A FREQUENCY ESTIMATION ALGORITHM FOR ADPLL DESIGNS WITH TWO-CYCLE LOCK-IN TIME

Chia-Tsun Wu, Wei Wang, I-Chyn Wey, An-Yeu (Andy) Wu

Graduate Institute of Electronics Engineering, and Department of Electrical Engineering,
National Taiwan University, Taipei 106, Taiwan, R.O.C.

***Abstract*—This paper presents a Frequency-Estimation Algorithm for the ADPLL designs instead of traditional binary frequency-search algorithm. With the proposed ADPLL architecture and synchronization process, the lock time can be optimized to two cycles. As the reference clock varies or frequency multiplication switches, lock time holds in two reference clock cycles. An implementation of proposed ADPLL design is realized in UMC 0.18 μm 1P6M CMOS technology with core area of 520×530 μm^2. The PLL has the frequency range of 140 MHz to 1030 MHz with 22ps DCO resolution.**

I. INTRODUCTION

PLL (Phase-Locked Loop) is widely used for various applications, especially in wire/wireless communications, CDR (Clock Data Recovery), and clock generators. In recent years, SoC (System-on-Chip) becomes a major trend. The ADPLL (All-Digital Phase-Locked Loop) applies digital design methodology to avoid mixed-signal design problems, and is suitable to be packed as hard IP, firm IP, and soft IP. These advantages make ADPLL very attractive to be integrated into SoC designs.

Another ADPLL advantage is that the ADPLL usually employs frequency-searching algorithm to achieve high-performance frequency lock-in time [1][2][4] as shown in Fig. 1. This fast–locking ADPLL is very useful for frequency hopping based FHSS (Frequency-Hopping Spread Spectrum) applications such as 802.11, Bluetooth, and FHMA (Frequency-Hopping Multiple Access) of UWB (Ultra Wide Band). Since the transceiver in a FHSS/FHMA system switches carrier frequency rapidly, a fast-switching and stable frequency synthesizer to generate the hopping carrier frequency is essential to FHSS/FHMA transmitters and receivers.

Most analog PLLs provide higher frequency resolution, but suffer from slow acquisition time. Lots of ADPLLs employ Binary Search Algorithm (BSA) as frequency-search algorithm to achieve faster lock-in time. The BSA [2][4] is a very simple, and useful algorithm. The search complexity is $O(\log_2 n)$. Using the BSA can reduce lock-in time to 16 reference clock cycles [3]. The lock-in time can be improved more than 40 times faster than traditional analog PLL [5]. Another approach for frequency-search uses TDC (Time-to-Digital Conversion), and developed DCO architecture [1] to achieve seven-cycle lock-in time.

This paper presents a Frequency-Estimation Algorithm (FEA) for frequency-acquisition in two-cycle lock-in time, with stable output frequency. We proposed a close-form analysis in time domain than others approach. A new VLSI architecture of ADPLL is also developed and realized in UMC 0.18 1p6m CMOS process. Cooperated with proposed scalable digital lock detector, the proposed ADPLL is suitable to fit demands of different design applications.

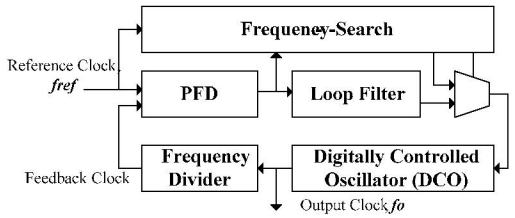


Figure 1. A typical ADPLL block diagram with frequency-search.

II. PROPOSED FREQUENCY-ESTIMATION ALGORITHM

The main feature of FEA is based on 1) Correlations between reference clock and DCO characteristics; 2) Analysis of DCO control code in time domain. The desired frequency output f_o can be obtained by further calculation instead of traditional search.

