

An Improved Solution for the Fast-Locking All Digital SAR DLL

Shibin Lu^{*1,2}, Tailong Xu², Junning Chen²

¹School of Electronics and Information Engineering, Hefei Normal University, Hefei, 230601, China

²School of Electronics and Information Engineering, Anhui University, 230601 Hefei, China

*Corresponding author, e-mail: shibinlu@yahoo.com.cn

Abstract

All digital successive approximation register-controlled delay-lock loops (SAR DLL) are widely used in system-on-chip to solve the clock generation and skew problems for their fast-locking characteristic. However, the conventional SAR DLL has the dead lock problem. So many improved solutions are proposed to solve the dead lock problem. Based on the resettable delay line, improved SAR controller and phase comparator are depicted. By these, a harmonic-free and fast-locking all digital SAR DLL without dead lock is implemented. Post-layout transistor-level simulation results show that the lock-in and relock-in time are both within N cycles of input clock for N -bit SAR controller. The dead lock problem of the conventional SAR DLL is eliminated very well.

Keywords: phase comparator, successive approximation register, SAR DLL, fast-locking, dead-lock

Copyright © 2013 Universitas Ahmad Dahlan. All rights reserved.

1. Introduction

Synchronization between the systems and sub-systems is becoming more challenging as the operating frequency of VLSI system is increasing. Delay-locked loops (DLLs) are widely used as de-skew buffers and clock generators in microprocessors, digital signal processors (DSPs), multi-core system-on-a-chips (SoCs), DRAM interfaces and application-specified integrated circuits (ASICs)[1-6]. DLLs can be roughly categorized into kinds, analog and digital DLLs. Although analog DLLs have better skew and jitter performance, they are process-sensitive and difficult to mitigate over different technologies. However, digital DLLs are well suit for mitigation and more easily scaled down in advanced technologies [7]. The scaling-down CMOS technologies can not only enable digital DLLs to operate at a lower supply voltage, but also offer the digital delay elements with fine delay resolution. Low supply voltage makes low power consumption achievable and the better delay resolution improves the digital DLLs' jitter and skew performance. Additionally, the digital DLLs have a significant advantage of short lock-in time. For these reasons, digital DLLs have become a better choice as the CMOS technologies advance continuously [3].

At present, the successive approximation register-controlled (SAR) DLL is much more popular because of short lock-in time and low hardware cost, among different types of digital DLLs [8]. The SAR DLL adopts binary search algorithm (BSA) to achieve fast lock, however, it has issues such as harmonic lock in the wide-range applications [9], the lock-in time is longer than the theoretical value and the dead lock i.e. the open-loop characteristic after locked [10]. The dead lock results in the conventional SAR DLL can not track the process, voltage, temperature, loading (PVTL) and operating frequency variations after the first lock-in. To meet various circuit specifications, the operating frequency range should be as wide as possible, and the locking time should be as small as possible especially in low-power SoCs with dynamic voltage and frequency scaling (DVFS) [11]. So many improved solutions are proposed to solve the problems of the conventional SAR DLL. Variable SAR algorithm [9] is adopted to avoid the harmonic lock and the SAR controller transforms into a counter after locked to track the PVTL variations. But the false SAR steps will waste many cycles, especially for the low-frequency input. A reversible SAR DLL is presented in [12] with an initial delay-range-search unit. This unit can overcome the harmonic lock and reduce the lock-in time to less than 42 cycles, but the lock-in time is also longer than the theoretical value [13], presented an improved scheme to shorten

the lock-time to theoretical value and eliminate the harmonic-locking using novel resettable delay units (RDUs). However, it did not resolve the dead-lock problem and its modified phase comparator cannot provide the lock signal that indicates the SAR DLL is in locked status. The lock signal of phase comparator is very important to resolve the dead-lock problem, because it can reflect whether the lock-out of DLLs occurs or not and trigger the DLLs to relock-in again.

So all previous improved solutions can not solve the three issues of the conventional SAR DLL very well. Due to the fastest lock and wide rang characteristics of the [13], based on it, improved SAR controller and phase comparator (PC) are proposed in this paper to solve the three issues of the conventional SAR DLL meanwhile.

