An FPGA-Based Phase Measurement System

Jubin Mitra, *Student Member, IEEE*, and Tapan K. Nayak

***Abstract*—Phase measurement is required in electronic applications where a synchronous relationship between the signals needs to be preserved. Traditional electronic systems used for time measurement are designed using a classical mixed-signal approach. With the advent of reconfigurable hardware such as field-programmable gate arrays (FPGAs), it is more advantageous for designers to opt for all-digital architecture. Most high-speed serial transceivers of the FPGA circuitry do not ensure the same chip latency after each power cycle, reset cycle, or firmware upgrade. These cause uncertainty of phase relationship between the recovered signals. To address the need to register minute phase shift changes inside an FPGA, we propose a design for phase measurement logic core having resolution and precision in the range of a few picoseconds. The working principle is based on subsample accumulation using systematic sampling over the phase detector signal. The phase measurement logic can operate over a wide range of digital clock frequencies, ranging from a few kilohertz to the maximum frequency that is supported within the FPGA fabric. A mathematical model is developed to illustrate the operating principle of the design. The VLSI architecture is designed for the logic core. We also discussed the procedure of the phase measurement system, the calibration sequence involved, followed by the performance of the design in terms of accuracy, precision, and resolution.**

***Index Terms*—Field-programmable gate array (FPGA), jitter, phase calculator, phase polarity, subsampling, successive approximation, synchronization, systematic sampling, VLSI, XOR-based phase detector.**

I. Introduction

EXPERIMENTS use phase information to calibrate and synchronize signals between different circuit elements. In certain experiments such as in high energy physics (HEP), preservation of phase relationship between critical signals throughout the experiment runtime is a necessary condition. In recent trend for compact implementation of full digital architectures, reconfigurable hardware technologies such as field-programmable gate arrays (FPGAs) play a very dominant role. Popular latency critical communication link standards used in HEP experiments such as gigabit transceiver and timing-trigger and control system over passive optical network technology [1], [2], are implemented in FPGAs directly. These links carry trigger and timing information needed for time-stamp generation and event building. It is essential that the latency critical protocols maintain constant phase differences in the recovered signals for the entire experiment's runtime. High-speed serial transceivers of FPGA's do not maintain constant phase shift with each round of power cycle, reset cycle, loss of lock in the transceiver [3], firmware upgrade, or aging of clock circuitry in phase-locked loop (PLL). A logic design for phase monitoring capability to register phase shift changes in the range of 20–100 ps is needed inside the FPGA circuitry. This would allow to extract the relative phase information and to recalibrate the system when needed, to maintain the constant phase relationships.

Several approaches for phase measurement have been discussed in the literature. The classical principle of using the over sampling technique is inadequate to measure a relative phase difference between the two high-frequency clocks inside an FPGA fabric, whose frequency exceeds the maximum limit supported by the fabric (<500 MHz). One of the solutions as mentioned in the works of Lu *et al.* [4], Trebbels *et al.* [5], and Rong *et al.* [6] is to sample it externally using an analog-to-digital converter (ADC) and then feed it back to the FPGA for computation. However, the technique needs an additional hardware to measure the phase difference of the internal digital clocks. Without the use of additional hardware, a phase measurement approach had been proposed by Hidvegi *et al.* [7] using the dynamic phase alignment (DPA) feature of the FPGA PLLs. The drawback of the method is the achieved resolution, which is limited to the 1/8th of the voltage controlled oscillator (VCO) frequency.

In this paper, we present a new approach for an accurate phase measurement in an FPGA using subsamples collected by the systematic sampling over XOR-based phase detector (PD) signal. The XOR-based PD introduces the least timing jitter because of the simplicity of its design [8]. It is known that use of subsampling technique causes spectral leakage. However, in our application, the use of averaging method for the phase measurement makes it less susceptible to interfere with the results [9]. A mathematical model is formulated to explain the operating principle of the phase measurement technique. The proposed technique is suitable to construct a logic core for relative phase measurement between clocks having very low to very high frequencies with precision, accuracy, and resolution in the range of a few picoseconds.

This paper is organized as follows. Section II discusses the principal components of the architecture, which include the PD, duty cycle computation, and the phase polarity detector. Sections III and IV present the details of the subsampling methodology. The procedure for phase value computation is outlined in Section V. In Section VI, we describe the instrumentation errors. This is followed by the calibration procedure in Section VII. The results are presented and

Manuscript received April 19, 2017; revised July 2, 2017 and August 21, 2017; accepted September 27, 2017. Date of publication October 25, 2017; date of current version December 27, 2017. This work was supported by CERN. *(Corresponding author: Jubin Mitra.)*

J. Mitra is with the Variable Energy Cyclotron Centre, Homi Bhabha National Institute, Kolkata 700064, India (e-mail: jm61288@gmail.com; jmitra@cern.ch).

T. K. Nayak is with the Variable Energy Cyclotron Centre, Homi Bhabha National Institute, Kolkata 700064, India, and also with CERN, CH-1211 Geneva, Switzerland.

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TVLSI.2017.2758807

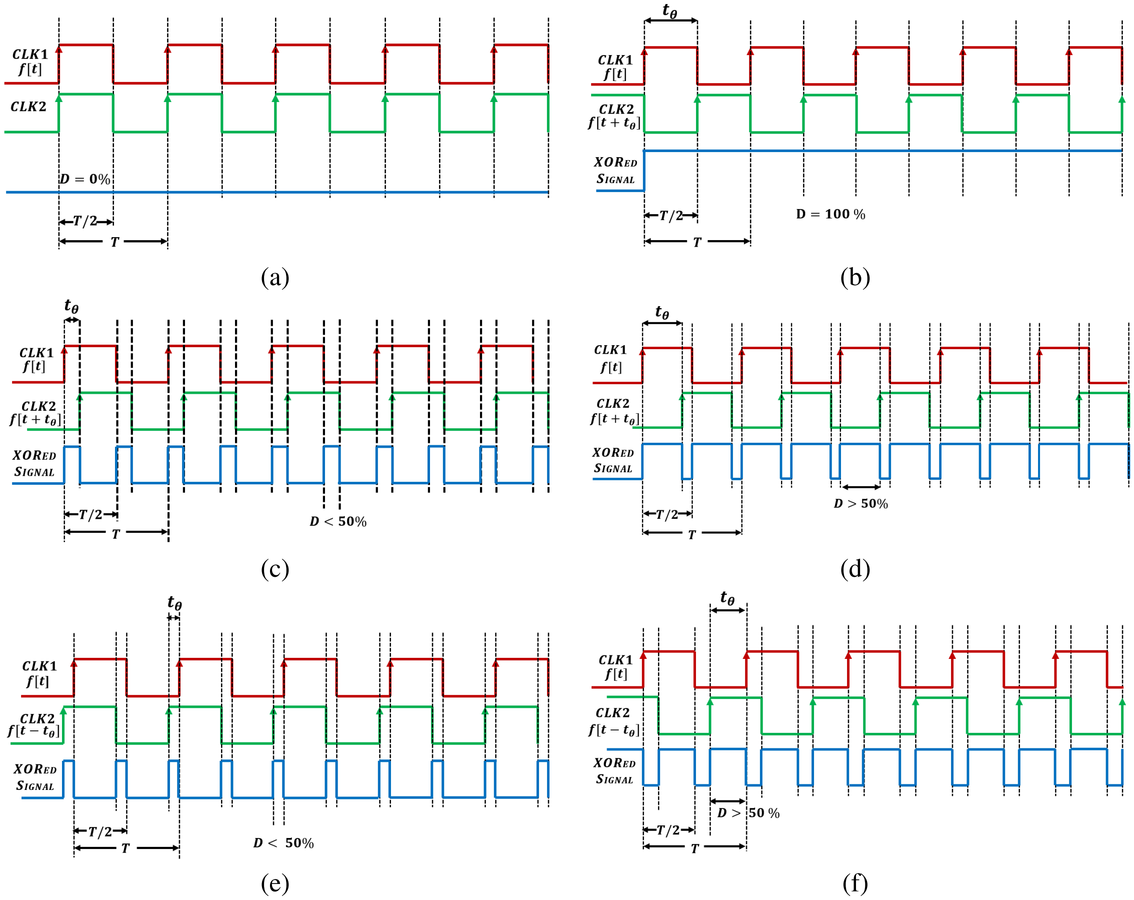


Fig. 1. Phase differences plotted as function of the duty cycle of XORed waveforms. For two signals in-phase or 180^o out of phase, the duty cycle is 0% or 100% as shown in (a) and (b), respectively. Leading phase duty cycles ($D_{\theta 1}$) are shown in (c) and (d), and lagging phase duty cycles in ($D_{\theta 2}$) (e) and (f) panels, respectively.