In this context, f_{ref} and f_o represent the frequencies of reference clock, and desired output frequency, respectively. N is multiplication factor. The frequency multiplication function of a PLL in locked state is described as

$$f_o = N \times f_{ref} . \tag{1}$$

Parameters are defined for clarification:

- τ_{ref} The cycle time of reference clock.
- τ_{min} The minimum cycle time of DCO when DCO operates at the maximum oscillating frequency.
- R_{max} The maximum cycle time ratio between reference clock and DCO clock. (DCO operated at τ_{min})
- τ_{max} The maximum cycle time of DCO when DCO operates at the minimum oscillating frequency.
- R_{min} The minimum cycle time ratio between reference clock and DCO clock. (DCO operated at τ_{max})

Then, the correlations between reference clock and DCO lock-in range are deduced as

$$R_{\min}\tau_{\max} = R_{\max}\tau_{\min} = \tau_{ref} . \quad (2)$$

Transformations from (2)

$$\tau_{\max} = \frac{\tau_{ref}}{R_{\min}}, \quad \tau_{\min} = \frac{\tau_{ref}}{R_{\max}} . \quad (3)$$

Consider a DCO provides T frequency acquisition steps. Then, we can derive the relation between resolution and difference of τ_{max}, and τ_{min} as

$$\Delta\tau = \frac{\tau_{\max} - \tau_{\min}}{T} = \frac{\frac{\tau_{ref}}{R_{\min}} - \frac{\tau_{ref}}{R_{\max}}}{T} . \quad (4)$$

where $\Delta\tau$ is the DCO acquisition resolution, and T is the total number of DCO acquisition steps.

After ADPLL closed loop converges, the cycle time of output frequency can be described as

$$\frac{1}{f_o} = F \times \Delta\tau + \tau_{\min} , \quad (5)$$

where F is the desired control code for accurate frequency.

From (1), and (5)

$$N \times \left(F \times \Delta\tau + \tau_{\min}\right) = \tau_{ref} . \quad (6)$$

Equation (6) denotes the relationship between reference clock and DCO parameters after closed loop converges.

From (3), (4), and (6) we have

$$N \times \left(F \times \frac{\frac{\tau_{ref}}{R_{\min}} - \frac{\tau_{ref}}{R_{\max}}}{T} + \frac{\tau_{ref}}{R_{\max}} \right) = \tau_{ref} . \quad (7)$$

After reorganization, the desired DCO control code F is expressed as

$$F = \frac{(T \times R_{\max} \times R_{\min}) - (T \times R_{\min} \times N)}{N \times \left(R_{\max} - R_{\min}\right)} . \quad (8)$$

Note that the τ_{max}, τ_{min}, and $\Delta\tau$ are parameters of process dependence, and difficult to be identified before fabrication. Also, the exact value of τ_{ref} varies between applications. But R_{max} and R_{min} can be obtained by frequency counters and dividers during run time. Also, the T is one of pre-defined parameters of the DCO, and N is the multiplication factor. Based on correlated dependency, F can be calculated by proposed FEA (8). An important feature of proposed FEA is, even under temperature fluctuations, power supply or process variations, the FEA gives exactly the desired control code.

Assume an ADPLL operates from 100MHz to 500MHz with *14* bits binary control code of DCO. The behavior simulations of proposed algorithm and BSA are shown within Fig. 2. When the BSA is applied in the frequency acquisition, the step size will be divided by two after for each frequency-search. Output frequency is unstable during convergence process in Fig. 2(a). The proposed FEA locks the reference clock in very few cycles with a stable convergence process as shown in Fig. 2(b).

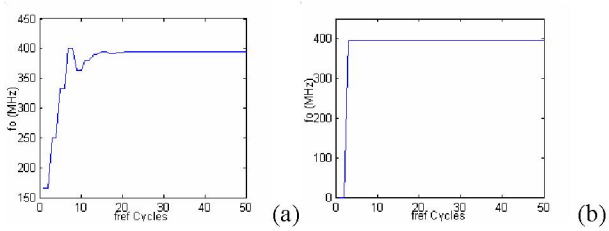


Figure 2. (a) Behavior simulation of binary-search algorithm. (b) Improved convergence of frequency-estimation algorithm.