2. The Dead-lock Problem of the SAR Controller and Improved Solutions

The block diagram of the conventional 6-bit SAR DLL is depicted in Figure 1(a) [8]. It consists of a phase comparator (PC), a digital-controlled binary weighted delay line, a divide-by-four frequency divider, an initialization circuit (IC) and a 6-bit SAR controller to provide the control word for the delay line. The PC gives phase information between input clock and output clock. If it is locked, the lock detect signal LD will become “1”. If it is out of lock, LD will be “0”. The signal Comp tells the phase of output clock lags or leads the input clock. The control word $b[5:0]$ of the delay line comes from the 6-bit SAR controller.

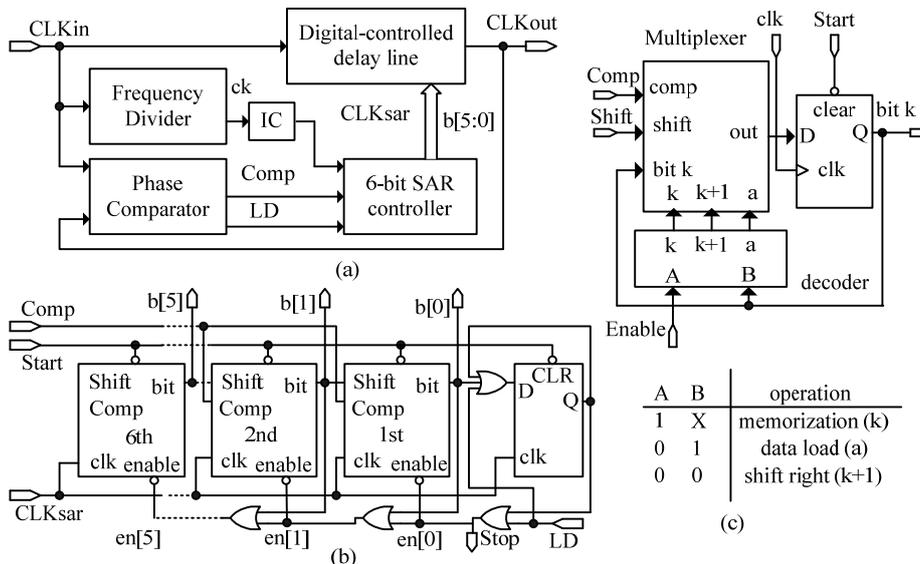


Figure 1. (a) The Diagram of the Conventional 6-bit SAR DLL, (b) The Conventional 6-bit SAR Circuit, and (c) The Internal Structure of the kth flip-flop.

Based on the output of the PC, the value of each bit of the output $b[5:0]$ of the SAR controller in the DLL is determined in a sequential and irreversible manner [12]. The conventional 6-bit SAR controller and internal structure of the kth flip-flop with truth table are illustrated in Figure 1(b) and (c) respectively. Whenever the flip-flop's are triggered, the kth flip-flop will have three different data inputs coming from: 1) the output of the (k+1)th flip-flop (shift right); 2) the output of the PC, Comp (data load); 3) the outputs of the kth flip-flop itself (memorization).

In the beginning, the signal Start sets the 6th flip-flop and clears the others. Therefore, the control word $b[5:0]$ equals to be 100 000 initially. Now the delay time is half of the whole delay line. When the SAR process begins, the signal Comp will decide the “1” of $b[5]$ to remain or change to “0”. In the delay time perspective of view, this “1” will be reserved if the delay time is not enough and will not if the delay time is sufficient. Likewise, $b[4]$ will be “1” (shifted from the

6th flip-flop). Then the signal Comp will examine this “1” of b[4] in the same way as b[5]. This SAR process will go on to b[0], until the SARDLL is locked (the signal LD is up).

However, the conventional SAR DLL has a dead-lock problem. When the signal LD changes to “1”, the signal Stop will be activated, through the OR gate feeding back to the flip-flop’s, it forces the SAR controller to be in memorization mode. Due to the output of the additional DFF in the right end of Figure 1(b) is fed back to its input through an OR gate, when the operation is stopped, the logic “1” output of this DFF will prohibit any possibility of re-starting the operation even the signal LD is pulled down again when the PC detects a large phase error for the operation condition variations. This denotes that relock-in process will never perform again even the PVTL and frequency variations are unacceptable. Then the SARDLL is useless.

