Published in IET Circuits, Devices & Systems
Received on 30th June 2009
Revised on 7th January 2010
doi: 10.1049/iet-cds.2009.0173

ISSN 1751-858X

Fast-locking all-digital phase-locked loop with digitally controlled oscillator tuning word estimating and presetting

G. Yu Y. Wang H. Yang H. Wang

Department of Electronic Engineering, Tsinghua National Laboratory for Information Science and Technology (TNList), Tsinghua University, Beijing 100084, People's Republic of China
E-mail: ygm05@mails.tsinghua.edu.cn

Abstract: Design of a fast-locking phase-locked loop (PLL) is one of the major challenges in today's wireless communications. A recently reported digitally controlled oscillator (DCO)-based all-digital PLL (ADPLL) can achieve an ultrashort settling time of 10 μs. This study describes a new DCO tuning word (OTW) presetting technique for the ADPLL to further reduce its settling time. Estimating the required OTW is the most crucial issue for presetting. Two methods are proposed here to estimate the required OTW. One method is using a foreground calibration block to eliminate the effect of DCO gain (K_{DCO}) estimation error (ε_K) and then directly calculating the required OTW for the process/voltage/temperature calibration (PVT-calibration) mode of the ADPLL. The other method is using a new counter-based mode switching controller (CB-MSC) to estimate the required OTW for the acquisition mode and tracking mode. This method is based on the ADPLL's inherent characteristic of frequency toggling and is independent of loop parameters. Furthermore, our proposed presetting technique can be used with the dynamic loop bandwidth control technique together. The ADPLL with the proposed OTW estimating and presetting block is designed using very-high-speed integrated circuit hardware description language and simulated in ModelSim environment. Simulation results demonstrate that a minimum settling time of 2.9 μs is achieved and the improvement is about 40–50% on average compared with the ADPLL without our techniques.

1 Introduction

The RF frequency synthesiser is an essential block of modern communication systems. It is traditionally implemented using a charge-pump phase-locked loop (CPPLL), which suffers from high-level reference spurs, poor phase noise caused by charge-pump mismatch and large die area because of integrated RC loop filter. Recently, an ADPLL has been presented in [1], which has a digitally controlled oscillator (DCO) [2] and a time-to-digital converter (TDC) [3] as key components. Fig. 1 shows the block diagram of the all-digital PLL (ADPLL). Compared with the traditional CPPLL, the ADPLL has many advantages. First, the ADPLL avoids analogue components and takes the advantage of nanometre-scale CMOS process. Second, the ADPLL is immune to the digital switching noise in a system-on-chip environment because all the signals in the ADPLL are digital. Third, the frequency acquisition process is faster in the ADPLL than in CPPLLs [4]. As reported in [5–7], the settling time is about 20–30 μs in CPPLLs, while it reaches 10 μs in the ADPLL [8]. Furthermore, the ADPLL can be implemented using automated CAD tools. Therefore the ADPLL has a faster design turnaround time and is easier to be integrated and migrated.

Settling time of the frequency synthesiser is an important design issue for today's wireless communication systems. For example, in the frequency hopping system, the settling time is required to be minimised to optimise system performance [9]. For ZigBee applications, it is important to minimise energy consumption of the wireless terminal node by minimising both the active power consumption

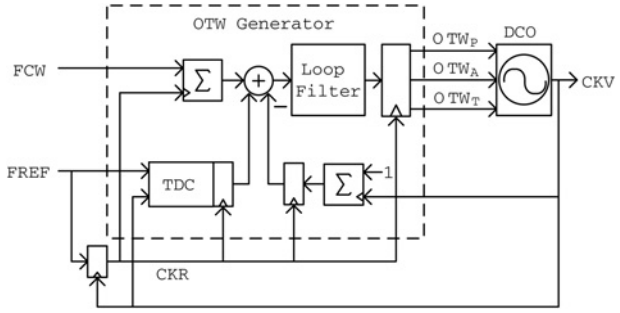

Figure 1 *Block diagram of the ADPLL in [1]*

and the active duty cycle that depends on the frequency settling time of the PLL [10].

In the ADPLL as shown in Fig. 1, the DCO output frequency is set by the OTW, which consists of three parts, namely OTW_P, OTW_A and OTW_T. Each of them controls a DCO capacitor bank, which are called process/voltage/temperature calibration (PVT-calibration) bank, acquisition bank and tracking bank, respectively. During ADPLL normal operation, an outer frequency reference (FREF) clock and a frequency command word (FCW) are sent to the ADPLL. Through an OTW generator, which is marked by the dashed line in Fig. 1, an OTW ($[OTW_P, OTW_A, OTW_T]$) is generated to change the DCO output (CKV) frequency. Then the CKV is fed back to the OTW generator to regenerate a new OTW. When the output frequency is settled to the desired frequency f^*_{CKV}

$$f^*_{CKV} = FCW \times FREF \tag{1}$$

the OTW is settled to OTW^* ($[OTW^*_P, OTW^*_A, OTW^*_T]$). So a useful concept is proposed that the locking process of the ADPLL is not only a frequency-locking process but also an OTW locking process. A fast frequency-locking is equal to a fast OTW locking.

Various techniques have been reported to reduce the settling time of PLL. One of the most popular techniques is dynamic loop bandwidth control. As presented in [11], the loop bandwidth was switched during the tracking mode. But the PVT-calibration mode and acquisition mode were neglected. Another popular technique for fast-locking is presetting. A VCO control voltage presetting technique was discussed in [12] for the CPPLL. For the ADPLL, a feed-forward technique was proposed in [13], which used the input reference signal to compensate the phase error directly. However, the achievable improvement of settling speed depended on the loop parameters. Other techniques such as binary search algorithms and two-stage TDC were studied in [14, 15]. Up to now, no literature that focuses on the OTW presetting in the ADPLL is reported.

In this paper, we propose novel techniques to estimate and preset OTW for the ADPLL to reduce its settling time. All the three operation modes are taken into account. For the PVT-calibration mode, we eliminate the effect of K_{DCO} estimation error (ε_K) using a foreground calibration block and then directly calculate the OTW_P for presetting. For the acquisition mode and tracking mode, a novel counter-based mode switching controller is proposed to estimate the OTW_A and OTW_T for presetting. Both of the proposed OTW estimating methods are independent of loop parameters. Simulation results demonstrate that the ADPLL with our proposed OTW estimating and presetting block achieves about 40–50% improvement of settling time on average compared with the ADPLL without our block. Furthermore, our analysis indicates that the proposed presetting technique can be used with the dynamic loop bandwidth control technique together. Consequently, a fast settling can be achieved without degrading the noise performance.

The rest of this paper is organised as follows: Section 2 presents the proposed fast-locking ADPLL architecture. Section 3 describes the principle of OTW calculating and analyses the effect of ε_K. The proposed OTW estimating and presetting methods are proposed in Section 4. Simulation results and discussions are presented in Section 5 and the conclusions are given in Section 6.

2 Architecture overview

Fig. 2 shows a diagram of the proposed fast-locking ADPLL. The OTW generator is the same as that presented in Fig. 1. The OTW estimating and presetting block consists of three parts: self-calibration, OTW_P calculation, and counter-based mode switching controller (CB-MSC).

