The Implementation of An Adaptive Bandwidth All-Digital Phase-Locked Loop

Chen-Feng Chen and Yawgeng A. Chau
Department of Communications Engineering
Yuan-Ze University
Chung-Li, Taiwan 320
eejfchen@saturn.yzu.edu.tw, eeyaw@saturn.yzu.edu.tw

***Abstract*—A circuit architecture of the adaptive-bandwidth all-digital phase-locked lopp (ADB-ADPLL) is designed, where a second-order system with proportional and integral (PI) gains is used to model the ADB-ADPLL. For the implementation of the ADB-PLL, a counter-based phase-frequency detector (PFD) is proposed and a time-to-digital converter (TDC) is used to transform the PFD output to a digital signal. In the ADB-ADPLL system, a digital loop filter (DLF) and a digital-controlled oscillator (DCO) are implemented based on the modulo-N algorithm and PI control. Simulation results are presented to illustrate the ADB-ADPLL performance.**

Keywords-Adaptive Bandwidth (ADB);Phase-Locked Loop (PLL);Second-Order System; Proportional Gain; Integral Gain; Time-to-Digtal Converter

I. INTRODUCTION

The phase-locked loop (PLL) has various applications in frequency synthesis, digital modulation, and synchronization, [1], [2]. The PLL can be implemented with analog/mixed-signal circuits, or.all-digital circuits [1]. The analog/mixed-signal PLL can supply high-frequency outputs, but its performance highly depends on the pressure-voltage-temperature manufacture conditions. In addition, the phase noise of an analog/mixed-signal PLL may is difficult to control. In contrast to the analog/mixed-signal PLL, the all-digital PLL (ADPLL) can be realized with a digital VLSI process, and its performance is easier to control. Moreover, the ADPLL can be easily integrated with a digital circuit system that is useful for many applications such as the baseband communications or electronic equipments.

On the other hand, to accommodate the noise in different frequency bands, a PLL with an adaptive bandwidth (AB) has been analyzed and designed in [3]-[5]. For an adaptive bandwidth PLL, the ratio of the loop natural frequency to the input reference frequency (denoted by the bandwidth ratio) must be maintained as a constant such that the locking time is adaptive to different input reference bands. In [3], the general rule of a mixed-signal PLL with an adaptive band was analyzed, where continuous- and discrete-time analyses were presented.. In [4], the mixed-signal PLL was designed. In the design of mixed-signal ADB-PLL, the leakage of CMOS capacitors and the sub-threshold leakage of CMOS switches pose potential problems to the realization of the PLL, and are difficult to control in the practical CMOS process.

In this paper, an ADB-ADPLL circuit is designed and its performance is illustrated with simulation. The general architecture of an APLL is illustrated in Fig. 1, where a PFD, a DLF, a frequency divider, and the DCO are employed.

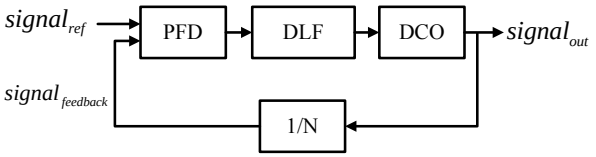


Fig. 1 The ADPLL system.

The paper is organized as follows. In Section II, the ADB-ADPLL model and its design criterion are reviewed In Section III, a circuit architecture that implements the ADB-ADPLL is proposed and analyzed. In Section IV, the simulation results of the ADB-ADPLL is presented. Conclusions are drawn in Section V.

II. ADB-ADPLL MODELING

In this section, the second-order ADB-ADPLL system based on its counterpart of an analog ADB-PLL is reviewed, and the key relations between the parameters of digital loop filters summarized.

For an ADB PLL, the following relations must be held [4]

$$\omega_n = C \cdot \omega_{ref} \quad \text{and} \quad \varsigma = \text{constant} \tag{1}$$

where C is a constant.

For a second-order PLL system, let K_P and K_I be the proportional and integral gains of a second-order PLL. It can be shown that

$$K_P = 2\varsigma\omega_n \text{ and } K_I = \omega_n^2 \tag{2}$$

where ς is the damping ration, and ω_n is the natural frequency of the second-order PLL system. Let T_s be the sampling period of a discrete-time PLL system. Then, with the z-transform, the discrete-time transfer function of a PLL is give by [5]

$$H_{DPLL}(z)=\frac{\left(K_PT_s+K_IT_s^2\right)z-K_PT_s}{z^2+\left(K_PT_s+K_IT_s^2-2\right)z+1-K_PT_s}$$
$$=\frac{\left(K_1+K_2\right)z-K_1}{z^2+\left(K_1+K_2-2\right)z+1-K_1} \quad (3)$$

where (K_1, K_2) are the parameters of the digital loop filter given by

$$K_1 = K_PT_s \text{ and } K_2 = K_IT_s^2. \quad (4)$$

The PI-based second-order ADPLL system with (K_1, K_2) is illustrated in Fig. 2 below.

