A Low-Power All-Digital Phase-Locked Loop Using Binary Frequency Searching[§]

Chua-Chin Wang[†] , *Senior Member, IEEE*, Tzung-Je Lee, and Sheng-Lun Tseng
Department of Electrical Engineering
National Sun Yat-Sen University
Kaohsiung, Taiwan 80424
email : ccwang@ee.nsysu.edu.tw

Abstract— **We propose a low power ADPLL (All-digital phase-locked loop) by using a controller which employs a binary frequency searching method in this paper. Glitch hazards and timing violations which occurred very often in the prior ADPLL designs are avoided by the control method and the modified DCO (Digital-controlled oscillator) with multiplexers. The proposed design is implemented by only using the standard cells of TSMC (Taiwan Semiconductor Manufacturing Company) 0.18 μm CMOS process. The feacture of power saving is verified by post-layout simulations, which show that the power consumption of the proposed ADPLL is merely 3 mW at 133 MHz output.**

Key Words—ADPLL, low power, glitch, timing violation, DCO

I. Introduction

Phase-locked loops (PLLs) are widely used circuits for frequency synthesis applications. A traditional PLL consists of a digital phase frequency detector (PFD), and an analog part including a charge pump, a loop filter, and a voltage controlled oscillator (VCO). Many parameters of the analog circuits in the traditional PLL are sensitive to temperature variation, supply voltage noise as well as process drift. They result in the design difficulty and the necessity of redesigning for each new technology. Moreover, capacitors and resistors, which are required in the loop filter in the traditional PLL, usually cause an area penalty.

On the contrary, ADPLL is made up with all digital components such that it possesses a high immunity to supply noise and temperature variation. Moreover, ADPLL can be designed by using hardware description language (HDL) with any standard cell library [5]. Thus, the portability over different processes is ensured and the time for redesign is reduced. Therefore, ADPLL has been received great attention to date [3], [2], [5]. However, ADPLL has a crucial disadvantage, i.e., large power consumption resulting from the digital-controlled oscillator (DCO), as shown in Table I. For instance, the ADPLL for high-speed clock generation proposed by Jong [1] consumes 100 mW at 500 MHz, which does not follow the trend of low power design. Moreover, it may introduce glitches into the oscillator due to switching delays.

Thus, we propose a novel control method for ADPLL to reduce the power consumption caused by DCO and avoid glitch hazards. The power consumption of the proposed ADPLL is found to be merely 3 mW at 133 MHz output.

II. The proposed ADPLL

Referring to Fig. 1, the proposed ADPLL is composed of a PFD, a frequency divider (FDIV), two DCOs (FB_DCO and OUT_DCO), and the controller for saving power (CSP). PFD detects the phase difference of the reference clock CLK_REF and the feedback clock CLK_FB. When CLK_FB lags CLK_REF, PFD generates a negative impulse on UP while DOWN remains at high for CSP to speed up the FB_DCO. On the contrary, a negative impulse on DOWN is generated for CSP to slow down FB_DCO if CLK_FB leads CLK_REF. CSP generates two signals, COARSE and FINE, for FB_DCO to select an oscillation frequency of CLK_FB_M. The frequency of CLK_FB_M is divided by the FDIV with a division ratio MOD to generate the divided signal CLK_FB which is sent back to PFD.

After the coarse tune and the fine tune procedures, a stable feedback loop is built up and the frequency of CLK_FB_M is adjusted to be MOD times of that of CLK_REF. Simutaneously, a signal LOCK is generated by CSP to indicate the frequency is locked successfully. In order to furtherly reduce the jitter caused by (1) PFD's deadzone, (2) the FB_DCO's finite resolution, and (3) the input jitter, CSP computes the averaged values of COARSE and FINE, i.e., AVG_COARSE and AVG_FINE, respectively, for OUT_DCO to generate the stable output signal CLK_OUT.

A. The PFD and the FDIV

Fig. 2 shows the schematic of PFD [1]. The PFD generates a low impulse on UP or DOWN according to the lag or lead of CLK_FB relative to CLK_REF, respectively. Notably, the impluse itself, not the width of the impulse, is used to the invoke the speed-up or slow-down event in the CSP.

