jitter terminology in literature, we define our terminology here.

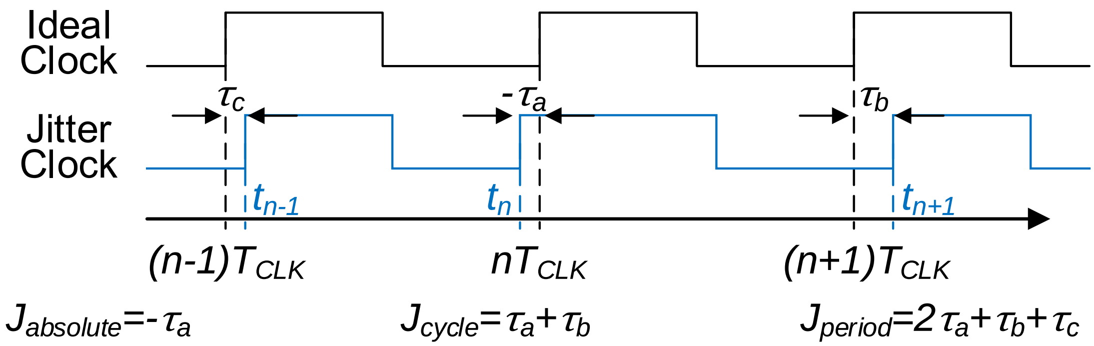



**Figure 2.11 Jitter terminology used in this thesis.**

Denoting the nominal period of the clock $T_{CLK}$ and the transition time of the $n^{th}$ rising edge of the clock $t_n$, we define the absolute jitter as the time difference between the actual transition time of the edge and its ideal time (Figure 2.11),

$$t_n - nT_{CLK}. \tag{2.11}$$

We define the cycle jitter to be the difference between the transition times of the adjacent rising edges of the clock compared to the nominal period $T_{CLK}$, i.e.

$$(t_{n+1} - t_n) - T_{CLK}. \tag{2.12}$$

In other words, the cycle jitter is deviation of the actual period from the nominal period.

We define the period jitter to be the difference between adjacent periods, i.e.

$$(t_{n+1} - t_n) - (t_n - t_{n-1}) = t_{n+1} - 2t_n + t_{n-1}. \tag{2.13}$$

Since the counting region in the delay measurement mode is deterministic, the counter output measures the proportion of the sampling edges that fall within that region. However, the counting region in the jitter measurement mode is probabilistic due to the clock jitter (Figure 2.7) and is different for each sampling edge. Therefore, the previous analysis describing delay measurement does not apply to how the clock jitter is measured. A more general way to look at the jitter measurement mode is that since the sampling edge may fall at any location on the unit period of the clock signal, it can sample any part of the clock jitter probability distribution. Even for a measurement with a single sampling edge (i.e. $N$=1), the count result is still representative of the amount of clock jitter because if the count is 1 instead of 0, it is more likely that the clock jitter is large rather than small. The result is highly unreliable for $N$=1 with large variance in the output count, but raising $N$ to a large number increases the confidence in the measurement accuracy and reduces the random error of the count.

More specifically, the probability that a sampling edge at $\tau$, which is defined in Section 2.3.2, is counted equals the cdf of the clock jitter at $\tau$ ($p = \text{cdf\_}CLK(\tau)$) because the sampling edge is counted as long as the clock jitter causes JIT CLK to be high at $\tau$ while REF CLK is low. Hence, the random variable $X$ describing this event is Bernoulli distributed, $X \sim Ber(\text{cdf\_}CLK(\tau))$, where the success, or 1, denotes the sampling edge being counted.

The mean of $X$, or the probability that a sampling edge at *any* point on the unit period of the clock signal is counted, is

$$
\begin{aligned}
\mu_X &= E[X] = 1^2 Pr(X=1) + 0^2 Pr(X=0) = Pr(X=1) \\
&= \int_0^{T_{CLK}} P(X|\tau) d\tau = \int_0^{T_{CLK}} cdf\_CLK(\tau)\, pdf\_\tau(\tau) d\tau \\
&= \int_0^{T_{CLK}} cdf\_CLK(\tau) \frac{1}{T_{CLK}} d\tau,
\end{aligned}
\tag{2.14}
$$

where pdf_$\tau(\tau) = 1/T_{CLK}$ describes the uniform distribution of the sampling edge on the unit period of the clock signal as described in Section2.3.2. Since the sampling signal jitter and the clock signal jitter are independent, $P(X|\tau) =$ cdf_$CLK(\tau)$pdf_$\tau(\tau)$.

Similarly,

$$
E[X^2] = 1 \cdot Pr(X=1) + 0 \cdot Pr(X=0) = E[X]. \tag{2.15}
$$

The standard deviation of $X$ is thus

$$
\sigma_X = \sqrt{Var[X]} = \sqrt{E[X^2] - E[X]^2} = \sqrt{\mu_X(1-\mu_X)}. \tag{2.16}
$$

After $N$ cycles of the sampling signal, the number of counted sampling edges is a random variable $C$ that is defined by $C = X_1 + X_2 + \cdots + X_N$. Since the sum of i.i.d Bernoulli distributions has Binomial distribution, $C$ has Binomial distribution with mean $\mu_C = N\mu_X$ and variance $\sigma_C^2 = N\mu_X(1-\mu_X)$. For a successful jitter measurement, the expected value of the counter output $\mu_C$ should be proportional to the rms jitter of the clock and the random error of the measurement, which is indicated by

$$\frac{\sigma_C}{N} = \sqrt{\frac{1}{N}\mu_X(1-\mu_X)}, \tag{2.17}$$

should be low. The exact values of the mean and the standard deviation of the counter output can be analytically estimated when the clock jitter distribution is known. The upper bound of (2.17) is when $\mu_X = 0.5$, which is a very unlikely case since it implies extremely large clock jitter, such as a deterministic jitter with peak to peak amplitude of more than a clock period. Therefore, the standard deviation of the jitter measurement with $N$=360,000 is much less than

$$\frac{\sigma_C}{N}T_{CLK} \ll \sqrt{\frac{1}{N}\frac{1}{2}\left(1-\frac{1}{2}\right)}T_{CLK} = \frac{T_{CLK}}{1200}, \qquad N = 360{,}000. \tag{2.18}$$

Hence, even without knowing the exact distribution of the jitter, the measurement resolution is estimated to be sub-picosecond.

The total jitter of a digital signal is composed of two types of jitter: random jitter and deterministic jitter. The random jitter is often assumed to be Gaussian since Gaussian noise processes inherent in devices, such as thermal noise and shot noise, are the major sources of random jitter. In addition, the accumulation of many different independent noise sources will result in a Gaussian random jitter by the central limit theorem.

The deterministic jitter may be further classified into data-dependent jitter and uncorrelated jitter. The data-dependent jitter is due to inter-symbol interference or duty cycle distortion in the data signal. Since this work focuses on the measurement of clock jitter, the analysis of data-dependent jitter is not considered. The uncorrelated jitter may be due to the coupling from other uncorrelated signals, such as an external clock or a periodic signal.

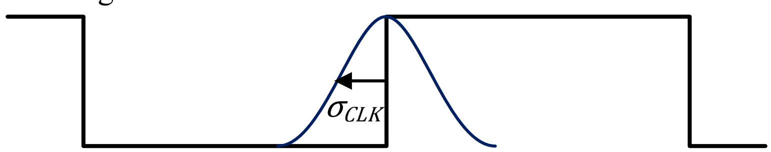



**Figure 2.12 Clock signal with random (Gaussian) jitter. Its rms/standard deviation value is represented as $\boldsymbol{\sigma_{CLK}}$.**

### 2.3.5 Measurement of clock jitter

When the clock jitter is Gaussian distributed with 0 mean and standard deviation $\sigma_{CLK}$ (Figure 2.12), the mean and standard deviation of the counter output can be calculated. Neglecting the falling edge jitter of the clock signals and focusing only on the rising edge jitter,

$$\mu_X = \int_{-T_{CLK}}^{0} cdf\_CLK(\tau) \frac{1}{T_{CLK}} d\tau = \int_{-T_{CLK}}^{0} \frac{1}{2}\left(1 + erf\left(\frac{\tau - 0}{\sqrt{2\sigma_{CLK}^2}}\right)\right) \frac{1}{T_{CLK}} d\tau$$

$$= \frac{1}{2} + \frac{1}{\sqrt{2\pi} T_{CLK}} \left(1 - e^{-\frac{T_{CLK}^2}{2\sigma_{CLK}^2}}\right) \sigma_{CLK} + \frac{1}{2} erf\left(-\frac{T_{CLK}}{\sqrt{2\sigma_{CLK}^2}}\right). \tag{2.19}$$

Assuming the clock jitter is small compared to its period $(T_{CLK} \gg \sigma_{CLK})$, $\mu_X$ is approximately

$$\mu_X \approx \frac{1}{\sqrt{2\pi}} \frac{\sigma_{CLK}}{T_{CLK}}. \tag{2.20}$$

The falling edge jitter doubles the probability of a VCO edge being counted, hence

$$\mu_X \approx \frac{2}{\sqrt{2\pi}} \frac{\sigma_{CLK}}{T_{CLK}}. \tag{2.21}$$

The mean and the standard deviation of the counter output are

$$\mu_C = N\mu_X \approx \frac{2N}{\sqrt{2\pi}} \frac{\sigma_{CLK}}{T_{CLK}}. \tag{2.22}$$

$$\sigma_C = \sqrt{N\mu_X(1 - \mu_X)} = \sqrt{N \frac{2}{\sqrt{2\pi}} \frac{\sigma_{CLK}}{T_{CLK}} \left(1 - \frac{2}{\sqrt{2\pi}} \frac{\sigma_{CLK}}{T_{CLK}}\right)}. \tag{2.23}$$

Equation (2.22) shows that the expected value of the counter output is linear with the clock jitter normalized to the period. This result is verified numerically by simulating the measurement in MATLAB, where a sampling signal (running at 400 MHz with 54 ps rms jitter) and clock signal with varying amounts of jitter are provided. Figure 2.13 shows the mean and standard deviation of the counter output for 6 GHz and 10 GHz clock simulations.

When the clock jitter has a bimodal distribution with two Gaussian random jitter distributions of $\sigma_{Rand}$ separated by deterministic $\tau_{\delta\delta}$ distance (Figure 2.14), it can be shown, see appendix, that the expected value of the counter output is

$$\mu_C \approx \frac{2N}{\sqrt{2\pi}} \frac{\sigma_{Rand}}{T_{CLK}} \left( \frac{\sqrt{2\pi}}{2} \frac{\tau_{\delta\delta}}{\sigma_{Rand}} erf\left( \frac{1}{\sqrt{2}} \frac{\tau_{\delta\delta}}{\sigma_{Rand}} \right) + e^{-\frac{1}{2}\frac{\tau_{\delta\delta}^2}{\sigma_{Rand}^2}} \right), \tag{2.24}$$

and for periodic jitter with peak deviation $\tau_{PJ}$, it is

$$\mu_C = \frac{2N}{\pi T_{CLK}} \tau_{PJ}. \tag{2.25}$$

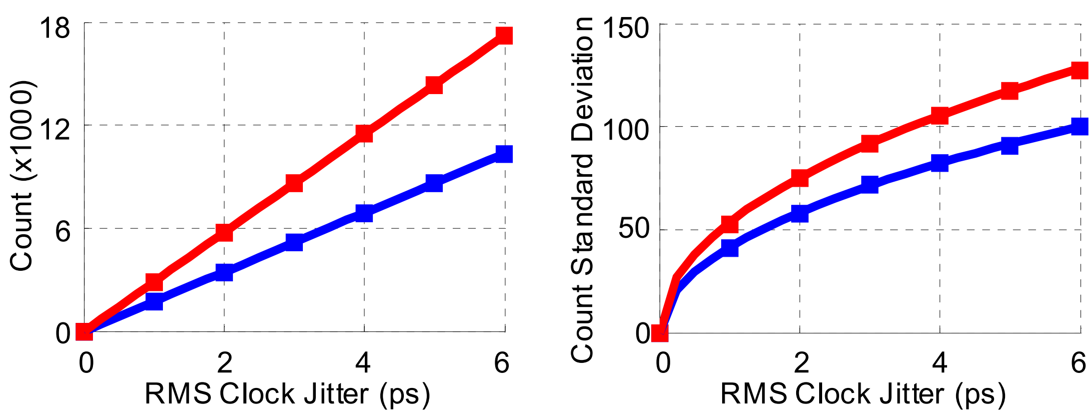


**Figure 2.13 Mean and standard deviation of the counter output for the simulations of 6 GHz (top/red) and 10 GHz (bottom/blue) clocks with random jitter. The simulated results (square points) match the theoretical estimates (lines) provided by (2.22) and (2.23).**

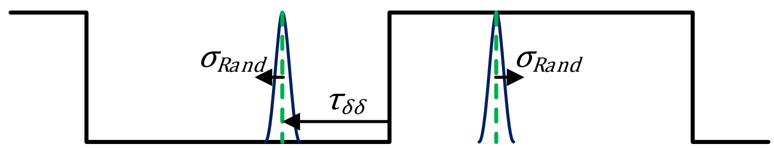



**Figure 2.14 Clock signal with bimodal/Dirac distributed deterministic jitter and small random (Gaussian) jitter. The distance between the ideal clock edge and the Dirac peaks shown in dotted lines is $\tau_{\delta\delta}$.**

Equation (2.24) shows that the expected value of the counter output is linear to the random jitter ($\sigma_{Rand}$) or the deterministic jitter ($\tau_{\delta\delta}$) when either of them dominates. Similarly, a linear relationship is shown by (2.25) for periodic jitter. The standard deviation of the counter output for bimodal and period jitter is also presented in the appendix.

### 2.3.6 Measurement error due to the internal noise of the circuit and the REF CLK jitter

The internal noise of the jitter/delay measurement circuit causes the sampling signal to sample *Input 1* and *Input 2* at slightly different times (excluding the deliberate delay using the delay block). Therefore, the edges of the JIT CLK and REF CLK will effectively have random relative variations and the counter output will be non-zero even if the input signals have no jitter. As the measurement circuit is designed to be isolated from supply, substrate and coupling noise, the internal noise is dominated by device noise and assumed white Gaussian. Since the input clock is external to the measurement circuit, the clock jitter and the sampling jitter are independent. Therefore, the random jitter that the circuit measures has standard deviation

$$\sigma_{Total} = \sqrt{\sigma_{Rand}^2 + \sigma_{INT}^2}, \tag{2.26}$$

where $\sigma_{INT}$ is the sampling jitter due to the internal noise of the measurement circuit.

When the deterministic jitter is negligible, the expected value of the counter output is

$$\mu_C \approx \frac{2N}{\sqrt{2\pi}} \frac{\sqrt{\sigma_{Rand}^2 + \sigma_{INT}^2}}{T_{CLK}}. \tag{2.27}$$

In order to get a linear relationship with the rms clock jitter, the internal jitter must be minimized. In addition, the internal jitter also causes the counter output to have slightly more variance ($\sigma_C^2$) since $\sigma_C$ increases with $\mu_C$. However, the measurement accuracy is still much better than the limit obtained in (2.18) because $\mu_C/N$ is still much smaller than the $\mu_X = 0.5$ assumed for the limit.

When the deterministic jitter is the dominant source of the clock jitter, it needs to be larger than both the random jitter of the clock as well as the internal jitter in order for the measurement to be linear with the deterministic jitter.

If REF CLK has jitter, the circuit measures the "relative" jitter between JIT CLK and REF CLK. In other words, the circuit measures how much the edges of JIT CLK vary with respect to the jittery edges of REF CLK. If REF CLK and JIT CLK have correlated jitter, such as when they come from the same source, the correlated component of the jitter is ignored by the measurement circuit. Therefore, the measurement of $\sigma_{INT}$ during calibration does not require the input clock signals to be jitter free. *Input 1* and *Input 2* can be shorted, thereby eliminating the clock jitter that can be measured by the circuit. The measurement results presented in Section 3.2 are adjusted for $\sigma_{INT}$ using this method.

If REF CLK jitter in externally referenced mode is uncorrelated with JIT CLK jitter, then REF CLK jitter is indistinguishable from the internal jitter. In other words, the standard deviations add in power similar to (2.26).

In self-referenced mode where REF CLK is one period delayed version of JIT CLK, the period jitter is measured since successive clock edges are measured.

### *2.4 Appendix*

### 2.4.1 Counter output when the clock jitter has bimodal distribution

In addition to random jitter, deterministic jitter can be a considerable source of timing uncertainty in clock signals. The simplest model for a deterministic jitter is the dual Dirac model, which assumes a bimodal distribution of jitter as shown in Figure 2.14. The clock edges are assumed to transition at either of two fixed positions from the ideal point due to a deterministic noise source such as coupling from an external clock signal. The timing of the edges may further vary due to random jitter.

The dual Dirac model does not perfectly represent the actual deterministic jitter in real life, but it is nonetheless widely accepted in industry [23]. Therefore, we analyze in this subsection the quality of the jitter measurement assuming the clock jitter fits the dual Dirac model. A more realistic model with periodic jitter is analyzed in Appendix 2.4.2.