III. Proposed ADPLL Architecture for Two-Cycle Locking

The VLSI architecture, Fast-Lock Engine (FLE), is developed to realize FEA algorithm in Fig. 3(a). With cooperation of FLE, an ADPLL design can provide fast-lock operation.

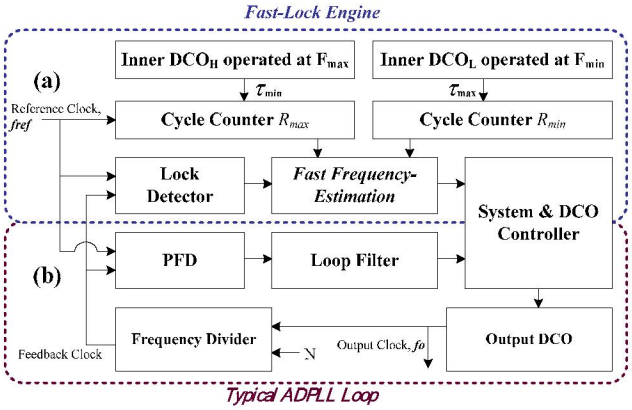


Figure 3. Proposed ADPLL architecture for two cycles lock.

When targeting at frequency acquisition with two-cycle lock-in time, R_{max}, and R_{min} should be concurrently available. Hence, two additional inner DCOs are utilized in proposed ADPLL configuration. Inner DCOH and DCOL were operated at the highest / slowest frequency, respectively. The generated clock signals are transmitted separately to cycle-counters. While system controller receives R_{max}, and R_{min} from two cycle-counters, the F can be calculated for desired frequency according to FEA.

A. *Fast-Lock Process*

Fig. 4(a) illustrates synchronization between reference clock, and f_o. Assume a frequency step is applied to the f_{ref} at cycle *1* in Fig. 4(b). The LD (Lock Detector) detects the frequency step during cycle *1* and activates trigger-signal (CycleSync). The R_{max} and R_{min} are both calculated simultaneously. The DFF suspends DCO on the rising edge of cycle *2* and FLE is now activated. FLE calculates the F during cycle *2*. At last, the DCO resumes with new control code F at the rising edge of cycle *3*. Based on this process, fast-lock frequency acquisition can be realized in two cycles.

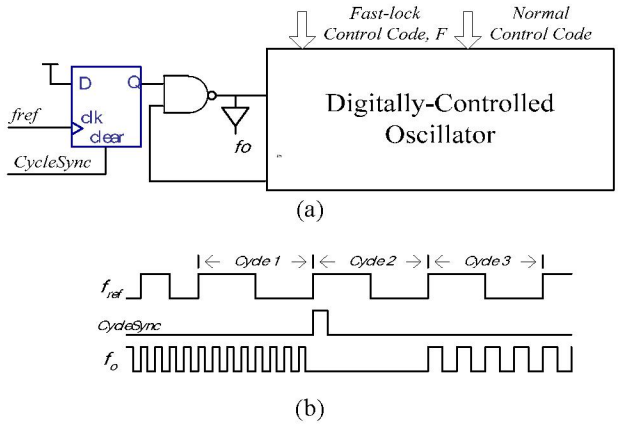


Figure 4. (a) Synchronization for DCO. (b) Fast-lock Process.

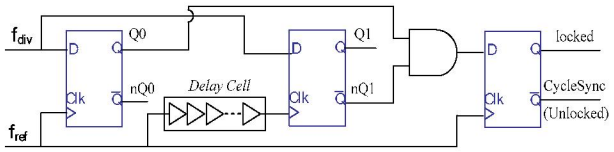
B. *Configurable Lock Detector (LD)*

A new LD configuration is utilized as phase /frequency lock detector as shown in Fig. 5(a). Phase error between f_{ref} and f_{div} and circuit status are shown in Fig. 5(b). The main function of LD is the detection of numerous phase errors. Once the phase error exceeds the threshold T_{LD}, the LD triggers CycleSync and activates the FLE for fast-lock operation. The delay cell can be constructed with a scalable buffer chain. Thus, the propagation delay of delay cell decides threshold of LD. The relation can be expressed as

$$T_{LD} = M \times \tau_{BUF}, \tag{8}$$

where T_{LD} is the LD threshold, M is the number of buffer stage, and τ_{BUF} is the gate delay of a buffer.

