A Fast Locking Digitally Controlled PLL for Constant-Gain Digitally Controlled Oscillator

Chen Xin and Zhang Wanqiao
College of Electronic and Information Engineering, Nanjing University of Aeronautics and Astronautics, Nanjing, Jiangsu, 210016, China
Email: xin_chen@nuaa.edu.cn, wanqiao_zhang@163.com

***Abstract*—A fast locking digitally controlled phase-locked loop (DCPLL) with a novel frequency search algorithm is presented in this paper. The proposed frequency search algorithm can predict the target code by two predetermined codes and the two corresponding digital phase errors. To implement the proposed frequency algorithm in the PLL, a DCPLL with a constant-gain digitally controlled oscillator is developed and implemented in SMIC 0.13μm 1P8M technology. Finally, the frequency acquisitions are simulated for the whole frequency range. The simulation results show that the maximum locking time of the DCPLL is four reference clock cycles.**

***Index Terms*—digitally controlled phase-locked loop (DCPLL), fast locking, digitally controlled oscillator (DCO)**

I. INTRODUCTION

Phase-locked loops (PLLs) have been widely used as clock generators in integrated circuits(ICs) and system-on-chip (SoC) processors[1], [2]. To reduce the power consumption, two power management methods which are dynamic frequency scaling(DFS) and dynamic voltage scaling(DVS) are usually implemented in SoCs [3], [4]. The principles of the two technologies are similar: the operating frequency or supply voltage of the IC varies with the workload, even shuts down the idle circuits to minimize the power consumption. Since the maximum operating frequency is voltage-dependent, the operating frequency should be adjusted in some cases for DVS applications. Therefore, both the two technologies require the PLL to provide the correct clock in a short time when the workload is changed, or a long wait time is needed. So designing a fast locking PLL is crucial for low power ICs.

To shorten the locking time, the typical method is adjusting the loop parameters dynamically. This method can be applied to both the traditional PLLs [5] and digitally controlled PLLs(DCPLLs)[6]. For this kind of PLLs, the PLL must be stable over a wide range of bandwidth. Therefore, balancing the locking speed and stability of the PLL emerges as a new problem.

Compared with traditional PLLs, more innovative fast locking methods are applied to DCPLLs. Reference [7] obtains fast frequency acquisition by employing a digital phase/frequency detector (DPFD) and a variable loop gain scheme. The minimum locking time is obtained when the loop gain $K_I K_o / \omega_{REF}$ equals one, where K_I is the proportional gain in the digital filter, K_o is the digitally controlled oscillator (DCO) gain and ω_{REF} is the angular frequency of the reference clock. Since K_o is a parameter of process, voltage and temperature (PVT) variation, it is difficult to precisely predict the optimum K_I to achieve the minimum locking time. Seven-cycle locking time is achieved by a high-linearity DCO [8]. However, the frequency search algorithm depends on the structure of the DCO heavily, and the frequency range of the DCO is hard to enlarge. Binary search algorithm (BSA) is widely used in DCPLLs [9]. The BSA is implemented easily and can achieve a fast locking. The maximum locking time is proportional to $O(\log_2 n)$ where n is the number of discrete frequency points in the DCO. Thereby, a trade-off among locking time, frequency range and DCO gain must be made when the BSA is utilized in a DCPLL.

In many DCO designs [9], [10], the period t_{DCO} of the DCO can be defined as follows.

$$t_{DCO} = t_{\max} - K_o C \tag{1}$$

where $t_{\max}$ is the maximum period of the DCO, K_o is the DCO gain measured in second, and C is the code for the DCO.

From Eq.(1), it is easily derived that the code C for the period t_{DCO} can be directly calculated if the values of K_o and $t_{\max}$ are obtained. However, the two parameters are PVT-dependent and difficult to measure. In this paper, a novel frequency search algorithm is proposed. With the proposed frequency search algorithm, C can be directly predicted by two predetermined codes and the corresponding digital phase errors. Then the DCPLL structure for the proposed search frequency is developed and the operation of the DCPLL is designed. The simulation results show that the locking time of the designed DCPLL can be decreased greatly.