Two jitter variables are considered in the model: the standard deviation of the Gaussian random jitter, $\sigma_{Rand}$, and the deviation of the deterministic Dirac displacement from the ideal position, $\tau_{\delta\delta}$. The cdf of the jitter is

$$cdf_{CLK(\tau)} = \frac{1}{4}\left(1 + erf\left(\frac{\tau - \tau_{\delta\delta}}{\sqrt{2\sigma_{Rand}^2}}\right)\right) + \frac{1}{4}\left(1 + erf\left(\frac{\tau + \tau_{\delta\delta}}{\sqrt{2\sigma_{Rand}^2}}\right)\right). \qquad (2.28)$$

Similar to the analysis in Section 2.3.5, the probability that a sampling edge is counted is

$$
\begin{aligned}
\mu_X &= 2\int_{-T_{CLK}}^{0} cdf\_CLK(\tau)\frac{1}{T_{CLK}}d\tau \\
&= 2\int_{-T_{CLK}}^{0}\left(\frac{1}{4}\left(1+erf\left(\frac{\tau-\tau_{\delta\delta}}{\sqrt{2\sigma_{Rand}^2}}\right)\right)\right. \\
&\qquad\left.+\frac{1}{4}\left(1+erf\left(\frac{\tau+\tau_{\delta\delta}}{\sqrt{2\sigma_{Rand}^2}}\right)\right)\right)\frac{1}{T_{CLK}}d\tau \\
&= 1+\frac{1}{2\kappa}\left(2\gamma\, erf(\gamma)+2\frac{e^{-\gamma^2}}{\sqrt{\pi}}+(\kappa+\gamma)\, erf(-\kappa-\gamma)\right. \\
&\qquad\left.-\frac{e^{-(\kappa+\gamma)^2}}{\sqrt{\pi}}-(-\kappa+\gamma)\, erf(-\kappa+\gamma)-\frac{e^{-(-\kappa+\gamma)^2}}{\sqrt{\pi}}\right),
\end{aligned}
\tag{2.29}
$$

where

$$
\kappa=\frac{1}{\sqrt{2}}\frac{T_{CLK}}{\sigma_{Rand}} \tag{2.30}
$$

and

$$
\gamma=\frac{1}{\sqrt{2}}\frac{\tau_{\delta\delta}}{\sigma_{Rand}} \tag{2.31}
$$

Assuming the clock period is much larger than the jitter (i.e. $T_{CLK} \gg \sigma_{Rand}$ and $T_{CLK} \gg \tau_{\delta\delta}$), (2.29) becomes

$$\mu_X \approx \frac{1}{\sqrt{\pi}\kappa}\left(\sqrt{\pi}\gamma\, erf(\gamma) + e^{-\gamma^2}\right). \tag{2.32}$$

The expected value of the counter output is

$$\mu_C = N\mu_X \approx \frac{N}{\sqrt{\pi}\kappa}\left(\sqrt{\pi}\gamma\, erf(\gamma) + e^{-\gamma^2}\right). \tag{2.33}$$

The plot of the expression inside the parentheses in (2.33) with respect to $\gamma$ is shown in Figure 2.15. If the deterministic jitter is larger than the random jitter (i.e., $\gamma > 1$), (2.33) becomes

$$\mu_C \approx \frac{N}{\sqrt{\pi}\kappa}\sqrt{\pi}\gamma = \frac{\tau_{\delta\delta}}{T_{CLK}}N, \tag{2.34}$$

which means the expected value of the counter output is linear with the deterministic jitter. For small values of $\gamma \ll 1$, (2.33) can be approximated using Taylor expansion as

$$\mu_C \approx \frac{N}{\sqrt{\pi}\kappa}(1+\gamma^2) = \frac{N}{\sqrt{\pi}T_{CLK}}\left(\sqrt{2}\sigma_{Rand} + \gamma\tau_{\delta\delta}\right), \tag{2.35}$$

which shows that the counter output is a measure of both random and deterministic jitter. If the random jitter is the dominating source of the total clock jitter ($\gamma \approx 0$), then

(2.33) matches (2.22) obtained in Section 2.3.5. MATLAB simulations numerically verifying this result is shown in Figure 2.16.

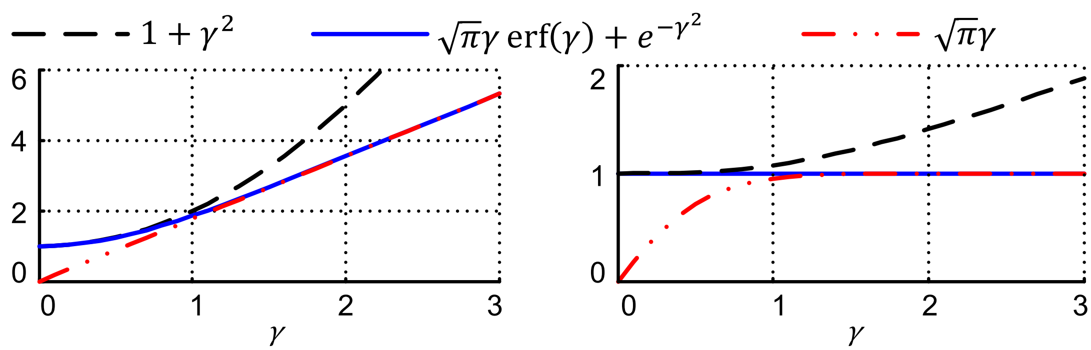



**Figure 2.15 The expression in parentheses in (2.33) and its approximations at small and large $\gamma$ values (left). The same plot normalized to the expression (right).**

### 2.4.2 Counter output when the clock jitter is periodic

An example of a clock with periodic deterministic jitter is analyzed in this subsection. The total clock jitter, as shown in Figure 2.17, is the convolution of Gaussian random jitter and periodic deterministic jitter. Assuming the periodic jitter is single tone, its pdf can be modeled as

$$pdf\_PJ(\tau) = \begin{cases} \dfrac{1}{\pi\sqrt{\tau_{PJ}^2 - \tau^2}} & |\tau| < \tau_{PJ} \\ 0 & |\tau| \geq \tau_{PJ} \end{cases}, \tag{2.36}$$

where $\tau_{PJ}$ is the peak deviation of the periodic jitter from the ideal location. Since calculating the convolution mathematically is infeasible, we calculate $\mu_X$ assuming the random jitter is negligible. The cdf of the periodic jitter is

$$cdf\_PJ(\tau) = \begin{cases} 0 & \tau \leq -\tau_{PJ} \\ \dfrac{1}{\pi}\left(arcsin\dfrac{\tau}{\tau_{PJ}} + \dfrac{\pi}{2}\right) & |\tau| < \tau_{PJ}. \\ 1 & \tau \geq \tau_{PJ} \end{cases} \tag{2.37}$$

Hence, we obtain

$$\begin{aligned} \mu_X &= 2\int_{-T_{CLK}}^{0} cdf\_PJ(\tau)\frac{1}{T_{CLK}}d\tau = 2\int_{-T_{CLK}}^{0}\left(\frac{1}{\pi}arcsin\frac{\tau}{\tau_{PJ}} + \frac{1}{2}\right)\frac{1}{T_{CLK}}d\tau \\ &= \frac{2}{\pi T_{CLK}}\tau_{PJ}. \end{aligned} \tag{2.38}$$

The expected value of the counter output is

$$\mu_C = N\mu_X = \frac{2N}{\pi T_{CLK}}\tau_{PJ} \tag{2.39}$$

and it has linear relationship with the periodic jitter amplitude. This result is verified numerically using MATLAB simulations (Figure 2.18). The case when the random jitter is non-negligible is also shown in Figure 2.18. Similar to the jitter with dual Dirac distribution, $\mu_C$ is linear to $\sigma_{Rand}$ if the random jitter is much larger than the deterministic jitter.

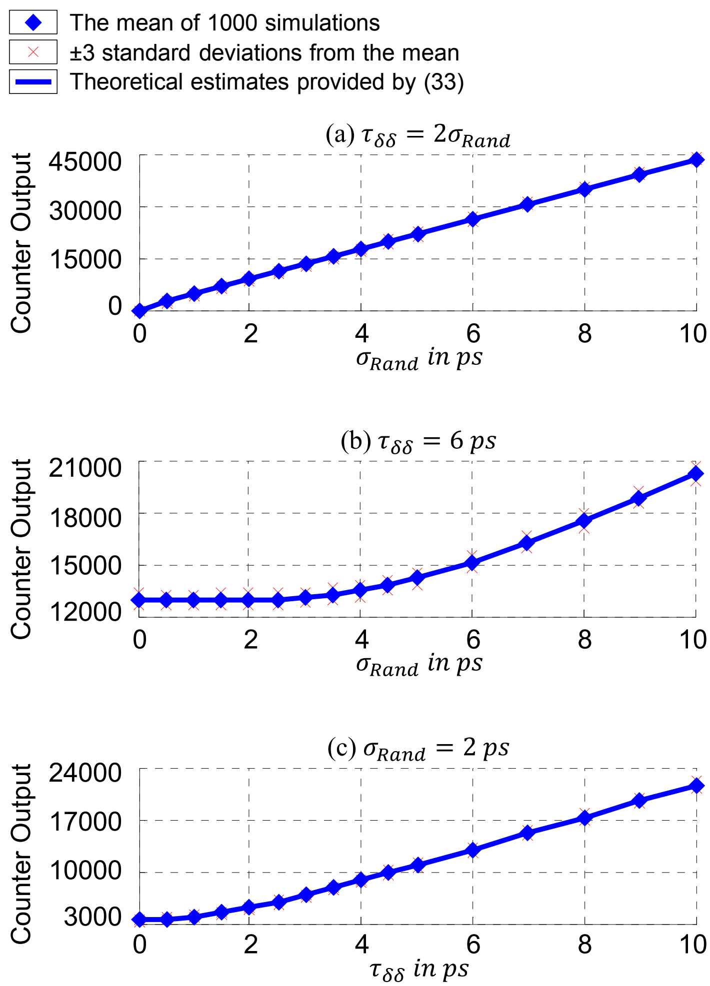



**Figure 2.16 Matlab simulation of the counter output when 6 GHz JIT CLK has a bimodal/Dirac distributed deterministic jitter and random Gaussian jitter as shown in Figure 2.14. (a) When the Dirac peak distance is proportional to the**

**rms jitter, the expected value of the count has linear relationship with $\sigma_{Rand}$ since the general shape of the jitter distribution does not change, but it only widens, increasing the counting region. (b) With constant $\tau_{\delta\delta}$, the expected value of the count has linear relationship with large $\sigma_{Rand}$ where the random jitter is dominant. For small rms jitter, the deterministic jitter is dominant and the expected value of the count is constant since $\tau_{\delta\delta}$ is set constant. (c) Similarly, with constant $\sigma_{Rand}$, the expected value of the count has linear relationship with large $\tau_{\delta\delta}$ where the deterministic jitter is dominant. For small deterministic jitter, the random jitter is dominant and the expected value of the count is constant since $\sigma_{Rand}$ is set constant.**

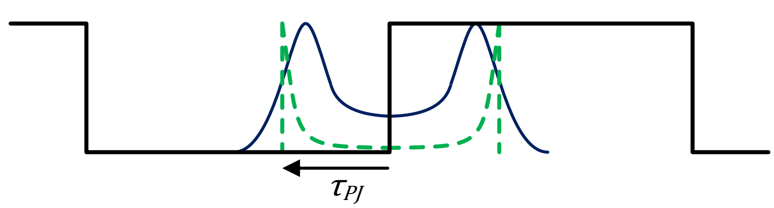



**Figure 2.17 Clock signal with single tone periodic jitter (shown in dotted line) and random Gaussian jitter (not shown). The convolution of the periodic jitter and the Gaussian jitter results in the distribution shown in solid line. The distance**

between the ideal clock edge and the farthest periodic jitter is $\tau_{PJ}$. The random jitter has standard deviation $\sigma_{Rand}$.

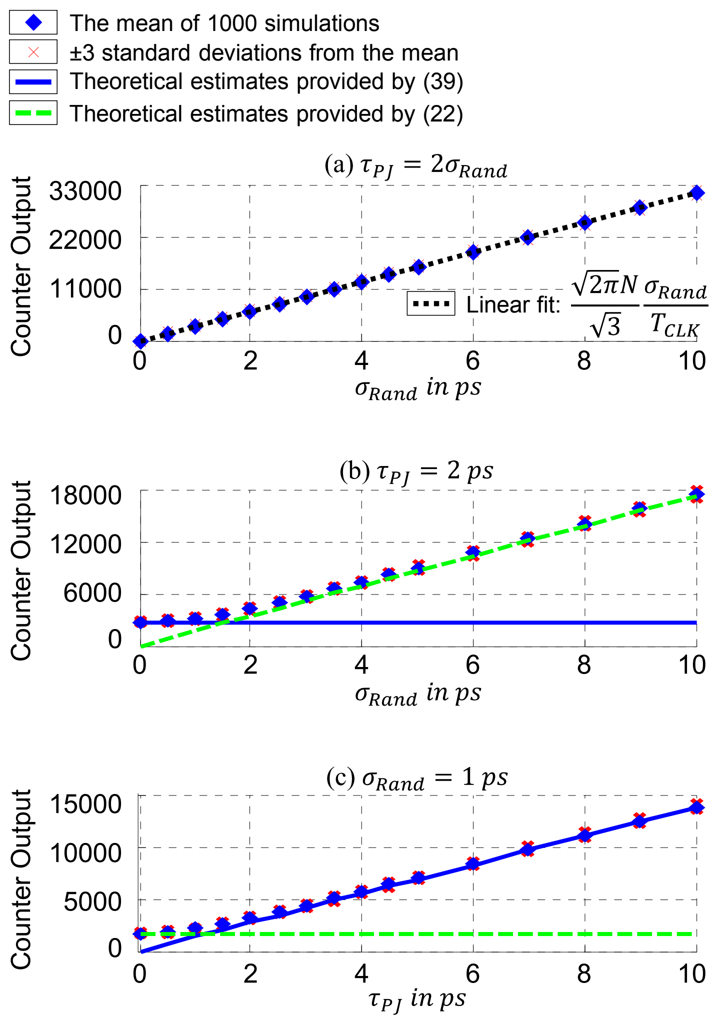



**Figure 2.18 Matlab simulation of the counter output when 6 GHz JIT CLK has a deterministic periodic jitter and random Gaussian jitter as shown in Figure 2.17.**

**Similar to Figure 2.16, (a) when periodic peak distance is proportional to the rms jitter, the expected value of the count has linear relationship with $\sigma_{Rand}$. (b) With constant $\tau_{PJ}$, the expected value of the count has linear relationship with large $\sigma_{Rand}$ where the random jitter is dominant. For small rms jitter, the periodic jitter is dominant and the expected value of the count is constant. (c) Similarly, with constant $\sigma_{Rand}$, the expected value of the count has linear relationship with large $\tau_{PJ}$ where the periodic jitter is dominant. For small periodic jitter, the random jitter is dominant and the expected value of the count is constant.**

Chapter 3

# ON-CHIP TIMING MEASUREMENT CIRCUIT

The previous chapter provides an analytical treatment of the stochastic measurement techniques for timing of clock signals. In this chapter, a circuit solution to demonstrate these techniques is introduced.

## *3.1 Overall architecture*

A simplified block diagram of a circuit implementation of the proposed technique is shown in Figure 3.1. The sampling signal is generated by a voltage controlled oscillator that is designed to have as much Gaussian jitter as possible. The edge detector block checks whether *Input 1* is high and *Input 2* is low when the sampling edge arrives. If they are, the *Edge* counter is incremented by the falling edge of the sampling signal. The *Reset* counter counts every edge of the sampling signal and when $N$ sampling edges are received, it triggers a register to save the output of *Edge* counter and then resets itself as well as the *Edge* counter. Since the measurement runs for $N = 360{,}000$ cycles of the sampling signal, the two counters are 19-bits. Although the implementation of a counter that counts to a value that is a power of 2 (e.g. $N = 2^{19}$) is simpler and larger $N$ improves accuracy, the measurement duration and energy consumption increase with $N$. Therefore, $N$ is kept as $360{,}000$ to obtain the shortest measurement time with good accuracy. The delay block (Figure 3.2a) is a variable delay line that can be used in self-referenced jitter measurement mode to delay the clock signal by one period to generate the reference signal. It is also used to calibrate the two input signals to have the same phase. A duty cycle calibration circuit is integrated in the edge detector block, which is further described in Section 3.1.2.

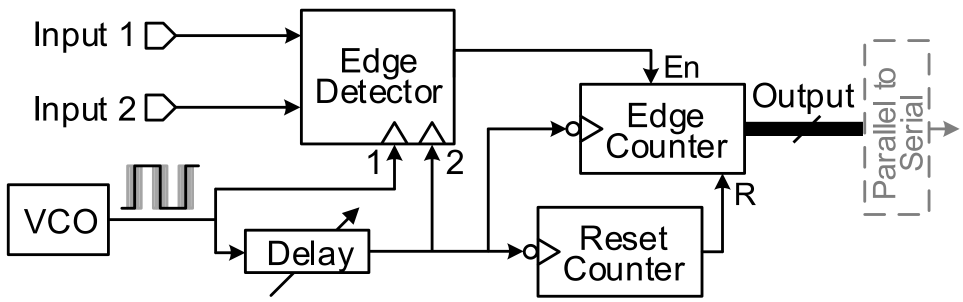



**Figure 3.1. Simplified block diagram of the proposed technique, which can be used for measurement of clock jitter, delay, and duty-cycle.**

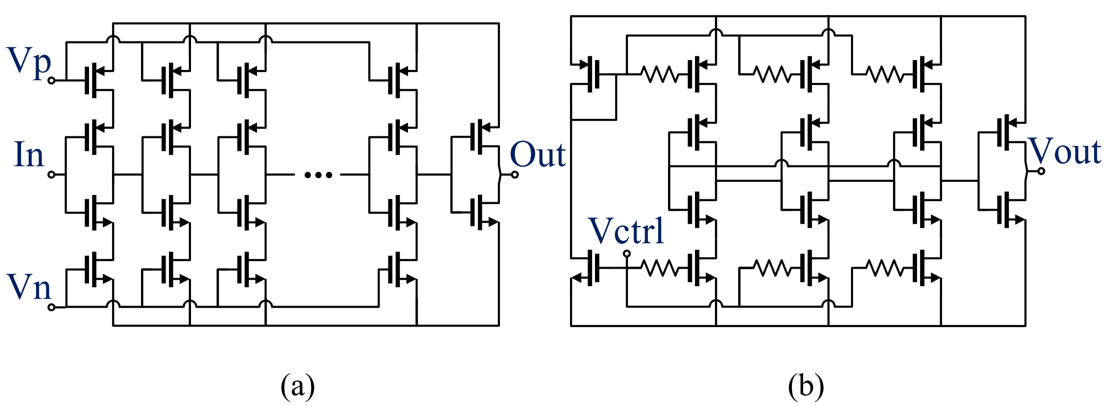
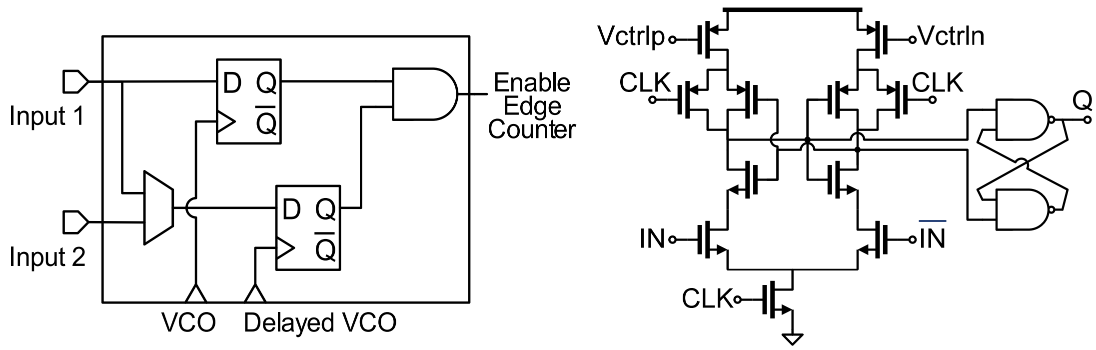



**Figure 3.2. (a) The delay block consisting of 8 stages of current starved inverters. The delay is adjusted by the current starving transistors. (b) High-jitter VCO.**

### 3.1.1 Jittery oscillator

As explained in Section 2.3.2, the sampling signal should have as much Gaussian jitter as possible. In order to achieve this, several design choices were made for the VCO. To minimize unwanted noise, the supply and control voltages are decoupled with large capacitors. The substrate is shielded from the other (mainly digital) circuitry on the same chip. The VCO is a current-starved, single-ended ring oscillator (Figure 3.2b). In order to maximize the jitter due to thermal noise, the VCO devices are set to be near-minimum size, consistent with the observations reported in [22]: jitter due to white noise in CMOS ring oscillator decreases with transistor width and length (below an optimum length). The minimum size also results in small current consumption, which helps with stabilizing the supply. Large resistors are added at the gates of the current starving devices to further increase the jitter using the thermal noise of the resistors. The final tweaking of device sizes with simulation was performed to maximize thermal noise while keeping the flicker noise corner low since flicker noise also increases with decreasing device area.

