Design of Configurable Multiphase Clock Generation and Frequency Measuring Circuit

[1]A.Madhuramalli, [2]Mr B. Muthukumaran

[1]PG Scholar, [2]Assistant Professor
SRM University, Kattankulathur
[1]Madhuramalli.a@gmail.com

***Abstract*—Design and implementation of process resilient multiple phase clock generator plays a major role in lots of applications over in communication and signal processing arena. we are planning to design a highly efficient multiple phase clock generator. The configurable platform utilizes logics efficiently and perform the required functions for various applications. If certain part of the function is required we can simply configure the bits and we implement the design. The repeatable, self healing technique adopts the signals to be generated in a constant manner , and the per phase delay is being reduced than 100ps.The application using the shifted clocks also developed here, we propose a cymmometer system where the unknown frequency is determined by using the standard frequencies and the phase shifted clocks are applied over here for range determination**

I. INTRODUCTION

A sequence of multiphase clock signals is often needed in a timing control unit to support multiple sampling operation used in numerous applications, such as the high-speed serial data-link, LCD controller, the timing controller for pulse-based radars, etc. A conventional multiphase clock generator (MPCG) is composed of delay-locked loops, or a voltage control oscillator . However, the multiphase signals in those conventional MPCG could be susceptible to the process variation that leads to larger phase errors. In this brief, we aim to develop a cell-based calibration methodology to alleviate this problem. Traditionally, an MPCG design can be realized by an architecture shown in Fig. 1. We assume that there are four phases of clock signals to be generated in this example. It incorporates four identical tunable delay elements (TDE) with the same digital tuning code. As a consequence, their delays are all the same hypothetically. Through a search process conducted by the controller, a proper delay of the four TDEs jointly will be found to make the final output signal $\varphi4$ in line with the original clock signal $\varphi0$. When this property holds true, it follows that the output signals of the four TDEs, namely $\{\varphi1,\varphi2, \varphi3, \varphi4\}$ form a sequence of clock signals with equally shifted timing intervals at their rising edges. This type of architecture has two major limitations. First, it is operated under the assumption that all TDEs exhibit an identical delay. In practice, their delays vary in silicon even under the same tuning code and their differences translate directly into the timing errors of the clock signals generated. If the timing margins for the errors are sufficiently large, this could be tolerated. In some other cases requiring higher timing resolution, however, this will pose as a serious limitation. Second, the minimum delay across a TDE is another limiting factor. For example the practical radar system-on-a chip (SoC) design we plan to support has 16 phases of clock signals with 100 ps of timing difference between any two consecutive phases of clock signals. Using low-cost 0.18 μm CMOS process technology, this cannot be easily achieved, as will be analyzed in Fig. 2 To boost the timing resolution (i.e., to have a smaller delay step we can control) in a MPCG design and make it adequate for many practical applications, simply upgrading the process technology may not be a panacea, as the process variation may kick in as another new limiting factor. Instead, we discovered in this brief that new methodology and architecture may turn the table more easily and complete the mission with only mature process technology. Conventional process calibration mostly uses a so-called relative phase detector to compensate for the phase mismatch. In an iterative process, every phase of clock signal continuously attempts to maintain a balanced phase position between its two neighboring phases of clocks, until the p hase errors of them converge to a small value Such a method may take large area and power consumption.