discussed in Section VIII. Comparison to other relevant work is also given in this section. Finally, this paper is concluded with a discussion and an outlook.

II. Design Implementation

The digital clock signals are defined by the continuous-time Heaviside step function [10], as denoted by $u(t)$. The digital clock (CLK1), having square waveform with periodicity T and pulsewidth DT centered at nT, is represented by

$$x[t] = \sum_{n=-\infty}^{+\infty} \left(u\left[t - nT + \frac{DT}{2}\right] - u\left[t - nT - \frac{DT}{2}\right]\right) \tag{1}$$

where duty cycle $D = 1/2$ or 50% and time period $T = (1/f_0)$.

Similarly, the digital clock (CLK2) having the same frequency (f_0) and shifted by the t_θ phase is given by $x[t + t_\theta]$. The phase difference between CLK1 and CLK2 is calculated by the XOR-based PD, shown in Fig. 1. The resulting XORed waveform $y[t]$ can be expressed as follows:

$$\begin{aligned} y[t] &= x[t] \oplus x[t + t_\theta] \\ &= \sum_{n=-\infty}^{+\infty} \left(u\left[t - \frac{nT}{2} + \frac{D_{\theta 1}T}{4}\right] - u\left[t - \frac{nT}{2} - \frac{D_{\theta 2}T}{4}\right]\right) \end{aligned}$$

where

$$\begin{cases} D_{\theta 1} = 0, \;\; D_{\theta 2} = \dfrac{t_\theta}{T/2} & : t_\theta < \dfrac{T}{2} \\ D_{\theta 1} = \dfrac{t_\theta}{T/2}, \;\; D_{\theta 2} = 0 & : t_\theta \geq \dfrac{T}{2} \end{cases}$$

and

$$D_\theta = D_{\theta 1} + D_{\theta 2}. \tag{2}$$

This is a pulse train of duty cycle D_θ, composed of two independent variables $D_{\theta 1}$ [shown in Fig. 1 (c) and (d)] and $D_{\theta 2}$ [shown in Fig. 1 (e) and (f)], referred to as the leading and lagging phase duty cycles, respectively. The full duty cycle D_θ is given by (2), as the ratio of the phase difference (t_θ) to the half-time period ($T/2$).

Continuous-time signals $y[t]$ and $x[t]$ are uniformly sampled by the sampling clock of time period T_s to obtain the discrete-time signals $y[kT_s]$ and $x[kT_s]$, respectively. The number of samples acquired for phase computation is referred as the sample population size (N). The ratio ($\mathcal{R}$) of the mean of N number of samples for the discretized XORed signal to the reference clock (CLK1) is given as

$$\mathcal{R} = \frac{\sum_{k=1}^{N} y[kT_s]}{\sum_{k=1}^{N} x[kT_s]}. \tag{3}$$

This ratio ($\mathcal{R}$) tends to a constant value as the sample population size (N) increases, as seen in the following equation:

$$\mathcal{R}_{N\to\infty} = \frac{D_\theta}{D} = \frac{D_{\theta 1} + D_{\theta 2}}{D}. \tag{4}$$

The phase difference t_θ can be calculated from eqrefeq:ratio, by replacing $D = (1/2)$ from (1) and $D_\theta = (t_\theta/T/2)$ from (2). The relationship between t_θ and $\mathcal{R}$ is shown below, where $\mathcal{R}$ can take any value within the range of [0–2]

$$t_\theta = \frac{T}{4} \times \mathcal{R}. \tag{5}$$

The phase shift measurement can be represented in an angular form by replacing $T/4$ with 90°

$$\tilde{\theta} = 360^\circ \times \frac{t_\theta}{T} = 90^\circ \times \mathcal{R} \tag{6}$$

where $\angle\tilde{\theta}$ lies in the range of [0°–180°].

From the measured phase difference we cannot distinguish between the leading or lagging phase shift. To resolve the phase polarity ambiguity, the reference clock (CLK1) is used to sample the phase shifted clock (CLK2) to determine whether the phase magnitude ($\tilde{\theta}$) is positive (leading) or negative (lagging) (see Fig. 1). Thus

$$\theta = \begin{cases} \tilde{\theta} & : f[kT + t_\theta] = 0 \\ -\tilde{\theta} & : f[kT + t_\theta] = 1. \end{cases} \tag{7}$$

III. Sampling Methodology

Commonly used stratified or random sampling [11] for signal processing holds no synchronous relationship between the sampling clock and the reference clock used to drive the logic signal. In this paper, we have used the concept of systematic sampling method [11] to sample the phase information signal or the XORed signal. The necessary condition for having systematic sampling is to maintain a synchronous relationship between the sampling clock (SAMPLE CLK) and the reference clock (CLK1).

Since we are systematically registering only a part of the possible sample set, we refer to the collected samples as "subsamples." This systematic sampling approach is illustrated in Fig. 2. Here, the progression through the subsamples is cyclic, which means that the subsample starts repeating itself after every Δth element in the selected sample frame.

The cyclic vector of subsamples $\mathbf{S}$ of order Δ is represented as $\mathbf{S} = \{s^0, s^1, s^2, \ldots, s^{\Delta-1}\}$, then the relationship between $s^\Delta = s^0$ holds. $\mathbf{S}$ forms a subset of the selected sampling frame or the sample population vector $\mathbf{P}$ of size N. To denote this mathematically, $\mathbf{S} \subset \mathbf{P}$ where $\Delta < N$.

From the isometric property of systematic sampling on the XORed signal, the generated sample population vector ($\mathbf{P}$) constitutes multiple sets of the cyclic vector of subsamples, $\mathbf{S}$, that repeats itself and are isomorphic to one another. The number of subsets of $\mathbf{S}$ possible from a total of $\mathbf{P}$ is expressed as the frequency ratio of the reference clock to the sampling clock, given by (8). The frequency ratio is reduced to its simplest form given by $\Delta : \Psi$, such that the greatest common divisor between the two factors is one and both factors belong to the set of natural numbers ($\mathbf{N}$). This is expressed in the form

$$\frac{F_s}{F} = \frac{\Delta}{\Psi}, \text{ where } \mathbf{gcd}(\Delta, \Psi) = 1 \text{ and } \{\Delta, \Psi\} \in \mathbf{N}. \tag{8}$$

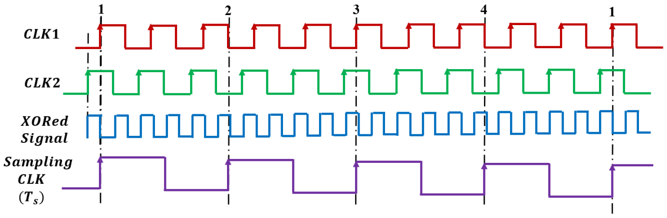


Fig. 2. Scanning of the XORed waveform using the sampling clock is illustrated for $F_s/F = 4/10$ or $\Delta/\Psi = 2/5$.