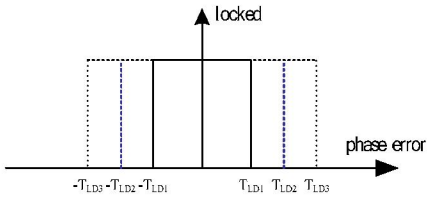
Also, the total umber of buffer stage, M, should be determined by application specifications. The relation of LD output versus different thresholds is illustrated in Fig. 5(c). For a popular application example of video, which referencing a HSYNC (horizontal synchronization) as f_{ref} to generate 27 MHz clock, 5 ns phase error may be tolerated. When referencing a HSYNC from VGA to generate 200 MHz clock for LCD monitor, the phase error may only be tolerated less than 500 ps and M should be reduced. The proposed LD is scalable and flexible to meet demands of various applications.



(a)

f_{input}	waveform	truth table			
f_{ref}		Q1	Q0	phase error	status
f_{div}		0	0	$< \frac{1}{2}\tau_{ref} + T_{LD}$	unlocked
		0	1	$\leq T_{LD}$	locked
		1	0	$\sim \frac{1}{2}\tau_{ref} + T_{LD}$	unlocked
		1	1	$> \frac{1}{2}\tau_{ref} + T_{LD}$	unlocked

(b)



(c)

Figure 5. (a) Lock detector architecture. (b) Charateristics of LD (c) Simulation of LD output v.s. different threshold.

C. *Working flow chart of the proposed ADPLL*

Fig. 6 shows the proposed ADPLL working flow chart. Once the phase error over the threshold, LD activates FLE, the system controller receives the desired control code F. The fast-lock acquisition is completed in two cycles. Otherwise, fine phase acquisition and are performed by phase detector (PD); phase / frequency maintenance are controlled by ADPLL closed loop.

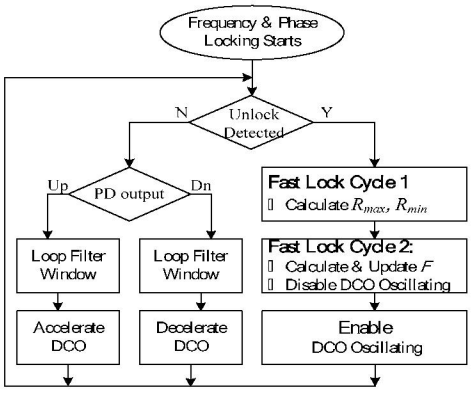


Figure 6. ADPLL working flow chart.

IV. Implementation and Simulation

Based on the proposed algorithm and architecture, a prototype ADPLL design is designed and realized using

UMC 0.18μm 1P6M CMOS technology, as shown in Fig. 7. For linear acquisition step is desirable in proposed frequency-estimation algorithm, a high-linearity DCO design [6] can be cited for purpose. With post-layout simulations, the DCO contains 272 frequency acquisition steps and resolution is about 22 ps. The DCO operates from 140 MHz to 1030 MHz. Three DCOs are replicas for FLE. The ADPLL core area is 520×530 μm^2.

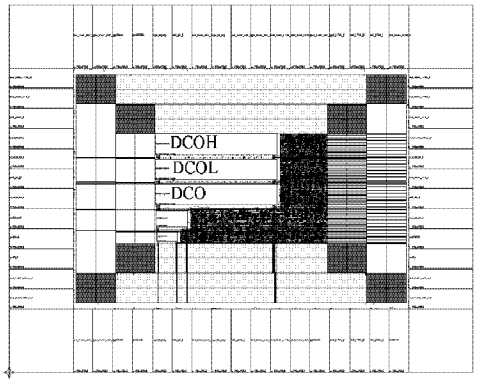


Figure 7. Chip layout of the prototype ADPLL. (where DCO: outer DCO, DCO_H: inner DCO operated at F_{max}, DCO_L: inner DCO operated at F_{min},)

For example of fast-lock operation is shown in Fig. 8. In the initial state, reference clock is 15.625 MHz, N is 32, output clock frequency is 500 MHz, and corresponding control code F is 50. As multiplication number N switches to 48; the FLE is activated; desired F should be 16. The output frequency turns out to be 750 MHz in two cycles and the acquisition process is exactly verified with simulation in Fig. 8.

