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 ($\delta_k$) 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
In the ADPLL as shown in Fig. 1, the DCO output frequency is set by the OTW, which consists of three parts, namely OTWP, OTWA, and OTWT. 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 ([OTWP, OTWA, OTWT]) 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 $f_{CKV}^*$

$$f_{CKV}^* = FCW \times FREF$$

the OTW is settled to $OTW^*$ ([OTWP*, OTWA*, OTWT*]). 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 ($e_K$) using a foreground calibration block and then directly calculate the OTWP for presetting. For the acquisition mode and tracking mode, a novel counter-based mode switching controller is proposed to estimate the OTWA and OTWT 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 $e_K$. 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, OTWP calculation, and counter-based mode switching controller (CB-MSC).
1. Self-calibration block works before ADPLL normal operation, as shown in Fig. 3. This block is used to eliminate the effect of $e_K$ 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 $\frac{F_{REF}}{D_f}$ (see in Section 3.1), will be used in the following OTWP calculation block. The error of $D_f$ 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 OTWP calculation block and the self-calibration block becomes idle. The algorithm of this block is presented in Section 4.1.

2. OTWP calculation block works in the PVT-calibration mode. Using the calibrated $R_P$ and the outer signals of $F_{REF}$ and $F_{CW}$, this block generates an OTW $P^*$ estimation value defined as $O_TWP^*$. Then OTW $P^*$ is added to the original OTW $P$ 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 $O_TWA$ and $O_TWT$ 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 $e_K$ 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 $\Delta f$. In the ADPLL, $\Delta f$ 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 $\Delta f_P$, which is 2333 kHz in our design. Finally, the OTWP 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 OTWP is locked to 6. The residual frequency difference, which is smaller than $\Delta f_P$, 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 $\Delta f_A$, which is 397 kHz in our design. At the end of the acquisition mode, the OTWA is locked to 4. The tracking mode is almost the same as the former ones except it has the finest frequency step of $\Delta f_T$, which is 23 kHz.

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

$$f_{CKV} - f_{free} = F_{CW} \times F_{REF} - f_{free} = O_TWP^* \times \Delta f_P + O_TWA \times \Delta f_A + O_TWT \times \Delta f_T + \delta_f \tag{2}$$

where $O_TWP^*$, $O_TWA$ and $O_TWT$ are integer. $\delta_f$ is the final residual frequency difference, which is smaller than $\Delta f_T$.  

![Figure 4 Example of locking process of the ADPLL](image-url)
For convenience, we define

\[ \xi_f \triangleq FCW - \frac{f_{\text{free}}}{\text{FREF}} \]  
\[ R_p \triangleq \frac{\text{FREF}}{\Delta f_p} \]  
\[ R_A \triangleq \frac{\Delta f_A}{\Delta f_A} \]  
\[ R_T \triangleq \frac{\Delta f_T}{\Delta f_T} \]  

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

\[ f_{\text{CKV}} - f_{\text{free}} = FCW \times \text{FREF} - f_{\text{free}} \triangleq OTW_p^* \times \Delta f_p + \xi_p \times \Delta f_p \]  

where \(|\xi_p| < 1\). This is because the residual frequency difference after the PVT-calibration mode is smaller than \(\Delta f_p\). So \( OTW_p^* \) is obtained by

\[ OTW_p^* = \left[ \left( FCW - \frac{f_{\text{free}}}{\text{FREF}} \right) \times \frac{\text{FREF}}{\Delta f_p} \right]_I = [\xi_f \times R_p]_I \]  

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

\[ \xi_p = \left[ \left( FCW - \frac{f_{\text{free}}}{\text{FREF}} \right) \times \frac{\text{FREF}}{\Delta f_p} \right]_F = [\xi_f \times R_p]_F \]  

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

\[ \xi_p \times \Delta f_p = [\xi_f \times R_p]_F \times \Delta f_p \] 
\[ = OTW_A^* \times \Delta f_A + OTW_T^* \times \Delta f_T + \delta_f \] 
\[ \triangleq OTW_A^* \times \Delta f_A + \xi_A \times \Delta f_A \]  

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

\[ OTW_A^* = [[\xi_f \times R_p]_F \times R_A]_I \]  