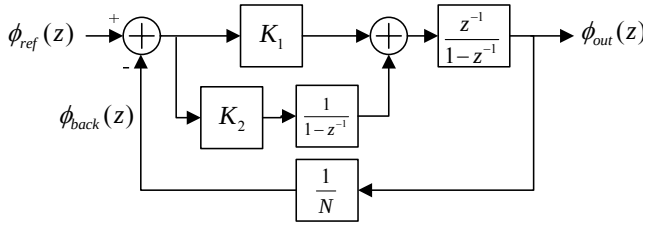


Fig. 2 The ADPLL system with the PI control.

With an ADB-ADPLL, for a range of the input reference frequency ω_{ref}, the loop parameters (K_1, K_2) should be able to adapt and adjust themselves in some way for different values of ω_{ref}. If we select the sampling frequency as $\omega_s = R \cdot \omega_{ref}$ with a constant ratio R, then

$$K_1 = 4\pi\varsigma\frac{\omega_n}{R\omega_{ref}} \quad (5)$$

and

$$K_2 = 4\pi^2\frac{\omega_n^2}{R^2\omega_{ref}^2} = \left(\frac{K_1}{2\varsigma}\right)^2. \quad (6)$$

III. ADB-ADPLL Circuit Architecture

In this section, the corresponding circuits of the ADPLL subsystems depicted in Fig. 2 are designed and described.

A. PFD

For the ADB-ADPLL, the function of the phase-offset detection is implemented by a PFD. This PFD circuit is plotted in Fig. 3, where the up/down signal is denoted by $singal_{ref}$ /$signal_{feedback}$. The analytical PFD model is given in Fig. 4.

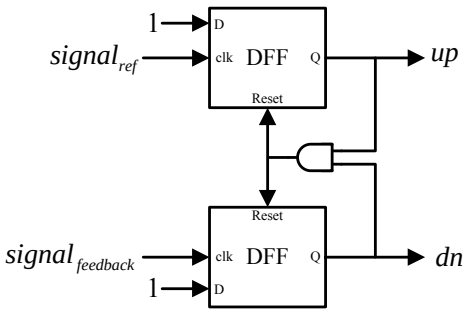


Fig. 3 The PFD circuit

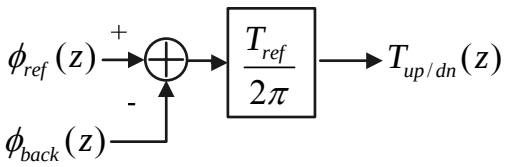


Fig. 4 The PFD analytical model.

In addition, an example of possible $singal_{ref}$ and $signal_{feedback}$ signals is illustrated in Fig. 5, where *up*/*dn* denotes the up/down signal.

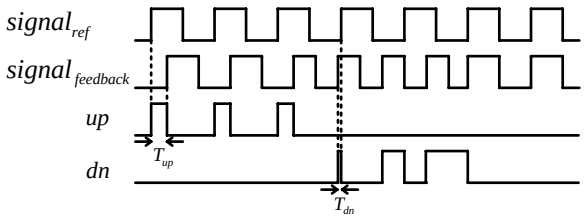


Fig. 5 The example of timing diagram for the inputs of PFD.

B. DLF

The output of the PFD is send to the DLF that in turn provides a digital level as the input of the following DCO. The DLF architecture is based on the PI control concept.

Particularly, in the DLF circuit, the PI gains are implementation by four modulo-N counters with enable (Modulo-N WE) that also enlarge the *up* and *dn* signals from the PFD. Then, two frequency control words (FCWs) are used to realize the PI control.

The implementation of the DLF is shown in Fig. 6, where to time-to-digital converters (TDCs) are employed to transform the outputs of the Modulo-N WE to digital levels that will function as the control signal of the DCO. In the DLF given in Fig. 6, the two sets of up/down signals (up_P, dn_P) with FCW_α and (up_I, dn_I) with FCW_β are the amplified controlled signals of the TDCs. Furthermore, the signal $singal_{ref}$ triggers the TDC_I and TDC_P employs the TDC_I output as an initial value to perform the consecutive count, and the output of TDC_P (i.e. DCO_{FCW}) will trigger the DCO.