Two digital pulse amplifiers (DPAs), as shown in Fig. 3, are used to increase the pulse width of INTU and INTD such

[§] This research was partially supported by National Science Council under grant NHRI-EX93-9319EI and NSC 92-2218-E-110-001.

[†]the contact author.

that the following D-filp-flops (DFFs), DFF3 and DFF4, can detect them [1]. Thus, the dead zone of the PFD can be effectively minimized. Moreover, in order to increase the sensitivity to the input phase difference, an asynchronous reset mode is adopted in DFF1 to DFF4. However, the asynchronous reset mode may cause the timing violation happened in these DFFs if CLK_FB and CDN possess contradictive values, as shown in Fig. 4. This problem can be avoided by the proposed control method in the following text.

FDIV is realized based on a 4-bit counter to provide a variable division ratio, MOD, by users.

B. The DCOs with MUX-based Switching

The DCOs, FB_DCO and OUT_DCO, are composed of COARSE_TUNE and FINE_TUNE delay cells, as shown in Fig. 5. The reset signals, RESET_A or RESET_B, receive a low impulse to initiate the DCOs. The control codes, COARSE and FINE, switch the COARSE_TUNE and FINE_TUNE delay cells, respectively, to select the desired oscillating frequency, as shown in Fig. 6. In Jong's design, tri-state buffers were used to switch the delay cells [1] which might cause timing violations in post-layout simulations due to the phase difference of the inputs of two switched tri-state buffer. Instead, we propose to utilize multiplexers to switch the delay cells to resolve this problem. A rising edge of the input signal of the multipexer is pre-set to wait for COARSE and FINE codes to switch the delay line such that the timing violation can be avoided.

Nevertheless, a glitch will appear at the output of DCOs if the switched inputs of the multiplexers does not rise simultaneously. What even worse is that the glitch would be accumulated over the feedback loop resulting in the chaos on the outputs of DCOs, as shown in Fig. 7. The glitches can be removed by the proposed control method in the following text.

C. CSP with Binary Frequency Searching

The state diagram of the proposed controller is divided into four states, START, COARSE_TUNE, FINE_TUNE, and LOCK, as shown in Fig. 8. The controller is initialized at the START state and then triggered by the UP and DOWN signals from PFD to move to the COARE_TUNE state. In the COARE_TUNE state, the 5-bit COARE codes are computed by a binary search, as shown in Fig. 9, to adjust the frequency of FB_DCO. According to the binary search, FB_DCO starts from its middle frequency. The frequency is increased if a low impulse on UP is received. Otherwise, the frequency is slowed down if a low impulse on DOWN is received. When each bit of COARSE is determined, the counter COARSE_STEP, which is used to count the binary search for COARSE, decreases by one from the number of the bit size of COARSE. This procedure continues until either of the following two events happen. Firstly, when the COARSE_STEP equals to 1 and [COARSE, FINE]=10 or 01, the controller moves to the FINE_TUNE state to comput the 4-bit FINE codes. Secondly, when UP and DOWN are both high, which means the target frequency is found, it goes to the LOCK state. The search step of the FINE_TUNE state is the same as the CORASE_TUNE state except for no counter required to count the search step.

The proposed low-power control method to switch the frequency of FB_DCO for the binary search is addressed as follows. Two cycles of CLK_FB_M are needed to complete a search step for the frequency of CLK_FB. At the rising edge of the first cycle, i.e., A in Fig. 10, CSP enables FB_DCO by RESET_A to attain the simultaneous rising edges of CLK_FB and CLK_REF. At the rising edge of the second cycle (B in Fig. 10), CSP receives the detected phase error of CLK_FB and CLK_REF from PFD. At the falling edge of the second cycle (C in Fig. 10), CSP sends the computed COARSE or FINE codes according to UP or DOWN received to select the frequency of CLK_FB_M, and disables FB_DCO at the same time. Repeatedly, FB_DCO is disabled every time when the delay line is switched. Thus, the glitch on CLK_FB_M is avoided. The power consumption is also saved due to the pause of FB_DCO.