The VCO frequency should be set low enough that enough jitter is accumulated to highly randomize the sampling edges. On the other hand, setting the frequency too slow prolongs the measurement since the measurement takes $N$ VCO cycles. We run the VCO at 400 MHz, which results in a measurement of about 0.9ms.




(a) (b)

**Figure 3.3. (a) Edge detector block (b) D-flip flop with duty-cycle tuning option.**

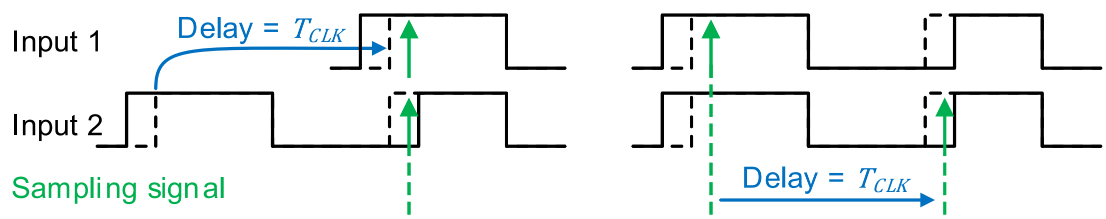



**Figure 3.4. Placing the adjustable delay block on the sampling signal path to 2nd D-flip flop (right) achieves the same result as placing it on the path of the high speed *Input 1* (left).**

### 3.1.2 Edge detector

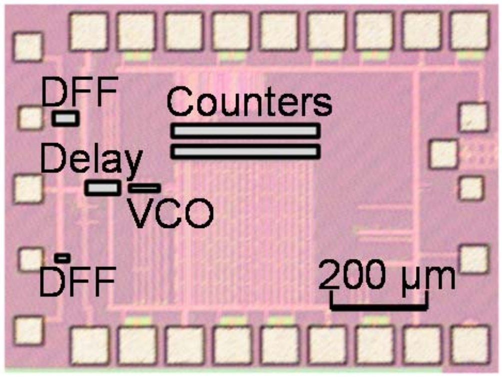
The edge detector comprises two D-flip flops, an AND gate and a multiplexer (mux) (Figure 3.3a). The D-flip flops check the state of the inputs when the sampling (VCO) edge arrives. The AND gate then produces an enable signal for the edge counter if the inputs are in the correct state. The mux connects *Input 1* to the second D-flip flop in self-referenced jitter measurement mode.

The main source of the internal jitter, $\sigma_{INT}$ as mentioned in Section 2.3.6, is the variation in the latching time of the flip flops. Therefore, in order to reduce the variation and speed up the latching time, the first stage of the flip flops is designed to be a dynamic comparator (Figure 3.3b). The simulated input-referred jitter of the flip flops is 32.8 fs.

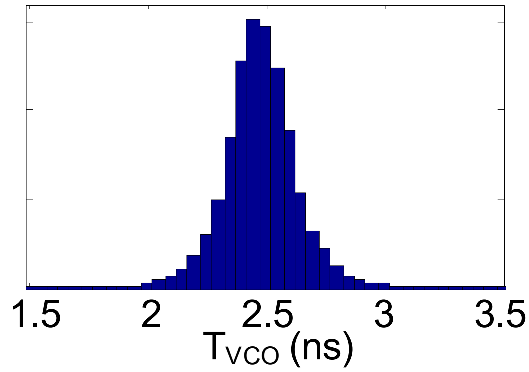
The duty cycle calibration is implemented in the dynamic comparator by adjusting the current available through each branch of the dynamic comparator.

### 3.1.3 Delay block

As mentioned in previous sections, adjusting the delay between the two inputs is necessary in self-referenced jitter measurement and calibration. Instead of placing a delay block in the path of the high-speed input signal, this design incorporates the adjustable delay block on the sampling path to the second D-flip flop. This results in delaying the sampling of *Input 2* and achieves the same results as placing the adjustable delay block in the path of *Input 1* (Figure 3.4).




(a)




(b)

**Figure 3.5. (a) Die photo. (b) VCO period histogram. Period jitter = 54.5 ps rms.**

### *3.2 Measurements*

A circuit demonstrating the application of this technique was fabricated in 65nm TSMC CMOS process and initial measurements were reported in [21]. Figure 3.5a shows the die photo of the chip. The edge detector blocks are positioned sparsely in order to match the pitch of the high-speed probe that was used to supply the clock signals. When the circuit is integrated into a system, the blocks can be placed in a compact area.

The setup for delay measurement mode includes Anritsu MP1763C signal generator, the fabricated chip and Keysight Infiniium 90000A oscilloscope. The signal generator is used to create two 6 GHz clocks with adjustable delay in 1ps steps. The chip outputs the digital counter value in series using a shift register. The digital oscilloscope is used to capture the chip output. The VCO is set to run at 400 MHz and

$N$=360,000. The measurement is repeated 200 times at each delay point in order to capture the random variation in counter output.

Out of the simulated period jitter of approximately 54 ps rms of the free running VCO, about 88% is from the resistors, 10% is from the current-starving devices, 1% is from the inverting devices and the other 1% is from the bias mirror devices. Here, the percentage is calculated as the ratio of the contributed variance to the total jitter variance since the jitter contributions add in variance (instead of standard deviation) as they are independent. The period jitter of the VCO is measured to be 54.5 ps rms (Figure 3.5b).

Each chip is calibrated before it is used for measuring the clock jitter and delay. A clock signal is supplied to one input while the other input is held constant (at either 0 or VDD) so that the counter output represents the duty cycle since the counting region equals the clock signal. The duty-cycle calibration circuitry is then tuned so that slight offset in the 50% clock duty cycle and the process mismatch in the edge detector are compensated. The internal and clock jitter will not affect the duty cycle measurement since the sampling edges capture both sides of the jitter distribution when the 2$^{nd}$ input is held constant. A clock with no added jitter is then supplied to both inputs of the circuit so that the delay block is calibrated by tuning the delay block until the counter output is minimum. The small but non-zero count corresponds to the effective jitter due to the noise in the measurement circuit (including the edge detector) and is measured to be 0.07 ps rms. This internal jitter is compensated for in the measurement plots shown below in order to show the linear relationship between counter output and the injected jitter amplitude.

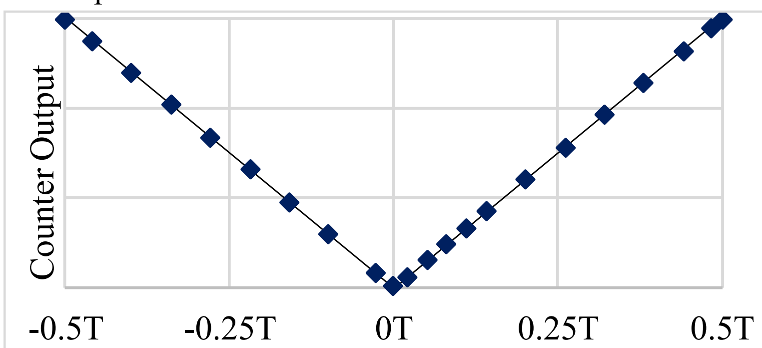



**Figure 3.6 On-chip delay measurement.**

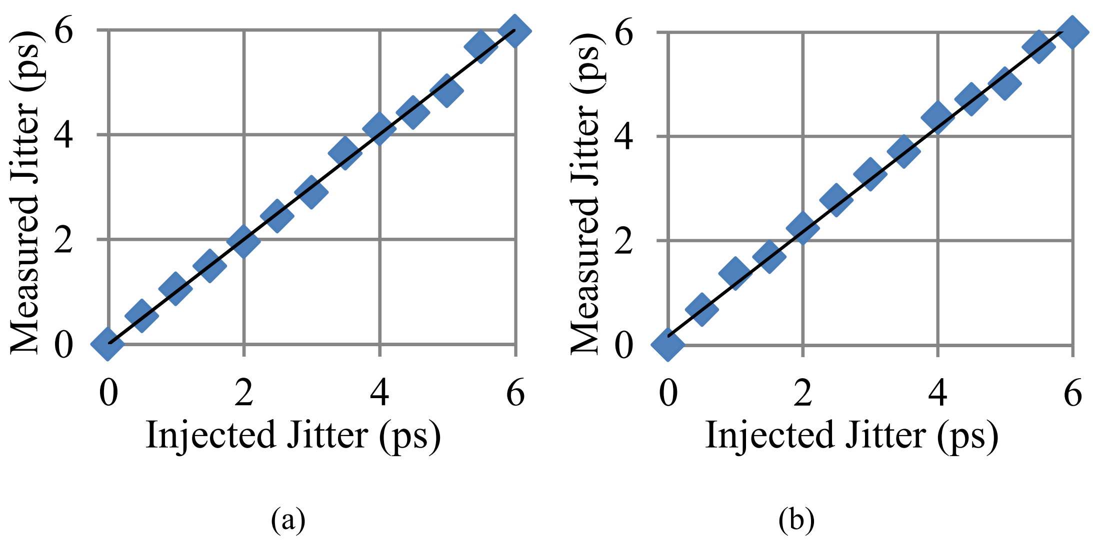


**Figure 3.7. On-chip jitter measurement in (a) external referenced mode and (b) self-referenced mode where the clock is delayed by one period. Counter outputs are converted to show the estimated rms jitter in ps.**

The delay measurement result is shown in Figure 3.6. As expected, the expected value of the counter output has linear relationship with the delay between the input signals. The standard deviation of the measurement at the delay of $T_{CLK}/2$ is 0.14 ps. The performance of the circuit in delay measurement mode is compared with the measured results of previous work in Table 3.1.

In externally-referenced jitter measurement mode, a Keysight N4903B is used to supply JIT CLK and REF CLK at 6 GHz. The JIT CLK is a jitter added version of REF CLK and the added rms jitter is swept from 0 ps to 6 ps, with the maximum limited by the equipment. In self-referenced jitter measurement mode, only JIT CLK is supplied by the equipment and it is connected to both D-flip flops internally on chip, with the second flip flop being sampled one clock period after the first one. The rest of the setup is the same as the delay measurement. Figure 3.7 shows the expected value of the measurement for different rms jitter values in externally referenced and self-referenced modes when the internal noise of the circuit is compensated.

The root mean square error (RMSE) of the measured data relative to the actual injected jitter amplitude is calculated to quantify the measurement error instead of reporting two metrics: the difference between the mean of the measured jitter and the actual injected jitter, which represents precision, and the standard deviation of the measurements, which represents accuracy. The RMSE for externally referenced mode is 0.102 ps and for self-referenced mode, it is 0.308 ps. The circuit consumes 0.89 mW. Table 3.2 compares the previous jitter measurement circuits with this work, which is the only technique that does not require off-chip post processing to provide an estimate of the jitter amplitude. Integrating the post processing on-chip would significantly increase the power and area costs of the other works.

### *3.3 Conclusion*

On-chip measurement of timing non-idealities is increasingly important for characterizing and improving high performance circuits where the high speed clock reduces the margin for timing errors. We presented a stochastic technique that can be used for on-chip measurement of clock jitter, delay and duty-cycle. We provided theoretical analysis showing that the stochastic measurement based on a simple noisy sampler that does not require timing and process accurate circuitry can be accurate and robust. We implemented the technique in a CMOS process and demonstrated the expected functionalities. The chip consumes 0.89 mW, occupies an active area of $1.5 \cdot 10^4 \mu m^2$ and achieves measurement error below 0.31 ps. To the best of our knowledge, it is the only work demonstrated to measure all three of the mentioned timing non-idealities. It is also the only fully on-chip jitter measurement circuit that does not require off-chip processing.

There are two ideas that I would have liked to try if I had more time and that have been left for the future. First, applying the timing measurement circuit in a feedback loop to adjust control parameters and demonstrating the calibrated/controlled outcome would have enhanced this work. For example, the proposed circuit could be used to monitor the output phase of phase interpolators (PIs) or delay locked loops (DLLs). With the feedback loop, the PIs and DLLs can generate the output signals with higher phase accuracy.

Second, it would be interesting to explore the application of the proposed circuit when it is deliberately made to be less precise. Currently, each measurement runs for $N = 360{,}000$ cycles, taking about 1 ms. Hence, any timing changes on a shorter time scale are averaged over the measurement period. This limits the application of the circuit to infrequent calibration or monitoring of slow variations such as aging effect and temperature change. However, if high precision is not crucial, $N$ can be significantly reduced with small degradation in precision because the measurement error scales with $\propto \sqrt{1/N}$, according to (2.9).

**Table 3.1: Comparison with Previous On-Chip Delay Measurement Circuits**

| Work | Test Clock Speed | Measurement Error | Counter Length | Measurement Time (μs) | Power | Area ($\mu m^2$) | Process |
|---|---|---|---|---|---|---|---|
| [7] | 0.2 GHz | 1 ps | $2.5 \cdot 10^5$ | N/A | N/A | $5.9 \cdot 10^5$ | 180 nm |
| [8] | 0.1 GHz | 50 ps[a] | $6.6 \cdot 10^4$ | N/A | N/A | $3.3 \cdot 10^3$ | 130 nm |
| [9] | 2.5 GHz | ±0.25 ps[b] | $5.0 \cdot 10^9$ | N/A (long[b]) | N/A | $1.0 \cdot 10^3$ | 32 nm |
| This Work | 6 GHz | 0.14 ps | $3.6 \cdot 10^5$ | 900 | 0.89 mW | $1.5 \cdot 10^4$ | 65 nm |

**Table 3.2: Comparison with Previous On-Chip Jitter Measurement Circuits**

| Work | Test Clock Speed | Measurement Error | Off-Chip Processing | Power | Area ($\mu m^2$) | Self Referenced | Process |
|---|---|---|---|---|---|---|---|
| [12] | 2.5 Gbps | 1.56 ps | Yes | N/A | $3.5 \cdot 10^4$ | No | 110 nm |
| [13] | 5 GHz | ≤1 ps | Yes | 132.8 mW | $3.2 \cdot 10^5$ | Yes | 65 nm |
| [14] | 2.5 GHz | 0.4 ps | Yes | 1 mW[a] | $3.2 \cdot 10^3$ | Yes[c] | 130 nm |
| [15] | 3.36 GHz | 0.7 ps | Yes | N/A | $4.9 \cdot 10^2$ | Yes | 65 nm |
| [16] | 6 GHz | 2.0 ps | Yes | 36.48 mW | N/A | No | 65 nm |
| This Work | 6 GHz | 0.10 ps<br>0.31 ps | No | 0.89 mW | $1.5 \cdot 10^4$ | No<br>Yes | 65 nm |

a.1 mW @ 200 MHz clock. Self-referenced version is published separately

Chapter 4

# PCO BASED SYNCHRONIZATION FOR PEER-TO-PEER NARROWBAND RF NETWORK

While the previous two chapters present a stochastic technique to measure and monitor timing non-idealities of clock signals at the small scale of integrated circuit chips, this and the next chapters present a technique to achieve and maintain timing at the large scale of P2P radio nodes communicating over tens to hundreds of meters. For the radio nodes in a symmetric P2P network (i.e. all nodes are identical with no distinctive master node) low-power long-range communication is only possible if the power-hungry RF front-end is duty-cycled so that the *average* power of transmission is much reduced. Aggressive duty-cycling for significant power savings requires the entire network to be synchronized. This work utilizes PCO for achieving the synchronization and maintaining network timing.

## *4.1 Pulse coupled oscillators*

Some species of fireflies, notably of the genus Pteroptyx found in Southeast Asia, are known to flash synchronously. Thousands of these fireflies are able to flash in perfect unison for their potential mates. From an engineering perspective, this phenomenon is very interesting because simple creatures that consume very small amount of energy and lack high intelligence are able to achieve synchronization at a massive scale.

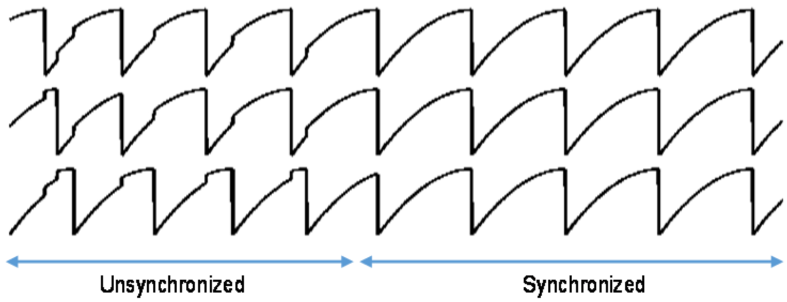



**Figure 4.1. Transient of the synchronization of 3 PCOs**

Mirrollo and Strogatz developed a mathematical model of this phenomenon in their work [18]. The model assumes each firefly has its own relaxation oscillator, or a body clock, that interacts through impulsive coupling, representing the flashing of the firefly. The oscillator is assumed to have a monotonically increasing and concave down phase function with respect to time. When the phase reaches a threshold, the firefly “fires” a coupling pulse and resets the phase. The coupling pulse, or the flash of light emitted by the firefly, causes the other oscillators’ phase to advance by a fixed amount, which is referred in this thesis as the coupling strength. Since the phase function is nonlinear, the fixed amount of phase advance translates to varying amount of time adjustment in how long the oscillator takes to reach the threshold.

Mirrollo and Strogatz proved that if the function ($S(t)$) mapping the phase variable to time is smooth, monotonically-increasing and concave-down ($S'>0$ and $S''<0$), a network of any number of ideal PCOs achieves synchronization. Here, the network consists of identical PCO nodes and does not require a distinctive master node. Each node receiving a coupling pulse, adjusting its own body clock/oscillator, and firing a coupling pulse when reaching the threshold causes the network self-organize and synchronize. Hence, it creates a naturally ad-hoc and scalable system.

An example of the synchronization process for three PCOs is shown in Figure 4.1. Each PCO starts at a random phase and follows the mathematical model described above. After exchanging several coupling pulses, the system achieves synchronization.

### *4.2 PCO for synchronization of IR-UWB radio nodes*

The mathematical proof by Mirrollo and Strogatz that any number of ideal PCOs can reach synchronization became the basis for PCO based synchronization of wireless nodes. PCO network in less ideal conditions was studied by X. Wang et al. and shown to be a promising scheme for scalable synchronization of ultra-wideband impulse radio (IR-UWB) networks [20]. They mapped the parameter space for physical implementation of IR-UWB radios in CMOS into the mathematical model of PCO and showed through simulation that robust synchronization is achieved.

Reference [20] presented several design requirements that is necessary for PCO synchronization in real-world applications:

1. The frequency mismatch between the PCOs should be large enough to reduce the relative jitter in the period of the oscillators.