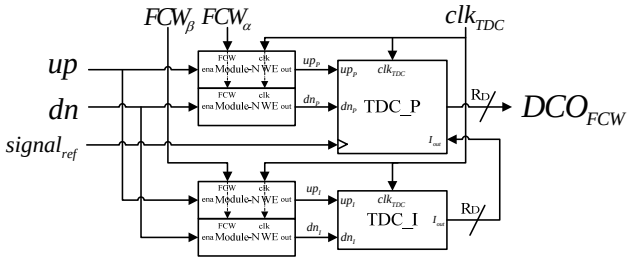


Fig. 6 The circuit of the Mudulo-N counter-based DLF.

In the DLF system, the Modulo-N WE circuits implement the addition and accumulation functions. In Fig. 6, the detailed Modulo-N WE circuit is shown. An example for the illustration of the timing relation of the Modulo-N WE is plotted in Fig. 7.

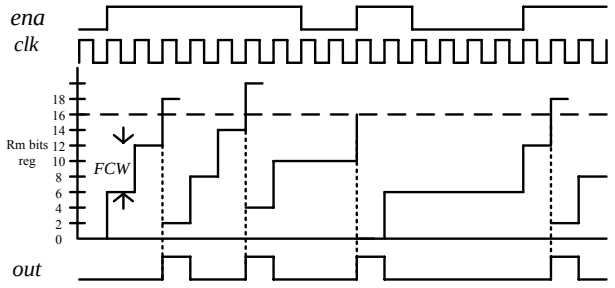


Fig. 7 An example of timing relation for the Modulo-N counter.

C. TDC

The TDC will convert the DLF output to the input trigger of the DCO that will generate the desired frequency for ADPLL, where the following relation between the time duration of signal T_{ena} and time duration of signal out T_{out}

$$T_{out} = \frac{\mathrm{FCW}}{2^{R_m}} \cdot T_{ena}, \quad 1 \le \mathrm{FCW} \le 2^{R_m} - 1. \tag{7}$$

Thus, we have

$$T_{up_P}(z) = \frac{\mathrm{FCW}_\alpha}{2^{R_m}} * T_{up}(z), \quad T_{dn_P}(z) = \frac{\mathrm{FCW}_\alpha}{2^{R_m}} * T_{dn}(z), \tag{8}$$

and

$$T_{up_I}(z) = \frac{\mathrm{FCW}_\beta}{2^{R_m}} * T_{up}(z), \quad T_{dn_I}(z) = \frac{\mathrm{FCW}_\beta}{2^{R_m}} * T_{dn}(z), \tag{9}$$

where T_{up} and T_{dn} are the timing durations of up and down signals, respectively.

The typical circuit of TDC_I and TDC_P is shown in Fig. 8 where clk_{TDC} is the counting clock for in TDC_P and TDC_I, In the TDC circuit, P_{out} and I_{out} will up count when $up_P = up_I = 1$, and down count when $dn_P = dn_I = 1$, where P_{out} will be sent to DCO_{FCW} when $signal_{ref}$ is triggered.

To prevent overflow of the TDC, its output is retained when the incrementer input reaches its maximum value, $2^{R_D} - 1$, or the decrementer input downs to its minimum value, zero.

For the TDC function, the following relations hold

$$DCO_{FCW}(z) = \frac{1}{T_{TDC}}(T_{up_P}(z) - T_{dn_P}(z)) + I_{out}(z) \tag{10}$$

and

$$I_{out}(z) = \frac{1}{T_{TDC}}(T_{up_I}(z) - T_{dn_I}(z)) + I_{out}(z)z^{-1}. \tag{11}$$

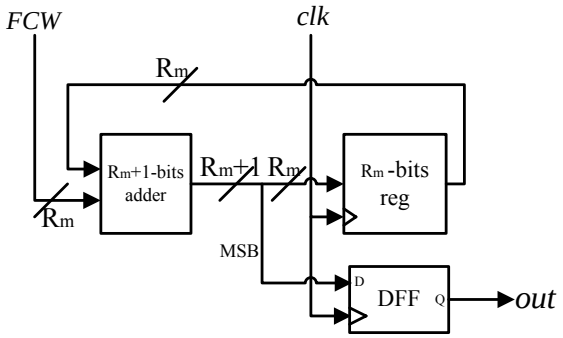


Fig. 8 The circuit of Modulo-N counters.