The paper is organized as follows. In Section II, the frequency search algorithm is introduced and the related mathematical analysis is given. Then the structure of the

Manuscript received Nov. 13, 2013; revised February 4, 2014.

doi: 10.12720/ijeee.2.2.95-100

proposed DCPLL is introduced in Section III and the operation of the proposed DCPLL is described in Section IV. Finally, simulation results are given in Section V.

II. Proposed Frequency Search Algorithm

A. The Deduction of the Algorithm

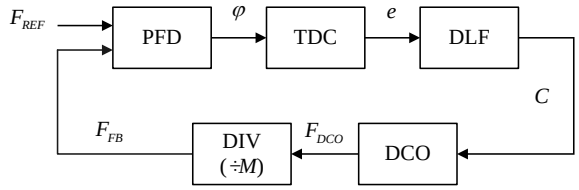


Figure 1. DCPLL block structure

A simplified block diagram of the DCPLL is shown in Fig. 1. The block structure of DCPLL consists of a phase/frequency detector (PFD) which is usually adopted, a time-to-digital converter (TDC), a digital loop filter (DLF), a DCO, and a frequency divider (DIV).

The operation of DCPLL is very similar to a charge-pump PLL. The PFD generates the phase error φ by comparing the falling edges of reference clock F_{REF} and feedback clock F_{FB}. The TDC circuit converts the phase error φ to the digital phase error e. The digital phase error e is sent to the DLF and filtered. Then the code C from the DLF tunes the DCO. Finally, the output of the DCO F_{DCO} is divided and sends to the PFD.

Therefore, if the previous phase error φ_0 is zero, it means that F_{REF} and F_{FB} falls at the same time $T = T_1$. Then the next falling time T_2 for F_{REF} is $T_2 = T_1 + t_{REF}$ where t_{REF} is the period of F_{REF}. The next falling time T_3 for F_{FB} is $T_3 = T_1 + t_{FB1} = T_1 + Mt_1$ in which M is the frequency divider ratio, t_{FB1} and t_1 are the periods of F_{FB} and the DCO for the code C_1, respectively. So the phase error φ_1 will be the period difference between t_{REF} and Mt_1 which is shown in Eq.(2).

$$\varphi_1 = T_3 - T_2 = Mt_1 - t_{REF} = e_1 \Delta_{TDC} \tag{2}$$

where e_1 is the corresponding digital phase error, Δ_{TDC} is the resolution of the TDC.

Under the same condition, the phase error φ_2 for the code C_2 is given in Eq.(3).

$$\varphi_2 = Mt_2 - t_{REF} = e_2 \Delta_{TDC} \tag{3}$$

where t_2 is the period of the DCO for C_2, e_2 is the corresponding digital phase error.

Eq.(2)-Eq.(3), Eq.(4) is obtained.

$$M(t_1 - t_2) = \Delta_{TDC}(e_1 - e_2) \tag{4}$$

If the DCO periods t_1 and t_2 are substituted by Eq.(1), Eq.(4) is rewritten as Eq.(5).

$$\frac{\Delta_{TDC}}{MK_o} = -\frac{C_1 - C_2}{e_1 - e_2} \tag{5}$$

When the PLL locks, the target period of the DCO is t_t. The corresponding code is C_t. Eq.(6) can be derived from Eq.(1).

$$t_{REF} = Mt_t = M(t_{\max} - K_o C_t) \tag{6}$$

Replacing t_1 and t_{REF} in Eq.(2) with Eq.(1) and Eq.(6), respectively, Eq.(2) is rewritten as Eq.(7) .

$$MK_o(C_t - C_1) = e_1 \Delta_{TDC} \tag{7}$$

Then C_t is given by Eq.(8). Seen from Eq.(8), it is found that to calculate C_t, Δ_{TDC} / MK_o is the key parameter.

$$C_t = C_1 + \frac{e_1 \Delta_{TDC}}{MK_o} \tag{8}$$

Substituting Eq.(5) into Eq.(8), Eq.(8) is rewritten as Eq.(9).

$$C_t = C_1 - \frac{e_1}{e_1 - e_2}(C_1 - C_2) \tag{9}$$

From Eq.(9), it is seen that C_t can be predicted based on e_1, e_2, C_1 and C_2. Compared with K_o and $t_{\max}$, the four parameters are much easier to obtain. In addition, because C_t is calculated by Eq.(9) for every initialization of the DCPLL, the proposed algorithm is PVT tolerant and can be applied to the entire frequency range of the DCPLL.