   It is shown that when the network is synchronized, the node with the highest PCO frequency resets on its own and drives the other nodes to reset. The period difference between this *leader* node and the second fastest node in the network should be large enough that the second node does not temporarily become the fastest node due to jitter.

2. The PCO design should include a blackout period longer than twice the largest propagation delay between two directly connected nodes.

   Blackout period is the amount of period right after a node resets, during which the node ignores any incoming coupling pulses. When a node resets on its own and sends out a coupling pulse, which causes a neighboring node to immediately reset and send a coupling pulse back, the original node should ignore the "echo" coupling pulse.

3. Coupling strength should be large enough to compensate for the frequency mismatch of the oscillators.

   Otherwise, the fastest node would not be able to drive the rest of the nodes to immediately reset, allowing the relative jitter of the oscillators to distort the synchronization quality.

These design requirements are applicable to the synchronization of any RF networks since propagation delay, oscillator jitter and frequency mismatch are inherent in real life systems.

### *4.3 PCO for synchronization of long-range narrowband RF network*

While PCO syncronization was demonstrated for UWB networks, UWB radios have significant limitations in range and tolerance to interferers. Implementing such a concept for narrowband radios requires complete rethinking of the system/circuits.

Since the range of this application is tens to hundreds of meters, the path loss is significant and the node is required to detect the received coupling pulse at much low SNR. The received signal also becomes susceptible to interference. In order to alleviate these issues, the synchronization scheme for long-range narrowband RF network must utilize a signature-signal/syncword as the coupling pulse so that the received signal can be correlated with the expected syncword for more reliable detection.

The longer range also results in longer propagation delay of the coupling pulse. In addition, due to the correlation operation, processing delay is added. These delays affect the synchronization quality.

Since the RF nodes operate in narrowband, phase and frequency mismatch of the carriers as well as the baseband clocks in the PCO nodes affect the detection of the coupling pulse. Hence, the PCO synchronization circuitry should be designed to handle these non-idealities.

While the IR-UWB in [20] operates with pulse rates of 150 kHz, most IoT applications want to sense the peers with a rate on the order of only one Hz [24], in part to further save power. This requires a PCO circuit design different from that of [20] as discussed in Section 5.3.

### 4.3.1 Investigation of sequences for the syncword

Various kinds of sequences with different properties are used in telecommunications, such as Gold code in CDMA and Zadoff-Chu sequence in LTE.

Several of these sequences are investigated for choosing the syncword. Coupling pulses consisting of each sequence and white Gaussian noise are generated in MATLAB. They are then correlated with the noise-free sequence. The correlation is repeated $10^8$ times to obtain the bit error rate (BER) at given signal to noise ration (SNR). The simulation result is show in Figure 4.2. While the Zadoff-Chu sequences provide the best performance, they also require complex processing as they are analog and complex signals. Hence, a 63-bit Kasami sequence, which is binary and real, is chosen for its simple and power efficient implementation (Section 5.2.1). It achieves BER of $10^{-5}$ at 3 dB SNR excluding any non-idealities in processing. An actual circuit implementation of the correlation and any coupling pulse processing will degrade this performance.

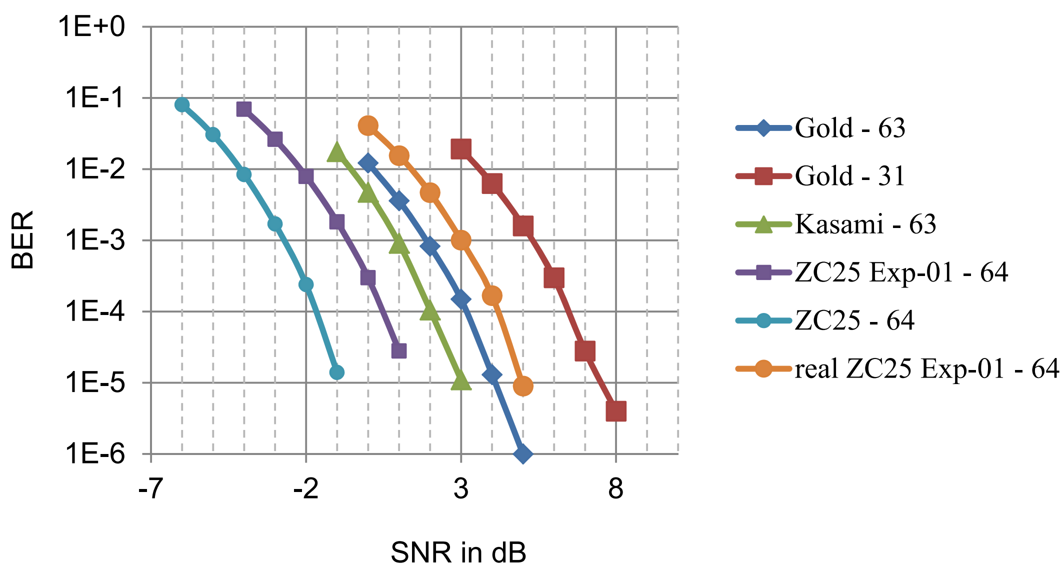



**Figure 4.2. Comparison of various sequences for syncword**

### 4.3.2 Range estimate

Using the SNR value from previous section, the maximum range between the RF nodes for synchronization with the given BER can be estimated assuming the applicable path loss model and the specs of the RF front end used.

The free-space (line of sight) path loss in dB at carrier frequency $f$ in Hz over distance $d$ in m is

$$PL = 20\,log\,d + 20\,log\,f - 147.55\;dB. \quad (4.1)$$

According to IEEE 802.11 Task Group project (IEEE 802.11ah) [25], the outdoor device to device path loss model for antenna height of 1.5 m and carrier frequency of 900 MHz is

$$PL = -6.17 + 58.6\,log\,d. \quad (4.2)$$

Note that this path loss increases much more quickly with distance (58.6) compared to the path loss of free-space (20) or outdoor macro deployment (37.6).

The signal power at the input of the correlator is

$$P_S = P_{TX} - PL + A_{RX}, \quad (4.3)$$

where $P_{TX}$ is the transmitted power and $A_{RF}$ is the gain of the RX front end. The noise power at the input of the correlator assuming room temperature is

$$P_n = -174\frac{dBm}{Hz} + 10\,log\,BW + NF_{RX} + A_{RX}, \tag{4.4}$$

where $BW$ and $NF_{RX}$ are the bandwidth and noise figure of the RX front end respectively. The bandwidth and input-power dependent $NF_{RX}$ and $A_{RX}$ can be obtained from the spec sheet of the RF front end that is used. Assuming transmit power of 0 dBm and a commercial receiver with BW of 2 MHz, the SNR at the input of the correlator given the transmission distance is plotted in Figure 4.3 in blue. The red line shows the SNR of 3 dB for BER = $10^{-5}$ with 63-bit Kasami sequence estimates about 80 m of transmission range using the path loss model in (4.2).

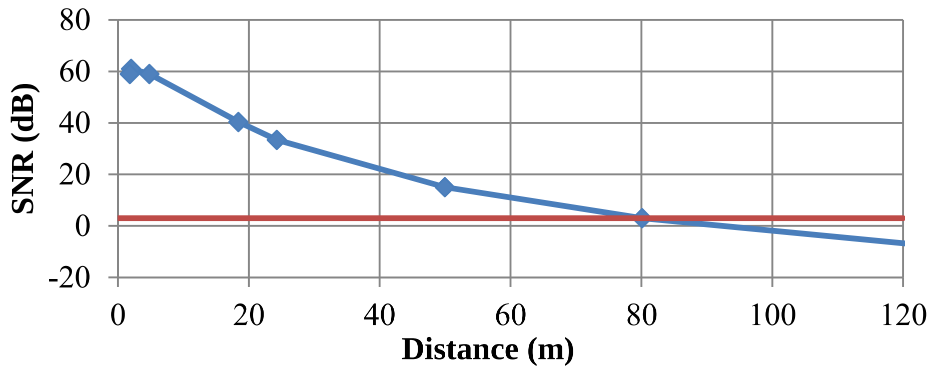



**Figure 4.3. Estimate of the transmission range**

### 4.3.3 PCO network simulation

Rigorous mathematical analysis of synchronization with the parameters of a real-word narrowband system is quite difficult in closed form. Consequently we conduct a numerical simulation for a qualitative analysis in MATLAB. We constructed an event-based simulator similar to the one presented in [20], which takes into account "frequency mismatch, jitter, propagation delay, variable coupling strengths" as well as "arbitrary network connectivity." The simulator we used for this work calculates the phase change using the digital oscillator model (Section 5.3), keeps track of the phase of the reference clock (as well as the oscillator) in each node, and includes the processing delay of syncword correlation in addition to the propagation delay. The simulator also keeps track of the arrival times of propagating CPs. Each event in the simulation is either (a) a node reaching its threshold and firing or (b) an in-flight CP reaching a destination node. At each simulation step, we elapse the time by the amount until the next soonest event. For (a), we reset the firing node's phase and generate a new CP. For (b), we advance the phase of the destination node – if the phase reaches threshold, then we also follow the steps in (a).

The simulator models a network of 20 nodes randomly scattered on a 290 m × 290 m grid with a coupling range of 100 m and $T_{PCO}$ = 1 s (Figure 4.4). We picked a blackout period of 100 ms and maximum coupling strength that immediately takes the nodes to threshold. The *period* jitter of the PCOs is set to be 4 µs rms. The coupling events are modeled as instantaneous. A frequency mismatch of ±50 ppm was chosen as a reasonable value for the reference clock. The exact frequency of each reference clock is drawn from a uniform distribution of ±50 ppm at the beginning of the simulations. The simulations simulates a transient of 100 cycles, which is long enough for the nodes the system to synchronize. Since the synchronization can be dependent on the initial state of the PCOs, each simulation is repeated 200 times with initial conditions for both the reference clock and the digital oscillator randomly chosen from a uniform distribution.

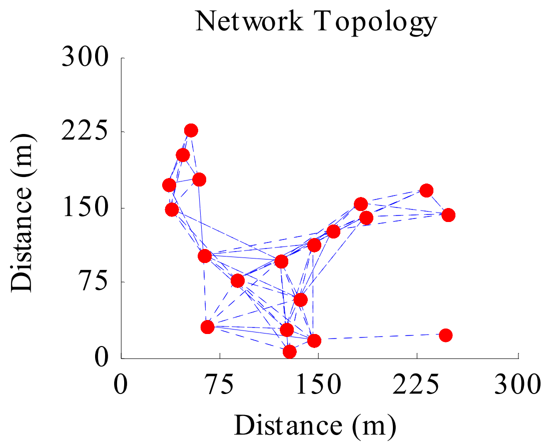



**Figure 4.4. Topology of the randomly located nodes. Nodes within 100 m are connected to each other.**

### 4.3.4 Simulation results and discussion

The impact of the narrowband specific non-idealities on the synchronization is measured by two attributes: how quickly the network synchronizes and how well it maintains synchronization. The speed of synchronization is important because the timescale of a PCO cycle is on the order of seconds. Thus, if the system requires many cycles to synchronize, there is a long absolute time, for which data communication is disabled and the front end consumes high power as it is not duty-cycled. A measure of synchronization quality is "*relative* jitter," amount by which the phase/period of the nodes varies relative to the phase/period of the leader node. Even if the leader has large *period* jitter, the synchronization is robust as long as the other nodes follow the phase of the leader. Maintaining low relative jitter synchronization is important because it allows the power hungry front end transceiver circuits to turn on and off more aggressively with very accurate timing precision and low error rates.

The speed of synchronization depends on two parameters, propagation delay $T_{prop}$ and $\Delta T_{PCO-1}$, which is the period difference between the fastest node and the second fastest node. Intuitively, if the fastest node is forced to reset by the second fastest node due to initial startup condition, then the fastest node lags the slower node by $T_{prop}$. Every cycle from then on, the fastest node reduces the lag by $\Delta T_{PCO-1}$ until it leads the slower node by $\Delta T_{PCO-1}$. At that point, the fastest/leader node starts to drive the slower node to threshold (i.e. the slower node is locked to the leader). Hence, we expect the number of cycles the system takes to synchronize ($N$) to be proportional to $T_{prop}$ and inverse proportional to $\Delta T_{PCO-1}$:

$$N \propto \frac{T_{prop}}{\Delta T_{PCO-1}}. \tag{4.5}$$

The propagation delay $T_{prop}$ is a combination of how long the coupling pulse travels between nodes and how long the syncword processing takes, $T_{Proc}$. Since the correlator has to wait until all of syncword is received, $T_{Proc}$ is the sum of the correlation circuit processing time and the period of syncword. In the first simulation, $\Delta T_{PCO-1}$ is swept and $N$ is recorded. $T_{Proc}$ is set to be 52 $\mu s$, which is representative of the delay in the circuit implementation in Chapter 5. Since initial conditions of the system, such as the starting phase and location of the oscillators, affect how quickly the system synchronizes, Figure 4.5 plots the average of $N$ over 200 iterations with different PCO starting phases. Similarly, in the second simulation, $T_{prop}$ is swept and $N$ is recorded. $\Delta T_{PCO-1}$ is set to be 40 $\mu s$ out of the PCO period of 1 s. The results are shown in Figure 4.6. With $T_{Proc} = 52\ \mu s$, average $N$ is 5.

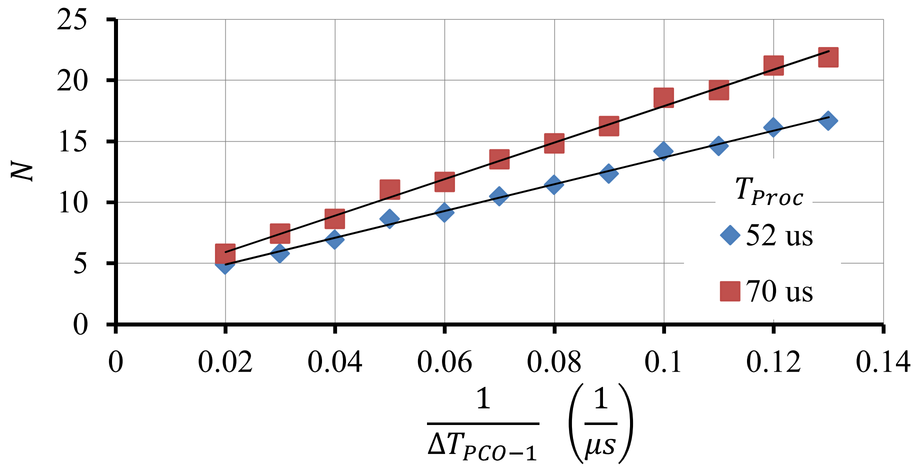



**Figure 4.5. Average number of cycles to synchronize vs. period difference between the leader node and the 2nd fastest node**

Both figures show results consistent with the linear relationship in (4.5). Again, the speed of synchronization depends on the initial conditions and the location of the nodes. In addition, we assumed the system reaches synchronization only if the leader node drives all the other nodes. During the synchronization process, the nodes may all synchronize to a slow node until the fastest node overcome the lag and become the leader. Hence, if $2T_{prop}$ is within the tolerance for synchronization quality, the system may be considered synchronized in a shorter amount of time.

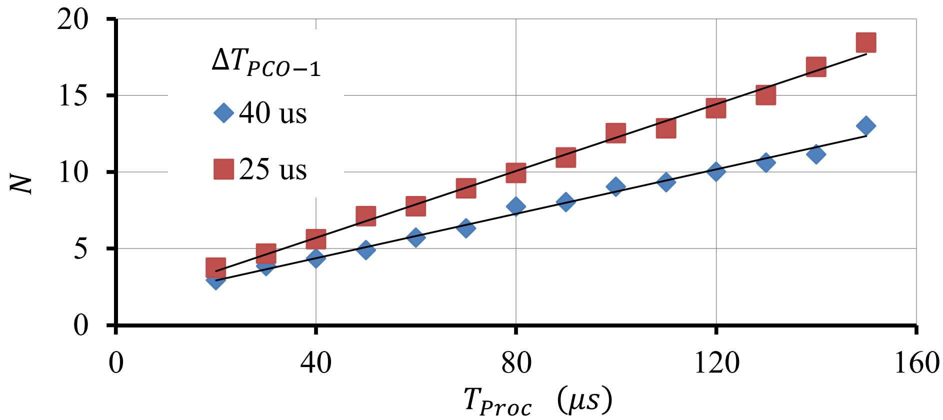



**Figure 4.6. Average number of cycles to synchronize vs. processing delay**

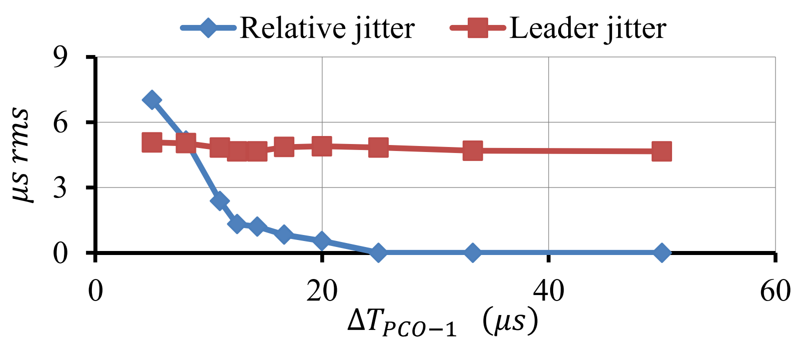



**Figure 4.7. Maximum relative jitter vs. period difference between the leader node and the 2nd fastest node. The jitter of the leader node is plotted as a reference.**

Once the system is synchronized, the jitter of some nodes may occasionally cause them to fire before the coupling pulse from the leader or a leader driven node arrives. However, the jitter must be large enough to overcome the period difference between the leader and the slow nodes. Hence, the faster the leader is than the rest, the less the system experiences relative jitter. Figure 4.7 shows that the maximum relative jitter is reduced for larger $\Delta T_{PCO}$. As long as $\Delta T_{PCO}$ is larger than the jitter of any node in the system, the high quality synchronization is maintained. Therfore, the criteria for keeping synchronization is

$$\Delta T_{PCO} \gg \sigma_{PCO}, \tag{4.6}$$

where $\sigma_{PCO}$ is the oscillator jitter of the PCO nodes.

In a multi-hop network where some nodes are indirectly driven by the leader node through some intermediary nodes, the firing lag between slow nodes and the leader node is integer multiples of $T_{prop}$. This should be taken into account when the framework for the data transmission scheduling is designed.

Chapter 5

# SYNCHRONIZER CIRCUIT FOR PEER-TO-PEER NARROWBAND RF NETWORK

In the previous chapter, a technique for emergent synchronization using PCO was introduced and its application in narrowband long-range P2P RF network was discussed. This chapter presents the design and implementation of the technique in CMOS circuits. The synchronization of a P2P network using the implemented circuits is also demonstrated.