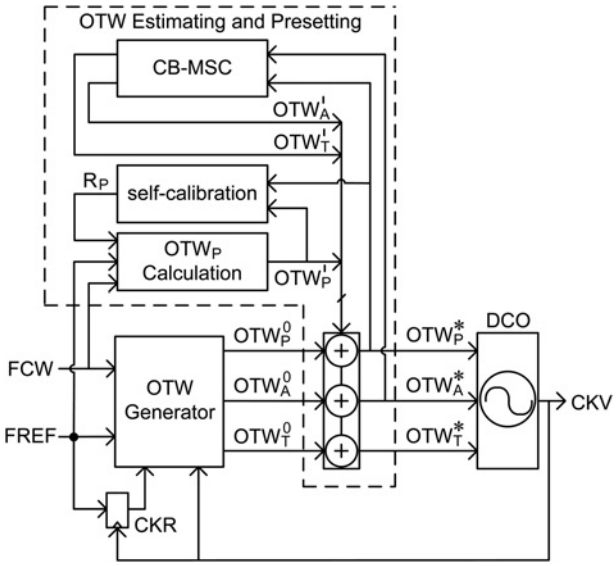

Figure 2 *Architecture of the proposed fast-locking ADPLL*

IET Circuits Devices Syst., 2010, Vol. 4, Iss. 3, pp. 207–217
doi: 10.1049/iet-cds.2009.0173

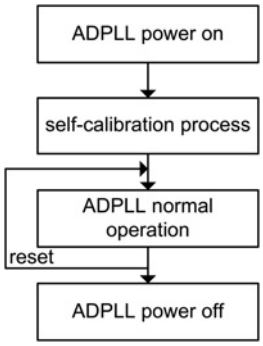

Figure 3 *Top level flowchart of ADPLL's operation*

1. Self-calibration block works before ADPLL normal operation, as shown in Fig. 3. This block is used to eliminate the effect of ε_K in the PVT-calibration bank, which is caused by some practical issues such as the capacitance deviation and mismatch in the DCO, inductance deviation, temperature variation and other non-ideal factors. An parameter R_P, which is defined as FREF/Δf_P (see in Section 3.1), will be used in the following OTW_P calculation block. The error of Δf_P can be expressed using the error of R_P. Therefore when the self-calibration process is completed, a calibrated R_P is generated for the OTW_P calculation block and the self-calibration block becomes idle. The algorithm of this block is presented in Section 4.1.

2. OTW_P calculation block works in the PVT-calibration mode. Using the calibrated R_P and the outer signals of FREF and FCW, this block generates an OTW_P estimation value defined as OTW'_P. Then OTW'_P is added to the original OTW_P^0 to generate a final OTW_P^* to the DCO. The OTW_P^* is very close to the required value. Therefore the locking process in the PVT-calibration mode is significantly accelerated. The implementation details of this block are provided in Section 4.2.

3. CB-MSC block has two functions. One, which is the basic function, is to control the ADPLL to traverse through the three operation modes which are the PVT-calibration mode acquisition mode and tracking mode. The other function is to generate OTW_A and OTW_T estimation values defined as OTW'_A and OTW'_T for the acquisition mode and tracking mode according to the different modes of frequency toggling. The detailed description of this block is given in Section 4.3.

All the three blocks are implemented using digital circuits, so the all-digital characteristic of the ADPLL is preserved.

3 Estimating the required OTW by theoretical calculation

How to obtain the required OTW for presetting is the most critical issue in ADPLL presetting techniques. In this section, we will first give a theoretical derivation to directly calculate the required OTW in ideal conditions. Then ε_K is taken into account and its effects on the OTW calculation is analysed.

3.1 Principle of theoretical calculation of the OTW

Fig. 4 shows an example of locking process of the ADPLL. The three operation modes are sequentially activated during the frequency-locking process. Each mode has a minimum frequency step of Δf. In the ADPLL, Δf is generally defined as K_{DCO}. The DCO free running frequency is f_{free} and the desired frequency is f^*_{CKV}. In the PVT-calibration mode as shown in Fig. 4, the DCO output frequency approaches f^*_{CKV} in a coarse frequency step of Δf_P, which is 2333 kHz in our design. Finally, the OTW_P reaches 6 and toggles between 6 and 7. The toggling is due to the frequency quantisation effect of the DCO. When the PVT-calibration mode is completed, the OTW_P is locked to 6. The residual frequency difference, which is smaller than Δf_P, is dealt with in the following acquisition mode. In the acquisition mode, the DCO output frequency approaches f^*_{CKV} in a medium frequency step of Δf_A, which is 397 kHz in our design. At the end of the acquisition mode, the OTW_A is locked to 4. The tracking mode is almost the same as the former ones except it has the finest frequency step of Δf_T, which is 23 kHz.

According to the discussion before, the initial frequency difference between f^*_{CKV} and f_{free} can be described as

$$\begin{aligned} f^*_{CKV} - f_{free} &= FCW \times FREF - f_{free} \\ &= OTW_P^* \times \Delta f_P + OTW_A^* \times \Delta f_A \\ &\quad + OTW_T^* \times \Delta f_T + \delta_f \end{aligned} \tag{2}$$

where OTW_P^*, OTW_A^* and OTW_T^* are integer. δ_f is the final residual frequency difference, which is smaller than Δf_T.

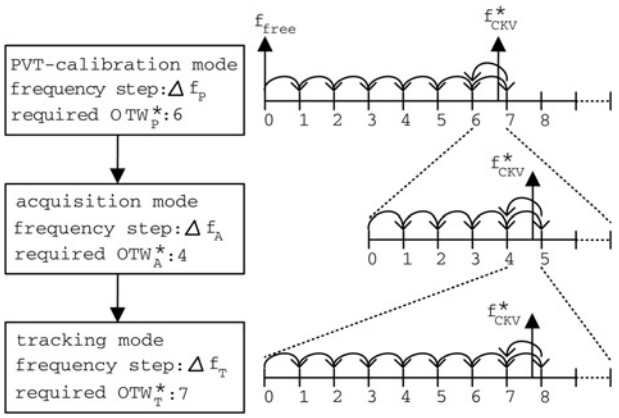

Figure 4 *Example of locking process of the ADPLL*

For convenience, we define

$$\xi_F \triangleq \mathrm{FCW} - \frac{f_{\mathrm{free}}}{\mathrm{FREF}} \tag{3}$$

$$R_P \triangleq \frac{\mathrm{FREF}}{\Delta f_P} \tag{4}$$

$$R_A \triangleq \frac{\Delta f_P}{\Delta f_A} \tag{5}$$

$$R_T \triangleq \frac{\Delta f_A}{\Delta f_T} \tag{6}$$

In order to obtain OTW_P^* in (2), we rewrite (2) as

$$\begin{aligned} f_{\mathrm{CKV}}^* - f_{\mathrm{free}} &= \mathrm{FCW} \times \mathrm{FREF} - f_{\mathrm{free}} \\ &\triangleq \mathrm{OTW}_P^* \times \Delta f_P + \xi_P \times \Delta f_P \end{aligned} \tag{7}$$

where $|\xi_P| < 1$. This is because the residual frequency difference after the PVT-calibration mode is smaller than Δf_P. So OTW_P^*, is obtained by

$$\mathrm{OTW}_P^* = \left[\left(\mathrm{FCW} - \frac{f_{\mathrm{free}}}{\mathrm{FREF}}\right) \times \frac{\mathrm{FREF}}{\Delta f_P}\right]_I = [\xi_F \times R_P]_I \tag{8}$$