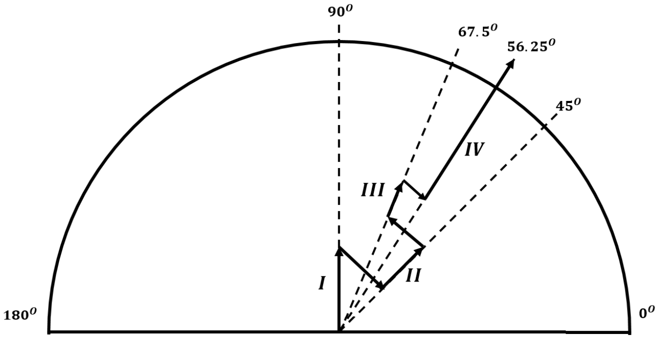


Fig. 3. Stepwise successive approximation of phase value computation for 56.25°.

Equation (8) states that the Δ unique subsamples of the XORed signal are registered by the sampling clock within the Ψ cycles of the measured reference clock. We refer to the ratio of Ψ/Δ as the sampling interval. The sampling period (T_s) is proportional to the sampling interval.

The present phase calculator is sensitive to phase magnitude variations within the range of $[0 - T/2]$, as can be inferred from (5). To register the smallest phase difference of t_θ, the minimum number of elements required in the subsample cyclic vector ($\mathbf{S}$) is given by

$$\Delta \geq \frac{T/2}{t_\theta}. \tag{9}$$

We constitute the following four constraints from the aforementioned equations that guide the proper choice of sampling frequency:

$$\left.\begin{array}{l} \Delta \geq \dfrac{1}{2Ft_\theta} \\ \mathbf{gcd}(\Delta, \Psi) = 1 \\ \Delta/\Psi = \begin{cases} > 1 & : \text{OVERSAMPLING} \\ \leq 1 & : \text{UNDERSAMPLING} \end{cases} \\ F_s = \dfrac{\Delta}{\Psi} \times F \end{array}\right\}. \tag{10}$$

The sampling of *CLK1* and *CLK2* with *SAMPLE CLK* involves asynchronous clock domain crossing, that can give rise to metastable output [12]. Hence, synchronization register chain or synchronizer is used for metastable signal to settle down, as shown in Fig. 4.

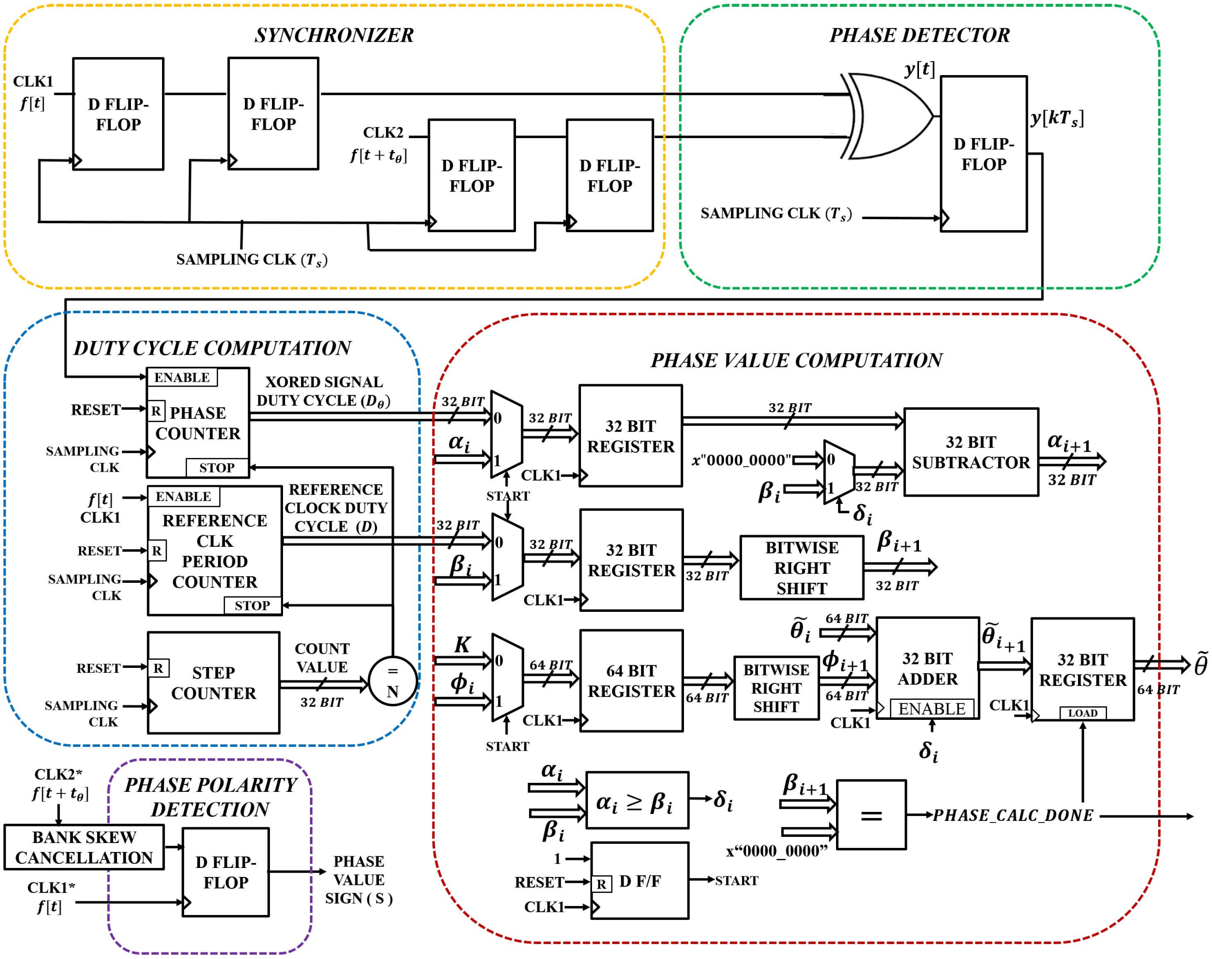
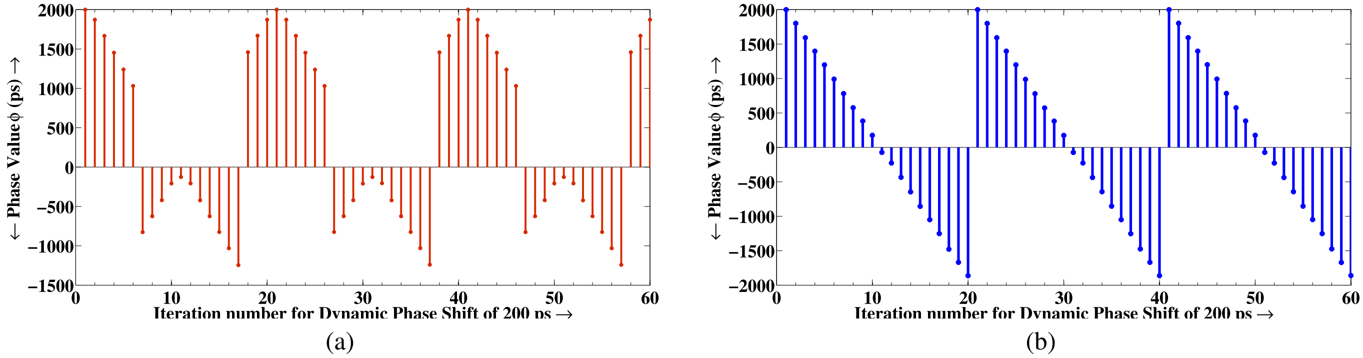


Fig. 4. VLSI architecture of phase measurement instrumentation inside FPGA.