After entering the LOCK state, the CSP computes the averaged value over the prior 64 cycles of COARSE and FINE. The two average values, AVG_COARSE and AVG_FINE, are sent to OUT_DCO to generate the output clock CLK_OUT. Besides, a signal LOCK is generated to enable OUT_DCO and to indicate that the target frequency is found.

III. Implementation and Simulation

The proposed ADPLL is carried out by using TSMC 0.18 μm 1P6M CMOS process standard cells with 1.8 V power supply, as shown in Fig. 11. In order to achieve the timing accuracy, we use Verilog hardware description language to directly select the cells from TSMC standard cells. The delay units of COARSE_TUNE cells and FINE_TUNE cells in DCOs must be designed carefully to balance the rising time and the falling time such that the 50% duty cycle in CLK_OUT and CLK_FB_M can be obtained.

The post-layout simulation in Fig. 12 shows the phase detection process of PFD, where no timing violation is occurred. Fig. 13 shows the simulation results of the proposed ADPLL, where the reference clock is 10 MHz, the division ratio MOD is 12, and the ourput frequency is 125 MHz. Moreover, there is no glitch or timing violation regarding CLK_FB_M and any other signal. Table II summarizes the comparison of the proposed ADPLL and several prior works. The frequency range of the proposed design is unavoidably shrunk due to the extra delay caused by the multiplexers.

IV. Conclusion

We have presented a low power ADPLL by using a binary frequency searching method. Moreover, the glitch hazard and the timing violation in the prior works are avoided by the control method and the modified DCOs with multiplexers. The feacture of saving power is verified by the post-layout simulation, which shows that the power consumption of the proposed ADPLL is merely 3 mW.

References

[1] J.-J. Jong, and C.-Y Lee, “An All-Digital Phase-Locked Loop for High-Speed Clock Generation,” *IEEE J. Solid-State Circuits,* vol. 38, no. 2, pp. 347-351, Feb. 2003.

[2] T. Olsson, and P. Nilsson, “A digitally controlled PLL for SoC applications,” *IEEE Journal Solid-State Circuits,* vol. 39, no. 5, pp. 751-760, May 2004.

[3] J.-S. Chiang, and K.-Y Chen, “The design of an all-digital phase-locked loop with small DCO hardware and fast phase lock,” *IEEE Trans. Circuits Systems II,* vol. 46, no. 7, pp. 945-950, July 1999.

[4] J. Dunning, G. Garcia, J. Lundberg, and E. Nuckolls, “An all-digital phase-locked loop with 50-cycle lock time suitable for high-performance microprocessors,” *IEEE J. Solid-State Circuits,* vol. 30, no. 4, pp. 412-422, Apr. 2004.

[5] T.-Y. Hsu, C.-C. Wang, and C.-Y. Lee, “Design and analysis of a portable high-speed clock generator,” *IEEE Trans. Circuits Systems II,* vol. 48, no. 4, pp. 367-375, Apr. 2001.

	Traditional PLL	ADPLL
Design time	Long	Short
Reusability	Bad	Good
Noise immunity	Bad	Good
Area	Large	Small
Power consumption	Small	Large

TABLE I

PERFORMANCES COMPARISON BETWEEN THE TRADITIONAL PLL AND ADPLL.

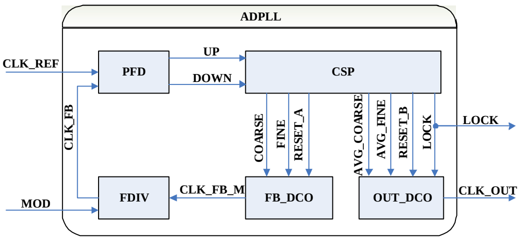

Fig. 1. The structure of the proposed ADPLL.