## *5.1 Overview of the synchronizer block*

This chapter presents a baseband synchronizer block for scalable synchronization and aggressive duty cycling of narrowband RF peer-to-peer network. The synchronizer block consists of an signal processor (SP), which detects if a coupling pulse containing the syncword is received, and a timing circuit, which implements the PCO using digital blocks to adjust local timing and synchronize the node clock to the network. For the system level demonstration, each node in the network consists of a commercial narrowband RF front end connected to the proposed synchronizer block. Figure 5.1 shows the peer-to-peer network, a node in the network, and the synchronizer block in addition to an example of the synchronization process.

### *5.2 Signal processor*

The goal of the proposed signal processor is to receive a baseband signal from a commercial RF front end and detect whether a predetermined syncword is embedded in it. Doing so requires correlating the signal with the expected sequence. However, practical duty cycled radios with long off-times must synchronize with 1) low latency sync detection to enable short duty cycles for low power, 2) tolerance to LO mismatch and random startup states of the RF front end due to LO between "on" cycles, and 3) high SNR for long range. Hence, the SP consists of circuit blocks designed to meet these challenges.

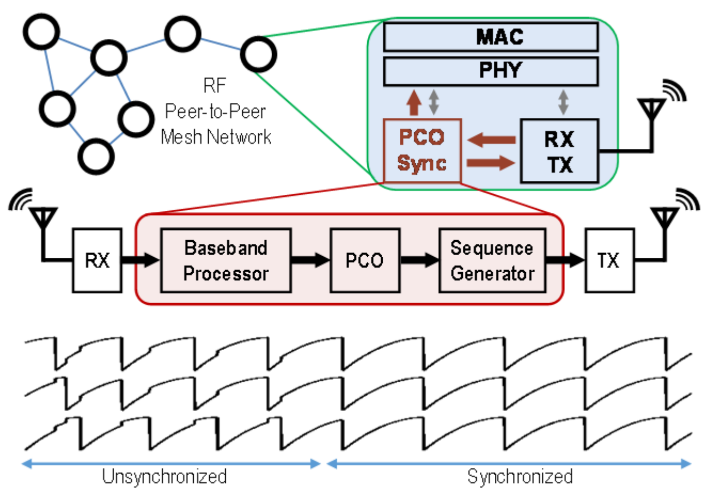



**Figure 5.1. Block Diagram of the system. Synchronizer block (PCO Sync) shown in red is designed and fabricated. Transient of PCO phase state is illustrated as an example of three node synchronization.**

Cross-correlation requires multiplication, summing, subtraction, delaying and comparing. While these can be done in the digital back-end, high performance ADCs and DSPs consume high power. Dedicated active analog circuits for some of these operations, such as banks of multipliers, also consume high power. In order to improve energy efficiency of these operations, we design a passive signal processor using only capacitors and switches along with few comparators. The passive topology allows low-power operation, quick on- and off-switching for aggressive duty-cycling, and predictable and short latency.

A block diagram of the signal processing circuit is shown in Figure 5.2. The main block is the correlator, which correlates the received signal with the expected sequence, programmed by the serial-to-parallel interface (SPI). A Kasami sequence of 63 bits is chosen for its good correlation performance and ease of implementation. The output of the correlator is resolved by the peak detector. Carrier frequency mismatch generates low frequency amplitude variation on the demodulated baseband signal. The differential detector mitigates this effect and enables detection in the presence of carrier offset. The amplitude detector automatically adjusts the peak detector threshold based on the received signal amplitude. Each of these blocks is discussed in greater detail in the subsections below.

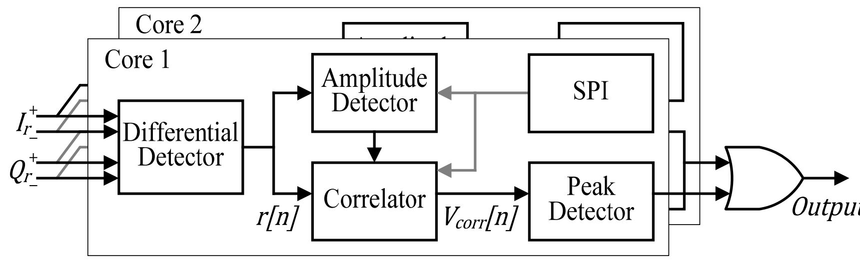



**Figure 5.2. Block diagram of the proposed signal processor.**

### 5.2.1 Correlator

This work uses a programmable 63 bit Kasami sequence for the syncword, which is used as the impulsive coupling for the narrowband system. The correlator block performs the cross-correlation of the expected Kasami sequence $s[k]$ and the received signal $r[n]$, where $k$ and $n$ represent the $k^{th}$ bit and $n^{th}$ sample, respectively.

$$s \star r[n] = \sum_{k=1}^{63} s[k]r[n+k] \tag{5.1}$$

As shown in (5.1), the correlation operation has three operations: slide, multiply and sum. If $r$ contains the sequence and is aligned with $s$, then the correlator generates a peak. However, the time point $n$ when the concealed sequence in $r$ starts is unknown. In conventional wireless circuits such as Wi-Fi and cellular radio, the starting point is estimated in backend digital circuitry, which is avoided in this work. This circuit instead emulates 63 multiplications in parallel, each with a different bit of $s[k]$, so that the need to detect the phase of the concealed sequence in $r$ is eliminated.

The correlator stores samples of the signal $r$ in sampling capacitors as $r$ is received, which is equivalent to the sliding operation of the correlation. Then, 63 capacitors containing the latest 63 samples of $r$ are shorted together (Figure 5.3) to average the stored charges, thereby performing the summing operation. Depending on whether the expected sequence bit $s[k]$ is 1 or $-1$, the capacitor with the $(n+k)^{th}$ sample of $r$ is shorted in positive or negative configuration with the rest of the 63 capacitors, which emulates multiplication operation. Notably, only one type of capacitor cell is used to perform all the operations of correlation.

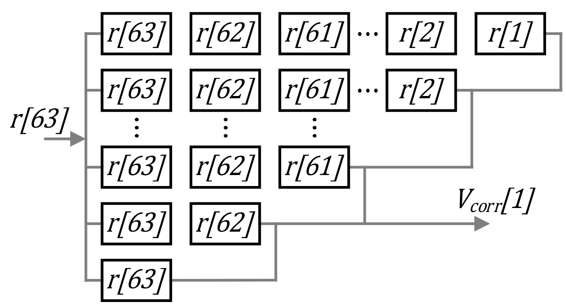



**Figure 5.3. The correlator consists of sampling cells that store the latest 63 bits of the decoded signal. The output of the differential detector, representing the current data bit, is connected to 63 cells. Half a bit period later, different cells containing the latest 63 bits are connected to the output.**

Each sample of the received signal $r$ is used once per bit period for 63 consecutive bit periods. On the other hand, shorting capacitors for the summing operation modifies their charge, making them unusable for future bit periods. Therefore, each received sample of $r$ must be stored on 63 different capacitors. The sampling capacitor multiplied with $s[63]$ is used for correlation during the same cycle, which allows it to be re-purposed during the next cycle. Similarly, sampling capacitor multiplied with $s[k]$ can be re-purposed every $[63-k]^{th}$ bit period. Thus, as shown on the left of Fig. 2, the correlator requires $1+2+\cdots+63=2016$ sampling capacitors.

The sampling capacitors in the correlator have four phases of operation (Figure 5.4): (i) storing the samples of $r$, (ii) sharing the stored charges to amplitude detector cells, to be discussed below, (iii) shorting 63 capacitors containing the latest 63 samples to the output node, and (iv) resetting the charges on the shorted capacitors.

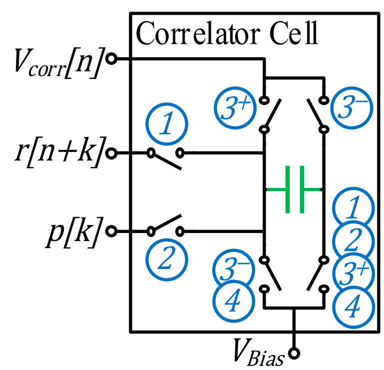



**Figure 5.4. The correlator cell consists of a sampling capacitor and switches connecting to other blocks. The numbers in circles represent which phases the corresponding switches turn on in. In phase (iii), the capacitor may be connected to the output ($V_{corr}$) in either positive or negative configuration based on $s[k]$.**

### 5.2.2 Peak detector

The peak detector consists of three identical dynamic comparators that are connected to the output node of the correlator (Figure 5.5). In phase (iii), when selected correlator capacitors are connected to the output node, the comparators are clocked to compare the correlator output with a threshold. If the comparator outputs are different due to random noise, the voting logic selects the majority vote. The peak detector outputs 1 if the correlator peak is larger than the threshold:

$$V_{corr}[n] = \frac{1}{63}\sum_{k=1}^{63} s[k]r[n+k] > V_{thresh}. \tag{5.2}$$

The amplitude detector adjusts the charges stored on the correlator capacitors in order to make $V_{thresh}$ effectively proportional to the input signal amplitude.

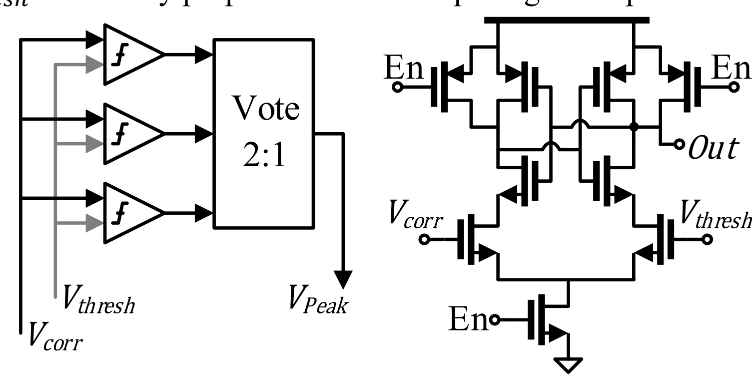



**Figure 5.5. (Left) The peak detector consists of three comparators and a logic block for majority vote for the output. (Right) Dynamic comparator used in the peak detector.**

### 5.2.3 Differential detector

For aggressively duty-cycled RF front ends with long off times, there will be phase and frequency mismatch between the transmitter carrier (TX LO) and the demodulation LO due to drift. This mismatch results in a low frequency amplitude modulation on the received baseband signal, causing the signal to be too small to detect and/or flipping the polarity of the binary code embedded in it. The differential detector solves this problem (a) by looking at the combination of I and Q channels so that if one of them has small amplitude due to the envelope, the other has large amplitude; and (b) by comparing two consecutive bits to compute a decoded bit (e.g. if the consecutive bits have the same polarity, then the decoded bit is 1 so that the actual polarity of the received bits does not matter).

Assume the baseband signals $X$ and $Y$ are transmitted on the $I$ and $Q$ channel, resulting in an RF signal of $X\cos(\omega t) + Y\sin(\omega t)$, where $\omega$ is the carrier frequency of the transmitter. When the signal is received at the receiver, it is mixed with carrier signals $\cos(\omega t + \Delta\omega t + \phi)$ and $\sin(\omega t + \Delta\omega t + \phi)$ for down-conversion, where $\Delta\omega$ and $\phi$ are the frequency and phase mismatch between the transmitter and receiver carriers. Hence, the baseband $I$ and $Q$ signals in the receiver are

$$\begin{cases} I_r = X\,cos(\Delta\omega t + \phi) - Y\,sin(\Delta\omega t + \phi) \\ Q_r = X\,cos(\Delta\omega t + \phi) + Y\,sin(\Delta\omega t + \phi)\,, \end{cases} \tag{5.3}$$

where the amplitude terms are removed for simplification. It can be seen from (5.3) that the sinusoidal terms vary from $-1$ to 1 over time, causing the problems mentioned above. Assuming the transmitter sent the same signal on $X$ and $Y$, (5.3) becomes

$$\begin{cases} I_r = \sqrt{2}X\, cos(\Delta\omega t + \phi') \\ Q_r = \sqrt{2}X\, sin(\Delta\omega t + \phi')\,, \end{cases} \tag{5.4}$$

where $\phi' = \phi + \pi/4$. Consider the following transformation for the consecutive samples of the received baseband signals:

$$\begin{aligned} I_r[n] \cdot I_r[n-1] + Q_r[n]Q_r[n-1] = \\ &= 2X[n]X[n-1] \\ &\quad \cdot (cos(\Phi) \cdot cos(\Phi + \Delta\omega T_{bit}) + sin(\Phi) \cdot sin(\Phi + \Delta\omega T_{bit})) \\ &= 2X[n]X[n-1]\, cos(\Delta\omega T_{bit}), \end{aligned} \tag{5.5}$$

where $\Phi = \Delta\omega t + \phi'$, $T_{bit}$ is the bit period of the Kasami sequence and $n$ refers to the $n^{th}$ sample of the baseband signals. The transformation in (5.5) is void of the low frequency amplitude modulation and the sinusoidal term is close to 1 since $\Delta\omega T_{bit}$ is small. However, it requires accurate analog multiplication.

The differential detector performs a variation of the transformation in (5.5) for more power-efficient implementation:

$$r[n] = I_r[n]sign(I_r[n-1]) + Q_r[n]sign(Q_r[n-1]), \tag{5.6}$$

where $r[n]$ is the output of the differential detector, which is also the input to the correlator in Section 5.2.1. More than 99% of the time, the sinusoidal terms on the consecutive samples of $I_r$ and $Q_r$ will have the same sign, e.g. $\text{sign}(\cos(\Phi)) = \text{sign}(\cos(\Phi)\Delta\omega T_{bit})$. In these cases, (5.6) becomes

$$\begin{aligned} r[n] &= \sqrt{2}X[n]sign(X[n-1]) \cdot (|cos(\Phi)| + |sin(\Phi)|) \\ &\approx \sqrt{2}X[n]sign(X[n-1]) \cdot \left(1 + \left(\sqrt{2}-1\right)|sin(2\Phi)|\right). \end{aligned} \tag{5.7}$$

The amplitude factor in (5.7) with the sinusoidal term varies only between 1 and $\sqrt{2}$. In the other $< 1\%$ of the time, the amplitude factor varies between $\sim 0.954$ and $\sqrt{2}$ with conservative assumptions of carrier frequency, mismatch, and signal bandwidth. Since output bit $r[n]$ depends on whether the consecutive two bits of the transmitted signal have the same sign, the Kasami sequence should be encoded accordingly on the transmitter side so that the decoded output of the differential detector is also a Kasami sequence.

As shown in Figure 5.6, the circuit implementation of (5.6) consists of sampling capacitors to sample $I_r$ and $Q_r$ and dynamic comparators with a delay register to produce $\text{sign}(X[n-1])$, which determines whether the capacitors containing $I_r$ and $Q_r$ samples are shorted to the output node in positive or negative configuration.

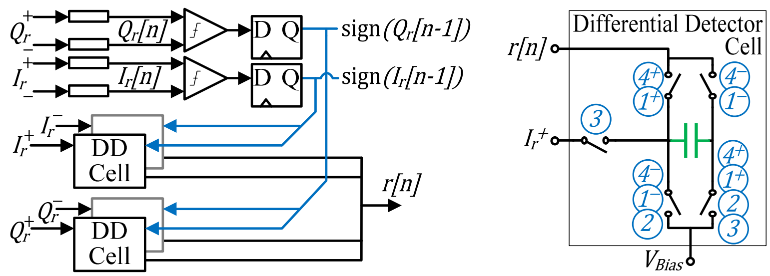



**Figure 5.6. (Left) Differential detector diagram and (Right) its sampling cell.**

### 5.2.4 Amplitude detector

Over time, the correlator output produces several peaks proportional to the received signal amplitude. While the largest peak occurs when the embedded sequence in the received signal aligns with the expected sequence, the correlator produces smaller peaks for misaligned sequences even if the received signal has no noise. Hence, too high a threshold in the peak detector ($V_{thresh}$) will miss the received sequence while too low a threshold produces false positives. Therefore, the threshold must be adjusted if the received signal amplitude changes and the correlator peaks change proportionally. The amplitude detector achieves this by removing a fixed percentage of the charges stored on the correlator capacitors, effectively making the threshold proportional to the received signal amplitude. Doing so avoids the need for analog envelope detector and its requirement for fine bandwidth adjustments. Define

$$V_{thresh} := \alpha V_{average} = \alpha \frac{1}{63}\sum_{k=1}^{63} |r[n+k]| \\ = \alpha \frac{1}{63}\sum_{k=1}^{63} r[n+k]sign(r[n+k]), \tag{5.8}$$

where $\alpha$ is a constant between 0 and 1. Substituting (5.8) into (5.2) and simplifying it yields

$$\frac{1}{63}\sum_{k=1}^{63} r[n+k](s[k] - \alpha \cdot sign(r[n+k])) > 0. \tag{5.9}$$

Defining $\alpha' = (1-\alpha)/(1+\alpha)$ and $r' = r(1+\alpha)$, (5.9) becomes

$$\frac{1}{63}\sum_{k=1}^{63} s[k]r'[n+k]\beta > 0, \tag{5.10}$$

where

$$\begin{cases} \beta = \alpha' < 1 & if \quad s[k] = sign(r[n+k]) \\ \beta = 1 & if \quad s[k] \neq sign(r[n+k]) \end{cases}. \tag{5.11}$$

If we let $r' = r$, (5.10) governs the operation of the correlator, and performs the original thresholding function in (5.2) except for $\beta$. Hence, the amplitude detector consists of charge-sharing capacitors that are shorted to the correlator capacitors at node $p[k]$ (Figure 5.4) in phase (ii) based on the condition in (5.11) so that the correlator voltage reduces from $r[n]$ to $\alpha' r[n]$.

### 5.2.5 Dual core

Since the transmitter and the receiver are not phase locked, the signal processing circuit may sample the received signal at a bit transition, causing errors. Therefore, a copy of the circuit, Core 2 (Figure 5.2), is added. It samples the received signals half a bit period after Core 1 so that if either one samples at a transition, the other core samples at the center of the bit period. The combined output produces 1 if either core outputs 1.

### *5.3 Digital PCO*

After the SP detects the syncword in the received signal despite phase and frequency offset and variations in signal amplitude, it generates a pulse, which is coupled into the digital PCO to advance the phase of the local clock.

As mentioned in Chapter 4, the period of the PCO in this application is on the order of a second. Analog oscillators, such as the RC relaxation oscillators that met the specifications for UWB PCO synchronization, cannot be used for one second timescale due to the very large components necessary to obtain the large time constant. An integrated capacitor of this scale has significant parasitics and leakage that prohibits correct operation.

In this work, we propose a digital oscillator circuit that can be integrated in a CMOS process as an alternative to the analog relaxation oscillator and investigate the impact of the quantization/discretization error inherent to the digital design on the synchronization performance of the PCO network. We show through simulation that when compared to the analog oscillator case, the trade-off between the synchronization speed and synchronization quality is much more pronounced with digital PCOs. We find that a well-designed digital PCO can achieve a better performance than an analog PCO if the opposing criterion is relaxed. We also find that the most important design decision in a digital PCO is how to time the reset after the oscillator reaches threshold due to coupling. This critical decision can considerably affect the synchronization performance, and is studied analytically in this paper before the results are applied to our final design. The circuit implementation of the proposed digital PCO design is discussed in Section 5.3.4.