B. The Mathematical Analysis of the Algorithm

In actual condition, the predicted code error will be introduced into the loop due to the following reasons. First, the resolution of the TDC is limited and quantization noises are introduced into the loop. Second, the DCO gain is hard to keep invariant during the whole frequency range of the DCO.

Considering the quantization noise of the TDC δ, Eq.(9) is rewritten as Eq.(10).

$$\begin{aligned} C_t &= C_1 - \frac{e_1 + \delta_1}{e_1 - e_2 + \delta_1 - \delta_2}(C_1 - C_2) \\ &= C_1 - \frac{e_1}{e_1 - e_2}(C_1 - C_2) - \Delta(C_1 - C_2) \end{aligned} \tag{10}$$

where $\Delta = \frac{e_1\delta_2 - e_2\delta_1}{(e_1 - e_2)(e_1 - e_2 + \delta_1 - \delta_2)}$, the subscript n of δ indicates that δ_n is for the code C_n.

To eliminate the effect of the quantization noise δ, the value of $\Delta(C_1 - C_2)$ should be smaller than one. Combining Eqs.(1)-(3), $\Delta(C_1 - C_2)$ is rewritten as follows.

$$|\Delta(C_1-C_2)| = \left|\frac{\Delta_{TDC}(\delta_2\varphi_1-\delta_1\varphi_2)(C_1-C_2)}{(\varphi_1-\varphi_2)[\varphi_1-\varphi_2+(\delta_1-\delta_2)\Delta_{TDC}]}\right| \approx \left|\frac{\Delta_{TDC}(C_1-C_2)}{\varphi_1-\varphi_2}\right| = \frac{\Delta_{TDC}}{MK_o} \quad (11)$$

Obviously, the value of Δ_{TDC} should be smaller than the half of MK_o, which can be easily achieved for most TDCs.

Besides, the non-uniform DCO gain will also result in the code mismatch. If the real DCO gain K_{or} is defined as $K_{or} = K_o(1+\gamma)$, Eq.(8) is rewritten as Eq.(12).

$$C_t = C_1 + (1+\gamma_t)\frac{e_1\Delta_{TDC}}{MK_{or}} \quad (12)$$

where γ_t is the non-uniform DCO gain error for the target code C_t.

Based on Eq.(12), if the code C_n calculated by Eq.(9) is assigned to control the DCO, the corresponding digital phase error e_n due to the non-uniform DCO gain can be given by Eq.(13). It is seen that e_n is proportional to e_1 and γ_t.

$$e_n \approx \frac{\gamma_t e_1\Delta_{TDC}}{MK_{or}} MK_{or}\frac{1}{\Delta_{TDC}} = e_1\gamma_t \quad (13)$$

C. The Implemented Algorithm

Seen from Eq.(11), the prediction code error resulted by the TDC can be ignored.

Thus, to decrease the effect of the non-uniform DCO gain on prediction, several methods should be implemented during the PLL design. First, considering the influence of PVT variation on K_o, K_o is calculated in the actual condition for every initialization of the DCPLL. Second, the middle code is chosen as the first code C_1, and the choice of the second code C_2 depends on e_1. So the value of the predicted code C_n is always between C_1 and C_2 which is helpful to reduce γ_n. Third, based on the analysis result of Eq.(13), the implemented frequency search algorithm should be slightly modified which is given as Eq.(14). If the sensed phase error exceeds the boundary which is defined as 5% of t_{REF} in this design, the DCPLL will recalculate the next code C_{n+1} by Eq.(14).