(a)

(b)

Fig. 5. Effect of phase polarity offset. (a) When no bank skew cancellation is in effect, the magnitude and polarity are misaligned leading to measurement ambiguity. (b) When bank skew cancellation is in effect the quantization steps are properly aligned.

IV. Generating the Subsamples

During systematic sampling, there are two main methods through which one can produce more subsamples to increase the resolution of the phase measurement. These two methods are discussed in the following sections.

A. Method of Cascaded PLL

When the multiplication factor exceeds the VCO maximum output frequency capability in PLL circuitry, then the PLL frequency synthesizer fails. In that scenario, PLL-to-PLL cascading is needed. Here, we can split the multiplication factor (Δ) between the two PLLs. The cascaded PLL having the same bandwidth settings has jitter amplification effect [13]. To minimize jitter effect, Altera (now part of Intel) [14] recommends the use of a low-bandwidth setting for the source (upstream) PLL, and a high-bandwidth setting for the destination (downstream) PLL. The first (upstream) PLL acts as a jitter filter when configured as low bandwidth and transfers a very little jitter to the downstream PLL.

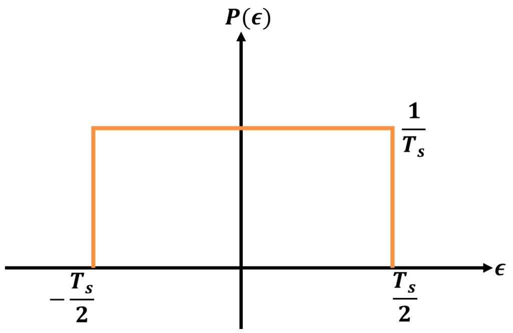


Fig. 6. PDF of quantization error (ϵ).

A high-bandwidth setting on the downstream PLL allows to track jitter from the first PLL.

To illustrate the implementation of this method, let us take, for example, a 120-MHz clock, whose phase shift of 1/100th of the total clock period is needed to be registered. Then according to (10), the number of subsamples required is 50 ($\Delta = 50$). However, the PLL multiplication factor of 50 requires VCO to operate at 6000-MHz frequency, that exceeds the maximum VCO output frequency limit of 1300 MHz in PLL circuitry for Intel Cyclone FPGAs. Hence, we split the multiplication factor between two PLLs, as $\Delta/\Psi|_{\text{PLL1}} = 5/7$ and $\Delta/\Psi|_{\text{PLL2}} = 10/17$, respectively.

B. Sampling Frequency for Subsamples Generation

The phase measurement design works for both oversampling and undersampling clock frequency to generate a particular number of subsamples (Δ), by choosing lower or higher PLL division factor (Ψ). The preference is given to the undersampling frequency, because it improves the phase noise performance of the $Sampling\ CLK$ significantly. According to the popular PLL manufacturers such as Texas Instruments [15] and Silicon Labs [16], one of the cardinal principles is that the multiplication of frequency by N causes degradation in phase noise performance by $20\log\ (N)$ with reference to the input clock, while dividing it by the same improves by the same factor [15]. In a classical PLL [17], due to the existence of the divide by M in the feedback path and the PD/CP, the divider noise amplification factor (in power) is M^2. However, the biasness toward undersampling can be avoided if subsampling technology-based PLL is used as proposed by Gao *et al.* [18].

V. Computation of Phase Magnitude

The computation of phase magnitude in time or in angular domain needs to be evaluated using (5) or (6), respectively. The generalized form of the equation is shown as

$$\mathcal{P} = K \times \mathcal{R} \tag{11}$$

where $\mathcal{P}$ is the phase value and K is the conversion factor. Depending on whether phase calculation is expressed in degrees or in time domain, the value of the conversion factor (K) is chosen as 90° or ($T/4$), respectively. The evaluation of (11) involves two operations, multiplication, and division. The division operation is used while computing the ratio $\mathcal{R}$ [see (3)] and multiplication operation is needed between K and R as given in (11).

For ease of formulation, lets represent the integrated values needed for division operation by $\alpha = \sum_{k=1}^{N} y[kT_s]$ and $\beta = \sum_{k=1}^{N} f[kT_s]$. The present value of dividend for each iteration step is denoted by (α_i) and the present value of divisor for each iteration is denoted by (β_i). The dividend is initialized by the value of α and divisor by the value of β. We have employed a shifted divisor binary-division technique [19], for calculating the division result of α/β. The steps of division method is kept tracked by the counter variable (i). The division is done by right shifting the register value of β with each iteration of the counter variable (i) and for each iteration, the divisor is stored in variable (β_i) by overwriting the past value as given in the following equation:

$$\beta_i = \beta \times 2^{-i} \tag{12}$$

The decision-making factor $\delta[i]$ in the following equation stores the comparison value between divisor (β_i) and dividend (α_i), which determines whether to continue with division operation:

$$\delta[i] = \begin{cases} \alpha_i \geq \beta_i & : -1 \\ \alpha_i < \beta_i & : 0 \end{cases} \tag{13}$$

The computed divisor value β_i is subtracted from the dividend α_i, based on evaluated value of the decision-making factor ($\delta[i]$) and at each iteration, the evaluated remainder overwrites the dividend (α_i) in the next clock cycle as denoted by α_{i+1} in

$$\alpha_{i+1} = \alpha_i + \beta_i \delta[i], \quad \alpha_0 = \alpha. \tag{14}$$

The second operation is to multiply the conversion factor (K) with the quotient from the division method. However, this would mean to wait for the quotient value computation to finish, before performing the multiplication. To make the operation parallelized, the quotient bit generated by the division method must be available simultaneously for the multiplication method to operate using the same bit. The intermediate quotient bit can be obtained by taking the modulus of the decision-making factor as given by $|\delta[i]|$. The following equation shows how multiplier value (K) is shifted synchronously with the divisor value (β) and stored in variable ϕ_i:

$$\phi_i = K \times 2^{-i}. \tag{15}$$

The sum of the product values between ϕ_i and quotient value $|\delta[i]|$ is continued for M number of iterations, to get the final result as shown in

$$\tilde{\theta} \approx \sum_{i=0}^{M} \phi_i |\delta[i]|. \tag{16}$$

The constant M indicates the number of iterations of counter variable (i) needed, so that the remainder (α_i) becomes negligible or zero, or to say it in other way, it is equal to the number of steps needed to shift bits rightward until β becomes 0.

