All-Digital Frequency Synthesizer Using a Flying Adder

Gang-Neng Sung, Szu-Chia Liao, Jian-Ming Huang, Yu-Cheng Lu, and Chua-Chin Wang, *Senior Member, IEEE*

***Abstract*—This brief presents an all-digital frequency synthesizer based on the flying adder (FA) architecture. The FA is a fascinating architecture for frequency synthesizer designs due to its simplicity and effectiveness. The FA-based frequency synthesizer can simply use a set of multiple phase reference signals to generate a desired frequency to achieve fast frequency switching. In the proposed work, the frequency synthesizer adopts an all-digital phase-locked loop to provide a steady reference signal for the FA. The proposed frequency synthesizer is implemented in a standard 0.18-μm CMOS cell-based technology, and the core area is 0.16 mm^2. The output frequency range is 39.38–226 MHz, and the peak-to-peak jitter is less than 130 ps.**

***Index Terms*—All-digital frequency synthesizer (ADFS), all-digital phase-locked loop (ADPLL), CMOS, flying adder (FA), low power.**

I. Introduction

THE FREQUENCY synthesizer is an essential component for many systems to generate a desired frequency for frequency conversion in RF transceivers or signal synchronization in system-on-a-chip (SOC) systems [1]. Therefore, a good frequency synthesizer design should meet the requirements of fast frequency switching and wide output frequency range. The most popular architecture of the existing solutions can be classified into phase-locked loop (PLL)-based frequency synthesizers and direct-digital frequency synthesizers (DDFSs). The PLL-based frequency synthesizers can easily generate a high output frequency, but they usually suffer from an inherent inability to simultaneously provide both fast frequency switching and high spectral purity [2]. In addition, the wide tunable output frequency range is still a challenging topic in the PLL designs. DDFSs adopt mathematical manipulations to directly synthesize the output frequency, where the implementation can be realized by either the memory-based phase-to-sine mapping or an algorithm-based phase-to-sine conversion [3]–[11]. Due to the absence of the feedback loop and voltage-controlled oscillator, DDFSs can accomplish the fast frequency switching and the wide output range more easily compared with the PLL-based solutions. DDFSs are also able to maintain the continuous phase during the process of frequency switching. The main drawback of the DDFS is that they need a high-resolution digital-to-analog converter (DAC) to convert the digital signals into an analog signal. Therefore, how to design a high-speed high-resolution DAC becomes another difficult topic.

The flying adder (FA) architecture was first introduced in [13] to overcome the problem associated with the conventional frequency synthesizers. By exploiting a set of n reference signals with the same frequency but a multiphase clock, i.e., with a fixed phase step of $360°/n$, the FA can generate the output frequency up to $n/2$ times of the reference frequency [12]. Notably, this architecture can generate a high output frequency using low-frequency reference signals, which is a favorably effective and cost-efficient scheme for the demand for high-speed clock generation in the modern SOCs. Prior works, [13]–[17], adopted the PLL to provide the reference signals. However, the traditional PLL design parameters of mixed-signal circuits seriously depend on CMOS technology, which results in a poor reusability, particularly in the trend of the scaling down of the CMOS technology. This brief presents an all-digital frequency synthesizer (ADFS) based on the FA architecture, where the reference signals is supplied by an all-digital PLL (ADPLL).

Manuscript received November 22, 2009; revised March 11, 2010; accepted May 6, 2010. Date of current version August 13, 2010. This work was supported in part by the Ministry of Economic Affairs, Taiwan, under Grants 98-EC-17-A-02-S2-0017,98-EC-17-A-07-S2-0010, 98-EC-17-A-01-S1-104, and 98-EC-17-A-19-S1-133, by the National Health Research Institutes under Grant NHRI-EX98-9732EI, and by the National Science Council under Grant NSC 96-2923-E-110-001-MY3. This work was recommended by Associate Editor M. Mar.