$$C_{n+1} = \begin{cases} C_1 - \dfrac{e_1}{e_1-e_2}(C_1-C_2) & n=2 \\ C_n - \dfrac{\alpha_n e_n}{e_n-e_x}(C_n-C_x) & n>2 \end{cases} \quad (14)$$

In Eq.(14), $e_x = \min\{abs(e_1), abs(e_2)\}$, C_x is the corresponding code, and α_n is the scale factor. The definition of α_n is given as follows.

$$\alpha_n = \begin{cases} 1 & n=2 \\ 0.5\alpha_{n-1} & n>2 \end{cases} \quad (15)$$

Based on Eq.(14), Eq.(13) is rewritten as Eq.(16). Seen from Eq.(16), it is obvious that even for an inappropriate DCO design, the frequency search algorithm can also find the target code after several reference clock cycles.

$$e_{n+1} = \begin{cases} e_1\gamma_t & n=2 \\ \alpha_n e_n\gamma_t & n>2 \end{cases} \quad (16)$$

III. Structure of the Proposed DCPLL

To implement the proposed algorithm presented as Eq.(14) in a DCPLL, four preconditions must be met.

1) Seen from Eqs.(5) and (8), two predetermined codes C_1 and C_2 must be assigned to control the DCO before prediction.

2) As is shown in Eqs.(2) and (3), the previous phase error should be zero when the DCO are tuned by C_1 or C_2.

3) To follow the definition in Eq.(1), the DCO gain should be a constant as much as possible.

4) Finally, due to several reasons such as the TDC quantization noise and non-uniform DCO gain, the prediction error isn't avoidable. The DCPLL should have the ability to eliminate the prediction error.

Considering the above requirements, the structure of the proposed DCPLL is developed which is drawn in Fig. 2. It consists of a PFD, a TDC, a first-order DLF, a digital controller (DC), a loop MUX(LMUX), a DCO, a DIV, and a zero phase restart controller (ZPRC).

The DC is utilized to implement the proposed algorithm described as Eq.(14), and control the operation of the DCPLL by the signal En_fre and the code C. The DC divides the operation of the DCPLL into four steps. The aims of the first two steps are assigning the two predetermined codes C_1 and C_2 to control the DCO successively based on Eqs.(2) and (3). The work of the third step is predicting the target code based on Eq.(14). Finally, the job of the fourth step is eliminating the prediction error with the help of the DLF.

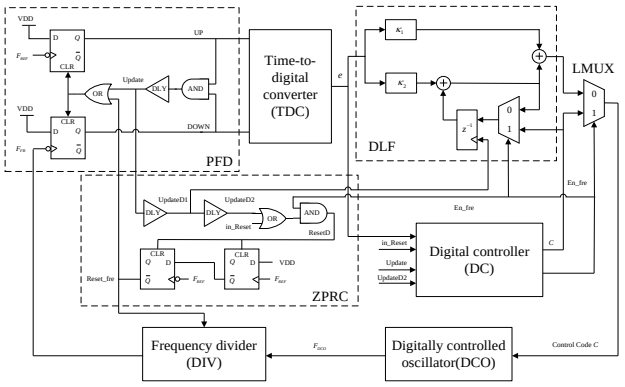


Figure 2. The proposed structure of the DCPLL

To make sure the previous phase error is zero when C_1 or C_2 is assigned to control the DCO, the module ZPRC drawn in Fig. 2 is developed. The principle of the ZPRC