Depending on the floating point precision required, ϕ_i must be initialized with conversion factor (K) padded with

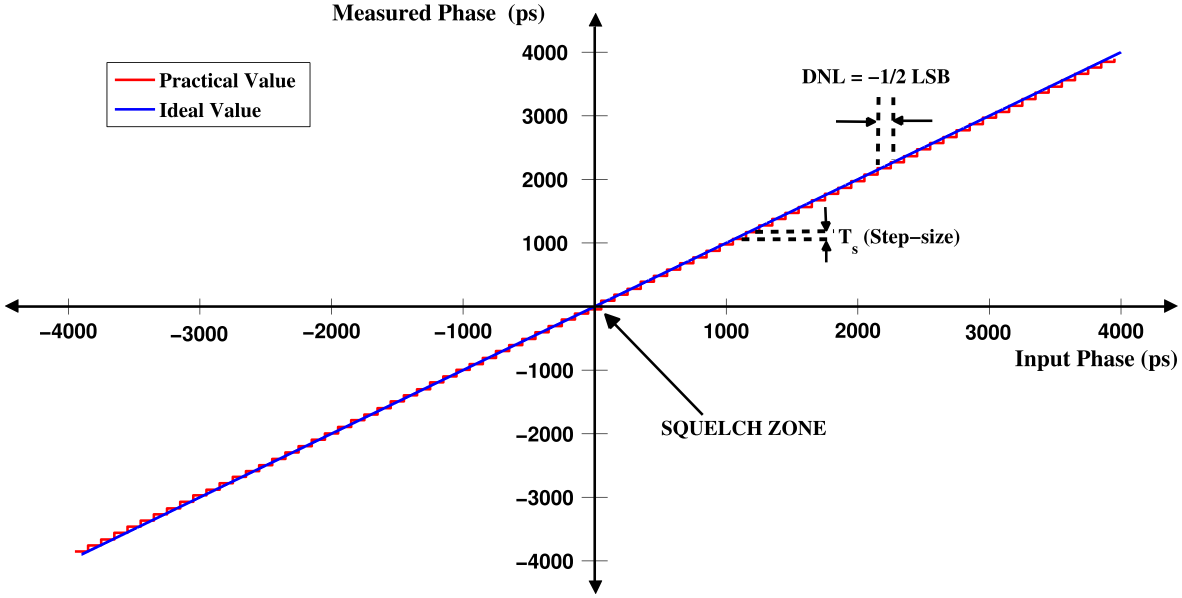


Fig. 7. Transfer characteristics of phase measurement logic core when operating in active low sample counting mode.

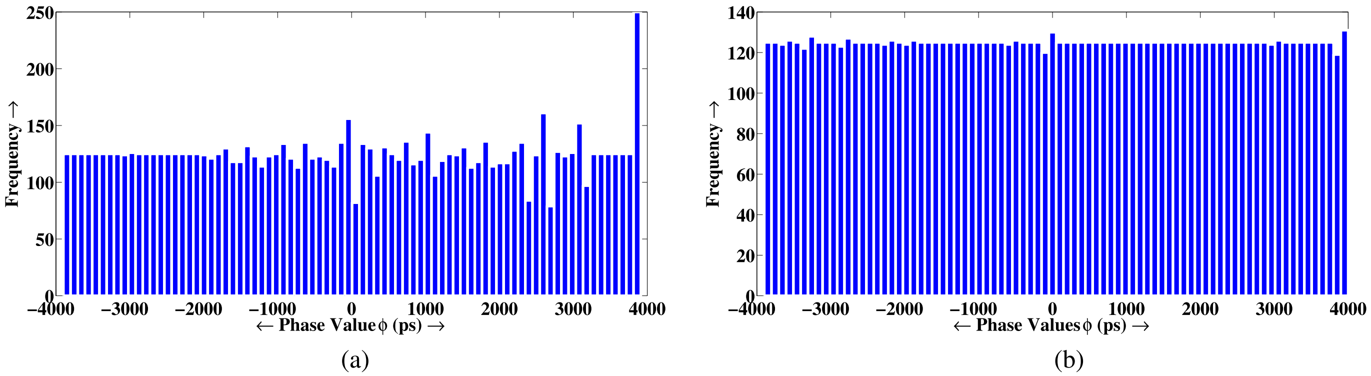


Fig. 8. Histogram of phase-stepped values. (a) Uncalibrated component. (b) Calibrated component.

zero bits. For maximum precision, the number of blank bits padded must be of equivalent size as divisor β. This method of successive approximation of phase value computation is illustrated in Fig. 3. The final phase value in our study, is stored in a 64-b register in little-endian format and represented in binary fractional form as

$$\theta = (-1)^{\text{sign}} \times \tilde{\theta} \times 2^{-32}. \tag{17}$$

We have used register size of 32 b for β. The VLSI architecture of the phase measurement system is shown in Fig. 4.

VI. Measurement Error

The phase calculator, like any other instrumentation technique, is subjected to certain kinds of errors. This can be listed as follows.

1) *Phase Magnitude Offset:* It occurs due to clock skew [$t_{\text{skew}(m)}$] between CLK1 and CLK2. The clock skew [20] is minimized by setting proper timing constraints to instruct the fitter place and route engine to do more robust timing closure for defined paths [21], and prevents variation of clock skew by large margin with each firmware upgrade.
2) *Phase Polarity Offset:* It occurs due to bank skew [$t_{\text{skew}(p)}$] between PD block and polarity detector block. Bank skew [20] is a type of skew that refers to the output skew between the signals with a single driving input terminal. Here, $t_{\text{skew}(p)} = \{t_{\text{CLK1}} - t_{\text{CLK1*}}\} - \{t_{\text{CLK2}} - t_{\text{CLK2*}}\}$. The error induced by polarity offset can be seen in Fig. 5(a).
3) *Nonideal Transfer Characteristic:* Our phase estimation logic is a typical mid-tread uniform quantizer. The quantization step is equal to the sampling interval T_s and the input–output relationship for the forward quantization step is expressed as

$$y_k = k.T_s \text{ where } k = \left\lfloor \frac{x}{T_s} \right\rfloor + \frac{1}{2}. \tag{18}$$

Finite time resolution in phase measurement introduces a quantization error between the input phase and the reconstructed output phase value. Assuming this quantization error of phase value is uniformly distributed over the step width

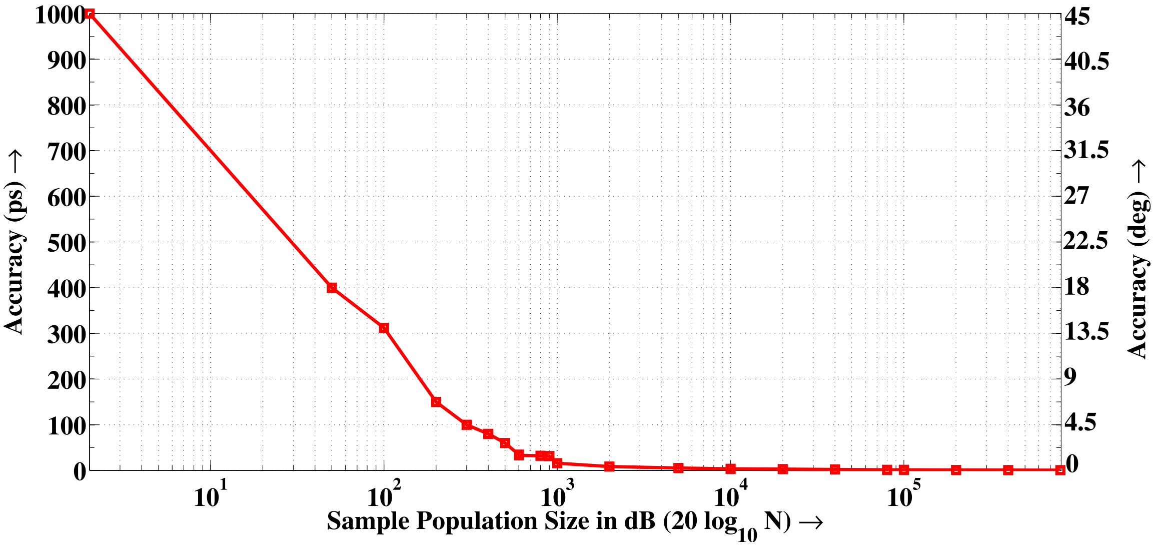


Fig. 9. Phase measurement accuracy, presented in picoseconds and in degrees, as a function of sample population size (N).

from $-1/2$ LSB to $+1/2$ LSB as shown in Fig. 6, the mean value is equal to 0 and the variance is

$$E(\epsilon^2) = \frac{1}{T_s}\int_{-T_s/2}^{T_s/2} \epsilon^2 \, d\epsilon = \frac{T_s^2}{12}. \tag{19}$$

Mid-tread quantizer shows symmetric dead-zone behavior around 0 or the end region zone, which is termed as squelch zone as shown in Fig. 7. In this zone, the samples collected is too less to make decisions, whether to consider it as, very low phase difference or suppress it as noise. There also exists another kind of nonlinearity, known as differential nonlinearity (DNL), where the actual step size shows deviation from the recorded step size during phase measurement. The nonlinearities in the transfer characteristics cannot be fully eliminated, however, the nature remains fixed for a particular sample population size (N). Using the least squares fit method to the acquired data set, we can fit a calibration curve that exhibits a linear response characteristics with a constant slope factor of K/D, as can be derived from (4) and (11), respectively.