For a practical industrial application, Fig. 9(a) shows the phase difference versus reference clock for 200 MHz frequency output. The phase error at the rising edge of cycle *3* is around 0.047 radian in Fig. 9(a). The desired output frequency is shown in Fig. 9(b).

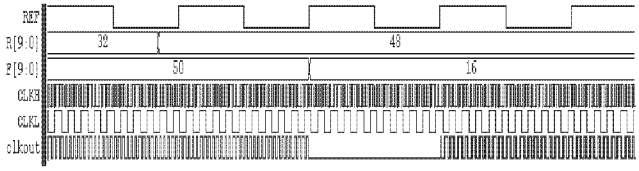


Figure 8. Simulation of frequency multiplication switch.

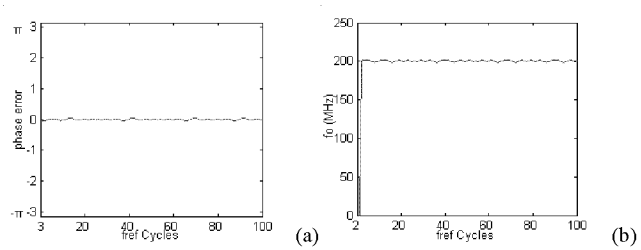


Figure 9. (a) Phase error v.s. reference clcok cycles. (b)ADPLL frequency out v.s reference clock cycles.

V. Conclusions

The Frequency-Estimation Algorithm for ADPLL and VLSI architecture are presented in this paper. Compared to existing algorithms, proposed algorithm realizes the fastest lock-in time of two cycles. Also, a new scalable LD is utilized for requirements of different applications. The prototype design is implemented in UMC 0.18 CMOS process with standard-cell library. The chip operates from 140 to 1030 MHz, and the core area is 520×530 μm^2. The fast-lock ADPLL design can be easily transferred and incorporated into system integration due to its digital feature.

Acknowledgment

The authors thank CIC (Chip Implementation Center) for CAD tool supports and technical discussions.

References

[1] Takamoto Watanabe and Shigenori Yamauchi, "An All-Digital PLL for Frequency Multiplication by 4 to 1024 With Seven-Cycle Lock Time," IEEE Journa of Solid-State Circuit, vol. 38, no. 2, February 2003.

[2] C. Chung and C. Lee, "An all-digital phase-locked loop for high-speed clock generation," IEEE International Symposium on Circuits and Systems, vol. 3, pp.26-29, May 2002.

[3] I. Hwang, S Lee, S. Lee, and S. Kim, "A digitally controlled phase-locked loop with fast locking scheme for clock synthesis application," IEEE Int. Solid-State Circuits Conf. Dig. Tech. Papers, Feb. 2000, pp. 168-169

[4] T.-Y. Hsu, C.-C. Wang, and C.-Y. Lee, "Design and analysis of portable high-speed clock generator," IEEE Trans. Circuits Syst. II, vlo. 48, pp. 367-375, Apr. 2001.

[5] H.-T Ahn and D.J. Allstot, "A low-jitter 1.9V CMOS PLL for ultra-SPARC microprocessor applications," IEEE K. Solid-State Circuits, vol. 35, pp. 450-454, May 1999.

[6] Chia-Tsun Wu, Wei Wang, I-Chyn Wey, and An-Yeu (Andy) Wu, "A Scalable DCO Design for Portable ADPLL Designs," IEEE International Symposium on Circuits and Systems, pp. 5449-5452, May 2005.