where $[\ldots]_I$ is an operation of taking integer part. Furthermore, ξ_P in (7) is obtained by

$$\xi_P = \left[\left(\mathrm{FCW} - \frac{f_{\mathrm{free}}}{\mathrm{FREF}}\right) \times \frac{\mathrm{FREF}}{\Delta f_P}\right]_F = [\xi_F \times R_P]_F \tag{9}$$

where $[\ldots]_F$ is an operation of taking fractional part. According to (2), (7) and (9), we obtain

$$\begin{aligned} \xi_P \times \Delta f_P &= [\xi_F \times R_P]_F \times \Delta f_P \\ &= \mathrm{OTW}_A^* \times \Delta f_A + \mathrm{OTW}_T^* \times \Delta f_T + \delta_f \\ &\triangleq \mathrm{OTW}_A^* \times \Delta f_A + \xi_A \times \Delta f_A \end{aligned} \tag{10}$$

where $|\xi_A| < 1$. This is because the residual frequency difference after the acquisition mode is smaller than Δf_A. So we obtain OTW_A^* by

$$\mathrm{OTW}_A^* = [[\xi_F \times R_P]_F \times R_A]_I \tag{11}$$

Furthermore, ξ_A is obtained by

$$\xi_A = [[\xi_F \times R_P]_F \times R_A]_F \tag{12}$$

According to (10) and (12), we obtain

$$\begin{aligned} \xi_A \times \Delta f_A &= [[\xi_F \times R_P]_F \times R_A]_F \times \Delta f_A \\ &= \mathrm{OTW}_T^* \times \Delta f_T + \delta_f \end{aligned} \tag{13}$$

So OTW_T^*, is obtained by

$$\mathrm{OTW}_T^* = [[[\xi_F \times R_P]_F \times R_A]_F \times R_T]_I \tag{14}$$

Up to now, the required OTW has been theoretically calculated according to (8), (11) and (14). However, in practice, because of some non-ideal factors, the ideal values of Δf_P, Δf_A and Δf_T are different from practical ones. This will affect OTW calculation values. Analysis on the effect are proposed in the following part.

3.2 Effect of ε_K on the OTW calculation

In practice, some practical issues, such as the capacitance deviation and mismatch in the DCO, inductance deviation, temperature variation and other non-ideal factors, will cause K_{DCO} estimation error (ε_K) and affect the OTW calculation. In order to investigate the effects, we take ε_K into account in derivations of OTW calculation.

Owing to ε_K in the PVT-calibration mode, an error term of ε_P should be added to R_P as

$$R_P' \triangleq R_P + \varepsilon_P \tag{15}$$

By replacing R_P in (8) with R_P', we obtain the OTW_P calculation value containing R_P error

$$\begin{aligned} \mathrm{OTW}_P' &= [\xi_F \times R_P']_I = [\xi_F \times (R_P + \varepsilon_P)]_I \\ &= [\xi_F \times R_P]_I + [\xi_F \times \varepsilon_P]_I \\ &\quad + [[\xi_F \times R_P]_F + [\xi_F \times \varepsilon_P]_F]_I \\ &= \mathrm{OTW}_P^* + [\xi_F \times \varepsilon_P]_I \\ &\quad + [[\xi_F \times R_P]_F + [\xi_F \times \varepsilon_P]_F]_I \end{aligned} \tag{16}$$

The third term of (16) has a value of 0 or 1 and can be neglected. The second term of (16) indicates that the error of R_P is amplified by ξ_F times, so it has a great effect on the OTW_P calculation.

Substituting (15) into (9), we obtain ξ_P'

$$\begin{aligned} \xi_P' &= [\xi_F \times R_P']_F = [\xi_F \times (R_P + \varepsilon_P)]_F \\ &= [\xi_P + [\xi_F \times \varepsilon_P]_F] \end{aligned} \tag{17}$$

If $0 \le \xi_P + [\xi_F \times \varepsilon_P]_F < 1$, then (17) can be rewritten as

$$\xi_P' = \xi_P + [\xi_F \times \varepsilon_P]_F \tag{18}$$

Substituting (18) into (11) we obtain

$$\begin{aligned} \mathrm{OTW}_A' &= [\xi_P' \times R_A]_I = [(\xi_P + [\xi_F \times \varepsilon_P]_F) \times R_A]_I \\ &= [\xi_P \times R_A]_I + [[\xi_F \times \varepsilon_P]_F \times R_A]_I + \alpha \\ &= \mathrm{OTW}_A^* + [[\xi_F \times \varepsilon_P]_F \times R_A]_I + \alpha \end{aligned} \tag{19}$$

IET Circuits Devices Syst., 2010, Vol. 4, Iss. 3, pp. 207–217
doi: 10.1049/iet-cds.2009.0173

where

$$\alpha = [[\xi_P \times R_A]_F + [[\xi_F \times \varepsilon_P]_F \times R_A]_F]_I \quad (20)$$

and the value of α is 0 or 1 and it has little effect on OTW_A calculation.

If $1 \le \xi_P + [\xi_F \times \varepsilon_P]_F < 2$, then (17) can be rewritten as

$$\xi'_P = \xi_P + [\xi_F \times \varepsilon_P]_F - 1 \quad (21)$$

Substituting (21) into (11) we obtain

$$\begin{aligned} OTW'_A &= [(\xi_P + [\xi_F \times \varepsilon_P]_F - 1) \times R_A]_I \\ &= [\xi_P \times R_A - (1 - [\xi_F \times \varepsilon_P]_F) \times R_A]_I \\ &= OTW^*_A - [(1 - [\xi_F \times \varepsilon_P]_F) \times R_A]_I - \beta \end{aligned} \quad (22)$$

where

$$\beta = \begin{cases} 1, & [\xi_P \times R_A]_F < [(1 - [\xi_F \times \varepsilon_P]_F) \times R_A]_F \\ 0, & [\xi_P \times R_A]_F \ge [(1 - [\xi_F \times \varepsilon_P]_F) \times R_A]_F \end{cases} \quad (23)$$

Equations (16), (19) and (22) indicate that ε_P not only affects the OTW_P calculation result, but also affects OTW_A calculation result. Furthermore, OTW_A calculation result will be affected by the error of R_A. It can be inferred that OTW_T calculation result will be affected by all the errors of R_P, R_A and R_T. We can use a set of equations to describe the effects of K_{DCO} estimation error on OTW calculation

$$\begin{cases} OTW'_P = f(P, \varepsilon_P) \\ OTW'_A = f(P, \varepsilon_P, \varepsilon_A) \\ OTW'_T = f(P, \varepsilon_P, \varepsilon_A, \varepsilon_T) \end{cases} \quad (24)$$

where P denotes loop parameters. ε_P, ε_A and ε_T are the errors of R_P, R_A and R_T. From (24) we can see that 'error propagation' is a major characteristic of the effect of ε_K on OTW calculation.

4 Proposed OTW estimation methods

According to the analysis in Section 3, we can see that the ε_K not only affects the OTW calculation result in the current mode, but also spreads into the following modes to affect the OTW calculation results. Owing to this phenomenon, two different methods are proposed for OTW estimation in different operation modes of the ADPLL. For the PVT-calibration mode, the self-calibration block is used to eliminate the effect of ε_P and then the OTW_P calculation block calculates the OTW'_P for presetting. For the acquisition mode and tracking mode, the CB-MSC block is used to generate OTW'_A and OTW'_T for presetting. Implementation details of the proposed OTW estimation methods are described in this section.