To overcome the dead-lock drawback, [14] loads the output of the SAR controller to an up/down counter. This problem is free at the cost of an extra counter, and the SAR controller is out of use after the first lock-in process. A variable successive approximation register-controlled (VSAR) algorithm is proposed in [9]. When the binary search is over, this VSAR controller can be transformed into a counter to avoid the open-loop nature without any extra counter. But the relock-in processes are counter-controlled, not SAR-controlled, i.e. the fast-locking characteristic of SAR is not kept in the relock-in processes. As a result, in the worst case, both of them will take $2N-1$ steps to be in the locked state again. So an improved SAR controller whose relock-in process is also SAR-controlled is proposed in this paper.

3. The Improved SAR Controller

From the description of section 2, the feedback of the DFF in the right end of Figure 1(b) from output to input is indeed the source of the dead-lock problem. So, in the improved SAR controller depicted in the Figure 2(a), a restarting module is added to break the feedback in the conventional SAR controller. During the initialization state, the signals Start and LD are logic “0”, so the outputs of DFF1 and DFF2, and the signals Start_in and Stop are all also logic “0”. The outputs b[5:0] of SAR controller are forced to be “100 000”. When the signal Start is changed to be logic “1”, the first lock-in process begins and it is the same as the operation of the conventional SAR controller. When the output of PC, LD transforms to be logic “1”, i.e. the SAR DLL is locked, the SAR controller goes into memorization mode.

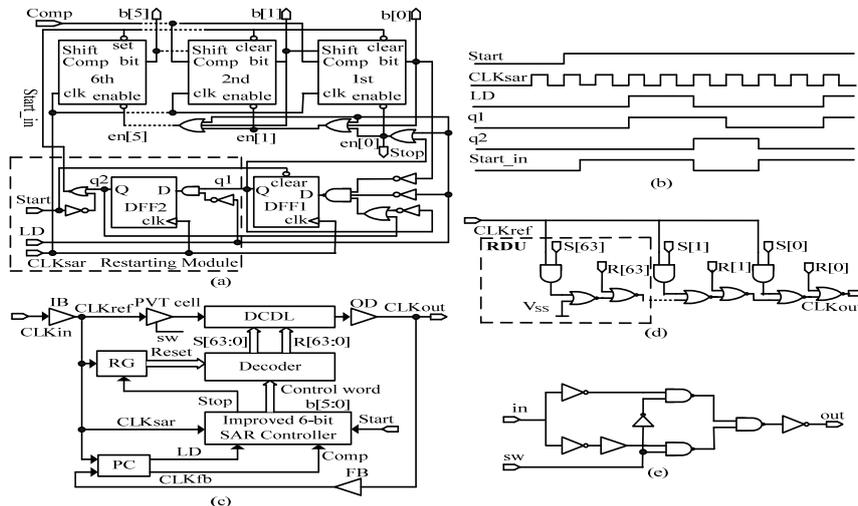


Figure 2. (a) The Schematic of the Improved SAR controller, (b) The Operating Timing Diagram of the Improved SAR Controller, (c) The Diagram of the Fast-locking and Relocking SAR DLL using the Improved SAR Controller, (d) The Schematic of the DCDL Proposed in [6] and (e) The Schematic of the PVT cell.

According to the schematic of Figure 2(a), different from the conventional SAR controller, after the SARDLL first lock-in, if the PVTL and operating frequency variations result in

lock-out, the signal LD will transit from “1” to “0”. And then, when the rising edge of CLKsar comes, the output of the D flip-flop DFF2, i.e. the signal q2 changes to be logic “1”. On the next rising edge of CLKsar, the output of the D flip-flop DFF1, i.e. the signal q1 changes to be logic “0”. On the third rising edge of CLKsar, the signal q2 changes to be logic “0” and the signal q1 keeps logic “0”. So the pulse width of the signal q2 equals to two cycles of the CLKsar, and in this period, the signal Start_in is forced to be in the logic “0”, the SAR controller exits from the memorization mode, the outputs of SAR controller b[5:0] are initialized to be “100 000” again. And then the relock-in process begins likewise the first lock-in process. The timing diagram is depicted in Figure 2(b) to illustrate how the improved SAR controller works.

Compared to [9, 14], the improved SAR controller in this paper reserves the SAR characteristic after the first lock-in, meanwhile, the first lock-in and relock-in process both only need N steps in the worst case.