Substituting (8) and (9) into (10) and (11), we have

$$DCO_{FCW}(z) = \frac{\mathrm{FCW}_\alpha}{2^{R_m} T_{TDC}}\left[T_{up}(z) - T_{dn}(z)\right] + I_{out}(z) \tag{12}$$

and

$$I_{out}(z) = \frac{\mathrm{FCW}_\beta}{2^{R_m} T_{TDC}}\left[T_{up}(z) - T_{dn}(z)\right] + I_{out}(z)z^{-1}. \tag{13}$$

D. DCO

In the context, a counter-based DCO is employed. In. Fig. 9, the circuit architecture of the DCO is plotted. In Fig. 9, an example of the timing diagram is illustrated, where the counter initial value is set by DCO_{FCW} for the frequency control of DCO output signal clk_{out}.

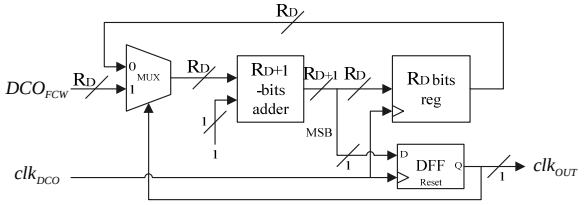
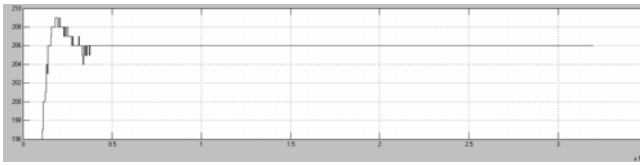


Fig. 9 The DCO circuit.

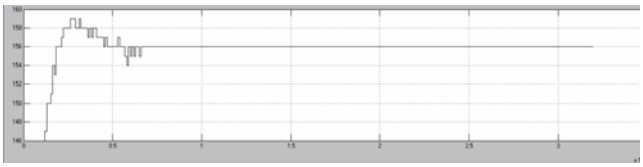
IV. Simulation Results

To demonstrate the ability of adaptive bandwidth for the ADPLL, extensive simulations are performed with MATLAB. For the simulation, the relevant parameters are set as $C = 0.01$, $\xi = 0.707$, $R_m = 10$, $R_D = 8$, $T_{DCO} = T_{TDC} = 1ps$, and $N = 1$.

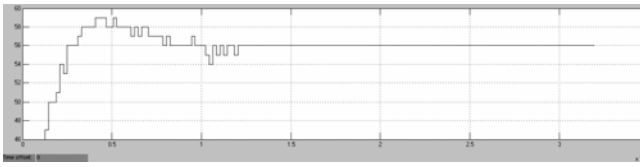
In Fig. 10, the bandwidth adaptability is shown by comparing the step change at different input frequency bands.



(a)



(b)



(c)

Fig. 10 The simulation results for the input frequency change of the ADB-ADPLL for different input bands: (a) in a 200MHz band (b) in a 150 MHz band (c) in a 50MHz band.

V. CONCLUSIONS

In this paper, the ADB-ADPLL system is implemented by digital circuits, where the counter-based PFD with a TDC is designed and modulo-N based DCO is proposed. From the simulation results, the ADB-ADPLL can adjust by itself to accommodate the changing of input frequency band to yield stable locking time.

ACKNOWLEDGMENT

The author would like to thank the National Chip Implementation Center, Taiwan for their kind assistance.

REFERENCES

[1] R. E. Best, *Phase-Locked Loops*, 5th ed, McGraw-Hill, 2003.

[2] V. F. Kroupa, *Phase Lock Loops and Frequency Synthesis*, Wiley, 2003.

[3] J. Kim, M.A. Horowitz, and G. Wei, "Design of CMOS adaptive-bandwidth PLL/DLLs: A general approach," *IEEE Trans. CIrcuits and Systems—II: Analogand Digital Signal Processing*, Vol. 50, pp. 860-869, Nov. 2003.

[4] S. Sidiropoulos, D. Liu, J. Kim, G. Wei, and M. Horowitz, "Adaptive bandwidth DLLs and PLLs using regulated supply CMOS buffers," *IEEE Symp. VLSI Circuits Dig. Tech. Papers,* June 2000, pp. 124–127.

[5] J. G. Maneatis, "Low-jitter process-independent DLL and PLL based on self-biased techniques," *IEEE J. Solid-State Circuits*, vol. 31, pp. 1723–1732, Nov. 1996.