### 5.3.1 Digital PCO Model

The fundamental design of the digital oscillator is based upon a modified counter driven by a reference clock (a crystal, a MEMS oscillator or a low jitter on-chip oscillator) and implements the monotonically-increasing concave-down function necessary for the PCO network. Our design consists of a main counter that counts a slowing *trigger clock*, each cycle of which is one reference clock period longer than the previous (Figure 5.7).

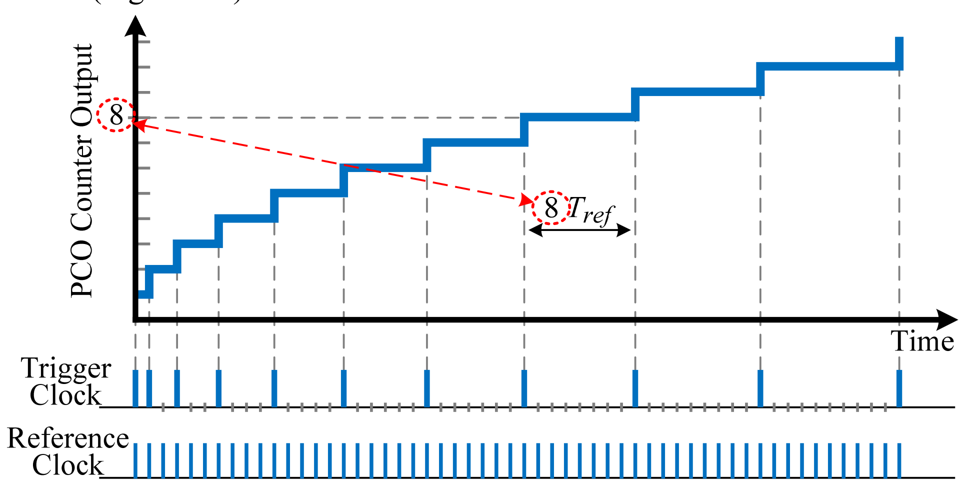



**Figure 5.7. Illustration of digital PCO state function. The state of the main counter lasts the same number of reference periods as its count.**

We can show that the normalized output of this counter is

$$V = \frac{\sqrt{T_{ref}} + \sqrt{T_{ref} + 8nT_{ref}}}{-\sqrt{T_{ref}} + \sqrt{T_{ref} + 8nT_{PCO}}}, \tag{5.12}$$

where $T_{ref}$ is the reference clock period, $T_{PCO}$ is the period of the digital oscillator and $nT_{ref}$ represents the quantized time. Since $\lim_{T_{ref}\to 0} V = \sqrt{t/T_{PCO}}$, the digital oscillator output is an approximation of $\sqrt{t/T_{PCO}}$ function. Although this function satisfies the monotonicity and the concavity conditions, the digital output sometimes has $S'=0$ (and $S''=0$) due to the discrete approximation. Therefore, it is important to study whether the system can synchronize with the non-smooth state function.

The critical design decision lies in how an oscillator responds to a CP that takes it to threshold. Ideally, the oscillator should self-reset $T_{PCO}$ after receiving the CP (Figure 5.8). However, due to the discretization, it may only reset at either $t=t_A$ or $t=t_B$. Thus, for that cycle, the digital oscillator has an effective period of $T_A = T_{PCO} + \tau$ or $T_B = T_{PCO} - (T_{ref} - \tau)$, respectively, where $\tau$ is the delay between the coupling and the following trigger clock. Since $\tau \le T_{ref}$, we see that $T_B \le T_{PCO} \le T_A$. As shown in Section 5.3.3, network synchronization is affected differently by the two different designs that provide $T_A$ and $T_B$, which we call design A and design B respectively.

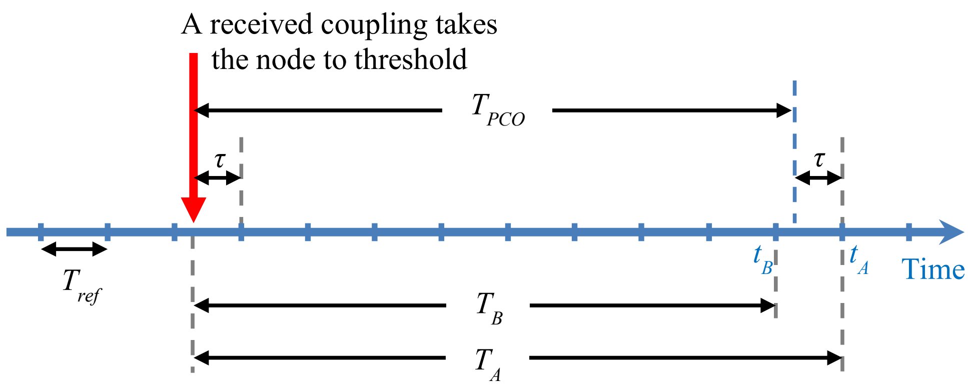



**Figure 5.8. Illustration of the difference between design A and design B.**

### 5.3.2 Network simulation with digital PCO

We ran the same MATLAB simulation from Section 4.3.3 to assess the impact of digital PCO on the synchronization. The digital PCO is different from analog PCOs since its phase function is discrete. The simulator calculates the phase change using the digital oscillator model and adjusts the period of each cycle with $\tau$. The simulator models a network of 20 nodes randomly scattered on a 500 m × 500 m grid with a coupling range of 200 m and $T_{PCO}$ = 1 s. In order to highlight the impact of just the digital PCO quantization error, the *period* jitter is set to be a small value of 0.1 ns. Simulations with a jitter of 1 µs are also done to illustrate that the network of digital PCOs can achieve synchronization with both jitter and the quantization error. When the digital oscillator resets either at the end of a cycle or due to coupling, its nominal period $T_{PCO}$ is adjusted to include $\tau$, if necessary, as well as the jitter. Simulations are run for 100 cycles for each parameter sweep.

### 5.3.3 Simulation results and discussion

An implication of design A is that if a CP takes the fastest node to threshold before the network synchronizes, the node will appear to have a longer period ( $T_A = T_{PCO} + \tau > T_{PCO}$ ), thereby hurting the *speed* of synchronization. A typical example of node dynamics of the network for design A is plotted in Figure 5.9, which shows the firing times of each node relative to the fastest node every time the fastest node fires. We observed that the network can be locked to a slower clock at the beginning as the fastest node reduces $\tau$ each cycle, eventually leading the synchronization.

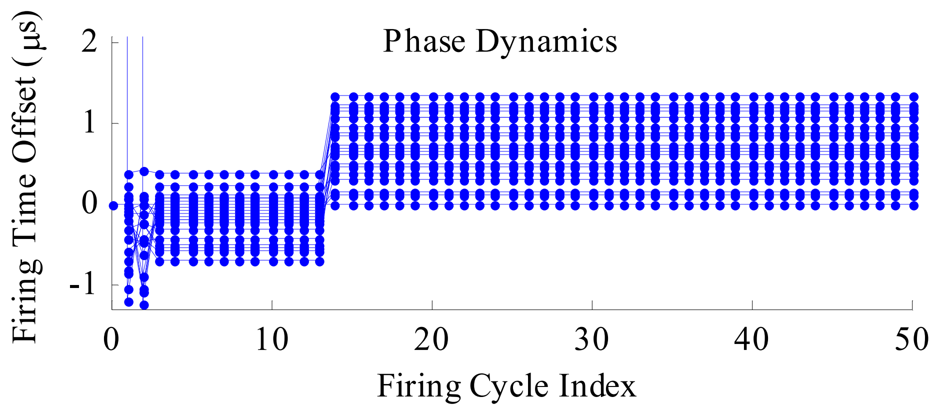



**Figure 5.9. Time of firing of all nodes relative to each time the fastest node fires (design A).**

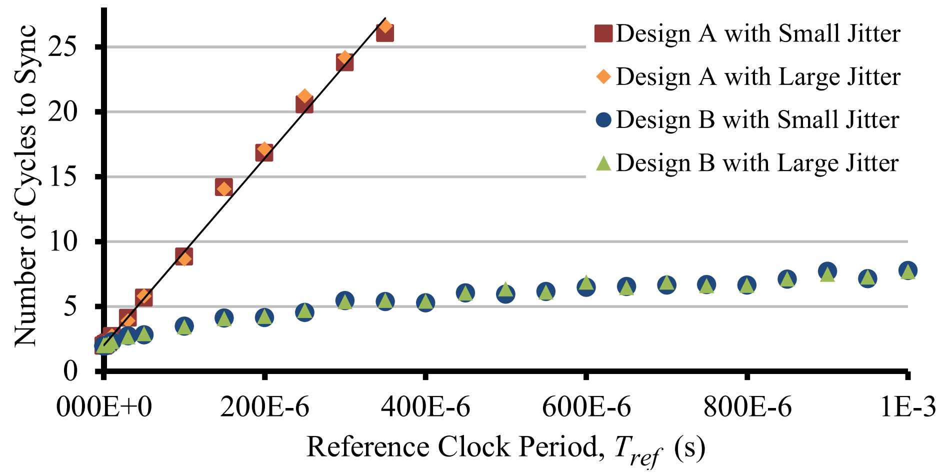



**Figure 5.10. Synchronization speed, i.e. the average number of cycles (across 100 different initial conditions) the leader node takes to synchronize. Large and small jitter refer to reference clock jitter of 1 µs and 0.1 ns, respectively.**

The synchronization time increases linearly (Figure 5.10) with the reference clock period. Figure 5.11 (Design A with Small Jitter) shows that design A does not introduce any considerable relative jitter to the system. In fact, one advantage of design A is that it helps to maintain the synchronization since once synchronized, all the slower nodes have even longer effective periods and thus their period jitter is less likely to take them out of lock. The orange curve shows an increased relative jitter with the introduction of an absolute jitter of 1 µs. However, as the effect of discretization increases with $T_{ref}$, design A reduces the relative jitter by reducing the contribution of the absolute jitter even below the analog oscillator case. The performance of analog oscillator is equivalent to that of the digital designs when $T_{ref}$ is sufficiently small (Figure 5.11).

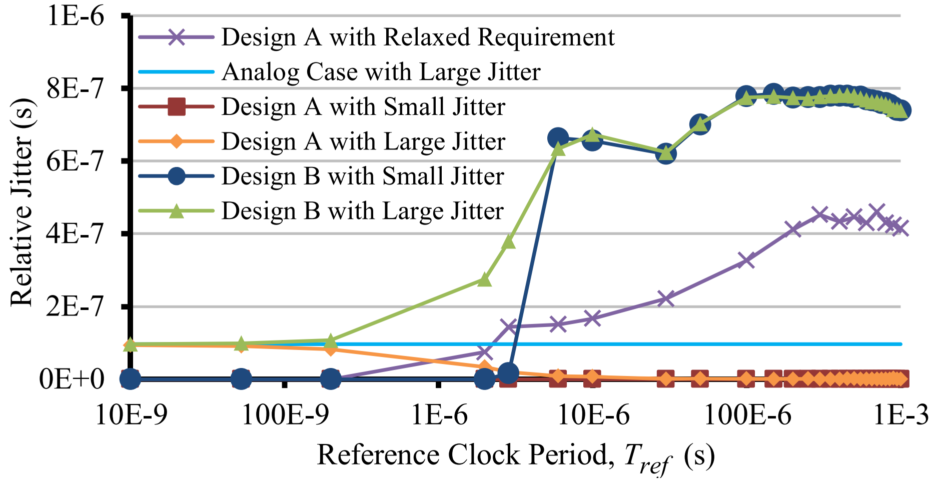



**Figure 5.11. Synchronization quality measured by the average of the maximum relative jitter. With the relaxed requirement, the network is considered synchronized if the period of each node is within $T_{ref}$ of the period of the fastest node.**

The impact of design B can be more serious because it affects the *quality* of the synchronization. A slow node locked to the leader node appears to have a shorter period ($T_B \leq T_{PCO}$) and if the leader (fastest) node is not fast enough, $T_B$ may become smaller than the period of the leader node, resulting in the slow node temporarily getting out of lock. Once out of lock, the slow node has its nominal $T_{PCO}$ (instead of

$T_B$) and synchronizes with the leader again. This constant loss of synchronization and re-locking can create a relative jitter that does not appear in design A. This effect is illustrated in the plot of a typical node dynamics in the pseudo-synchronized state (Figure 5.12). The relative firing times of many nodes exhibit triangular shapes, which means these nodes are getting in and out of lock. This large relative jitter is illustrated in Figure 5.11. The synchronization speed of design B appears faster than design A (Figure 5.10) because design A takes longer to reach synchronization with a much stricter relative jitter criterion. However, design A network achieves synchronization faster than design B for the same synchronization quality (Figure 5.13 - Design A with Relaxed Requirement).

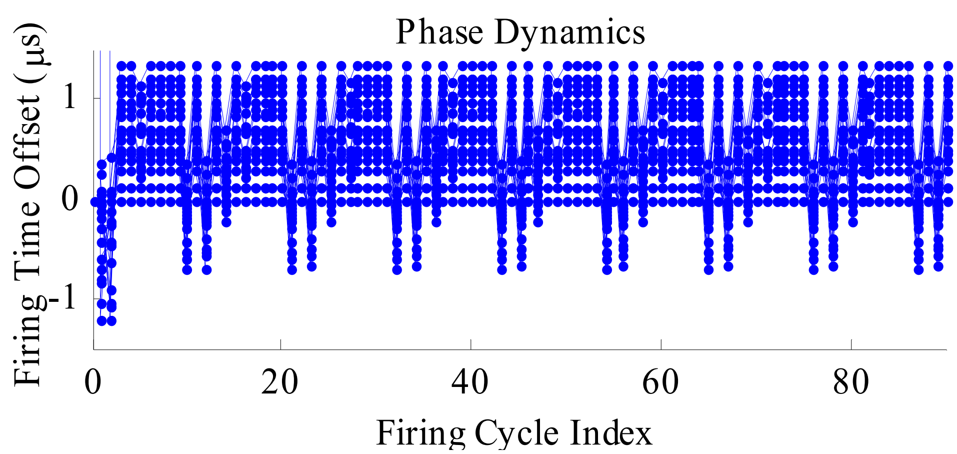



**Figure 5.12. Time of firing of all nodes relative to each time the fastest node fires (design B).**

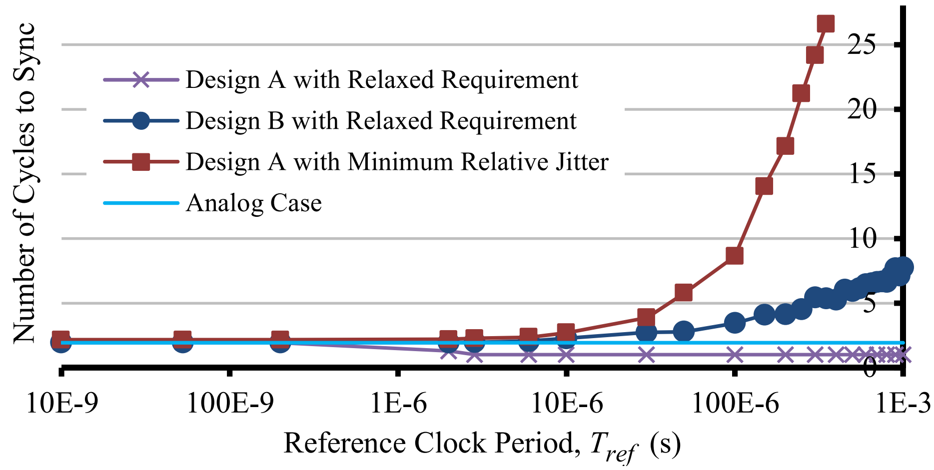



**Figure 5.13. Synchronization speed. With fast reference clocks (small $T_{ref}$), the discretization error is negligible and the performance of design A and design B approaches that of analog oscillator.**

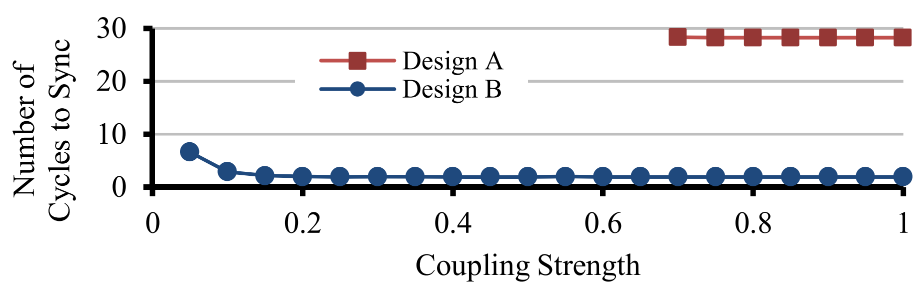



**Figure 5.14. Synchronization speed with varying coupling strength**

We investigated whether the coupling strength alters the impact of the digital PCO on synchronization speed. As shown in Figure 5.14, the synchronization speed is constant as long as the coupling strength is large enough. We also performed the same simulations with 5 nodes and the results were qualitatively similar.

In summary, design A is the better choice. First, while design B adds relative jitter on the order of 1 μs, design A not only introduces zero relative jitter contribution from the discretization, but also reduces the contribution from the period jitter of the reference clock, achieving better synchronization quality than even the analog oscillator design. However, this robust synchronization is at the expense of synchronization speed. Depending on the system parameters, the network can take as many as hundreds of cycles or more to achieve synchronization. However, if the system does not have a strict relative jitter requirement, design A is also the better choice because if we can tolerate the same relative jitter as in design B, only 2-3 cycles are necessary for design A network to pseudo-synchronize, which is fewer than that of the analog oscillator and much fewer than the tens of cycles of design B.

### 5.3.4 Digital PCO implementation

To meet the needs of this system, a PCO based on a relaxation oscillator described in design A was implemented using standard counters and digital logic. The slowing trigger clock is generated using a counter driven by the reference clock (Figure 5.15) and a static comparator logic that resets the counter when its count equals that of the main counter, which represents the oscillator. The output pulse from the comparator serves as the trigger clock signal for the main counter. A main counter that resets at count $M+1$ takes $T_{ref}(1+2+...+M) = T_{ref}M(M+1)/2$ time to reset, which is within $MT_{ref}$

of the desired $T_{PCO}$. To be able to set the oscillator period in steps of $T_{ref}$, we included a dedicated counter in Trigger Clock Timing block that blocks the reference clock for up to $MT_{ref}$ at the beginning of each cycle. The blackout counter tracks the blackout period. The main counter comprises two identical counters, one tracking the oscillator phase and the other tracking the coupling-adjusted oscillator phase so that the PCO can switch to the coupling-adjusted counter for instantaneous coupling. The former counter resets to 1 and the latter resets to a value equal to the coupling strength. The difference between design B and design A is that once a coupling to threshold is received, the PCO counters are either reset immediately (design B) or at the next reference clock (design A).