Furthermore, \(\xi_A\) is obtained by

\[ \xi_A = [[\xi_f \times R_p]_F \times R_A]_F \]  

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

\[ \xi_A \times \Delta f_A = [[\xi_f \times R_p]_F \times R_A]_F \times \Delta f_A \] 
\[ = OTW_T^* \times \Delta f_T + \delta_f \]  

So \( OTW_T^* \) is obtained by

\[ OTW_T^* = [[\xi_f \times R_p]_F \times R_A]_F \times R_T]_I \]  

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 \(\Delta f_p\), \(\Delta f_A\) and \(\Delta f_T\) 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 \(\epsilon_K\) 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 \( K_{\text{DCO}} \) estimation error \((\epsilon_K)\) and affect the \( OTW \) calculation. In order to investigate the effects, we take \(\epsilon_K\) into account in derivations of \( OTW \) calculation.

Owing to \(\epsilon_K\) in the PVT-calibration mode, an error term of \(\epsilon_p\) should be added to \( R_p \) as

\[ R_p' \triangleq R_p + \epsilon_p \]  

By replacing \( R_p \) in (8) with \( R_p' \), we obtain the \( OTW_p \) calculation value containing \( R_p \) error

\[ OTW_p' = [\xi_f \times R_p']_I = [\xi_f \times (R_p + \epsilon_p)]_I \] 
\[ = [\xi_f \times R_p]_I + [\xi_f \times \epsilon_p]_I \] 
\[ + [[\xi_f \times R_p]_F + [\xi_f \times \epsilon_p]_F]_I \] 
\[ = OTW_p^* + [\xi_f \times \epsilon_p]_F \] 
\[ + [[\xi_f \times R_p]_F + [\xi_f \times \epsilon_p]_F]_I \]  

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 \(\xi_f\) times, so it has a great effect on the \( OTW_p \) calculation.

Substituting (15) into (9), we obtain \(\xi_p\)

\[ \xi_p = [\xi_f \times R_p']_F = [\xi_f \times (R_p + \epsilon_p)]_F \] 
\[ = [\xi_f \times R_p]_F + [\xi_f \times \epsilon_p]_F \]  

If \(0 \leq \xi_p + [\xi_f \times \epsilon_p]_F < 1\), then (17) can be rewritten as

\[ \xi_p = \xi_f + [\xi_f \times \epsilon_p]_F \]  

Substituting (18) into (11) we obtain

\[ OTW_A' = [\xi_f \times R_A]_I = [[\xi_f + [\xi_f \times \epsilon_p]_F] \times R_A]_I \] 
\[ = [\xi_f \times R_A]_I + [[\xi_f \times \epsilon_p]_F \times R_A]_I + \alpha \] 
\[ = OTW_A^* + [\xi_f \times \epsilon_p]_F \times R_A]_I + \alpha \]
where

\[ \alpha = [(\xi_F \times R_A)_F] + [(\xi_F \times e_P)_F \times R_A)_F] \]  

and the value of \( \alpha \) is 0 or 1 and it has little effect on OTW calculation.

If \( 1 \leq \xi_P + [\xi_F \times e_P)_F < 2 \), then (17) can be rewritten as

\[ \xi_P = \xi_P + [\xi_F \times e_P)_F - 1 \]  

Substituting (21) into (11) we obtain

\[
\text{OTW}'_A = [(\xi_P + [\xi_F \times e_P)_F - 1) \times R_A)_F] \\
= [(\xi_P \times R_A) - (1 - [\xi_F \times e_P)_F \times R_A)_F] \\
= \text{OTW}_A - ([1 - (1 - [\xi_F \times e_P)_F \times R_A)_F] - \beta \\
\]

where

\[
\beta = \begin{cases} 
1, & \xi_P \times R_A)_F \leq ([1 - \xi_F \times e_P)_F \times R_A)_F \\
0, & \xi_P \times R_A)_F \geq ([1 - \xi_F \times e_P)_F \times R_A)_F 
\end{cases} 
\]

Equations (16), (19) and (22) indicate that \( e_P \) not only affects the OTW calculation result, but also affects OTW calculation result. Furthermore, OTW calculation result will be affected by the error of \( R_A \). It can be inferred that OTW 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 \( \Delta \) estimation error on OTW calculation.

\[
\begin{align*}
\text{OTW}'_P &= f(P, e_P) \\
\text{OTW}'_A &= f(P, e_P, e_A) \\
\text{OTW}'_T &= f(P, e_P, e_A, e_T) 
\end{align*}
\]

where \( P \) denotes loop parameters, \( e_P, e_A \) and \( e_T \) 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 \( e_K \) on OTW calculation.