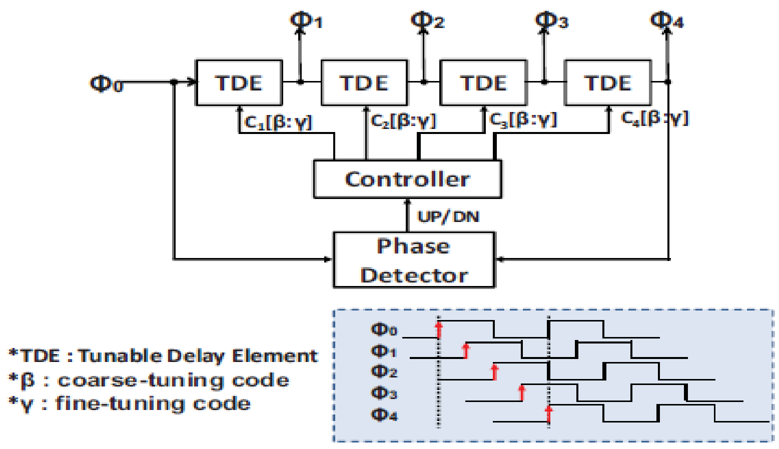
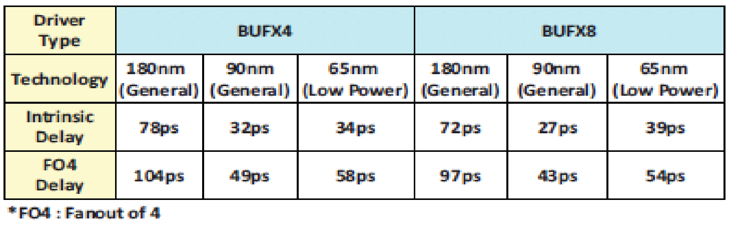


Fig. 1. Traditional MPCG—architecture and output signals.



Driver Type	BUFX4			BUFX8		
Technology	180nm (General)	90nm (General)	65nm (Low Power)	180nm (General)	90nm (General)	65nm (Low Power)
Intrinsic Delay	78ps	32ps	34ps	72ps	27ps	39ps
FO4 Delay	104ps	49ps	58ps	97ps	43ps	54ps

*FO4 : Fanout of 4

Fig. 2. Characteristics of TDE under different process technologies.

The methodology to be presented in this brief has a simpler architecture. Also, it uses only standard cells and thus can lend itself to automation in the future.

II. PRELIMINARIES

More formally, the MPCG problem is to design a circuit that can generate a sequence of clock signals $\{\varphi 0, \varphi 1, \ldots, \varphi N\}$. The parameter N is used to represent the total number of phases. In some cases, it is as small as just 2. But in some other cases, it could be a larger number, say 16 as in our Radar SoC. The multiphase clock signals have the same frequency, denoted as F, while their clock cycle time (or clock period) is denoted as T . Next, we will define some related terminologies more formally, so as to make our presentation more clear.

Definition 1: The rising edges of every two consecutive clock signals, e.g., $\{\varphi i$ and $\varphi i{+}1\}$, are apart by a fixed timing interval, referred to as per-phase delay, denoted as δnotch. For simplicity
without losing generality, we assume that the clock period is equal to N times the per-phase delay, i.e., $T = N \times \delta$notch. In some sense, a clock period is now divided into N notches evenly. This assumption leads to a phenomenon that the clock signal of the last phase (φN) ought to be edge-aligned with the clock signal of the initial phase ($\varphi 0$).

Definition 2: Relative phase or simply phase of a clock signal is referred to the relative timing position of the rising edge of a clock signal as compared to that of $\varphi 0$ (which is considered as a reference clock signal). Based on this definition, we will say that the phase of $\varphi 0$ is zero, and the phase of φi is $i \times \delta$notch.

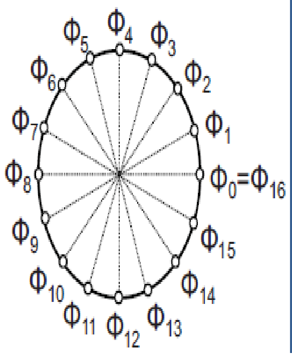
Definition 3: Two clock signals, say $\{\varphi x , \varphi y\}$, are denoted by $\varphi x \equiv \varphi y$, if their phases are the same. Sometimes, we also depict that these two signals are in-phase or edge-aligned, and use these phrases interchangeably.
1) Cyclic Property of a Clock Signal's Phase: In general, a phase is a concept similar to the time delay except that the phase is cyclic. That is, for a phase larger than a reference period, say T , we can modulo T to derive its equivalent phase

Equivalent − Phase = Original_Phase % T.