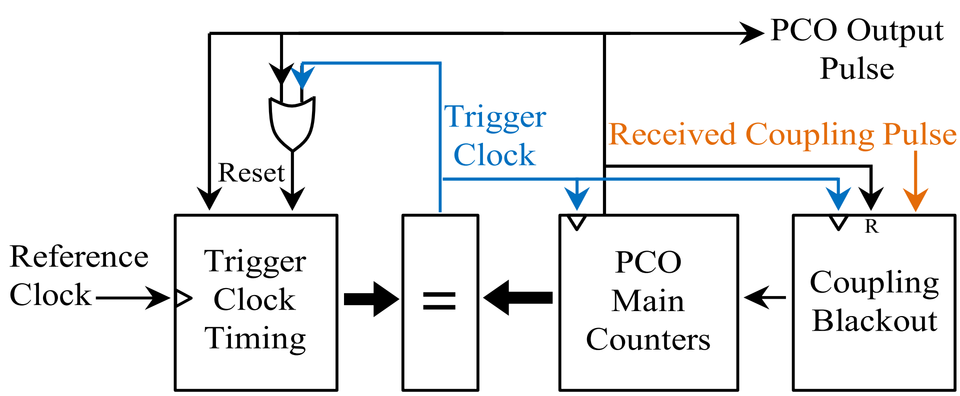



**Figure 5.15. Basic block diagram of the digital PCO**

### *5.4 Sequence generator*

The sequence generator block outputs the Kasami sequence when the PCO resets. It consists of 63 shift registers, which can be programmed to output any binary sequence. The output of this block becomes the baseband input of the commercial TX used in the RF node.

### 5.5 Measurement results

The synchronizer block was fabricated in a TSMC 65nm CMOS process and occupies 3.63 $mm^2$. The P2P RF node setup is shown in Figure 5.16. The control and clock signals are generated by an FPGA and delivered to the synchronizer chip through a motherboard and daughterboard. The commercial RF TX and RX chips (ADRF6701 and ADRF6850) sit on a custom PCB that is connected to the baseband daughterboard. In addition to testing each block in the fabricated chip, we demonstrated synchronization of a 3 node wireless mesh network in the 915 MHz ISM band using the RF node setup in Figure 5.16.

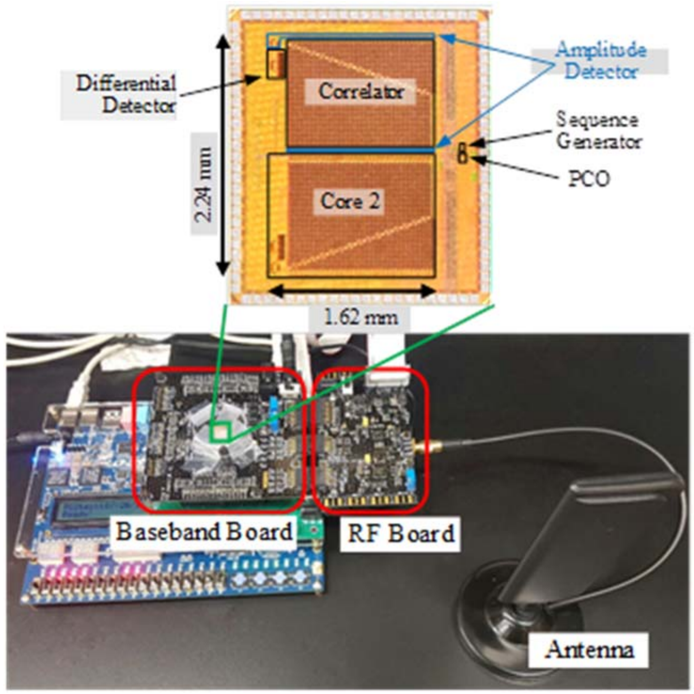



**Figure 5.16. Die and PCB photo**

The signal processor consumes $1.11\ mW$ when fully on and $13.19\ \mu W$ when operating at a duty cycle of 0.007% (Table 5.1). This is dominated by leakage, which accounts for about 7 µW.

**Table 5.1: Power consumption of synchronizer block**

| | Clock Running | SP Clock Off | Duty-cycled 70us/1s |
|---|---|---|---|
| Signal Processor | 1.04 mW | 6.19 uW | 6.26 uW |
| PCO | 6.93 uW | 6.93 uW | 6.93 uW |
| Total | 1.11 mW | 13.12 uW | 13.19 uW |

An estimate for a duty-cycled power consumption of a commercial RF front end is provided in Table 5.2. The duty-cycling is assumed to be 100us/1s in order to account for slow turn on time. The VCO in the synthesizer is assumed to take 1.5 ms to turn on and stabilize. The total power of maintaining the synchronization is $13.2\ \mu W + 41\mu W = 54.2\ \mu W$.

**Table 5.2: Power consumption of a duty-cycled RF front end**

| RF | Duty-cycled | Notes |
|---|---|---|
| TX | 10 uW | Duty-cycled 100us/1s |
| RX | 24 uW | Duty-cycle: VCO 1.5ms/1s, the rest 100us/1s |
| Xtals | 7 uW | No duty-cycle |
| Total | 41 uW | |

The data rate for the input baseband signal is $1.25\ Mbps$ for the measurements presented below ($T_{bit} = 0.8\ \mu s$). The latency is less than $2T_{bit} = 1.6\ \mu s$. More specifically, the time it takes for the peak detector to output the result once the last bit of the sequence is received at the processor input is between $T_{bit}$ and $2T_{bit}$, depending on the phase offset between the transmitter and the receiver.

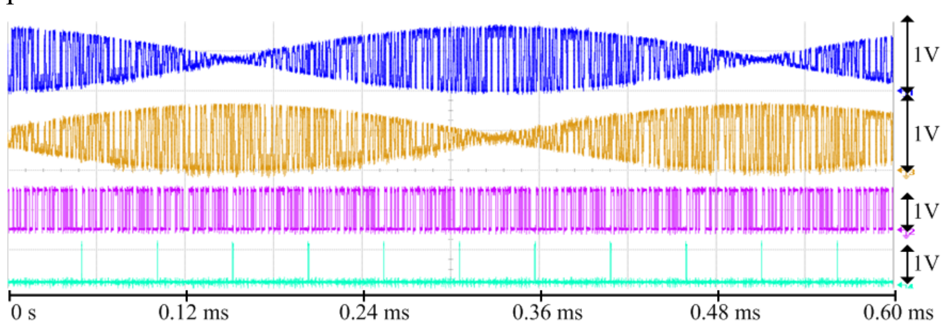



**Figure 5.17. Differential detector: 1st and 2nd waveform – I and Q inputs, 3rd waveform – digitized output, 4th waveform – peak detector output. Differential detector correctly decodes the baseband signals with carrier frequency offset**

**modulation. The envelope wavelength is deliberately shortened to show the bit transitions clearly.**

Figure 5.17 demonstrates the functionality of the differential detector. The top two waveforms are the baseband received signals on I and Q channels with an envelope due to the carrier frequency mismatch. The figure shows that the encoded sequence embedded in them flips polarity from one eye to the next. Regardless of this polarity switch, the sequence is decoded correctly as shown by the third waveform, which is the digitized output of the differential detector in Core 1. The decoded sequence is also correct when the received signal has small amplitude at the edges of the eye. The fourth waveform is the peak detector output confirming that the differential detector sent the correct decoded sequence to the correlator.

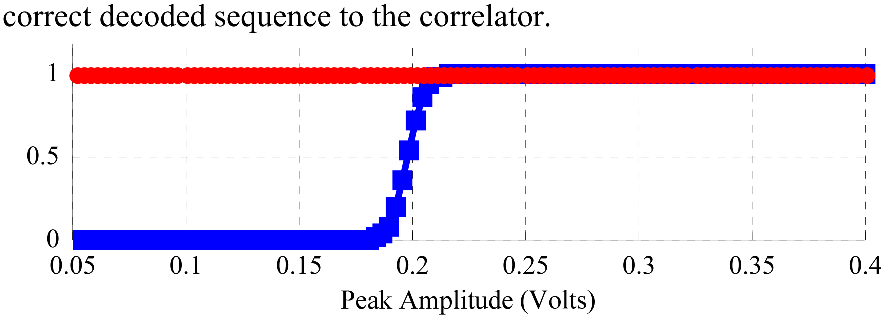



**Figure 5.18. Correlation success rate vs. input amplitude. Blue/bottom waveform has fixed $V_{thresh}$. Red/top waveform has amplitude detector turned on.**

Measurement of the amplitude detector is presented in Figure 5.18. The red/top waveform shows the success rate of the correlation when the amplitude detector is turned on. Since the threshold automatically adjusts based on the input amplitude, the correlation works for a large range of input voltages. The blue/bottom waveform plots the case without the amplitude detector, i.e. with fixed $V_{thresh}$. If the input power is too small (or too large, which is not shown in the figure), the correlation fails.

Figure 5.19 shows the dual core design. When there is a dead-zone at the output of one core due to it sampling the input signals during the transitions, the other core samples at the right phase and produces the correct output.

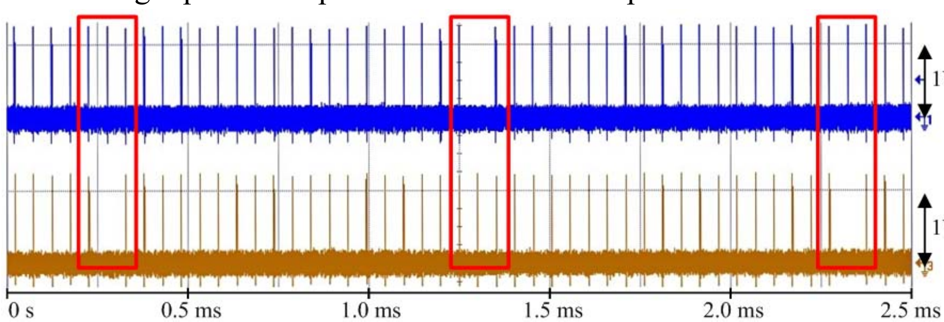



**Figure 5.19. Correlator output of Core 1 (top) and Core 2 (bottom). When one of the cores samples the input signal at bit transitions and skips a pulse (inside red rectangles), the other core samples at the center of the bits, producing the correct peaks.**

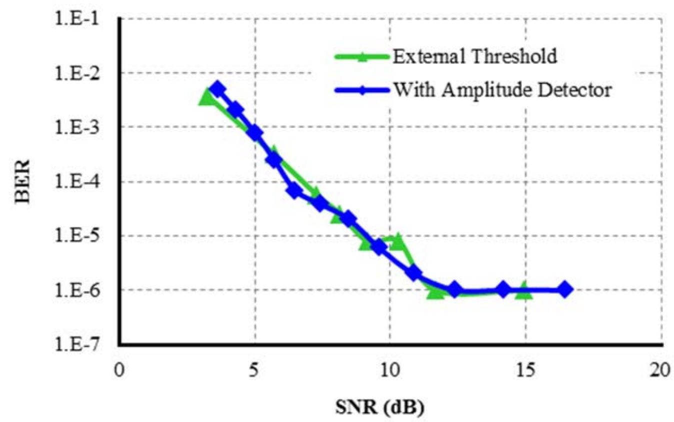



**Figure 5.20. BER vs SNR at the chip input.**

Figure 5.20 plots correlation BER vs input SNR. Since our signal processor does not include an RF front end, the BER is measured against SNR of the baseband. The setup for this measurement is shown in Figure 5.21. Assuming a commercial RF front-end with $NF = 5\ dB$ and $2\ MHz$ bandwidth, the SNR of $5\ dB$ at BER of $10^{-3}$ translates to a sensitivity of $-101\ dBm$. If the PCO nodes are placed outdoors in an urban environment at 1.5 m height and set to transmit with 0 dBm power, the path loss model in (4.2) applies and the PCO nodes should be within 67.4 m of each other in order to synchronize with BER = $10^{-3}$.

Figure 5.22 shows tolerance to in-band interference from multipath and other nodes. For distances within the 120m range (path delay less than ~400ns, i.e. half bit period) there is negligible degradation of performance from multipath reflection. This is due to the fact that the bit period is relatively long and the echo signals still arrive within a bit period, which does not degrade the correlation significantly.

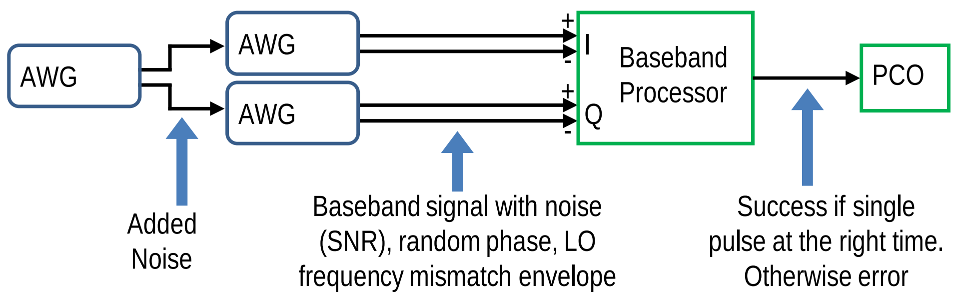



**Figure 5.21. Setup for BER measurement.**

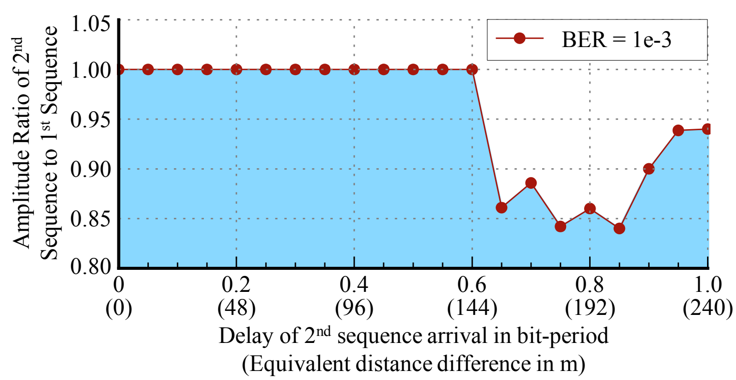



**Figure 5.22. BER in the presence of multipath. Multipath interference of equal magnitude does not significantly degrade performance up to distances of 144m of system range.**

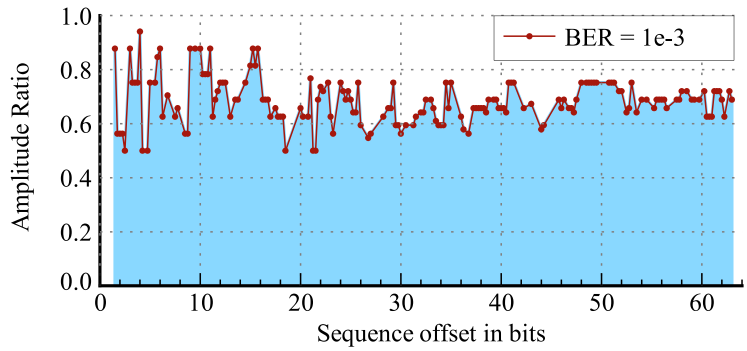



**Figure 5.23. Large in-band interferers with signal ratios outside of the blue region and with offsets between 0.6 and 63 bits can degrade detection.**

However, if the PCO period difference between the fastest (i.e. shortest PCO period) two nodes in the system is smaller than syncword length, or <51ppm in our system, their transmitted signals overlap, degrading detection (BER>1e-3) if the amplitude ratio is large (above the blue region in Figure 5.23). This can be resolved by ensuring that one node in the network has a 51ppm faster PCO and drives the nodes to synchrony.

Figure 5.24 shows the transient circuit simulation output of the main counter of the digital PCO, showing the monotonically-increasing concave-down curve. Figure 5.25 shows the transient of the digital PCO set to reset at count 16 for simplicity purposes. The top waveform is external coupling pulse. Its first pulse is ignored as it is in the blackout period (3rd waveform indicates blackout, active low). The second pulse causes the PCO (4th waveform) to reset early. 2nd waveform indicates when the PCO starts from 0.

Figure 5.26 shows the output of the sequence generator (3rd waveform). It can be seen that the sequence generator outputs the same sequence as the 1st waveform, which is the correct sequence.