4. The Improved Fast-locking SAR DLL Scheme

Theoretically, the number of bits of the control SAR determines the number of clock periods needed for the SAR DLL to achieve lock. However, the conventional SAR DLL, after receiving a decision signal from the PC, the DLL will need time to respond before receiving the next decision result. The needed response time is mainly determined by the time that the input clock propagates through the delay line. So if the response time is reduced within an input clock period, the lock time will equal the theoretical value. Based on the ideal, [13] presents a resettable delay line to shorten the lock time to the theoretical value.

The resettable delay line is depicted in Figure 3(a), which consists of RDUs. For every RDU, when the signal Rcode is logic “1”, the output is logic “0”, and when the signal Scode is logic “1”, the input clock feeds into delay line from the RDU. Before the input clock feeding into delay line, the signal Rcode forces the output of the delay line to be logic “0”, and then the input clock goes through the delay line. After a period of the input clock, if the delay time of the delay line is too long, the output of the delay line keeps logic “0”, and on the contrary, the output of the delay line changes to be logic “1”. So the response time of the DLL can be thought to shorten within a period of the input clock.

However, the output of delay line may be not a regular pulse signal, described in Figure 3(c). The conventional D flip-flop phase comparator cannot deal with such case. Hence a modified phase comparator is presented in [13], which depicted in Figure 3(b). The FF2 is a D flip-flop phase comparator. The FF1 extends the pulse width of the output clock so that it can be sampled by FF2 as shown in Figure 3(b). The input clock CLKin is converted into the pulse signal by a pulse generator. Then the pulse signal is delayed by the matched delay cell 1 and acts the reset signal of FF1. The matched delay cell 2 is used to approximate the clk-Q delay of the FF1, in order to reduce the error of the PC operation. But such modified phase comparator can only provide the phase relation between the input clock CLKin and output clock CLKout and cannot provide the lock signal indicates the DLL is in lock status. So its application is limited, for example, some applications need the lock signal to realize more functions. So in this paper, an improved PC that can provide lock signal is presented.

5. The Improved Fast-locking SAR DLL Scheme

The schematic as well as the operation principle of the PC is shown in Figure 4. Two D-flip-flops, FF1 and FF2 are used to sample the output clock of the delay line. The clock inputs of the two sampling DFF's have a timing difference equal to the unit delay of the delay line, thus forming a lock detecting window. When the output clock is located outside the window, the SARDLL is said unlock and LD is always “0” in this situation. If the output clock leads the input one, both outputs of the DFF's are “1”, and Comp is “1”. If the output clock lags the input clock, both outputs of the DFF's are “0”, and Comp turns to be “0”. Once the output clock enters the window, the outputs of the DFF's will be “0” and “1”, respectively, and LD will become “1” immediately, i.e. the SARDLL is locked. Such functions are described in the truth table of Figure 4(c). When the LD transits from “1” to “0”, the DLL goes into lock-out from lock-in status, the SAR must restart to synchronize the output and input clocks again.

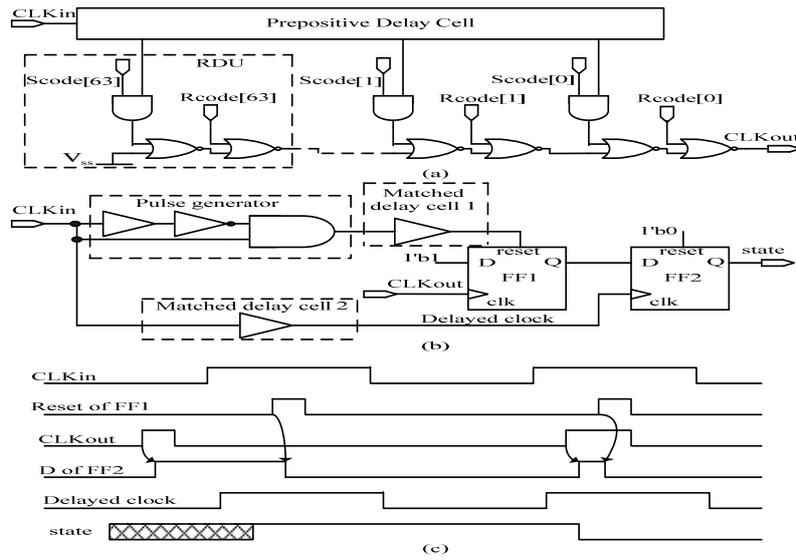


Figure 3. (a) Resettable Delay Line, (b) Structure of Phase Comparator and (c) Timing Diagram of the PC of the [13]