This property is based on the observation that the Waveform of a clock signal with a phase of "p" and the waveform of a clock signal with a phase of "p plus a number of clock periods" do not have any ifference in their waveforms. By applying the above modulo operation, we can confine the phase of a clock signal to [0, T].

Definition 4: We define the phasor plot of a MPCG problem, as
illustrated in Fig. 3. The N phases of clock signals are mapped to their positions on a unit circle evenly Angle$(\varphi i) = i \times(360/N)$.

This phasor plot can describe the cyclic property in a more illustrative way. For example, a clock signal, say φN+1, with a phase of $(N+1)\times\delta$notch, is a clock signal with an equivalent phase of δnotch. It thus implies that it is also in-phase with φ1.



Notation:

Φ_0	dk 0°	Φ_9	dk 202.5°
Φ_1	dk 22.5°	Φ_{10}	dk 225°
Φ_2	dk 45°	Φ_{11}	dk 247.5°
Φ_3	dk 67.5°	Φ_{12}	dk 270°
Φ_4	dk 90°	Φ_{13}	dk 292.5°
Φ_5	dk 112.5°	Φ_{14}	dk 315°
Φ_6	dk 135°	Φ_{15}	dk 337.5°
Φ_7	dk 157.5°	Φ_{16}	dk 360°
Φ_8	dk 180°		

Fig. 3. Phasor plot of a 16-phase clock signal.

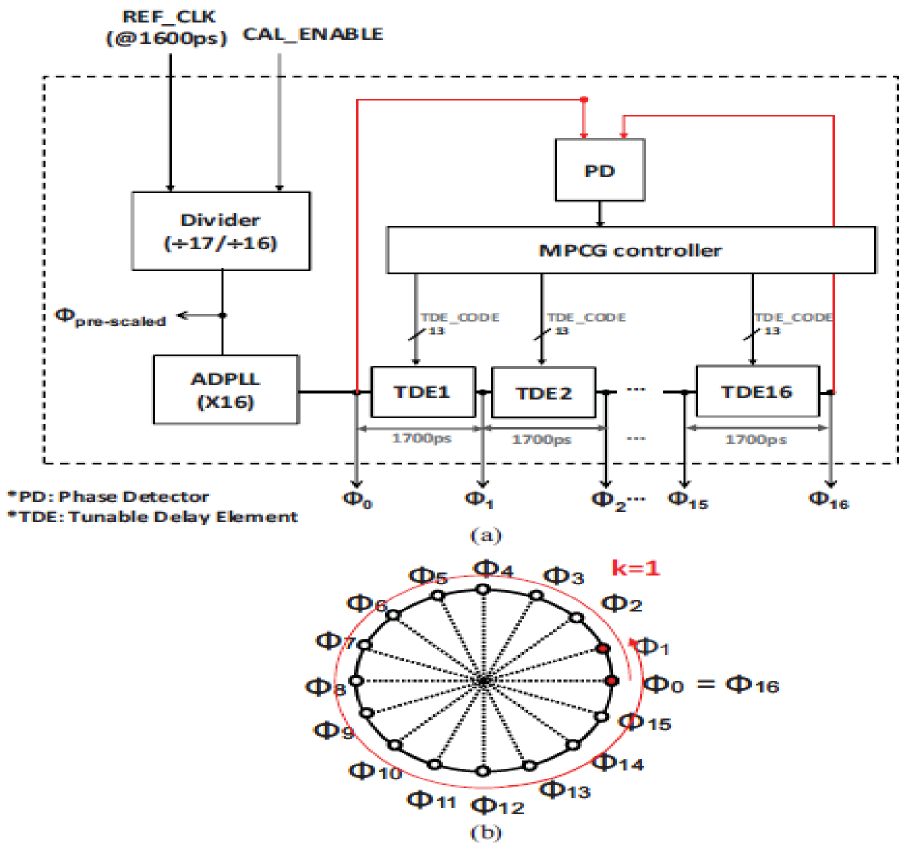


Fig. 4. (a) Basic cyclic MPCG architecture and (b) its phasor plot.