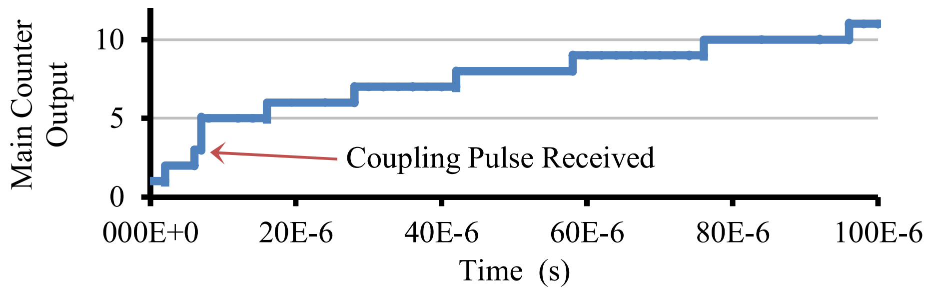



**Figure 5.24. Digital PCO circuit output – Cadence transient simulation. A coupling pulse received at *t* =7 µs advances the oscillator state**

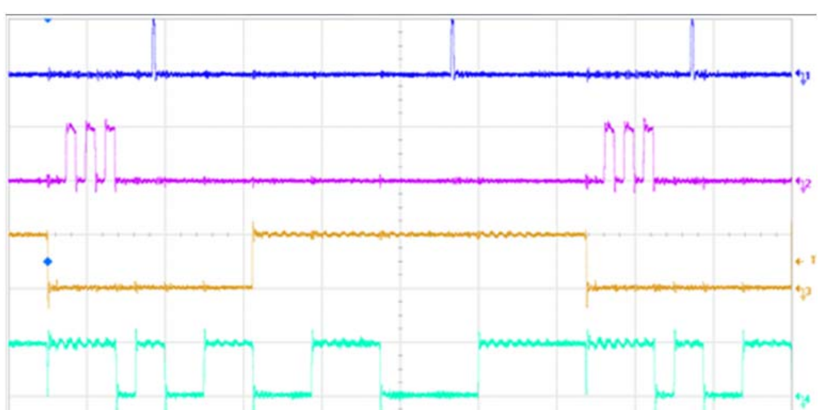


**Figure 5.25. Transient of digital PCO.**

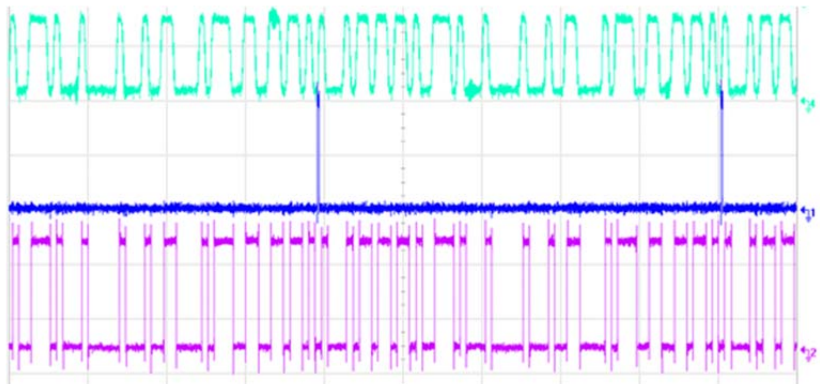


Figure 5.26. Transient of the sequence generator

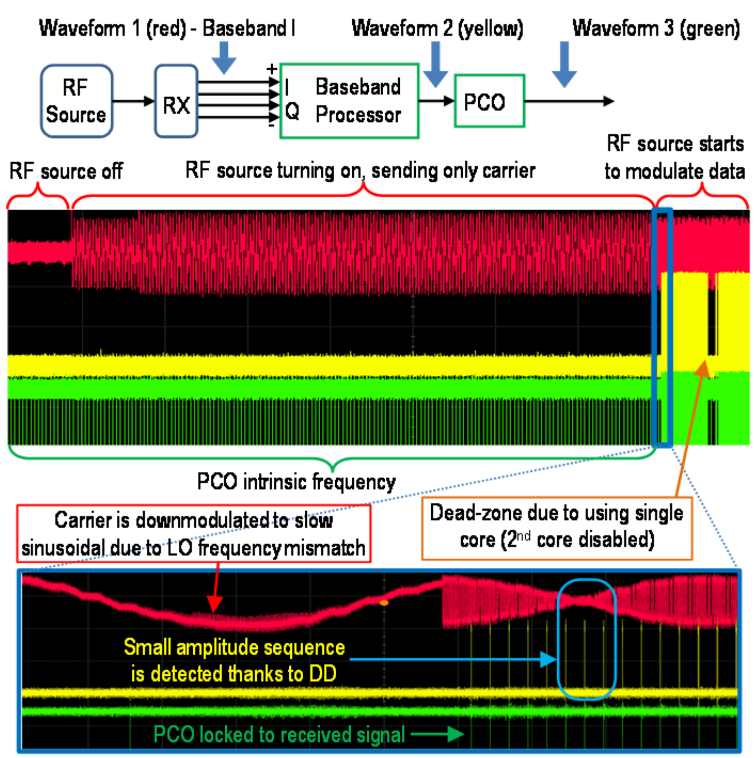



Figure 5.27. Transient of a node locking to a received signal

Figure 5.27 shows the functionality of the synchronizer. An RF source is configured to (1) turn on, (2) send carrier signal, and then (3) modulate the baseband Kasami sequence to the carrier frequency. The signal from the RF source, demodulated down to baseband, is seen as the red/top waveform in the figure. It can be seen that signal processor detects the expected sequence successfully despite phase and frequency mismatch and noise. Disabling the 2nd core resulted in the dead-zone in correlator output, which implies the baseband clock of the RF source has also different frequency compared to the FPGA generated baseband clock. The green/bottom waveform shows that the PCO is able to lock to the received signal.

Figure 5.28 shows synchronization of 3 nodes in a mesh network through an RF link at 915 MHz with off-the-shelf RX and TX chips. The top waveform shows the initial state of the three PCO's at their natural frequencies and phases. The bottom shows the synchronized PCOs after the links are activated and coupling between nodes proceeds.

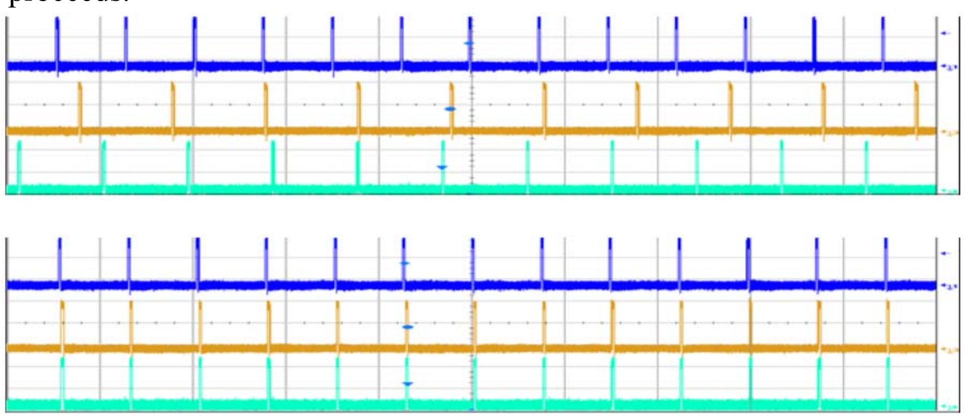


**Figure 5.28. Wireless synchronization of three nodes**

The synchronization jitter is shown in Figure 5.29. Jitter is measured to be 4e-5% of period and does not significantly degrade duty cycle. Most of the jitter is due to the baseband clock mismatch, which causes the synchronized edge to drift over half bit period (400 ns) as it is sampled by the two SP cores that sample half bit period apart.

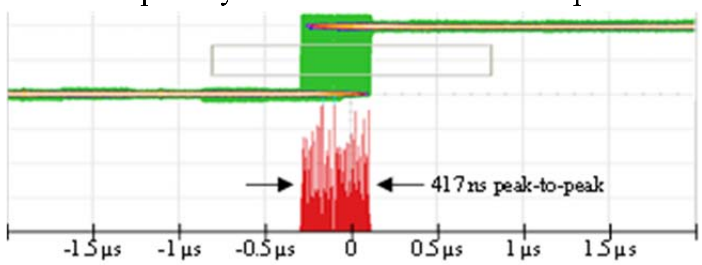



**Figure 5.29. The reset signal of synchronized PCO and the histogram of its rising edge.**

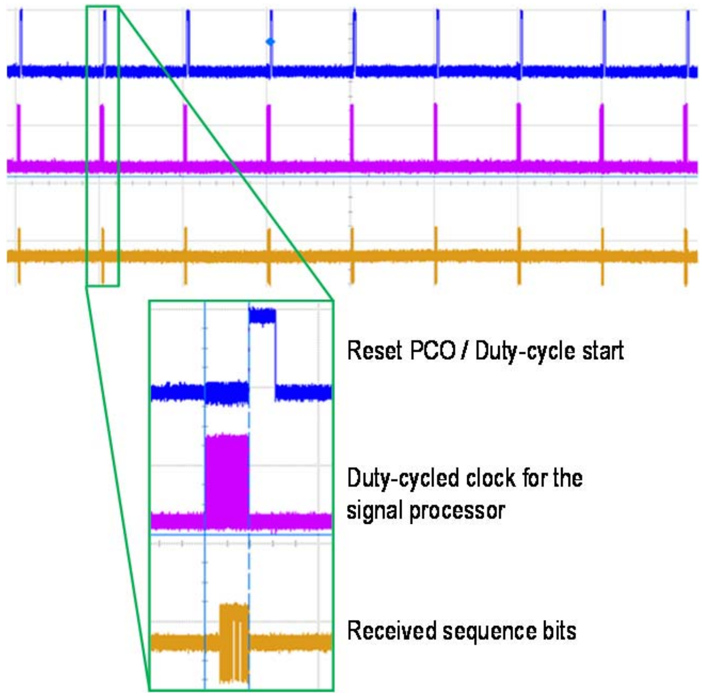



**Figure 5.30. Duty-cycled operation. System locks with low latency despite long "off" cycle and random "on" states.**

Finally, once an RF node is synchronize to the network, it duty-cycles the signal processing block and the sequence generator block to save power. The digital PCO is always on since it provides the timing for the duty-cycling. In this implementation, the duty-cycling is accomplished by stopping the clock signals. Figure 5.30 shows that the node turns on the clock right before it expects to receive a coupling pulse, receives the pulse and advances its PCO phase to threshold/reset, maintaining the network timing.

Since this architecture is unique in that it supports scalable mesh networks, the closest comparisons in performance (sensitivity, SNR, power) are made to the wake-up radios as shown in Table 5.3. Estimates of NF, RX power, BW are based upon commercial radio front end.

**Table 5.3: Comparison with the state-of-the-art wake-up radios.**

| | Sensitivity (dBm) | Power (uW) | Modulation | Data rate (kbps) | LO Freq (MHz) | Tech (nm) |
|---|---|---|---|---|---|---|
| This Work | -101[a] | 44.2[b] | BPSK | 1250 | 915 | 65 |
| [26] | -97 | 99 | OOK | 10 | 2400 | 65 |
| [27] | -87 | 45.5 | GFSK | 50 | 924.4 | 65 |
| [28] | -70 | 44 | FSK | 200 | 402 | 130 |
| [29] | -69 | 0.0045[c] | OOK | 0.3 | 113.5 | 180 |
| [30] | -56.5 | 0.236 | OOK | 8.192 | 2400 | 65 |
| [31] | -45 | 0.116 | OOK | 12.5 | 402 | 130 |

* Comparison with the state-of-the-art wake-up radios. Comparison is to RX only since other designs do not include TX power.

[a]Estimated from baseband SNR assuming conventional RX with NF=5dB and BW=2MHz

[b]13.19uW baseband chip + 31uW for RX front end duty-cycled at 0.01%, synthesizer assuming 1.5ms turn-on time, and crystal.

[c]Assumes clock and input data are synchronized

### *5.6 Conclusion*

In this chapter, we demonstrated a low-power baseband synchronizer block for emergent synchronization of P2P RF network. The architecture supports a low-latency detection of a programmable syncword in wireless nodes. It is compatible with commercial RF front ends, and can enable aggressive duty-cycling for power savings in such systems. Our measurement results show a total power consumption of 1.11 mW while processing 1.25 Mbps syncword in a fully-on mode, stand-by power of 13.12 μW, a sensitivity of -101 dBm, and latency of less than two bit periods, enabling 0.007% duty-cycled power consumption of 13.19 μW.

Although the hardware that synchronizes narrowband P2P radio is developed in this thesis, fully functional wireless P2P network with duty-cycled communication requires future work. First, a communication framework that uses the PCO reset signal as a trigger to schedule the data communication should be developed. In addition, the current duty-cycling algorithm should be expanded to handle turning on the radio during data transmission. This should take into account the fact that multi-hop nodes may reset $T_{prop}$ before/after their neighbors.

Design trade-offs affecting power consumption should be more thoroughly studied to improve power savings or increase performance. For example, reducing the syncword length reduces $T_{prop}$ and saves power and area in the signal processing block. However, it will degrade BER performance, requiring higher TX power or shortened range. If better BER performance is required, the syncword should be extended to achieve more correlation gain. However, expanding the proposed analog processor to handle longer sequence would quickly increase the power and area costs due to parasitics. Hence, all digital architecture for the signal processor should be investigated. With an ADC at the input, the exact same architecture for the differential detector, correlator and amplitude detector can be implemented in digital. Alternatively, traditional signal processing may be performed in DSP. The power and other metrics should be evaluated to determine which signal processor implementation is better.

Chapter 6 REFERENCES


[1] X. Wang, M. Tehranipoor and R. Datta, “A novel architecture for on-chip path delay measurement,” *2009 Int. Test Conf.*, Austin, TX, 2009, pp. 1-10.

[2] A. Jain, A. Veggetti, D. Crippa and P. Rolandi, “On-chip delay measurement circuit,” *17th IEEE Eur. Test Symp.*, Annecy, 2012, pp. 1-6.

[3] R. Datta, A. Sebastine, A. Raghunathan and J. A. Abraham, “On-chip delay measurement for silicon debug”, *Proc. Great Lakes Symp. VLSI*, Boston, MA, 2004, pp. 145-148.

[4] M. C. Tsai, C. H. Cheng and C. M. Yang, "An All-Digital High-Precision Built-In Delay Time Measurement Circuit," *Proc. 26th IEEE VLSI Test Symp.*, San Diego, CA, 2008, pp. 249-254.

[5] S. Ghosh, S. Bhunia, A. Raychowdhury and K. Roy, "A Novel Delay Fault Testing Methodology Using Low-Overhead Built-In Delay Sensor," in *IEEE Trans. Comput.-Aided Des. Integr. Circuits Syst.*, vol. 25, no. 12, pp. 2934-2943, Dec. 2006.

[6] K. Kato and S. Choomchuay, “An on-chip delay measurement using adjacency testable scan design,” *7th Int. Conf. Inform. Technol. and Elect. Eng.*, Chiang Mai, 2015, pp. 508-513.

[7] S. Maggioni, A. Veggetti, A. Bogliolo and L. Croce “Random sampling for on-chip characterization of standard-cell propagation delay”, *Proc. 4th Int. Symp. Quality Electron. Des.,* 2003, pp. 41-45.

[8] R. Z. Bhatti, M. Denneau and J. Draper, “Phase measurement and adjustment of digital signals using random sampling technique,” *2006 IEEE Int. Symp. Circuits and Syst.*, Island of Kos, 2006, pp. 4.

[9] M. Mansuri, B. Casper and F. O'Mahony, "An on-die all-digital delay measurement circuit with 250fs accuracy," *Symp. VLSI Circuits*, Honolulu, HI, 2012, pp. 98-99.

[10] T. Hashimoto, H. Yamazaki, A. Muramatsu, T. Sato and A. Inoue, "Time-to-digital converter with Vernier delay mismatch compensation for high resolution on-die clock jitter measurement", *IEEE Symp. VLSI Circuits*, Honolulu, HI, 2008, pp. 166-167.

[11] K. Nose, M. Kajita and M. Mizuno, "A 1-ps resolution jitter-measurement macro using interpolated jitter oversampling", *IEEE J. Solid-State Circuits*, vol. 41, no. 12, pp. 2911-2920, Dec. 2006.

[12] M. Ishida, K. Ichiyama, T. Yamaguchi, M. Soma, M. Suda and T. Okayasu, "On-chip circuit for measuring data jitter in the time or frequency domain", *IEEE Radio Frequency Integrated Circuits Symp.*, Honolulu, HI, 2007, pp. 347-350.

[13] J. Liang, M. S. Jalali, A. Sheikholeslami, M. Kibune and H. Tamura, "On-Chip Measurement of Clock and Data Jitter With Sub-Picosecond Accuracy for 10 Gb/s Multilane CDRs", *IEEE J. Solid-State Circuits*, vol. 50, no. 4, pp. 845-855, Apr. 2015.

[14] K. A. Jenkins, A. P. Jose and D. F. Heidel, "An on-chip jitter measurement circuit with sub-picosecond resolution", *Proc. Eur. Solid-State Circuits Conf.*, 2005, pp. 157-160.

[15] K. Niitsu, M. Sakurai, N. Harigai, T. J. Yamaguchi and H. Kobayashi, "CMOS Circuits to Measure Timing Jitter Using a Self-Referenced Clock and a Cascaded Time Difference Amplifier With Duty-Cycle Compensation", *IEEE J. Solid-State Circuits*, vol. 47, no. 11, pp. 2701-2710, Nov. 2012.

[16] J. D. Schaub, F. H. Gebara, T. Y. Nguyen, I. Vo, J. Pena and D. J. Acharyya, "On-chip jitter and oscilloscope circuits using an asynchronous sample clock," *Proc. Eur. Solid-State Circuits Conf.,* 2008, pp. 126-129.

[17] X. Wu *et al*., "FlashLinQ: A Synchronous Distributed Scheduler for Peer-to-Peer Ad Hoc Networks," in *IEEE/ACM Trans. Netw.*, vol. 21, no. 4, pp. 1215-1228, Aug. 2013.

[18] R. E. Mirollo and S. H. Strogatz, "Synchronization of pulse-coupled biological oscillators," *SIAM J. Appl. Math*, vol. 50, pp. 1645-1662, Dec. 1990.

[19] Y. W. Hong and A. Scaglione, "A scalable synchronization protocol for large scale sensor networks and its applications," *IEEE J. Sel. Areas Commun.*, pp. 1085–1099, May 2005.

[20] X. Wang, R.K. Dokania and A. Apsel, "PCO Based Synchronization for Cognitive Duty-Cycled Impulse Radio Sensor Network," *IEEE Sensors J.,* vol. 11, no. 3, pp. 555 - 564, Mar. 2011.

[21] E. Gantsog, D. Liu and A. B. Apsel, "0.89 mW on-chip jitter-measurement circuit for high speed clock with sub-picosecond resolution," in *Proc. Eur. Solid-State Circuits Conf.*, Sep. 2016, pp. 457-460.

[22] J. A. McNeill and D. Ricketts, "Low jitter VCO design examples" in *The Designer's Guide to Jitter in Ring Oscillators*. Boston, MA, USA: Springer-Verlag US, 2009, ch. 9, sec. 1, pp. 245-246.

[23] S. H. Hall, H. L. Heck, "Modeling and Budgeting of Timing Jitter and Noise," in *Advanced Signal Integrity for High-Speed Digital Designs*, Hoboken, NJ, USA: John Wiley & Sons, 2009, ch. 13, pp. 549-603.

[24] M. S. Corson, R. Laroia, J. Li, V. Park, T. Richardson and G. Tsirtsis, "Toward proximity-aware internetworking," in *IEEE Wireless Commun.*, vol. 17, no. 6, pp. 26-33, December 2010.

[25] *IEEE P802.11 Wireless LANs, TGah Channel Model*, IEEE 802.11-11/0968r4, Mar. 2015.

[26] C. Salazar, A. Kaiser, A. Cathelin and J. Rabaey, "13.5 A −97dBm-sensitivity interferer-resilient 2.4GHz wake-up receiver using dual-IF multi-N-Path architecture in 65nm CMOS," *2015 IEEE Int. Solid-State Circuits Conf. – Dig. Tech. Papers*, San Francisco, CA, 2015, pp. 1-3.

[27] T. Abe *et al*., "An ultra-low-power 2-step wake-up receiver for IEEE 802.15.4g wireless sensor networks," *2014 Symp. VLSI Circuits Dig. Tech. Papers*, Honolulu, HI, 2014, pp. 1-2.

[28] J. Pandey, J. Shi and B. Otis, "A 120μW MICS/ISM-band FSK receiver with a 44μW low-power mode based on injection-locking and 9x frequency multiplication," *2011 IEEE Int. Solid-State Circuits Conf.*, San Francisco, CA, 2011, pp. 460-462.

[29] H. Jiang *et al*., "24.5 A 4.5nW wake-up radio with −69dBm sensitivity," *2017 IEEE Int. Solid-State Circuits Conf.*, San Francisco, CA, 2017, pp. 416-417.

[30] N. E. Roberts *et al*., "26.8 A 236nW ???56.5dBm-sensitivity bluetooth low-energy wakeup receiver with energy harvesting in 65nm CMOS," *2016 IEEE Int. Solid-State Circuits Conf.*, San Francisco, CA, 2016, pp. 450-451.

[31] S. Oh, N. E. Roberts and D. D. Wentzloff, "A 116nW multi-band wake-up receiver with 31-bit correlator and interference rejection," *Proc. IEEE 2013 Custom Integrated Circuits Conf.*, San Jose, CA, 2013, pp. 1-4.