4.1 RP-calibration process

As presented in (24), OTW_P calculation result is only affected by the error of R_P. So if we can calibrate the error of R_P, the required OTW^*_P can be obtained. Fig. 5 shows the algorithm of R_P-calibration process. Before ADPLL normal operation, a known FCW is sent to the ADPLL for R_P-calibration. A large FCW is chosen because at this frequency a large OTW_P calculation error is produced according to (3) and (16). In the R_P-calibration process, when the PVT-calibration mode is completed, a required OTW^*_P is generated by the ADPLL itself. Meanwhile, OTW'_P, the estimation value of OTW^*_P, is calculated by the OTW_P calculation block. Then we compare OTW'_P with OTW^*_P, if OTW'_P is larger than OTW^*_P, then R_P is decreased by a fixed small step of δ. Else, if OTW'_P is smaller than OTW^*_P, R_P is increased by the fixed small step of δ. Using the updated R_P, a new OTW'_P is calculated and sent to be compared with OTW^*_P again. This operation cycles until OTW'_P is equal to OTW^*_P. Then the latest R_P is fixed as the calibrated one and will be used in the PVT-calibration mode during ADPLL normal operation. When the R_P-calibration process is completed, the self-calibration block becomes idle.

4.2 OTW'_P calculation

Once R_P is calibrated, R_P estimation error ε_P (see (15)) is very small and can be considered as 0, so (16) can be rewritten as

$$\begin{aligned} OTW'_P = OTW^*_P &+ [\xi_F \times \varepsilon_P]_I + [[\xi_F \times R_P]_F \\ &+ [\xi_F \times \varepsilon_P]_F]_I = OTW^*_P \end{aligned} \quad (25)$$

Equation (25) indicates that after R_P-calibration, the calculated OTW'_P is equal to the required OTW^*_P. Consequently, we can calculate OTW'_P using (8). For

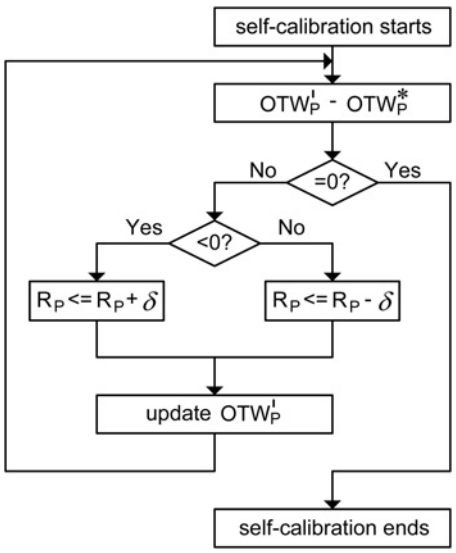

Figure 5 *Calibration algorithm of R_P*

convenience, we rewrite (8) here as

$$\mathrm{OTW}_{\mathrm{P}}^{*} = \left[\left(\mathrm{FCW} - \frac{f_{\mathrm{free}}}{\mathrm{FREF}}\right) \times \frac{\mathrm{FREF}}{\Delta f_{\mathrm{P}}}\right]_{\mathrm{I}} = [\xi_{\mathrm{F}} \times R_{\mathrm{P}}]_{\mathrm{I}} \tag{26}$$

where ξ_{F} is a 23 bits fixed-point number, R_{P} is a 6 bits fixed-point number that consists of 3 bits integer part and 3 bits fractional part. Hardware implementation of $\mathrm{OTW}_{\mathrm{P}}$ calculation block is very simple that only adder and shift-register are needed.

4.3 Counter-based mode switching controller

As discussed in Section 3.2, $\mathrm{OTW}_{\mathrm{A}}$ calculation is affected by both errors of R_{P} and R_{A}, and $\mathrm{OTW}_{\mathrm{T}}$ calculation is affected by all the errors of R_{P}, R_{A} and R_{T}. So in the acquisition mode and tracking mode, it is hard to obtain the required $\mathrm{OTW}_{\mathrm{A}}^{*}$ and $\mathrm{OTW}_{\mathrm{T}}^{*}$ by directly calculating and a simple calibration process is useless. So a novel CB-MSC is proposed to estimation the required OTW for the acquisition mode and tracking mode. This technique is based on the ADPLL's inherent characteristic of frequency toggling.

As shown in Fig. 6, because of the frequency quantisation effect, the DCO output frequency will toggle between the upper (f_2) and lower (f_1) frequency levels around the desired frequency (f_{CKV}^{*}). Furthermore, when f_{CKV}^{*} is close to f_2, the time (T_{high}) that the output frequency stays at f_2 is longer than the time (T_{low}) that at f_1, as shown in Fig. 6*a*. This is because when the DCO output frequency is f_2, the frequency difference between f_2 and f_{CKV}^{*} is very small, so it takes a long time for the ADPLL to accumulate an phase error that is big enough to change the DCO output frequency from f_2 to f_1. Contrarily, when the DCO output frequency is f_1, the frequency difference between f_1 and f_{CKV}^{*} is large, so it takes a short time for the ADPLL to accumulate a phase error that is big enough to change the DCO output frequency from f_1 to f_2. The other two modes of frequency toggling are similar to the one described and are shown in Figs. 6*b* and *c*, respectively.

We can use two counters to record T_{high} and T_{low} in the unit of CKR clock cycle number. The counters are incorporated with the mode switching controller and are implemented using a finite state machine (FSM), as shown in Fig. 7. '○', '*' and 'X' in Fig. 7*a* denote different frequency levels. When the output frequency holds at the same frequency level, the state of the FSM remains unchanged and the counter increases by 1. When frequency jumps from one level to another, the state of the FSM switches and the counter is reset to 0.

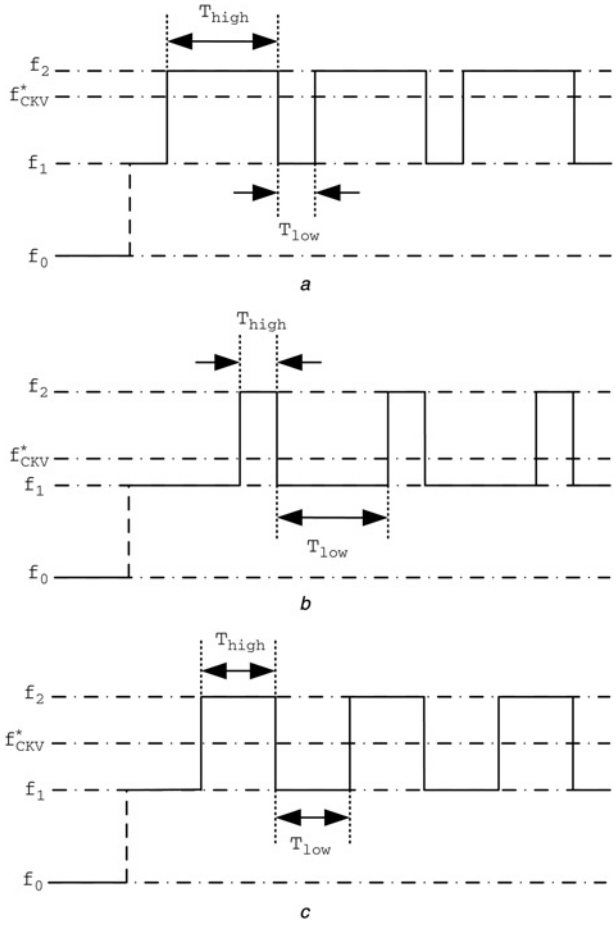

Figure 6 *Three major modes of frequency toggling between the upper and lower frequency levels*