III. INTRODUCTION TO MPCG

A. Basic Cyclic MPCG Architecture

Fig. 4 reveals the architecture of a cyclic MPCG design. For simplicity, we assume that there are 16 phases of clock signals to be produced and the per-phase delay, δnotch, is 100 ps. The clock period, T, of the clock signals are all 100 × 16 = 1600 ps, while the clock frequency is roughly 625 MHz. At the first glance, this requirement is almost unattainable for a 0.18-μm CMOS process technology in that we cannot make a TDE with the delay as small as 100 ps. But as will be shown later, it could be easily achievable by taking advantage of the cyclic property discussed previously. The basic idea is that any TDE with the delay of ($k\times T$ +100 ps) will produce desired MPCG output signals. Here, k is zero or any positive integer.

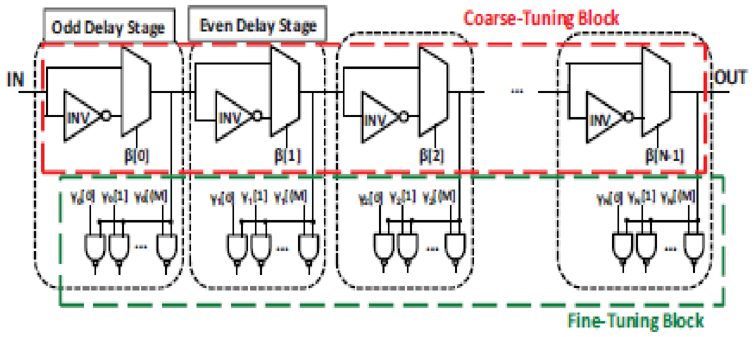


Fig. 5. Architecture of proposed TDE.

In Fig. 4, we show an example of using $k = 1$, in which the desired delay across each TDE is $(1 \times 1600 + 100) = 1700$ ps. On the phasor plot as shown in Fig. 4(b), the clock signal produced by TDE1, denoted as $\varphi 1$, travels the entire circle once and then settles down to the correct position

B. TDE

The proposed TDE is shown in Fig. 5. To minimize the variation of the duty cycle when a clock signal passes through the delay path, we use a pair of the delay stages, which are composed of inverters, one labeled as odd delay stage and the other is labeled as even delay stage. When one clock signal passes through the odd delay stage, the pulse width of the clock signal will be inversed and modified (in a way of being either shrunk or sometimes expanded). However, the inversed modified pulse can be recovered when passing through the even delay stage. Overall, the duty cycle is almost invariant through the two delay stages combined. The resolution of the TDE used in proposed architecture is mainly dictated by the two-input NAND gates used as variable loading proposed in at the outputs of the multiplexers. The delay across a delay stage can now be finely tuned by controlling a $(M + 1)$-bit thermometer γ -code, with each bit of this code controlling the sideinput of a NAND gate. This is due to the fact that each extra turned-on NAND gates will cause a slight change on the loading effect at the output node it connects to. The tuning resolution of a TDE is about 3.14 ps in a typical 0.18-μm CMOS process, while it can be improved to 1.91 ps in a 90-nm CMOS process.

IV. PROCESS CALIBRATION

The cyclic MPCG architecture only works under ideal conditions. Certain problems must first be solved for it to work correctly in silicon. The most prominent problem that has not been fully addressed in the literature is how to equalize the true delay of all TDEs in silicon—a process-calibration problem—by using only standard cells If we apply the same tuning code to all TDEs, chances are their delays vary from one to another. This problem is aggravated when we use a TDE with a longer delay. Assuming that there is 30% process variation in the worst case, then it translates to only 30 ps for a 100 ps TDE, while 510 ps for a 1700 ps TDE. A 510 ps deviation from the ideal timing position is a total disaster, since 510 ps is already more than 5 times the perphase delay (which is 100 ps). That is, the error is so large that it could make the generated clock signals $\{\varphi 1, \varphi 2, \ldots, \varphi 16\}$ totally out of order and thus useless. This worst-case scenario will completely invalidate the proposed cyclic MPCG architecture if not addressed properly In some sense, we have not yet overcome the challenge of deigning a 100 ps TDE yet. We simply convert it into another problem for which we need to propose some process calibration technique to solve. From this point of view, our method takes two stages:

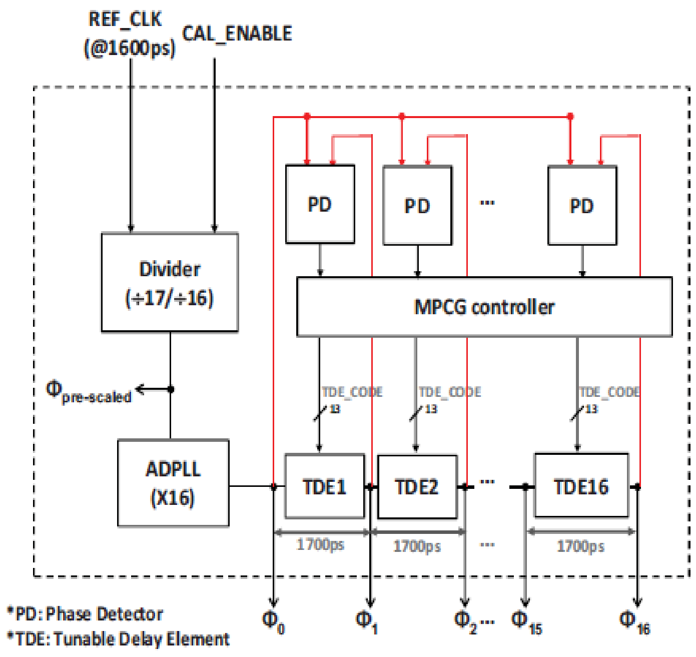


Fig. 6. Modified MPCG architecture supporting process calibration.

1) we relax the stringent requirement of the TDE, and 2) we find a way to control each of them separately but precisely.

A. *Process-Resilient Architecture*

Fig. 6 depicts the architecture of a cyclic MPCG design that can adapt to the true silicon speed of a manufactured chip and make the delay of each TDE equal as much as possible. In addition to the original chain of TDEs, the distinctive feature added is the incorporation of an all-digital phase-locked loop (ADPLL) responsible for generating the calibration clock signals. The following paragraph describes our basic idea. Unlike the traditional MPCG design, we need to precisely control the edge position of not just the final phase of clock signal, $\varphi16$, but also every internal phase of clock signal, i.e., $\{\varphi1, \varphi2, \ldots, \varphi16\}$. This may seem quite challenging, but indeed achievable since the delay of each TDE has been much larger after we have applied the cyclic property. In practice, this can be done by adjusting the frequency (or the clock period in other words) of the input clock signal $\varphi0$. For example, if we adjust the clock period in our working example from the original 1600 to 1700 ps, then the goal of process calibration is to make all phases of clock signals in-phase with $\varphi0$ in our cyclic MPCG design. If this criterion is achieved, then we will have successfully placed the edge position of each phase of clock signal at its right timing location within a very small error range roughly determined by the tuning resolution of a TDE.
The modified architecture operates in two different modes: 1) the process calibration mode after power up, and 2) the functional mode. These two operational modes accept a reference clock signal of different frequencies (or clock periods). In our working example, the clock period is 1700 ps in the calibration mode, whereas 1600 ps in the functional mode. These two requirements can be integrated as an ADPLL, explained below.

1. The input is an external clock signal, denoted as φref , with the functional clock period (e.g., 1600 ps in our example).
2. There is a frequency divider (as a pre-scaler) that divides the frequency of φref by (N +1) in the calibration mode, and N in the functional mode. In terms of clock period, this is equivalent to stretching φref by (N + 1) or N times, producing a clock signal called φpre−scaled
3. The basic function of the ADPLL is to multiply the frequency of clock signal φpre−scaled by a multiplicative factor equaling the number of phases, N, and $N = 16$ in our example. In terms of clock period, φpre−scaled is shrunk by N times when producing clock signal $\varphi0$.