## 4 Proposed OTW estimation methods

According to the analysis in Section 3, we can see that the \( e_K \) 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 \( e_P \) and then the OTW calculation block calculates the OTW for presetting. For the acquisition mode and tracking mode, the CB-MSC block is used to generate OTW and OTW 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 calculation result is only affected by the error of \( R_P \). So if we can calibrate the error of \( R_P \), the required OTW can be obtained. Fig. 5 shows the algorithm of RP-calibration process. Before ADPLL normal operation, a known FCW is sent to the ADPLL. A large FCW is chosen because at this frequency a large OTW calculation error is produced according to (3) and (16). In the RP-calibration process, when the PVT-calibration mode is completed, a required OTW is generated by the ADPLL itself. Meanwhile, OTW is the estimation value of OTW is calculated by the OTW calculation block. Then we compare OTW with OTW, if OTW is larger than OTW then \( R_P \) is decreased by a fixed small step of \( \delta \). Else, if OTW is smaller than OTW, \( R_P \) is increased by the fixed small step of \( \delta \). Using the updated \( R_P \), a new OTW is calculated and sent to be compared with OTW again. This operation cycles until OTW is equal to OTW. 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 RP-calibration process is completed, the self-calibration block becomes idle.

### 4.2 OTW calculation

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

\[
\text{OTW}'_P = \text{OTW}'_P + [\xi_F \times e_P)_F] + [(\xi_F \times R_P)_F \\
+ [\xi_F \times e_P)_F] = \text{OTW}'_P 
\]

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

![Figure 5](image-url)
convenience, we rewrite (8) here as

\[
\text{OTW}_p^* = \left[\left(\text{FCW} - \frac{f_{\text{free}}}{\text{FREF}}\right) \times \frac{\text{FREF}}{\Delta f_p}\right]_1 = [\xi_F \times R_p]_1
\]

(26)

where \(\xi_F\) 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 \(\text{OTW}_p^*\) 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, \(\text{OTW}_A\) calculation is affected by both errors of \(R_p\) and \(R_A\), and \(\text{OTW}_T\) 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 \(\text{OTW}_A\) and \(\text{OTW}_T\) 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 \((f_{\text{CKV}})\). Furthermore, when \(f_{\text{CKV}}\) is close to \(f_2\) or \(f_1\) the time \((T_{\text{high}})\) that the output frequency stays at \(f_2\) is longer than the time \((T_{\text{low}})\) that it at \(f_1\), as shown in Fig. 6a. This is because when the DCO output frequency is \(f_2\), the frequency difference between \(f_2\) and \(f_{\text{CKV}}\) is very small, so it takes a long time for the ADPLL to accumulate a 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 \(f_{\text{CKV}}\) 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. 6b and 6c, respectively.

We can use two counters to record \(T_{\text{high}}\) and \(T_{\text{low}}\) 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. \(\ast\) and \(X\) in Fig. 7a 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.

As shown in Fig. 7a, 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. 6c and the \(\text{OTW}_A\) (or \(\text{OTW}_T\)) 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
ADPLL. This condition means the DCO output frequency stays at either \( f_1 \) or \( f_2 \) for a long time, which corresponds with Figs. 6a or 6b. In this condition, the DCO output frequency is close to the desired frequency \( f_{\text{CKV}}^\ast \), 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 \( \Sigma \Delta \) 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.