The authors are with the Department of Electrical Engineering, National Sun Yat-Sen University, Kaohsiung 80424, Taiwan (e-mail: ccwang@ee.nsysu.edu.tw).

Color versions of one or more of the figures in this paper are available online at http://ieeexplore.ieee.org.

Digital Object Identifier 10.1109/TCSII.2010.2056011

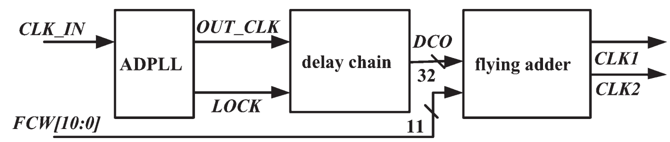

Fig. 1. Block diagram of the proposed ADFS.

II. ADFS Architecture

Fig. 1 shows the block diagram of the proposed ADFS architecture. The ADPLL [18]–[22] is employed to track the phase of the reference clock CLK_IN and generate a phase-locked clock signal OUT_CLK to the delay chain block. The delay chain produce 32 reference signals with an equal phase difference. The FA uses the multiple phase reference signals to synthesize the desired frequency according to the frequency control word FCW. The details of each function block in Fig. 1 are described as follows.

A. ADPLL

The detailed block diagram of the ADPLL is shown in Fig. 2, where the ADPLL is composed of a phase

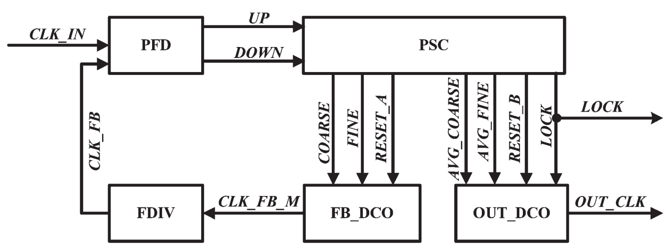

Fig. 2. Block diagram of the ADPLL.

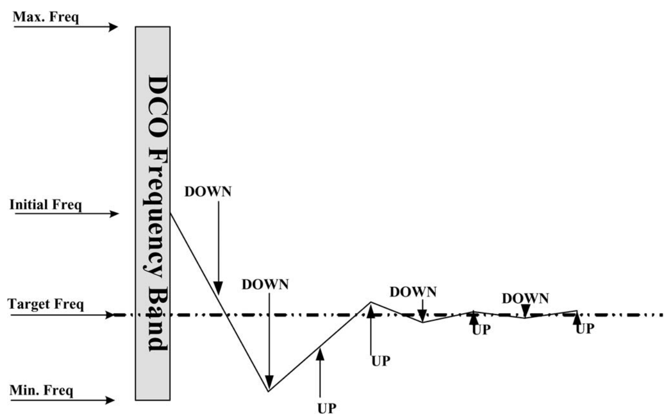

Fig. 3. Binary search for the target frequency [22].

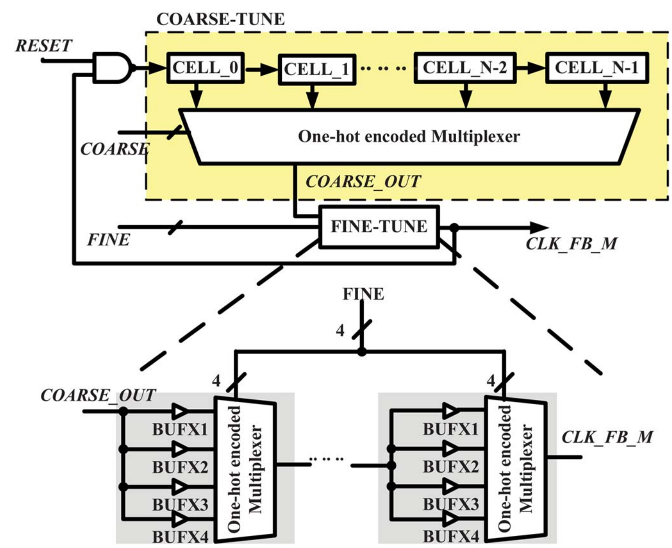

Fig. 4. Block diagram of the FB_DCO.