is simple and direct. The ZPRC generates the reset signal Reset_fre and sends it to the DIV and the feedback path of the PFD. The reason relies on the operations of the PFD and the DIV. The PFD generates the phase error by comparing the falling edges of F_{REF} and F_{FB}. The clock F_{REF} is sent by the outside and can not be changed. The clock F_{FB} is generated by the DIV. The DIV generates the feeback clock F_{FB} by counting F_{DCO} when Reset_fre is low. When the counter value is the half of M, F_{FB} rises. If the counter value is M, F_{FB} falls and the counter value is reset to zero. Therefore, if F_{REF} falls at $T = T_1$, the next falling time T_2 for F_{REF} is $T_1 + t_{REF}$. Meanwhile, if the DIV is reset and restart to work at $T = T_1$, the counter of the DIV counts from zero, and the next falling time T_3 for F_{FB} is $T_1 + Mt_1$. Then the phase error is $Mt_1 - t_{REF}$ which is shown in Eq.(2). Thereby, although F_{FB} doesn't falls at $T = T_1$, it also looks like that the previous error between F_{REF} and F_{FB} is zero. Besides, the PFD also has the function of detecting frequency error. So the PFD is reset by Reset_fre to make sure Eq.(2) is tenable. Finally, considering the DCO needs a stable time when the code is changed, Reset_fre turns to low from high only after one rising and one falling edge of the reference clock F_{REF} which can be deduced by the ZPRC shown in Fig. 2.

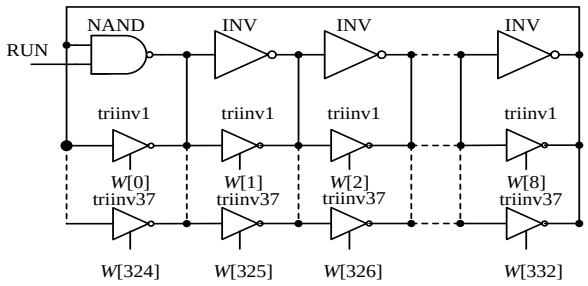


Figure 3. The structure of the DCO

The DCO gain of several DCO structures can be designed as a constant [9]-[11]. In this paper, the DCO structure [11] is adopted for convenience which is drawn in Fig. 3. It is shown that the DCO has nine stages. To change the frequency of the DCO, each stage of DCO is in parallel with a set of 37 tri-state inverters.

Due to the discussion in Eq.(11), it is seen that most TDC structures are suitable for the proposed DCPLL structure. The TDC structure proposed is adopted in this paper for convenience [12].

The DLF is a common first-order digital loop filter except a MUX is inserted in the integral path. The extra MUX is used to insert the predicted code by Eq.(14) in the DLF when the DCPLL enters the last step.

IV. Operation of the Proposed DCPLL

The operation of the DCPLL is divided into four steps which are named as step0~3. The details are described in the following.

A. Step 0

Seen from Fig. 4, when the system reset signal in_Reset is high, En_fre is reset to high and remains high until the DCPLL enters step3, the status of the DCPLL is reset to step0, the PFD and the DIV stop working due to the high level of Reset_fre, and the DC assigns the middle code as the first code C_1 to control the DCO. When in_Reset is low, Reset_fre turns low after one rising and one falling edge of F_{REF} at $T = T_1$, the DIV begins to divide F_{DCO} and sends F_{FB} to the PFD. Then the falling time T_2 for F_{REF} is $T_1 + t_{REF}$. The falling time T_3 for F_{FB} is $T_1 + Mt_1$. After the phase error φ_1 shown in Eq.(2) is sensed, the DCPLL enters step1 when Update rises.

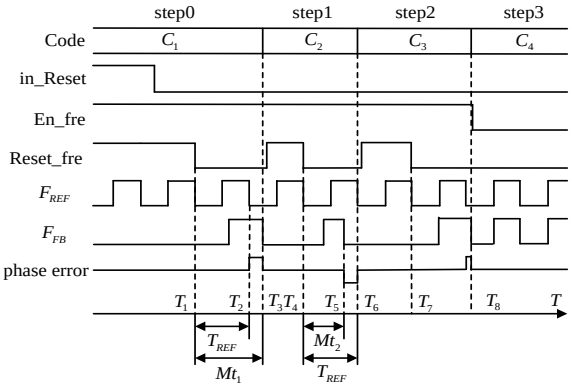


Figure 4. The waveforms of the related signals

B. Step 1

When the DCPLL enters step 1, ResetD turns high which is resulted by the positive pulse of Update shown in Fig. 2. Reset_fre turns to low until $T = T_4$.