a Close to the upper level
b Close to the lower level
c Almost in the middle

As shown in Fig. 7*b*, there are two conditions which indicate that the frequency has been locked in the current operation mode and the ADPLL needs to switch to the next mode. One condition is that the DCO output frequency jumps continuously between two adjacent frequency levels for M times. In our design M is set to 2. In this condition, the FSM traverses all the available states starting from '00' to '01', '10' and '11' sequentially and then back to '00'. This means that the DCO output frequency stays at neither f_1 nor f_2 for a long time. This corresponds with Fig. 6*c* and the $\mathrm{OTW}_{\mathrm{A}}'$ (or $\mathrm{OTW}_{\mathrm{T}}'$) is assigned a value at the middle of its available range for presetting. The value of M should be regarded. In our design, M is set to 2. Although a bigger value could be used, it would take longer time for the ADPLL to switch from one mode to another. Setting M to 1 is not recommended. Although it takes a shorter time for the ADPLL to switch from one mode to another, it may cause a wrong frequency-locking judgment. The other condition that indicates the frequency has been locked is that the counter reaches the upper bound N, which is 16 in our design. In this condition, the FSM switches to '00' immediately and generates a mode switching signal to the

doi: 10.1049/iet-cds.2009.0173

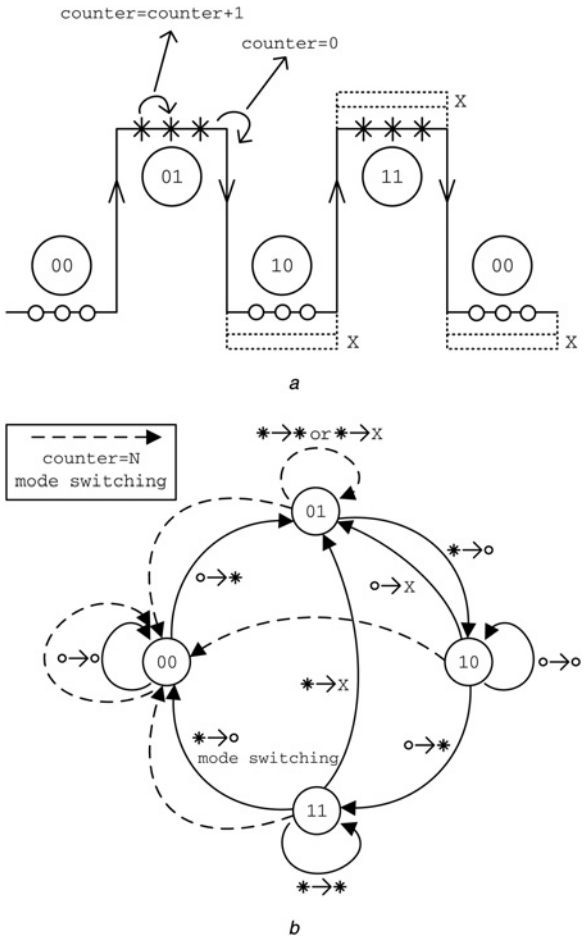

Figure 7 *Principle of the counter-based mode switching controller*

a Is a schematic plan of frequency toggling
b Is state switching based on *a*

ADPLL. This condition means the DCO output frequency stays at either f_1 or f_2 for a long time, which corresponds with Figs. 6*a* or *b*. In this condition, the DCO output frequency is close to the desired frequency f^*_{CKV}, and a small OTW is expected for the next mode. So the OTW'_A (or OTW'_T) is assigned zero for presetting.

5 Simulation results and discussions

In this work we use very-high-speed integrated circuit hardware description language and ModelSim to design and simulate the proposed fast-locking ADPLL shown in Fig. 2. The parameters of the ADPLL are shown in Table 1. As a comparison, the ADPLL in Fig. 1 is also realised with the same parameters. A ΣΔ modulator is contained to increase the frequency resolution. However, it has nothing to do with the settling time and is idle all the time in this work. The modelling method is the same as that introduced in [16]. It should be noted that 6 bits OTW_T are binary weighted. In circuit implementation, a decoder is needed to transmit 6 bits binary weighted OTW_T to 64 bits unit-weighted OTW_T. The OTW_P calculation block is combinational logic circuit meanwhile the CB-MSC and self-calibration block are sequential circuits working at the low rate clock of FREF. Furthermore, the self-calibration block is idle during ADPLL normal operation. Consequently, the power consumption of the OTW presetting and estimating block occupies a very small portion of the ADPLL's total power consumption.

Table 1 ADPLL simulation parameters

reference frequency FREF	13 MHz
DCO free running frequency f_{free}	2398 MHz
FCW	23 bit ($I_{8bit} + F_{15bit}$)
output frequency range	2.4–2.5 GHz
loop filter gain	2^{-7}
loop bandwidth	16.164 kHz
TDC resolution	30 ps
Δf_P	2333 kHz
Δf_A	397 kHz
Δf_T	23 kHz
OTW_P	8 bit
OTW_A	8 bit
OTW_T	6 bit
ΣΔ modulator	MASH 1-1-1

5.1 Results of R_P-calibration process

Fig. 8 shows the simulation result of R_P-calibration process. The FCW sent to the ADPLL is 192.30767822265625 so that the DCO output frequency is 2.5 GHz. When the ADPLL is turned on, R_P is initialised to 5.25 according to

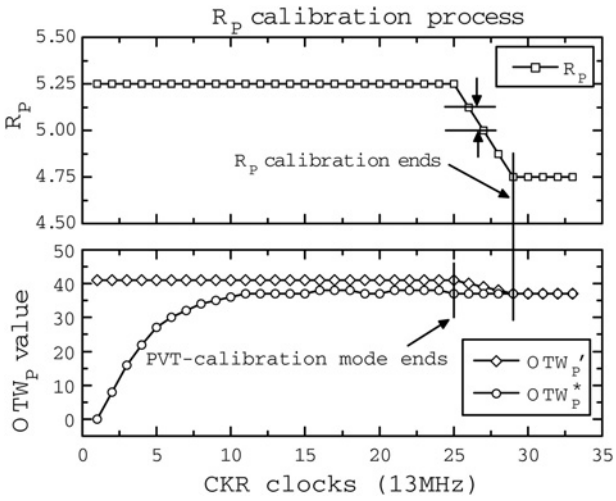

Figure 8 *R_P calibration process*

the ideal case and the calculated OTW'_P is 41. When the PVT-calibration mode is completed, the ADPLL itself generates a required OTW^*_P which is 37. Then the R_P-calibration process starts, as presented in Fig. 5. For $OTW'_P > OTW^*_P$, R_P decreases in a small step of δ, which is 0.125 in our design. The new R_P is used by OTW_P calculation block to update OTW'_P. When OTW'_P is equal to OTW^*_P, the calibration process terminates and R_P is finally fixed to 4.75 as shown in Fig. 8. The whole R_P-calibration process costs 29 CKR clock cycles (about 2.3 μs). It should be noted that the R_P-calibration process could be carried out during idle time of the ADPLL, therefore it does not increase the settling time of the ADPLL in normal operation.