The final clock period of $\varphi0$ can be calculated as follows. 1) In the calibration mode, Period($\varphi0$) = $(1/N)$ × Period(φpre−scaled) = $(1/N)$ × $(N + 1)$ × Period(φref) = $(N + 1)/N$ × Period(φref). In our example, this corresponds to Period($\varphi0$) = 17/16 × 1600 ps = 1700 ps, or one notch larger than the target clock period. 2) In the functional mode, Period($\varphi0$) = $(1/N)$ × Period(φpre−scaled) = $(1/N)$ × (N) × Period(φref) = Period(φref). In our example, this corresponds to Period($\varphi0$) = 1600 ps, which is simply the target clock period. It is worth mentioning that ADPLL has been a mature design target with numerous related brief. In this brief, we use an ADPLL compiler which can automatically realize a desired ADPLL with only standard cells .

B. Process Calibration Procedure

After powering up, our MPCG design will go through a calibration procedure before getting ready for the normal operation. Basically it first sets up the ADPLL to produce a calibration-purpose clock signal φpre−scaled (with the clock period one notch higher than the target one) and then decides the tuning code for each TDE, one at a time, following the order from $\varphi 1$ to $\varphi 2$, and all the way to $\varphi 16$. As previously mentioned, all these phases of clock signal should be edge-aligned with $\varphi 0$ during the calibration mode. Once the calibration is done, a nominal tuning code has been set to each TDE. Typically, deciding a tuning code for a TDE is itself a mini-search process. After that, we switch the MPCG design to the functional mode, in which the clock period of $\varphi 0$ returns to the functional one, say 1600 ps in our example

C. Dynamic Tracking

During the normal operation, we still need to adjust the tuning code of each TDE slightly to make it adaptive to potential supply voltage or temperature variation. Note that at the moment there is no way to align the internal clock signals $\{\varphi 1, \varphi 2, \ldots, \varphi 15\}$. We can judge by the phase of the final output clock signal, $\varphi 15$, by comparing it to the initial clock signal $\varphi 0$. We change the turning codes of the TDEs based on phase comparison results. Since we are in a phasemaintenance stage, we will increment or decrement the fine-tuning code of only one TDE at a time to minimize the amount of phase change, albeit in a round robin order. That is, each TDE must be tuned in turn.

V. EXPERIMENTAL RESULTS

The proposed cell-based cyclic MPCG is implemented by following the general cell-based design flow with the TSMC 0.18-μm CMOS technology. We use an IC-Compiler from Synopsys as our automatic place and routing tool. The layout of the chip is demonstrated in Fig. 7. Fig. 8 shows the waveforms of the 16 phases of clock signals before and after the process calibration. It can be seen that the rising edges are all precisely aligned after the process calibration as expected. The phase errors of the 16 phases of the clock signals $\{\varphi 1, \varphi 2, \ldots, \varphi 16\}$ are also profiled in Fig. 9. It shows that the phase error can be as high as 188 ps without the process calibration due to the variation of the wire lengths, while reduced to less than 13 ps after applying the proposed calibration scheme

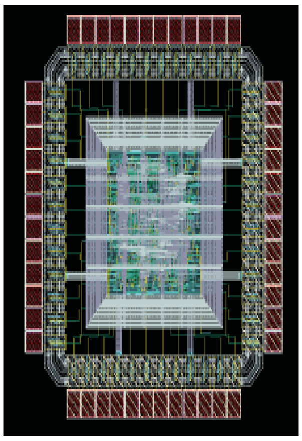
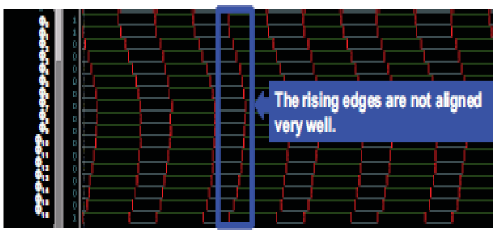


Fig. 7. Layout of our target 16-phase design.