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 192.306782265625 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

<table>
<thead>
<tr>
<th>Table 1 ADPLL simulation parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>reference frequency ( F_{\text{REF}} )</td>
</tr>
<tr>
<td>DCO free running frequency ( f_{\text{free}} )</td>
</tr>
<tr>
<td>FCW</td>
</tr>
<tr>
<td>output frequency range</td>
</tr>
<tr>
<td>loop filter gain</td>
</tr>
<tr>
<td>loop bandwidth</td>
</tr>
<tr>
<td>TDC resolution</td>
</tr>
<tr>
<td>( \Delta f_P )</td>
</tr>
<tr>
<td>( \Delta f_A )</td>
</tr>
<tr>
<td>( \Delta f_T )</td>
</tr>
<tr>
<td>OTW( _P )</td>
</tr>
<tr>
<td>OTW( _A )</td>
</tr>
<tr>
<td>OTW( _T )</td>
</tr>
<tr>
<td>( \Sigma \Delta ) modulator</td>
</tr>
</tbody>
</table>

Figure 7 Principle of the counter-based mode switching controller

\( a \) is a schematic plan of frequency toggling
\( b \) is state switching based on \( a \)

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 calculation block to update OTW_P. When OTW_P is equal to OTW_b, 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 188.461733984375 MHz so the desired output frequency is 2450 MHz. Fig. 9a 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. 9b–d respectively. In the PVT-calibration mode, as shown in Fig. 9b, 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. 9b. 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. 9c 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. 9a, Fig. 10a shows the transient response of the proposed ADPLL with OTW presetting. The zoom views of the three operation modes are shown in Figs. 10b–d, from which we can see the details of our techniques. In the PVT-calibration mode as shown in Fig. 10b, the calculated OTW′_P is directly sent to the DCO. Compared with Fig. 9b 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. 6c. So the presetting OTW′_P for the acquisition mode is 2, as shown in Fig. 10c. Although the final required OTW∗_A is 4, which is different from our presetting value, it is still better than using an initial OTW_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 output frequency is close to the desired frequency, which corresponds with Fig. 6a. So the ADPLL switches to the tracking mode immediately and the OTW_T is preset to 0, as shown in Fig. 10d. 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]
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 \(K_F\) 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 \(\Delta f_T\), 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 \(\mu\)s. 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 \(\Delta f_P\) (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

<table>
<thead>
<tr>
<th>Frequency (GHz)</th>
<th>T(_a) ((\mu)s)</th>
<th>T(_b) ((\mu)s)</th>
<th>Improvement (%)</th>
</tr>
</thead>
<tbody>
<tr>
<td>2.4</td>
<td>11.4</td>
<td>10.0</td>
<td>12.3</td>
</tr>
<tr>
<td>2.41</td>
<td>15.9</td>
<td>3.6</td>
<td>77.4</td>
</tr>
<tr>
<td>2.42</td>
<td>11.6</td>
<td>7.3</td>
<td>37.1</td>
</tr>
<tr>
<td>2.43</td>
<td>10.6</td>
<td>6.5</td>
<td>37.1</td>
</tr>
<tr>
<td>2.44</td>
<td>11.0</td>
<td>5.3</td>
<td>51.8</td>
</tr>
<tr>
<td>2.45</td>
<td>10.7</td>
<td>9.0</td>
<td>15.9</td>
</tr>
<tr>
<td>2.46</td>
<td>13.7</td>
<td>7.0</td>
<td>48.9</td>
</tr>
<tr>
<td>2.47</td>
<td>16.3</td>
<td>7.5</td>
<td>54.0</td>
</tr>
<tr>
<td>2.48</td>
<td>10.8</td>
<td>8.8</td>
<td>18.5</td>
</tr>
<tr>
<td>2.49</td>
<td>11.4</td>
<td>7.0</td>
<td>38.6</td>
</tr>
<tr>
<td>2.5</td>
<td>11.2</td>
<td>7.2</td>
<td>35.7</td>
</tr>
</tbody>
</table>

\(a\) 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\textsubscript{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 $\Delta f$. 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 3** Setting time of both of the ADPLLs and the improvement. The capacitors in the DCO are with 1% mismatch

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

$^a$Setting time of the ADPLL without OTW presetting (Fig. 1).

$^b$Setting time of the ADPLL with OTW presetting (Fig. 2).

$^c$Obtained by $(T(a) - T(b))/T(a) \times 100\%$.

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

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

$^a$Setting time of the ADPLL without OTW presetting (Fig. 1).

$^b$Setting time of the ADPLL with OTW presetting (Fig. 2).

$^c$Obtained by $(T(a) - T(b))/T(a) \times 100\%$.

**Figure 11** Setting 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.
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 resetting. 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