Meanwhile, if $e_1 > 0$, it indicates that t_{REF} is smaller than t_{FB}. Then based on the definition in Eq.(1), the DC generates the maximum code as C_2 to tune the DCO. If $e_1 < 0$, the DC assigns the minimum code as C_2. Then F_{FB} falls at $T = T_5$ and F_{REF} falls at $T = T_6$. After φ_2 is sensed, the DCPLL turns to step 2.

C. Step 2

In step 2, The DC calculates the code based on Eq.(14) and sends it to the DCO as C_3. If the sensed phase error exceeds the boundary which is defined as 5% of t_{REF} in this design, the DCPLL will stay at the step2 and recalculate the next code C_{n+1} based on Eq.(14). Or else, the DCPLL enter the step3 at $T = T_8$.

D. Step 3

The operation of the DCPLL in step 3 is very similar to the operation of the traditional PLL. The PFD senses the

phase error φ between F_{FB} and F_{REF} . The phase error φ is converted to e by the TDC and sent to the DLF. Then the output of the DLF C tunes the DCO. Finally, the DCO clock is divided by the DIV and is fed back to the PFD.

Based on the operation of the traditional PLL, the phase error will be increased if the prediction error exists. If the phase error due to the prediction error can be sensed by the TDC, the prediction error will be eliminated by the DCPLL in step 3.

V. Simulation Results

To verify the proposed algorithm, a DCPLL is designed. In the DCPLL, The DCO and TDC are implemented in SMIC 0.13μm 1P8M technology. The period and γ of the DCO with the code are drawn in Fig. 5, where γ is defined by Eq.(12). Seen from Fig. 5, the frequency range is 183MHz~621MHz. The average DCO gain is 11.5ps and the range of γ is from -0.28 to 0.52. The average resolution of the TDC is 1ns. To reduce the verification time, the simulations are made by the HSIM-VCS DKI co-simulation method which is widely used in the simulation of the mixed-signal circuits. In the co-simulation method, The DCO and TDC are simulated by spice simulator HSIM. The other blocks are described with synthesizable codes in Verilog and simulated by verilog simulator VCS.

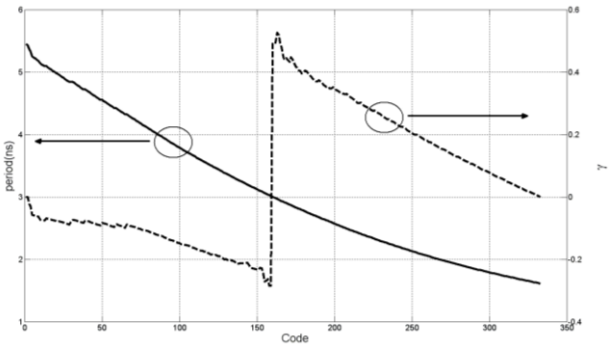


Figure 5. The period and γ of the DCO

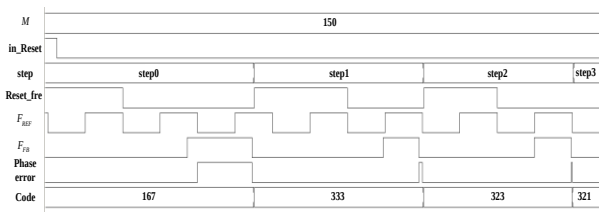


Figure 6. An example of the frequency acquisition @600MHz

An example of the frequency acquisition when the DCPLL locks to 600MHz is also shown in Fig. 6. In the Fig.6, the frequency of the reference clock is 4 MHz, the frequency divider ratio is 150. In the simulations, the locking time is defined as from the time when Reset_fre falls in step0 to the time when the code is predicted where the corresponding phase error is smaller than 5% of t_{REF} in step2. From the figure, it is seen that with two sensed phase errors, the DCPLL predicts the desired code is 323. Then the DCPLL enters step3 to eliminate the remaining frequency error. The maximum phase error is 2.88ns which is less than 2% of T_{REF} .