As explained previously, the output clock of the delay line, CLKout is an irregular pulse signal whose pulse width may be narrow, because of adopting the RDUs. As Figure 4(b) depicts, when the output clock leads the input clock, if the width of the output clock CLKout is too narrow, the sampling DFF's will wrongly sample two "0", it should sample two "1" according to the correct operation principle. And when the output clock enters the lock detecting window, if the width of the output clock CLKout is too narrow, the FF2 will sample "0" not "1", such case is wrong. So a D-flip-flop, FF3 is inserted between the CLKout and the data inputs of the two sampling DFF's to extend the pulse width of the CLKout, so that it can be sampled by FF1 and FF2, as depicted in Figure 4(a). The CLKin is converted into the pulse signal by a pulse generator, and then the pulse signal is delayed by the matched delay cell 1 and acts the reset signal of FF3. The matched delay cell 2 is used to approximate the clk-Q delay of the FF3, in order to reduce the error of the PC operation. The operation principle of the extending circuit and the whole PC are described in Figure 4(b).

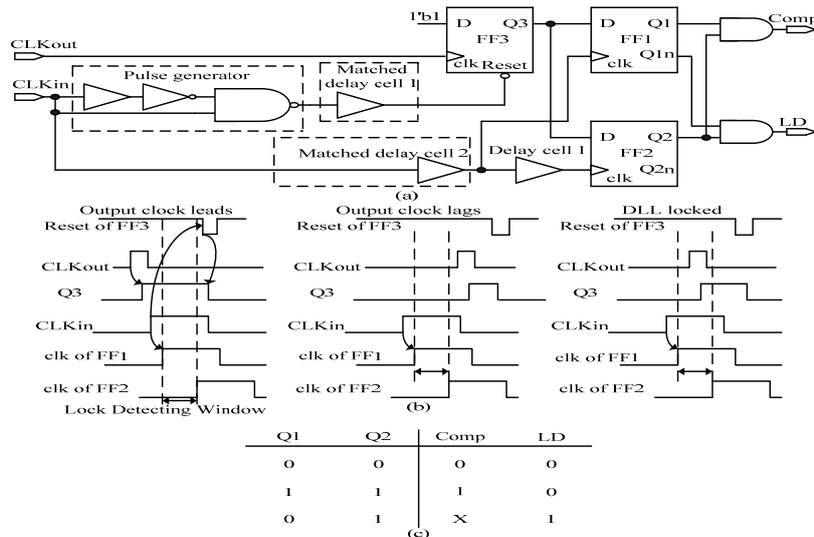


Figure 4. (a) Schematic of the PC, (b) Operation Principle, (c) Function truth table.

Since the PC is realized with two DFF's to detect the rising edge of the output clock signal, the metastable problem i.e. dead zone is of great concern because the clock and the data input can switch exactly at the same time in this case. When the metastable problem occurs, the SAR DLL can still be locked although the decision result of the PC is ambiguous. However, the static phase error between input and output clocks would be larger than one unit delay.

6. Simulation

In order to validate the improved SAR controller and PC proposed in this paper, as described in Figure 2(c), an all digital SAR DLL using the improved SAR controller and PC is designed, which comprises input buffer (IB), PVT cell, digital-controlled delay line (DCDL), output driver (OD), reset signal generator (RG), decoder, clock driver (CD), improved SAR controller, phase comparator (PC) and feedback buffer (FB). As shown in the Figure 2(d), the scheme of DCDL proposed in [13] is adopted for its fast-locking and harmonic-free characteristic, which consists of resettable delay units (RDUs). The SAR DLL using the scheme of DCDL can lock-in within N cycles of input clock [13]. The signals CLK_{in} and CLK_{out} denote the input and output clocks respectively.