frequency detector (PFD), a frequency divider (FDIV), two digital-controlled oscillators (FB_DCO and OUT_DCO), and a phase searching controller (PSC). PFD detects the phase difference between the reference clock CLK_IN and the feedback clock CLK_FB. When CLK_FB lags CLK_IN, PFD generates a negative impulse on UP, whereas $DOWN$ remains at high to inform PSC to speed up FB_DCO. On the contrary, a negative impulse on $DOWN$ is generated to slow down FB_DCO if CLK_FB leads CLK_IN. The PSC generates two signals, i.e., COARSE and FINE, for FB_DCO to select an oscillating frequency of the multiplied clock signal in the feedback loop CLK_FB_M. The frequency of CLK_FB_M is divided by FDIV to generate the divided signal CLK_FB, which is sent back to PFD.

If the phase of the CLK_IN is clocked, a signal $LOCK$ is generated by PSC to indicate that the frequency is successfully locked. To further reduce the jitter caused by the dead zone of the PFD and the finite resolution of FB_DCO, the PSC 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 OUT_CLK. The PSC uses a binary search scheme, as shown in Fig. 3, to control the DCOs to generate the oscillating frequency with desired phases for the locking process.

Referring to Fig. 2, there are two DCOs, namely, FB_DCO and OUT_DCO. The FB_DCO is located at the feedback loop, which is employed to lock the phase of the reference signal, and the OUT_DCO generates the output signal when the phase is locked. Fig. 4 shows the block diagram of the FB_DCO. The control words $COARSE$ and $FINE$ select the appropriate delay cells of the COARSE-TUNE and FINE-TUNE blocks, respectively, to generate the desired oscillating frequency. The operation of the OUT_DCO is similar to the FB_DCO. In [23], the tristate buffers were used to switch the delay cells. However, using the tristate buffer might cause timing violations. The reason is that the "high-Z" state caused by the tristate buffers will be fed back to feedback loop of the ADPLL when the control signals to switch tristate buffers do not arrive at the same time. As a result, it will cause the timing hazard during the locking process. Instead, we adopt multiplexers to switch the delay cells to resolve this problem. Additionally, a rising edge of the input signal of the multiplexer is preset to wait for COARSE and FINE signals and then switch the delay line such that the timing violation can be avoided.

B. Delay Chain

If the ADPLL successfully locks the phase of the reference clock CLK_IN, the output of the ADPLL is passed to the delay chain to generate the multiple phase signals for the FA. The delay chain is composed of a series of the buffer cells, as shown in the Fig. 5(a). Fig. 5(b) depicts the resultant reference signals, where $\triangle$ is the propagation delay of each buffer. A larger number of the resultant reference signals give a finer frequency resolution. The number of the buffers is 32 in our design. For instance, if we need 120 MHz as the ADPLL output frequency, $\triangle$ should be 0.2625 ns. Therefore, the delay chain in the proposed design preferably consists of 32 buffers with a 0.2625-ns propagation delay.

C. FA

The FA architecture is adopted in the proposed frequency synthesizer. Fig. 6 shows the block diagram of the FA [14], where PART_A and PART_B are similar circuits, where the major difference between these two function blocks is the size of the adders. Referring to PART_B, when $CLK2$ is high, the summation of ADD_1 and reg_1 is stored in reg_2, and the value previously stored in reg_2 is passed to a 32-to-1 multiplexer, i.e., MUX1, to select one of the delay chain output signals, i.e., DCO[31:0], as the input signal of the