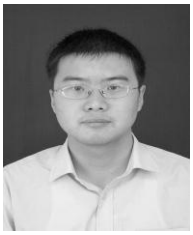
Finally, the frequency acquisitions for the whole frequency range are simulated by HSIM-VCS DKI co-simulation method, and the maximum locking time is $4t_{REF}$ for the whole frequency range. Compared with the presented methods [3, 5-7], the proposed DCPLL has the shortest locking time.

Acknowledgment

This Project is supported by the National Natural Science Foundation of China (No.61106029).

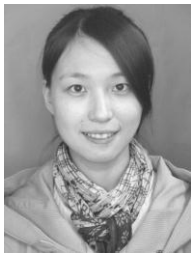
References

[1] S. Dasgupta and A. Yakovlev, "Comparative analysis of GALS clocking schemes," *IET Computers & Digital Techniques*, vol. 1, pp. 59-69, Feb 2007.
[2] L. Lu, J.V. McCanny, and S. Sezer, "Reconfigurable system-on-a-chip motion estimation architecture for multi-standard video coding," *IET Computers & Digital Techniques*, vol. 4, pp. 349-364, May 2010.
[3] J. Choi and H. Cha, "System-level power management for system-on-a-chip -based mobile devices," *IET Computers & Digital Techniques*, vol. 4, pp. 400-409, May 2010.
[4] E. Hadi, B. Emily, St. A. Renée, S. Karthikeyan, B. Doug., "Dark silicon and the end of multicore scaling," *IEEE Micro*., vol. 32, pp. 365-376, Mar. 2012.
[5] S. L. Huang and Z. H. Wang, "An adaptive PLL architecture to achieve fast settling time," *Journal of Electronics & Information Technology*, vol. 29, pp. 1492-1495, June 2007.
[6] P. H. Hsieh, M. Jay, and CK. K. Yang, "A phase-selecting digital phase-locked loop with bandwidth tracking in 65-nm CMOS technology," *IEEE J. of Solid-State Circuits*, vol. 45, pp. 781-792, Apr. 2010.
[7] I.-C. Hwang, S. H. Song, and S. W. Kim, "A digitally controlled phase-locked loop with a digital phase-frequency detector for fast acquisition," *IEEE J. Solid-State Circuits*, vol. 36, pp. 1574-1581, Oct. 2001
[8] T. Watanabe and S. Yamauchi, "An all-digital PLL for frequency multiplication by 4 to 1022 with seven-cycle lock time," *IEEE J. Solid-State Circuits*, vol. 38, pp. 198-204, Feb. 2003.
[9] P. L. Chen, C.C. Chung, J. N. Yang, *et al.* "A clock generator with cascaded dynamic frequency counting loops for wide multiplication range applications," *IEEE J. Solid-State Circuits*, vol. 41, pp. 1275-1284, June 2006.
[10] S. D. Chung, C. C. Lee, and C. Yi, "An ultra-low-power and portable digitally controlled oscillator for SoC applications," *IEEE Trans. Circuits Syst. Express Briefs*, vol. 54, pp. 954-958, Nov. 2007.
[11] X. Y. Deng, J. Yang, and J. H. Wu, "Contributions to the analysis of deterministic noise on ADPLL jitter performance," *Analog Integrated Circuits and Signal Processing*, vol. 67, pp. 331-338, Mar. 2011.
[12] O. Thomas and N. Peter. "A digitally controlled PLL for SoC applications," *IEEE J. Solid-State Circuits*, vol. 39, pp. 751-759, May 2004.



Xin Chen received the B.S. degree in electrical engineering from Southeast University, China in 2005, and Ph.D. degrees in microelectronics and solid electronics from Southeast University, China in 2010. In 2010, he joined the faculty of college of electronic and information engineering, Nanjing University of Aeronautics and Astronautics. His research interests mainly include mixed-

signal circuits design, and digitally controlled phase-locked loop.



Wanqiao Zhang received the B.S. degree in information engineering from JinCheng College, Nanjing University of Aeronautics and Astronautics, China in 2012. She is currently pursuing the M.S. degree in microelectronics and solid electronics in college of electronic and information engineering, Nanjing University of Aeronautics and Astronautics. Her research interests mainly include digital circuits design, and SoC design.