VII. Calibration

The phase measurement technique needs a calibration procedure to eliminate the offset due to random and systematic effects. A linear phase step method is used to calibrate the component. In Intel chips, DPA technology [22] allows phase step resolution of (1/8th) of VCO frequency. Dynamic phase shift method is used to detect the amount of bank skew correction needed to compensate for phase polarity offset

$$\text{Bank skew cancellation delay} = -t_{\text{skew}(p)}. \tag{20}$$

Fig. 5(a) shows the effect of phase polarity offset. This is corrected by adding bank skew cancellation as indicated in Fig. 4 to get linear response characteristics as shown in Fig. 5(b). In our test case, we have used −250-ps bank skew cancellation.

The effect of nonlinearity in the phase measurement transfer characteristic is indicated in Fig. 7. The readings are taken at a step interval of 100 ps over 10 000 repetition cycle. Fig. 8(a) shows without calibration how phase magnitude demonstrates nonuniform distribution, due to squelch zone and DNL. The quantization error is corrected using a software-based lookup table method. Then after calibration, Fig.8(b) demonstrates uniform phase magnitude distribution.

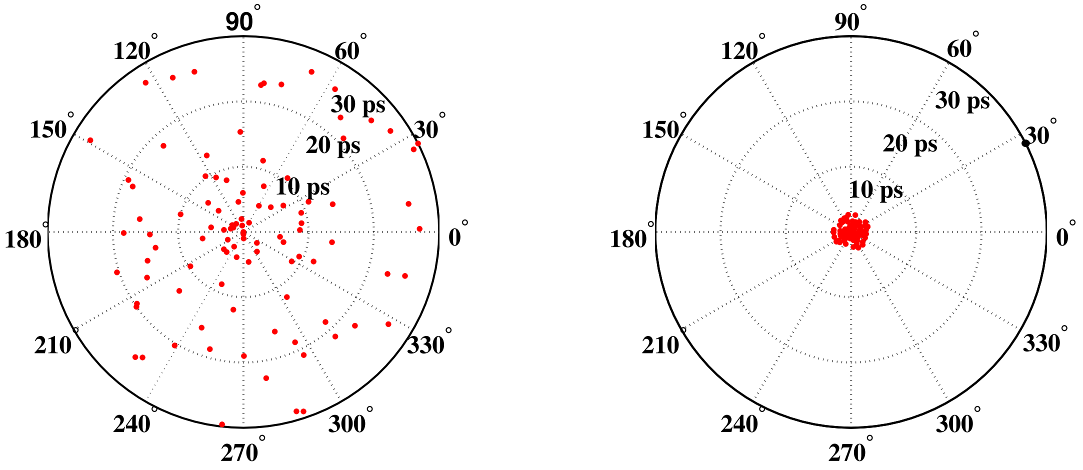
Static calibration methods are used to compensate for systematic errors. Dynamic calibration method is needed to compensate for random errors, caused by the timing skew shift due to temperature and voltage drift. The procedure for dynamic calibration is not discussed, as in our case, the board is kept in controlled environment. However, the averaging process involved in computation step suppresses the errors due to random effects.

VIII. Results

A. Measurement Setup

The course phase shift of minimum granularity in the order of 70–100 ps is asserted by the DPA feature of the PLL, where the minimum phase shift allowed limited by the 1/8th of the VCO frequency within FPGA. The finer phase shift in the order of 8–30 ps is inserted by variation of the cable length to adjust the propagation delay. For characterizing our phase measurement system in low-end FPGA, we have used Intel Cyclone IV FPGA board to register phase shift in *250-MHz clock* with a minimum resolution of 100 ps. Our phase calculator is set to collect subsamples (Δ) of 119 within a sample population size (N) of 10 000. We have quantified the phase measurement system performance using accuracy, precision, and resolution.

1) Accuracy estimates the closeness of the measured value to the actual or true value. Systematic errors affect the accuracy involved in a system [23]. Sample population size (N) is one of the system parameters that plays



(a)

(b)

Fig. 10. Graphical bullseye representation of accuracy and precision. (a) High accuracy and low precision. (b) High accuracy and high precision.

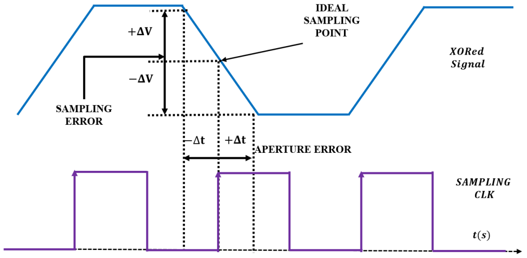


Fig. 11. Effect of sampling on precision of the system.

a dominant role in determining the operating accuracy of the phase measurement system as shown in Fig. 9. The total system response time increases in the order of $O(N + \log_2 N)$ as N increases. The response time includes the combined effect of sample accumulator and phase computation block as shown in (3) and (16), respectively.

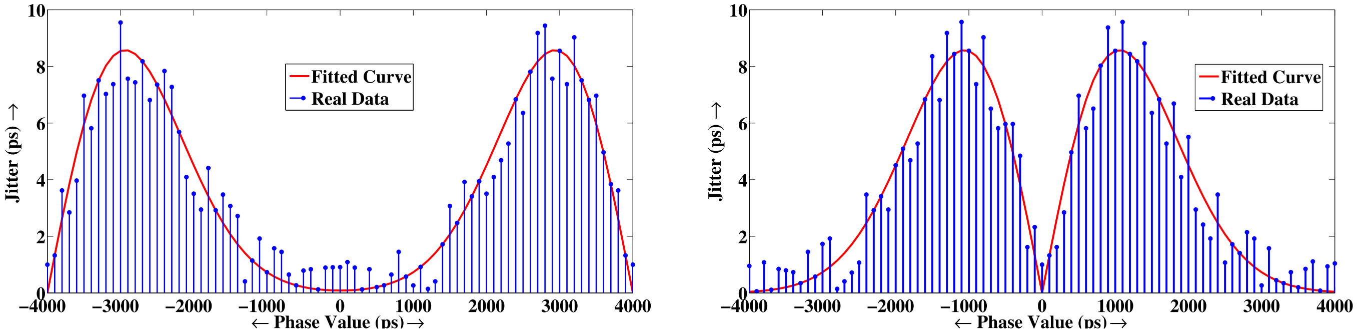
2) Precision measures the spread of the readings when they are repeated. Random errors limit the precision of a system; hence, it can never be zero [23]. However, at higher sample population, the precision of the system improves. The effect is visualized in a more graphical way by a bullseye representation showing only 100 readings. For lower sample population size of $N = 10^3$, the accuracy is moderately high but precision is low as shown in the polar plot in Fig. 10(a), and for higher sample population size of $N = 10^4$, both the accuracy and the precision is high as plotted in Fig. 10(b).