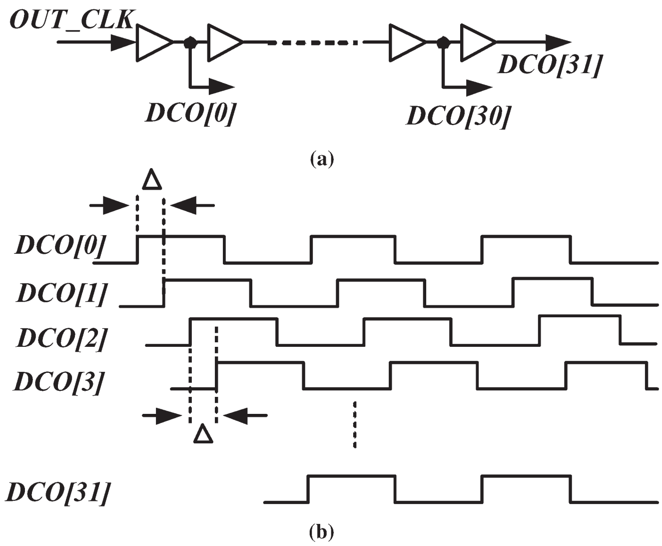

Fig. 5. (a) Schematic of the delay chain. (b) Resultant multiple phase reference signals.

2-to-1 multiplexer, i.e., MUX3. The accumulation step in PART_B is determined by the frequency control word $FCW1$. The operation of PART_A is similar to PART_B, except that reg_4 is triggered by $CLK1$. Therefore, MUX3 always selects the signal generated at the last half-clock cycle, which can avoid the glitch caused by the transition of the input signal propagating through the MUXs. Eliminating the glitch can prevent the D-type flip-flop (DFF) from being triggered by false signals.

The desired frequency is synthesized by triggering the DFF with the signal selected from DCO. The relation between the frequency control word FCW and the desired frequency f_{out} can be expressed as follows:

$$\frac{1}{f_{\text{out}}} = \frac{T_{DCO}}{N} \times FCW \tag{1}$$

where T_{DCO} is the period of the DCO signals, and N is the number of the DCO signals [24]. For example, to obtain an output frequency of 150 MHz given that the number and the frequency of the DCO signals are 32 and 150 MHz, respectively, the required FCW is around $25.371 = (011001.01011)_2$. The longer length of the FCW's decimal part results in a finer frequency resolution. The length of the FCW in the proposed frequency synthesizer is 11 bits, where FCW[10:5] and FCW[4:0] represent the integer and decimal parts, respectively.

III. Implementation and Measurement

The proposed ADFS is carried out by the standard 0.18-μm CMOS technology to verify the performance. All of the process corners, namely, $[0\ ^\circ\text{C}, +100\ ^\circ\text{C}]$, and (SS, TT, FF) models, are simulated. Fig. 7 shows the waveform of the synthesized frequency. Referring to Fig. 7(a), when the phase of CLK_IN is locked, the ADPLL generates an 80-MHz clock signal OUT_CLK, and the synthesized frequency $CLK1$ is changed from 39.38 to 170 MHz according to the change of FCW,