5.2 Results of OTW estimating and presetting

An example is given in this part to show the details of OTW estimating and presetting. From the example we can see how the proposed techniques speed up the locking process. The input FCW is 188.461517333984375 so the desired output frequency is 2450 MHz. Fig. 9*a* shows the transient response of the ADPLL without OTW presetting. The *x*-axis is the time evolution in CKV clock units (about 417 ps/cycle). The *y*-axis is the time deviation expressed in femtoseconds from an initial value of 417 ps, which is the free running DCO cycle. In order to observe the details of locking process, zoom views of the three operation modes are shown in Figs. 9*b*–*d* respectively. In the PVT-calibration mode, as shown in Fig. 9*b*, the DCO output frequency jumps slowly to the desired frequency, and finally reaches the locking state. Owing to frequency quantisation effect of the DCO, the output frequency toggles between two adjacent frequency levels around the desired frequency. The toggling is marked in the elliptic line in Fig. 9*b*. When the PVT-calibration mode is completed, the OTW_P is locked to 19. The same processes take place in the acquisition mode and tracking mode, as shown in Figs. 9*c* and *d*. When the output frequency is locked, the OTW_A and OTW_T are locked to 3 and 15, respectively. The total settling time is about 13.2 μs.

As a comparison with Fig. 9*a*, Fig. 10*a* shows the transient response of the proposed ADPLL with OTW presetting. The zoom views of the three operation modes are shown in Figs. 10*b*–*d*, from which we can see the details of our techniques. In the PVT-calibration mode as shown in Fig. 10*b*, the calculated OTW'_P is directly sent to the DCO. Compared with Fig. 9*b* we can see that the PVT-calibration process is significantly accelerated. At the end of PVT-calibration mode, the output frequency toggles between two adjacent frequency levels for two times, which corresponds with Fig. 6*c*. So the presetting OTW'_A for the acquisition mode is 2, as shown in Fig. 10*c*. Although the final required OTW^*_A is 4, which is different from our presetting value, it is still better than using an initial OTW^0_A of 0. At the end of the acquisition mode, the OTW_A stays at the value of 4 for a long time. This means the current

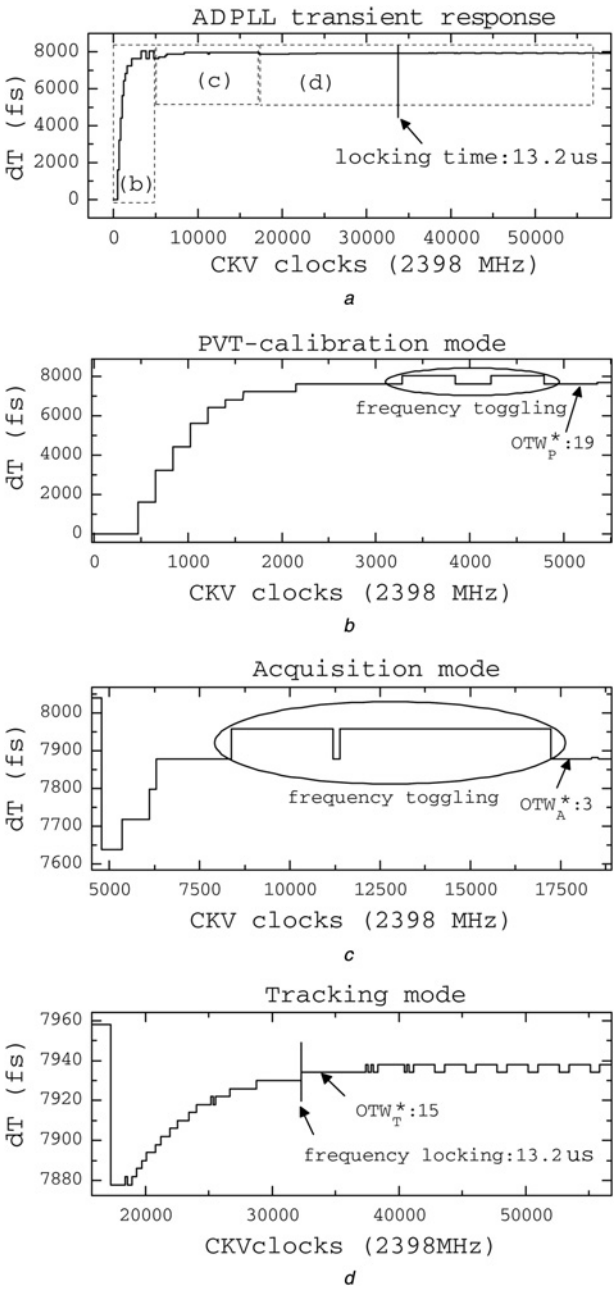

Figure 9 *Transient response of the ADPLL without OTW presetting (a), PVT-calibration mode (b), acquisition mode (c), and tracking mode (d)*

output frequency is close to the desired frequency, which corresponds with Fig. 6*a*. So the ADPLL switches to the tracking mode immediately and the OTW'_T is preset to 0, as shown in Fig. 10*d*. In the tracking mode, the OTW_T jumps one step to the final OTW^*_T, of −1. The total settling time is merely 3.9 μs. It can be seen that at the end of the tracking mode, the output frequency is still toggling. This is due to the finite frequency resolution of the DCO, which is 23 kHz in our design. A further finer frequency resolution could be achieved by using a ΣΔ modulator [2].

Furthermore, as shown in Figs. 9 and 10, the final OTWs ($[OTW^*_P, OTW^*_A, OTW^*_T]$) of both ADPLLs are [19, 3, 15]