The precision distribution of the phase measurement system [24] is affected by two categories of errors as illustrated in Fig. 11. The sampling error occurs when the XORed signal undergoes systematic sampling by the *Sampling Clk* and leads to subsample variation. The aperture error happens when erroneous sample values accumulate over a set of sample population acquisition and lead to cumulative error in the computed average value.

According, to the central limit theorem, the sampling distribution of the sampling means approaches the Gaussian or Normal distribution as the sample population size increases, irrespective of the probability distribution [25]. The probability distribution function (PDF) of the sampling error distribution be represented as $P(x)$. The PDF of aperture error be represented as $P(y)$. For simplicity, let us assume that both the PDFs are a Gaussian distribution with mean 0 and standard deviation σ, denoted by $N(0, \sigma)$. Since both the error sources are statistically independent Gaussian distribution, their joint PDF is written as $P(x, y) = P(x)P(y) = (1/2\pi\sigma^2)e^{-(x^2+y^2/2\sigma^2)}$. The envelope of the sum of two quadrature Gaussian noise signals obeys a Rayleigh distribution. Solving for the joint probability distribution, by transforming it into polar coordinates (r, ϕ), we get

$$\begin{aligned} P(r,\phi) &= P(\phi)P(r) \\ &= (\text{Uniform Distribution}) \times (\text{Rayleigh Distribution}) \\ &= \frac{1}{2\pi} \times \frac{r}{\sigma^2}e^{-\frac{r^2}{2\sigma^2}} : \quad r \in [0, \infty], \phi \in [-\pi, \pi]. \end{aligned} \tag{21}$$

Histogram of precision distribution plotted over population size of 10 000 in the range of [−4000 to 4000 ps] is shown in Fig. 12. The skewness of the precision distribution curve exhibits different behavior for active low sample counting mode and for active high sample counting mode. For active low sample counting mode, the skewness is toward the minimum value of 0 ps, and for active high sample counting mode, the skewness is toward full-scale reading value of 4000 ps as plotted in Fig. 12(a) and (b). Either of the plots shows symmetric behavior across center phase value of 0 ps because the magnitude response remains same, whereas only the polarity sign changes. To obtain the nature of the distribution, we have chosen a sample population size of 50×10^3, owing to the fact that for more number of



(a)

(b)

Fig. 12. Precision distribution for 50×10^3 sample population size operating in (a) active high sampling counting mode and (b) active low sampling counting mode.

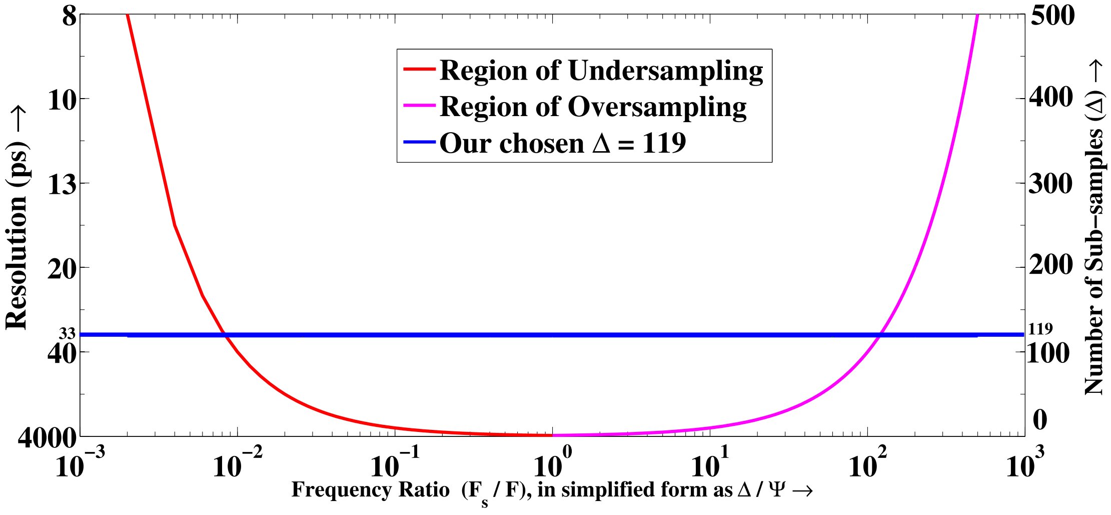


Fig. 13. Dependence of resolution (ps) and number of subsamples (Δ) with frequency ratio (F_s/F).

TABLE I

FPGA-Based Phase Detection Methods

Parameters	**Proposed Method**		**Hidvégi *et. al.*** [7]	**Lu *et. al.*** [4]	**Trebbels *et. al*** [5]	**Rong *et. al.*** [6]	**Guntoro *et. al.*** [26]
Sampling Unit	FPGA based Internal Sampling		FPGA based Internal Sampling	ADC based Dual Channel Data Sampling	Delta Modulator (DM)	ADC based Sampling	ADC based Sampling
Phase Detection Technique	Sampling Clock Scanning		Step-wise Dynamic Phase Drift	Phase Difference between I and Q channels	DM-Integrator Triggered Feedback	I-Q channel based PD using CORDIC	Phase Difference between I and Q channels
Sampling Technique	Under-Sampling / Over-Sampling		Under-Sampling	Over-Sampling	Over-Sampling / Under-Sampling	Over-Sampling	Over-Sampling
Max. Freq. Used	250 MHz		50 MHz	200 MHz	10 MHz	1 MHz	100 MHz
Detection Range	$-180^0 - 180^0$		$0 - 180^0$	$0 - 360^0$	$0 - 180^0$	$-180^0 - 180^0$	$-180^0 - 180^0$
No. of Samples Collected	33×10^4		33×10^4	0.24×10^4	100×10^4	0.013×10^4	0.1024×10^4
Max. Sampling Rate	Any (Depend on FPGA Fabric)		33.33 MHz	3 GSamp / sec	500 MHz	100 MHz	400 MHz
Accuracy	**CYCLONE IV** 0.27^0 ($3\ ps$)	**ARRIA 10** 0.09^0 ($1\ ps$)	0.1^0 ($2.78\ ps$)	0.01^o ($0.13\ ps$)	-	0.05^0 ($139\ ps$)	0.00273^0 ($0.076\ ps$)
Precision	0.63^0 $(7\ ps)^{\#}$	0.18^0 $(2\ ps)^{\#}$	0.044^0 ($2.4\ ps$)	0.0032^o ($0.044\ ps$)	-	0.0094^0 ($26\ ps$)	0.00314^0 ($0.872\ ps$)
Resolution	3.06^0 $(34\ ps)^{\S}$	0.72^0 $(8\ ps)^{\S}$	1.8^0 ($\geq 100\ ps^*$)	0.021^o ($0.3\ ps$)	0.01^0 ($2.78\ ps$)	0.05^0 ($139\ ps$)	0.00195^0 ($0.054\ ps$)

Depends on sample population size, § Depends on number of sub-samples , * Bounded by Phase Drift Step size of DPA

N.B.: Here the absolute measurement in ps should be considered, not the degrees for comparison metrics

samples averaging effect suppresses the relative uncertainty and makes it hard to decipher the distribution, while for less number of samples, too little statistics are gained to determine the nature of the distribution.

3) Resolution indicates the ability of the phase measurement system to detect and display the smallest changes in the characteristic of the results. It is limited by the step size of the quantization step. The number of quantization

steps involved in the measurement is determined by the number of subsamples (Δ) collected by the system. For finer resolution, more number of subsamples need to be collected and quantization error of the system also reduces. To operate the phase measurement system in a particular resolution, the user can operate both in undersampling or in oversampling mode by properly choosing the frequency ratio, as shown in Fig. 13.