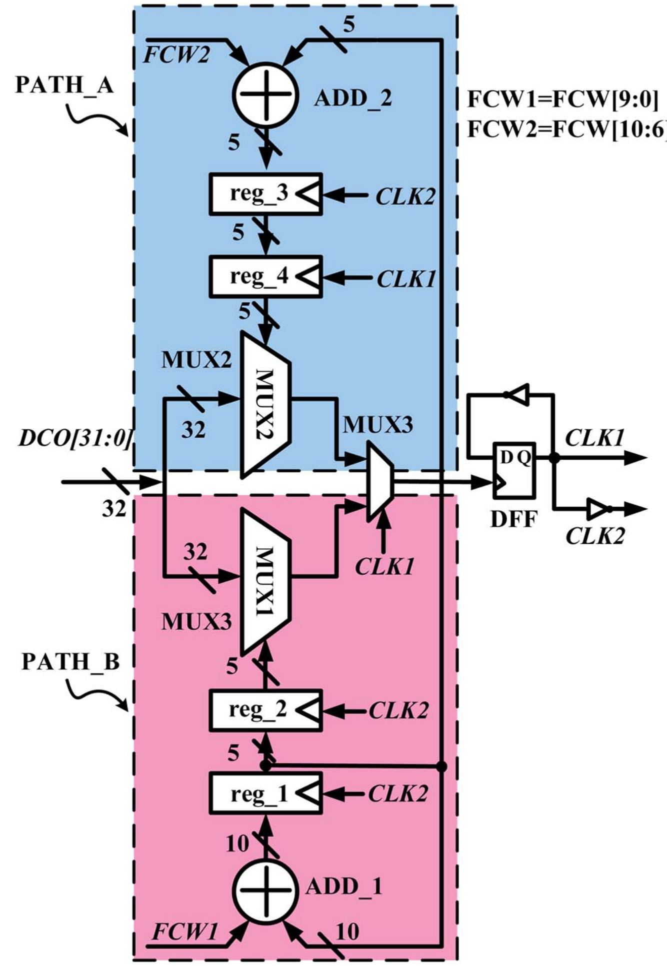

Fig. 6. FA.

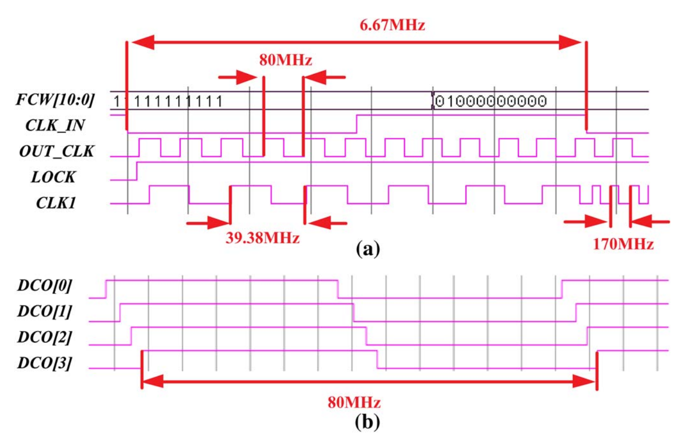

Fig. 7. Waveform of the generated output signal. (a) Output frequency changed from 39.38 to 170 MHz. (b) Multiple phase reference signals generated by the delay chain.

where the frequency switching latency is two $CLK1$ cycles. Fig. 7(b) shows a part of the multiple phase reference signals produced by the delay chain.

Fig. 8 shows the die photo of the proposed frequency synthesizer. Fig. 9 reveals that, when the frequency of the ADPLL clock signal is locked, where the locked frequency is 187.5 MHz, the third harmonic frequency is 570 MHz. At the same locked frequency, the voltage and time of the eye

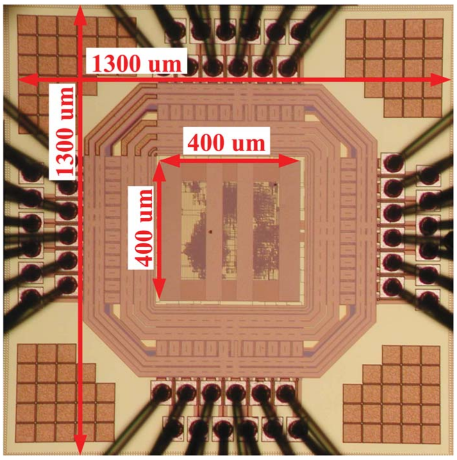

Fig. 8. Die photo of the proposed frequency synthesizer.