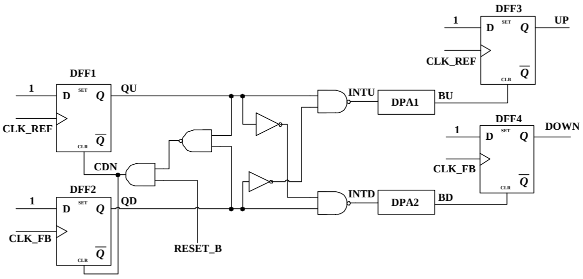

Fig. 2. Schematic of the PFD.

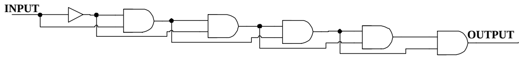

Fig. 3. Schematic of the digital pulse amplifier.

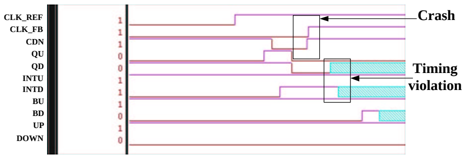

Fig. 4. Timing violation on QU and QD.

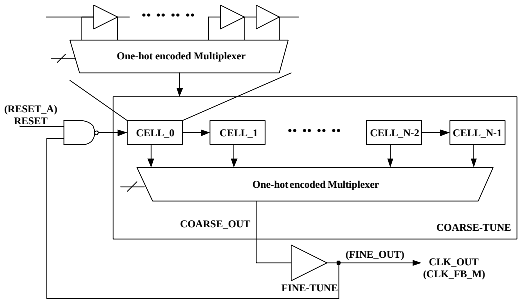

Fig. 5. Schematic of the DCOs.

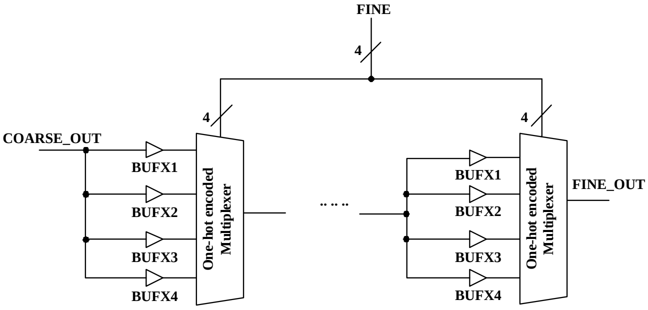

Fig. 6. Schematic of FINE_TUNE cells.

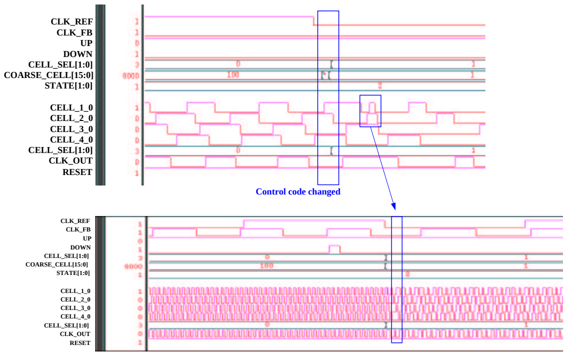

Fig. 7. The glitch on CLK_FB_M due to the accumulated clash.

	ours	[1]	[2]	[4]	[5]
Technology	0.18 μm CMOS	0.35μm CMOS	0.35μm CMOS	0.5μm CMOS	0.6μm CMOS
Design approach	cell-based	cell-based	cell-based	Full-custom	cell-based
Area(mm^2)	0.06	0.71	0.07	1.1	2.75
Power comsumption	3 mW @133 MHz	100 mW @500 MHz	8.1 mW @152 MHz	39.6 mW @100 MHz	315 mW @800 MHz
Max. freq.	158 MHz	510 MHz	336 MHz	550 MHz	800 MHz
Min. freq.	70 MHz	45 MHz	152 MHz	50 MHz	360 MHz
Supply voltage	1.8 V	3.3 V	3.0 V	3.3 V	3.3 V
Output jitter	300 ps	70 ps	1.2 ns	125 ps	60 ps

TABLE II

SPECIFICATIONS COMPARISON OF THE PROPOSED ADPLL.