B. Comparison

We have used Intel's Cyclone IV FPGA to characterize our phase component in low-end FPGAs with 10 000 data sets, where measurement resolution we can achieve is around 34 ps. For high-end FPGAs such as Stratix V and Arria 10, the resolution goes to 8–10 ps depending on the input clock jitter. High-end FPGA circuitry got better jitter performance leading to better performance of our phase measurement system. A detailed comparison of our proposed design competency against other recent works is shown in Table I. Our presented architecture outperforms other *in situ* FPGA-based PD implementation in resolution and also in maximum operational frequency. The performance results are reasonably competitive with other *ex situ*-based implementations.

IX. Conclusion

In this paper, we have discussed the development of a sensitive phase detection logic core for FPGA, having precision, accuracy, and resolution in the range of a few picoseconds. This can be used within FPGA as a monitoring device of phase relationship between digital clock pulses, without any additional circuitry. The design is modularized in a way that allows designers to modify different components for more robustness of the design, like replace XOR-based PD with other phase comparator. The concept of using systematic sampling for subsample collection can also be extended to map other complex analog domain problem to digital domain.

References

[1] J. Mitra, S. A. Khan, S. Mukherjee, and R. Paul, "Common readout unit (CRU)—A new readout architecture for the ALICE experiment," *J. Instrum.*, vol. 11, no. 3, p. C03021, 2016.

[2] J. Mitra *et al.*, "GBT link testing and performance measurement on PCIe40 and AMC40 custom design FPGA boards," *J. Instrum.*, vol. 11, no. 3, p. C03039, 2016.

[3] X. Liu, Q.-X. Deng, and Z.-K. Wang, "Design and FPGA implementation of high-speed, fixed-latency serial transceivers," *IEEE Trans. Nucl. Sci.*, vol. 61, no. 1, pp. 561–567, Feb. 2014.

[4] S. Lu, P. Siqueira, V. Vijayendra, H. Chandrikakutty, and R. Tessier, "Real-time differential signal phase estimation for space-based systems using FPGAs," *IEEE Trans. Aerosp. Electron. Syst.*, vol. 49, no. 2, pp. 1192–1209, Apr. 2013.

[5] D. Trebbels, D. Woelki, and R. Zengerle, "High precision phase measurement technique for cell impedance spectroscopy," *J. Phys., Conf. Ser.*, vol. 224, no. 1, p. 012159, 2010.

[6] L. Rong *et al.*, "FPGA-based amplitude and phase detection in DLLRF," *Chin. Phys. C*, vol. 33, no. 7, p. 594, 2009.

[7] A. Hidvégi *et al.*, "FPGA based phase detector for high-speed clocks with pico-seconds resolution," in *Proc. IEEE Nucl. Sci. Symp. Med. Imag. Conf. (NSS/MIC)*, Oct./Nov. 2013, pp. 1–4.

[8] D. W. Kuddes, "Method and system for aligning the phase of high speed clocks in telecommunications systems," U.S. Patent 5 638 410, Jun. 10, 1997.

[9] H. Choi, A. V. Gomes, and A. Chatterjee, "Signal acquisition of high-speed periodic signals using incoherent sub-sampling and back-end signal reconstruction algorithms," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 19, no. 7, pp. 1125–1135, Jul. 2011.

[10] A. V. Oppenheim, R. W. Schafer, and J. R. Buck, *Discrete-Time Signal Processing*. Delhi, India: Pearson Education India, 1999.

[11] S. K. Thompson, *Sampling*. Hoboken, NJ, USA: Wiley, 2012.

[12] J. Stephenson, D. Chen, R. Fung, and J. Chromczak, "Understanding metastability in FPGAs," Altera Corp., San Jose, CA, USA, White Paper WP-01082-1.2, 2009.

[13] D. T. L. Chow, V. K. Tsui, L. H. Lim, and M. O. Wong, "Jitter estimation in phase-locked loops," U.S. Patent 8 170 823, 2012. [Online]. Available: http://www.google.sr/patents/US8170823

[14] *Altera Phase-Locked Loop (Altera PLL) IP Core User Guide*, Altera Corp., San Jose, CA, USA, 2015.

[15] C. Barrett, "Fractional/integer-N PLL basics," Wireless Commun. Bus. Unit, Texas Instrum., Dallas, TX, USA, Tech. Brief SWRA029, 1999.

[16] "Clock division with jitter and phase noise measurements," Silicon Labs, Austin, TX, USA, White Paper Clock-Division-WP-Rev 1.0, 2015.

[17] *Fundamentals of Phase Locked Loops (PLLs), MT-086 Tutorial*, Analog Devices, Norwood, MA, USA, 2009.

[18] X. Gao, E. A. M. Klumperink, M. Bohsali, and B. Nauta, "A low noise sub-sampling PLL in which divider noise is eliminated and PD/CP noise is not multiplied by N^2," *IEEE J. Solid-State Circuits*, vol. 44, no. 12, pp. 3253–3263, Dec. 2009.

[19] K. R. Wadleigh and I. L. Crawford, *Software Optimization for High Performance Computing: Creating Faster Applications*. Englewood Cliffs, NJ, USA: Prentice-Hall, 2000.

[20] *Definition of Skew Specifications for Standard Logic Devices*, JEDEC Solid State Technol. Assoc., Arlington, VA, USA, 2003.

[21] "Using timing analysis in the Quartus II software," Altera Corp., San Jose, CA, USA, Appl. Note A-AN-123-02, 2001.

[22] "The need for dynamic phase alignment in high-speed FPGAs," Altera Corp., San Jose, CA, USA, White Paper WP-STXGXDPA-1.1, Feb. 2004.

[23] J. R. Taylor, *Introduction to Error Analysis: The Study of Uncertainties in Physical Measurements*. Berkeley, CA, USA: University Science Books, 1997.

[24] R. S. Turgel and D. F. Vecchia, "Precision calibration of phase meters," *IEEE Trans. Instrum. Meas.*, vol. IM-36, no. 4, pp. 918–922, Dec. 1987.

[25] V. K. Rohatgi and A. K. M. E. Saleh, *An Introduction to Probability and Statistics*. Hoboken, NJ, USA: Wiley, 2015.

[26] A. Guntoro, P. Zipf, O. Soffke, H. Klingbeil, M. Kumm, and M. Glesner, "Implementation of realtime and highspeed phase detector on FPGA," in *Proc. Int. Workshop Appl. Reconfigurable Comput.*, 2006, pp. 1–11.

Jubin Mitra (S'15) received the B.Tech. degree in electronics and communication engineering from the Heritage Institute of Technology, Kolkata, India, in 2011 and the M.Tech. degree from Bengal Engineering and Science University, Kolkata, in 2013. He is currently working toward the Ph.D. degree at the Variable Energy Cyclotron Centre under the aegis of the Homi Bhabha National Institute, Kolkata.

He is a Visiting Research Engineer with CERN, Geneva, Switzerland, where he is involved in firmware development with Common Readout Unit for ALICE collaboration at CERN.

Tapan K. Nayak received the Ph.D. degree in physics from Michigan State University, East Lansing, MI, USA, in 1990.

He was a Postdoctoral Research Fellow at Columbia University, New York, NY, USA, and a Guest Scientist at CERN, Geneva, Germany, and GSI, Darmstadt, Germany. He has been a Scientist with the Variable Energy Cyclotron Centre, Kolkata, India, and a Professor of the Homi Bhabha National Institute, Mumbai, India, a deemed university. He is currently a Scientific Associate with CERN, where he holds the position of Deputy Spokesperson of the ALICE collaboration with the CERN Large Hadron Collider.