doi: 10.1049/iet-cds.2009.0173

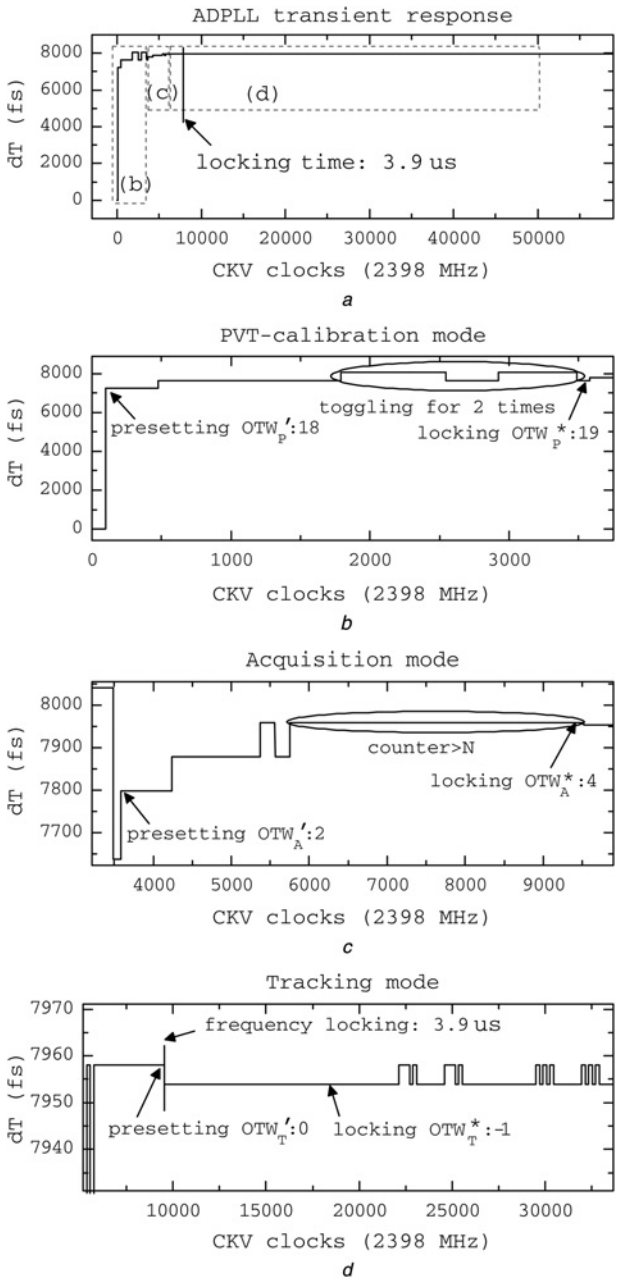

Figure 10 *Transient response of the ADPLL with OTW presetting (a), PVT-calibration mode (b), acquisition mode (c), and tracking mode (d)*

and [19, 4, −1]. This indicates that for the same desired output frequency, the required OTW is not unique. Our proposed techniques can automatically choose the proper OTW to achieve a shorter settling time.

From Fig. 10 we can see that the OTW presetting occurs at the beginning of each operation mode. On the other hand, the dynamic loop bandwidth control technique, which was called 'gear-shifting' in [11], was used during the tracking mode of the ADPLL. So the two techniques can be used together to achieve a short settling time and also a good noise performance. Firstly, we use OTW presetting technique in the three operation modes for fast settling. Secondly, when the desired frequency is acquired in the tracking mode, the 'gear-shifting' can be used to reduce the loop bandwidth to reduce phase noise.

5.3 Settling time comparisons of both ADPLLs

Both of the ADPLLs shown in Figs. 1 and 2 are realised and simulated at different output frequencies from 2.4 to 2.5 GHz in a step of 10 MHz. The effect of ε_K is taken into account by adding 1% mismatch and 10% deviation to the capacitors in the DCO. The simulation results are presented in Tables 2–4. Figures based on these data are drawn as shown in Figs. 11 and 12. Settling time is defined as the time elapsed between the ADPLL being reset and the output frequency reaching the desired frequency. The frequency tolerance is Δf_T, which is the frequency resolution of the tracking mode. It can be seen from Fig. 11 that our proposed techniques are effective during a large frequency band and are immune to non-ideality of the capacitors. The settling time at all frequency points is no more than 10 μs. Fig. 12 shows the settling time improvements of the proposed fast-locking ADPLL compared with the one without OTW presetting block. The improvement on average is about 40–50%.

In the three cases with different capacitors, the minimum improvement appears at 2.4 GHz. This is because at 2.4 GHz frequency point, the initial frequency difference is 2 MHz (2.4–2.398 GHz), which is smaller than Δf_P (2.333 MHz), so the required OTW_P^*, is 0. The effect of

Table 2 Settling time of both of the ADPLLs and the improvement. The capacitors in the DCO are ideal ones

Frequency (GHz)	$T(a)$[a] (μs)	$T(b)$[b] (μs)	Improvement[c] (%)
2.4	11.4	10.0	12.3
2.41	15.9	3.6	77.4
2.42	11.6	7.3	37.1
2.43	10.6	6.5	38.7
2.44	11.0	5.3	51.8
2.45	10.7	9.0	15.9
2.46	13.7	7.0	48.9
2.47	16.3	7.5	54.0
2.48	10.8	8.8	18.5
2.49	11.4	7.0	38.6
2.5	11.2	7.2	35.7

[a]Settling time of the ADPLL without OTW presetting (Fig. 1).
[b]Settling time of the ADPLL with OTW presetting (Fig. 2).
[c]Obtained by $(T(a) - T(b))/T(a) \times 100\%$.

Table 3 Settling time of both of the ADPLLs and the improvement. The capacitors in the DCO are with 1% mismatch

Frequency (GHz)	$T(a)$[a] (μs)	$T(b)$[b] (μs)	Improvement[c] (%)
2.4	11.0	9.0	18.2
2.41	13.0	7.3	43.8
2.42	9.3	7.0	24.7
2.43	10.7	4.2	60.7
2.44	9.6	7.3	24.0
2.45	10.3	5.6	45.6
2.46	11	5.3	51.8
2.47	13.1	2.9	77.9
2.48	11.7	5.5	53.0
2.49	11.3	3.4	69.9
2.5	11.3	3.3	70.8

[a]Settling time of the ADPLL without OTW presetting (Fig. 1).
[b]Settling time of the ADPLL with OTW presetting (Fig. 2).
[c]Obtained by $(T(a) - T(b))/T(a) \times 100\%$.

OTW_P presetting can be neglected in this condition. So the improvement at this frequency point is smaller than at others. The maximum improvement in the three cases appears at different frequency points. The reason is that in the three cases, the capacitors in the DCO are different, which causes different Δf. So the same output frequency corresponds to different OTWs. Therefore the improvements are different at the same output frequency point in the three cases and the maximum one appears at an uncertain place.

Table 4 Settling time of both of the ADPLLs and the improvement. The capacitors in the DCO are with 1% mismatch and 10% deviation

Frequency (GHz)	$T(a)$[a] (μs)	$T(b)$[b] (μs)	Improvement[c] (%)
2.4	8.9	7.4	16.9
2.41	15.8	5.3	66.5
2.42	9.0	3.5	61.1
2.43	13.3	7.7	42.1
2.44	11.3	4.7	58.4
2.45	13.2	3.9	70.5
2.46	11.4	5.2	54.4
2.47	14.2	5.5	61.3
2.48	8.9	5.5	38.2
2.49	12.6	6.3	50.0
2.5	10.1	3.5	65.3

[a]Settling time of the ADPLL without OTW presetting (Fig. 1).
[b]Settling time of the ADPLL with OTW presetting (Fig. 2).
[c]Obtained by $(T(a) - T(b))/T(a) \times 100\%$.