(a)

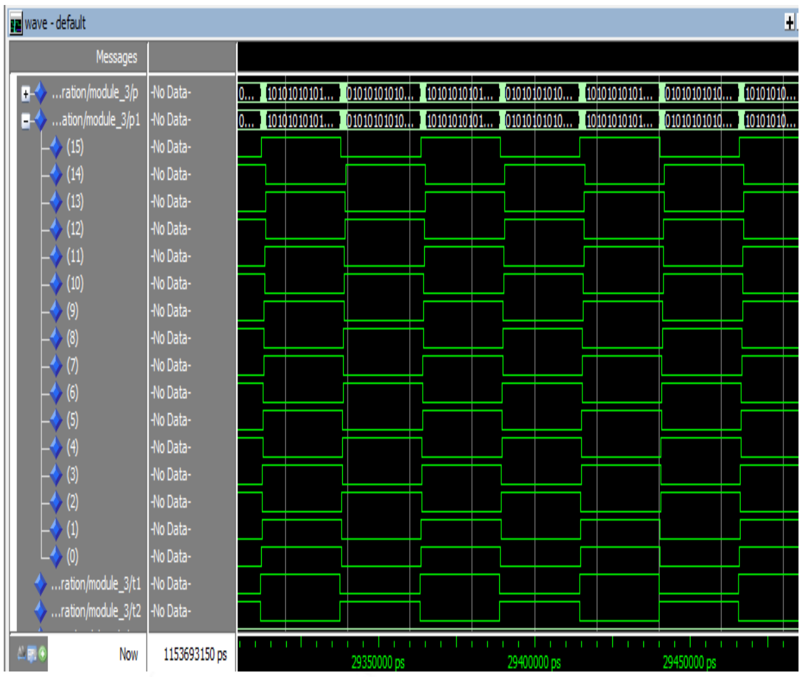


Fig. 8. Post-layout simulation waveforms of 16-phase clock signals (a) before and (b) after calibration.

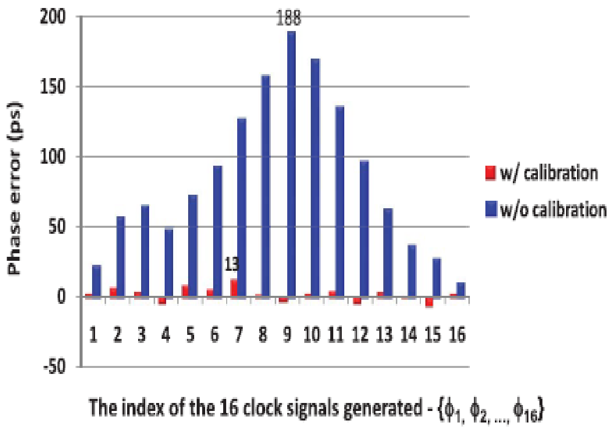


Fig. 9. Phase errors of the 16 phases of clock signals.

Table I shows the performance comparison with some previous works. Since our design can be easily portable to any cell library ready process, we designed three versions for comparison: 1) a 4-phase MPCG using a 0.18 μm process; 2) a 4-phase MPCG using a 90 nm process; and 3) a 16-phase MPCG for our in-house radar IC using a 0.18 μm process. It is shown that ours is still quite competitive in the achievable minimum per-phase delay even though it is made of standard cells. As for area, ours is much smaller. The listed first two works, i.e., are 5.6× and 35× larger than ours, respectively, after area normalization. In a head-to head comparison with the work in we are only inferior in the phase error (e.g., our 11 ps versus their 3.5 ps). This is mostly due to the limitation imposed by the cell-based phase detector. If smaller phase error is required, then we can resort to one of the following two solutions: 1) we can fully customize a 1-ps-precision phase detector as a new cell and add it