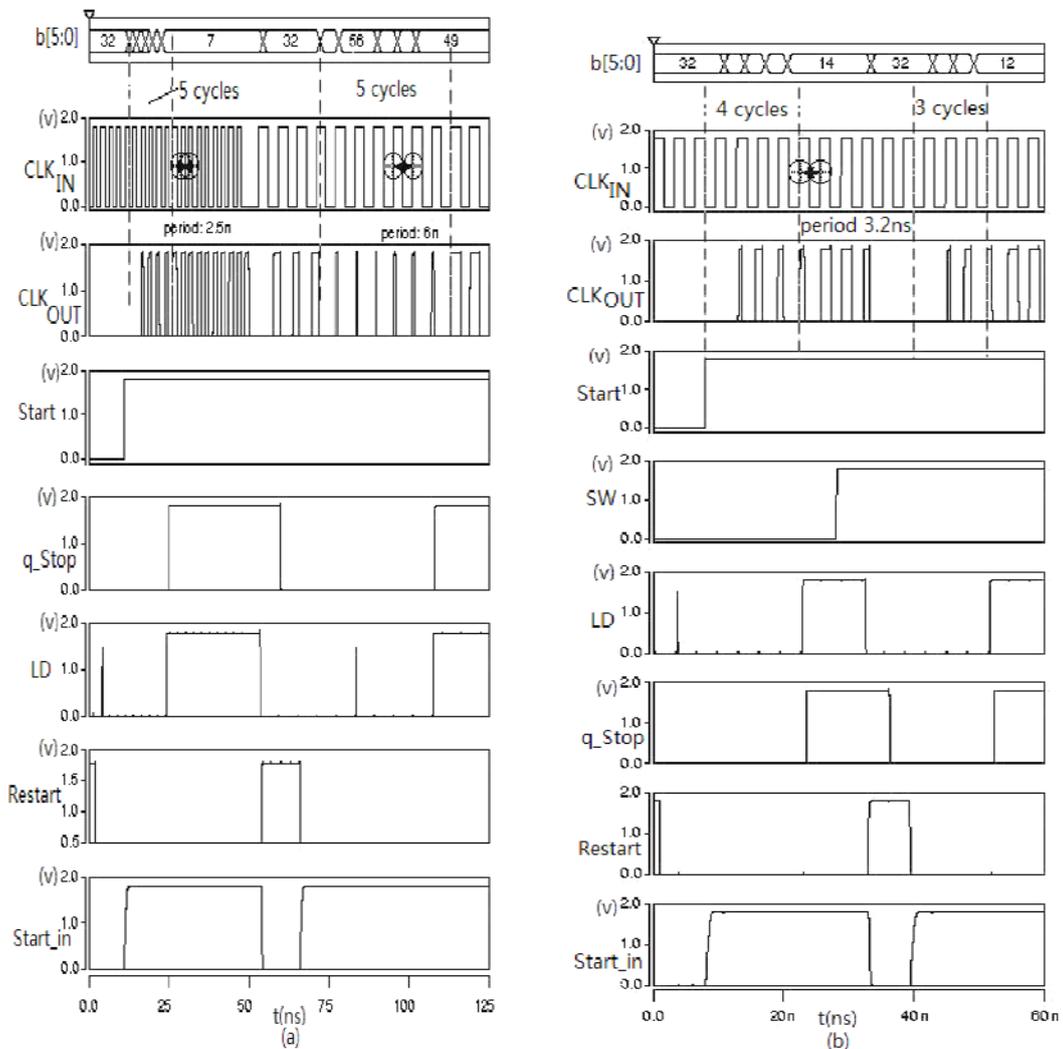


Figure 5. The Relock-in Process (a) When the Period of the Input Clock is Transited from 2.5ns to 6ns and (b) When the PVTL Variations Result in Lock-out.

In order to simulate the impact of PVTL variations, a PVT cell is added into the clock path, because of the impact of PVTL variations can be equivalently represented as a variable delay element to be added in the clock path [14]. The circuit of the PVT cell is shown in Figure 2(e), during the normal operation, the control signal *sw* for the cell is set to “0”, and the PVT cell provides an intrinsic delay to the clock path. This extra delay will be considered automatically in the locking process and will not affect the function of the SAR DLL. After lock-in, *sw* can be pulled high intentionally at a particular time. Now, the PVT cell will provide a much larger delay.