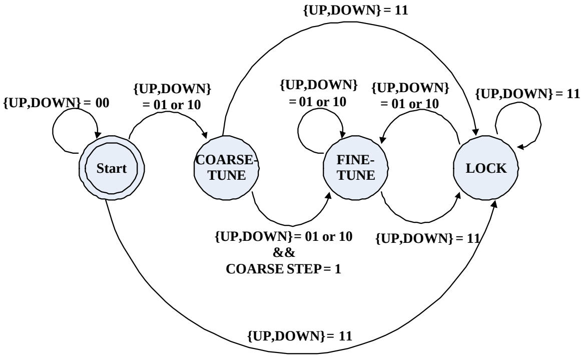

Fig. 8. The state diagram of the proposed CSP.

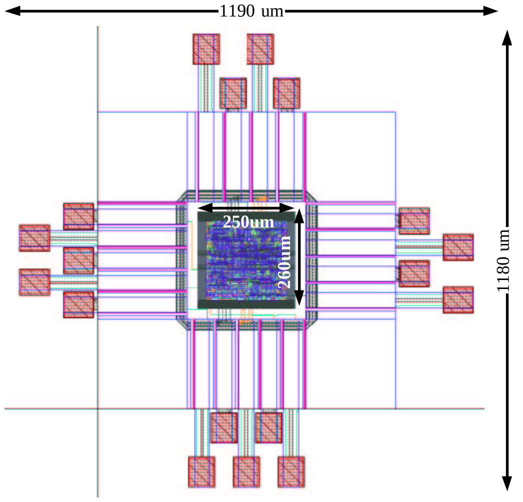

Fig. 11. Layout of the proposed ADPLL.

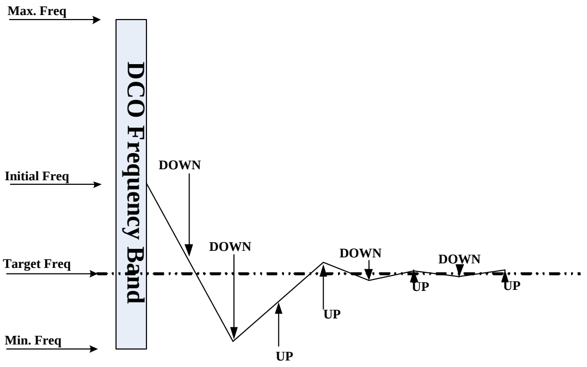

Fig. 9. The binary search for the target frequency.

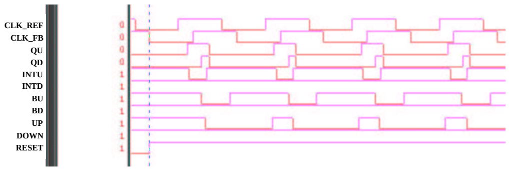

Fig. 12. The post-layout simulation result of PFD.

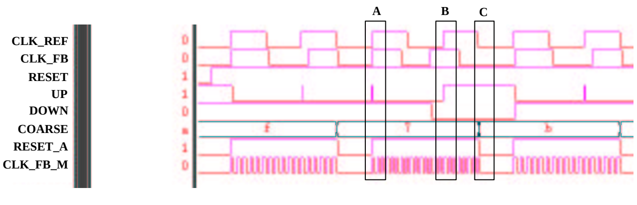

Fig. 10. The simulation result of the proposed CSP.

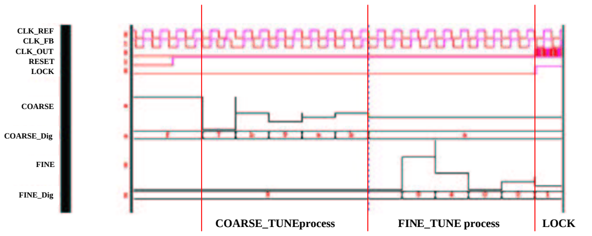

Fig. 13. The post-layout simulation result of the proposed ADPLL.