TABLE I
PERFORMANCE SUMMARY

Parameter	ISSCC'03 [10]	JSSC'06 [11]	This Brief (post-layout simulation)		
Type	Analog	Digital	Cell-based		
Process	0.35 μm	0.18 μm	90 nm	0.18 μm	
Power supply voltage	3.3 V	1.8 V	1.0 V	1.8 V	
Target frequency rate	150 MHz	2 GHz	2 GHz	2 GHz	625 MHz
Phase number	10	4	4	4	16
Per-phase delay	666.7 ps	125 ps	125 ps	125 ps	100 ps
$\text{Jitter}_{\text{RMS}}$	3.41 ps	2.45 ps	1.6 ps	2.37 ps	2.55 ps
$\text{Jitter}_{\text{pk-to-pk}}$	26.38 ps	18.9 ps	11.5 ps	16 ps	22 ps
Max. phase error Before calibration	422 ps	20.4 ps	27 ps	45 ps	188 ps
Max. phase error After calibration	113 ps	3.5 ps	4 ps	11 ps	13 ps
Core area	2.25 mm^2	1.55 mm^2	0.01 mm^2	0.044 mm^2	0.161 mm^2
Power	18 mW	81 mW	3.4 mW	22.9 mW	49.2 mW

Standard cell library to improve the accuracy, or 2) we can simply use a more advanced process, if that is available. For example, the phase error in our design can be reduced from 11 ps to only 4 ps after shifting from a 0.18 μm process to a 90 nm process. The power consumptions of our 4-phase clock generator design are 22.9 mW for 0.18 μm process and 3.4 mW for 90 nm process, respectively. In terms of power consumption, the work in is 81/22.9 = 3.53 × that of ours. It is also notable that the work in consumes only 18 mW. This is partly due to its low operating frequency at 150 MHz. After scaled by the supply voltage factor (1.82/3.3)2 and the clock frequency factor (2000 MHz/150 MHz), the power consumption in will translate to roughly 71 mW, which is still (71/22.9) = 3.1 × that of our version in the 0.18 μm process

VI. CONCLUSION

When it comes to the design of a MPCG, the selected process technology may impose some limitations that do not seem to be reconcilable by standard cells—e.g., the amount of small delays one wish to control precisely across a delay cell. One may choose to use pricer and more advanced processes to alleviate this problem, but the process variation still may make process calibration necessary. In this brief, we have found that we can actually boost the performance of a MPCG design tremendously via architectural innovations. The design was robust (even for advanced processes) and highly accurate with small phase errors. Even better, it lent itself to automation since it only used standard cells. Post-layout simulation validated that this method can achieve much smaller area and power consumption than previous works. We also used this method to asily generate a 16-phase MPCG design meeting the requirement of an in-house radar SoC design using just 0.18-μm CMOS process technology.

REFERENCES

[1] C.-C. Chung and C.-Y. Lee, "A new DLL-based approach for all-digital multiphase clock generation," *IEEE J. Solid-State Circuits*, vol. 39, no. 3, pp. 469–475, Mar. 2004.

[2] C.-C. Chen, J.-Y. Chang, and S.-I. Liu, "A DLL-based variable-phase clock buffer," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 54, no. 12, pp. 1072–1076, Dec. 2007.

[3] C.-N. Chuang and S.-I. Liu, "A 0.5–5-GHz wide-range multiphase DLL with a calibrated charge pump," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 54, no. 11, pp. 939–943, Nov. 2007.

[4] C.-N. Chuang and S.-I. Liu, "A 20-MHz to 3-GHz wide-range multiphase delay-locked loop," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 56, no. 11, pp. 850–854, Nov. 2009.

[5] C.-H. Park, O. Kim, and B. Kim, "A 1.8-GHz self-calibrated phaselocked loop with precise I/Q matching," *IEEE J. Solid-State Circuits*, vol. 36, no. 5, pp. 777–783, May 2001.