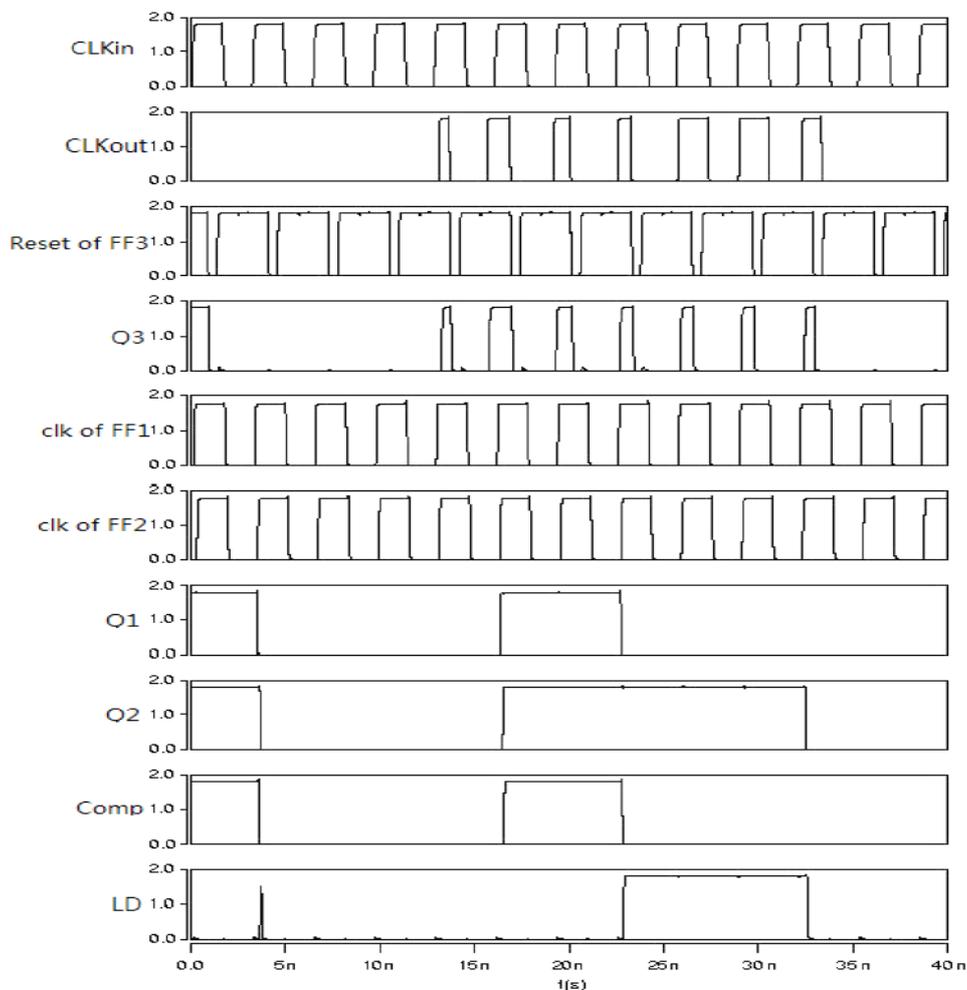


Figure 6. The Transistor-level Post-layout Simulation Results of the Improved PC

The all digital SAR DLL with the improved 6-bit SAR controller and PC is implemented in 0.18 μ m CMOS. HSPICE is chosen as the simulator. The post-layout transistor level simulation results, which are based on SPICE models, TT corner at 1.8V/25 $^{\circ}$ C and the load is 0.005pF, are in Figure 3. The signal *sw* of PVTL cell is set to logic “0”, Figure 5(a) shows that the SAR DLL can track the operation frequency variation. In the Figure 5(b), when the signal *sw* is set “0”, the DLL can get locked, and the signal *LD* goes high. And then, the signal *sw* is set “1” to turn on the PVT cell intentionally, as a result, the DLL becomes unlocked, and the signal *LD* is pulled low. Now the restarting module is activated, the control word *b*[5:0] is fine tuned continuously, finally, the DLL gets locked again, and the signal *LD* goes high again. Both the first lock-in and the relock-in time are within 6 cycles of input clock, i.e. the dead-lock issue of conventional SARDLL is eliminated and the advantages of the SAR algorithm are kept after the first lock-in. The values of *b*[5:0] in the Figure 5 are all decimal.