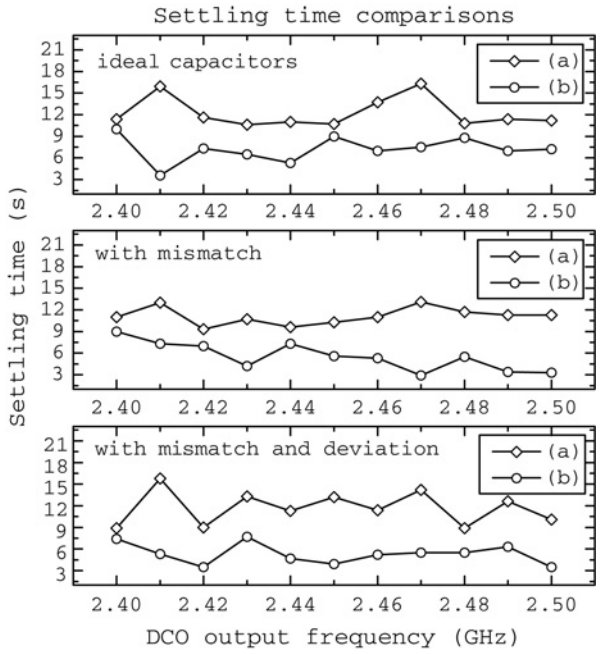

Figure 11 *Settling time comparisons of the two ADPLLs: (a) Is the ADPLL without OTW presetting and (b) Is the proposed ADPLL with OTW presetting*
The non-ideality of the capacitors in the DCO is considered

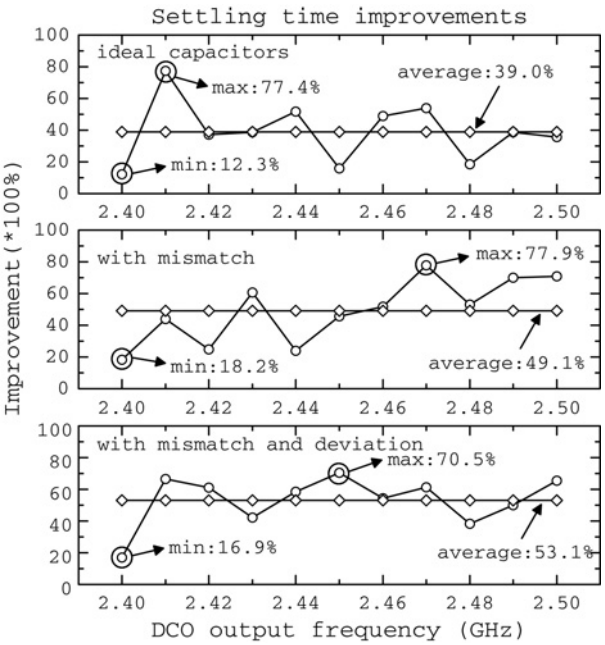

Figure 12 *Improvements at different output frequency points*

doi: 10.1049/iet-cds.2009.0173

6 Conclusion

We propose techniques that estimate and preset the OTW for the ADPLL to reduce its settling time in this article. All the three operation modes of the ADPLL are taken into account. For the PVT-calibration mode, a direct OTW calculation method is used with a foreground calibration block to eliminate the effect of K_{DCO} estimation error, which ensures the validity of the proposed method with a large DCO capacitance deviation and mismatch. For the acquisition mode and tracking mode, a counter-based mode switching controller is proposed to accelerate the mode switching process and estimate the OTW for presetting. This method is based on the ADPLL's inherent characteristic of frequency toggling, and consequently independent of loop parameters. Simulation results demonstrate that our proposed techniques can reduce the settling time by about 50% on average even there are large DCO capacitance deviation and mismatch. The settling time of the proposed fast-locking ADPLL is several microseconds over a large output frequency band. Furthermore, the proposed presetting technique can be used with the dynamic loop bandwidth control together. So a fast settling can be achieved without degrading the noise performance.

7 Acknowledgments

This work was sponsored by the National Key Technological Program of China under contracts, No. 2008ZX01035-001, National Natural Science Foundation of China (No.60870001), 863 program of China (No. 2009AA01Z130) and Tsinghua National Laboratory for Information Science and Technology (TNList) Cross-discipline Foundation.

8 References

[1] STASZEWSKI R.B., BALSARA P.T.: 'Phase-domain all-digital phase-locked loop', *IEEE Trans. Circuits Syst. II: Express Briefs*, 2005, **52**, (3), pp. 159–163

[2] STASZEWSKI R.B., HUNG C.M., LEIPOLD D., BALSARA P.T.: 'A first multigigahertz digitally controlled oscillator for wireless applications', *IEEE Trans. Microw. Theory Tech.*, 2003, **51**, (11), pp. 2154–2164

[3] STASZEWSKI R.B., VEMULAPALLI S., VALLUR P., WALLBERG J., BALSARA P.T.: '1.3 V 20 ps time-to-digital converter for frequency synthesis in 90-nm CMOS', *IEEE Trans. Circuits Syst. II: Express Briefs*, 2006, **53**, (3), pp. 220–224

[4] JEON H.S., YOU D.H., PARK I.C.: 'Fast frequency acquisition all-digital PLL using PVT calibration'. IEEE Int. Symp. on Circuits and Systems, 2008, pp. 2625–2628

[5] HSU C.H., STRAAYER M.Z., PERROTT M.H.: 'A low-noise wide-BW 3.6-GHz Digital$ DeltaSigma$ fractional-N frequency synthesizer with a noise-shaping time-to-digital converter and quantization noise cancellation', *IEEE J. Solid-State Circuits*, 2008, **43**, (12), pp. 2776–2786

[6] SWAMINATHAN A., WANG K.J., GALTON I.: 'A wide-bandwidth 2.4 GHz ISM band fractional-N PLL with adaptive phase noise cancellation', *IEEE J. Solid-State Circuits*, 2007, **42**, (12), pp. 2639–2650

[7] WOO K., LIU Y., NAM E., HAM D.: 'Fast-lock hybrid PLL combining fractional', *IEEE J. Solid-State Circuits*, 2008, **43**, (2), p. 379

[8] STASZEWSKI R.B., WALLBERG J.L., REZEQ S., *ET AL.*: 'All-digital PLL and transmitter for mobile phones', *IEEE J. Solid-State Circuits*, 2005, **40**, (12), pp. 2469–2482

[9] ZHANG B., ALLEN P.: 'Feed-forward compensated high switching speed digital phase-locked loop frequency synthesizer'. IEEE Int. Symp. on Circuits and Systems, 1999, pp. 371–374

[10] SHIN S., LEE K., KANG S.M.: '4.2 mW CMOS frequency synthesizer for 2.4 GHz ZigBee application with fast settling time performance'. IEEE MTT-S Int. Microwave Symp. Digest, 2006, pp. 411–414

[11] STASZEWSKI R.B., BALSARA P.T.: 'All-digital PLL with ultra fast settling', *IEEE Trans. Circuits Syst. II: Express Briefs*, 2007, **54**, (2), pp. 181–185

[12] KUANG X., WU N.: 'A fast-settling PLL frequency synthesizer with direct frequency presetting'. IEEE Int. Solid-State Circuits Conf., 2006. Digest of Technical Papers, pp. 741–750

[13] CHAIVIPAS W., MATSUZAWA A., OH P.C.: 'Feed-forward compensation technique for all digital phase locked loop based synthesizers'. IEEE Int. Symp. on Circuits and Systems, 2006, pp. 3209–3212

[14] DUNNING J., GARCIA G., LUNDBERG J., NUCKOLLS E.: 'An all-digital phase-locked loop with 50-cycle lock time suitable for high-performance microprocessors', *IEEE J. Solid-State Circuits*, 1995, **30**, (4), pp. 412–422

[15] SHENG D., CHUNG C.C., LEE C.Y.: 'A fast-lock-in ADPLL with high-resolution and low-power DCO for SoC applications'. IEEE Asia Pacific Conf. on Circuits and Systems, 2006, pp. 105–108

[16] STASZEWSKI R.B., BALSARA P.T.: 'All-digital frequency synthesizer in deep-submicron CMOS' (Wiley-Interscience, 2006), pp. 189–206