As shown in Figure 6, The improved PC not only correctly gives the signal Comp that reflects the phase relations between input clock and output clock, but also gives the signal LD that indicates whether the DLL is locked or not. And when the PVTL variations result in DLL lock out, the LD can turn to be "0" to show the DLL is not locked. Using such case can realize more functions and applications of all digital fast-locking SARDLLs.

7. Conclusion

Based on the resettable deadly line, improved SAR controller and PC solution are presented for the the SAR DLL. By these, the dead lock, harmonic lock, and the long lock time issues of the conventional SAR DLL are solved very well and simultaneously. The transistor-level post-layout SPICE simulation results show that the proposed solutions are valide.

Acknowledgments

This work was financially supported by Natural Science Foundation of Anhui High Education (KJ2012B143), Youth research fund of Anhui University (KJQN1011), Anhui provincial excellent youth research fund (2012SQRL013ZD) and NSFC (61076086).

References

- [1] S Moorthi, D Meganathan, N Krishna, et al. A Novel 14~170 MHz All Digital Delay Locked Loop with Ultra Fast Locking for SoC Applications. *IEEE Recent Advances in Intelligent Computational Systems*. 2011: 076-080.
- [2] DH Jung, K Ryu JH. Park, et al. A Low-Power and Small-Area All-Digital Delay-Locked Loop with Closed-Loop Duty-Cycle Correction. European Solid-State Circuits Conference. France. 2012: 1-4.
- [3] S Chen, H Li, K Jia. A Fast-Lock-in Wide-Range Harmonic-Free All-Digital DLL with a Complementary Delay Line. *IEEE International Symposium on Circuits and Systems*. Korea. 2012: 1803-1806.
- [4] JD Garside, SB Furber, S Temple, et al. An Asynchronous Fully Digital Delay Locked Loop for DDR SDRAM Data Recovery. *IEEE International Symposium on Asynchronous Circuits and Systems*. 2012: 49-56.
- [5] S Maheshwari, M Srinivasarao, HS Raghav, et al. Harmonic Free Delay Locked Loop having Low Jitter with Wide-range Operations. International Conference on Advances in Engineering, Science and Management. 2012: 577-581.
- [6] Jinyeong, Moon, Hye-young Lee. A Dual-Loop Delay Locked Loop with Multi Digital Delay Lines for GHz DRAMs. *IEEE International Symposium on Circuits and Systems*. 2012: 313-316.
- [7] RJ Yang, SI Liu. A 2.5 GHz All-Digital Delay-Locked Loop in 0.13 μm CMOS Technology. *IEEE Journal of Solid-State Circuits*. 2007; 42(11): 2338-2347.
- [8] GK Dehng, JM Hsu, CY Yang, et al. Clock-Deskew Buffer Using a SAR-Controlled Delay-Locked Loop. *IEEE Journal of Solid-State Circuits*. 2000; 35(8): 1128-1136.
- [9] RJ Yang, SI Liu. A 40-550 MHz Harmonic-Free All-Digital Delay-Locked Loop Using a Variable SAR Algorithm. *IEEE Journal of Solid-State Circuits*. 2007; 42(2): 361-373.
- [10] YJ Wang, SK Kao, SL. Liu. All-digital delay-locked loop/pulsewidth-control loop with adjustable duty cycles. *IEEE Journal of Solid-State Circuits*. 2006; 41(6): 1262-1274.
- [11] CC Chung, CL Chang. A wide-range all-digital delay-locked loop in 65nm CMOS technology. *IEEE international symposium on VLSI design automation and test (VLSI-DAT)*. 2010: 66-69.
- [12] L Wang, LB. Liu, HY Chen. An Implementation of Fast-Locking and Wide-Range 11-bit Reversible SAR DLL. *IEEE Transactions on Circuits and Systems II: Express Briefs*. 2010; 57(6): 421-425.
- [13] K Huang, ZK Cai, X Chen, et al. A Harmonic-Free All Digital Delay-Locked Loop Using an Improved Fast-Locking Successive Approximation Register-Controlled Scheme. *IEICE Trans. Electron*. 2009; E92-C(12): 1541-1544.
- [14] JS Wang, YM Wang, CY Cheng, etc al. An improved SAR controller for DLL applications. *IEEE International Symposium on Circuits and Systems*. 2006: 3898-3901.