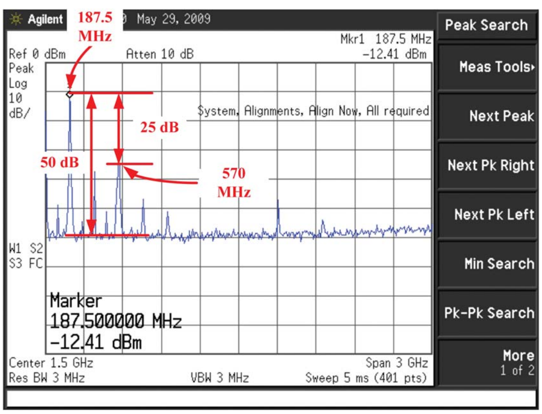

Fig. 9. Waveform of the output frequency locked on 187.5 MHz.

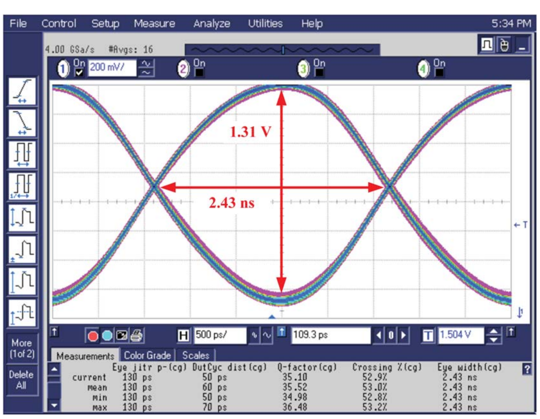

Fig. 10. Eye diagram for the output synthesized frequency, where the locked frequency is 187.5 MHz.

diagram depicted in Fig. 10 are 1.31 V and 2.43 ns, respectively. Referring to Fig. 11, which is a measurement result on Agilent SOC 93000, $CLK1$ varies from 80 to 40 MHz when FCW is updated. It takes only four OUT_CLK cycles to relock. The performance comparisons of our design and several prior works are summarized in Table I. The proposed ADFS attains the edge of low power and short lock time (cycles).

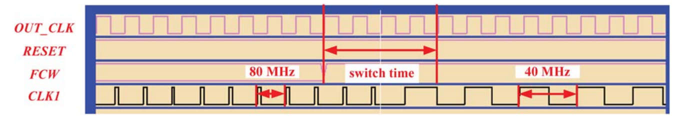

Fig. 11. Waveform of the output frequency switching from 80 to 40 MHz.

TABLE I
PERFORMANCE COMPARISONS WITH SEVERAL PRIOR WORKS

ADPLL	[25]	[26]	[18]	[27]	Ours
Process (μm)	0.18	0.18	0.18	0.18	0.18
Year	2005	2008	2009	2009	2009
Maximum Frequency (MHz)	1500	725	250	560	226
Minimum Frequency (MHz)	520	70	87	53	39.38
Supply Voltage (V)	1.8	1.8	0.9	N/A	1.8
Maximum Lock Time (cycles)	$\leq$96	$\leq$72	$\leq$72	N/A	$\leq$4
Jitter (ps)	76	N/A	152	51	130
Power Dissipation (mW)	26.8	27	5.4	25.2	3.6
Area (mm^2)	0.27	0.174	0.76	0.14	0.16

IV. CONCLUSION

The FA architecture was developed to resolve challenges encountered in the design of the conventional PLL-based frequency synthesizer. Although the FA architecture only needs a simple PLL to provide multiphased signals, the integration of the mixed-signal circuits still needs efforts on both the design and the simulation stages. In addition, analog circuits are much more sensitive to the variation of the technology compared with the digital circuits. In this brief, we propose an ADFS based on the FA architecture, where the mixed-signal PLL is replaced with a low-power ADPLL. Moreover, the glitch hazard on both the FA and the ADPLL is eliminated to ensure the stability of the frequency synthesizer.

ACKNOWLEDGMENT

The authors would like to thank the Chip Implementation Center, National Applied Research Laboratories, Taiwan, for their thoughtful chip fabrication service.

REFERENCES

[1] R. Best, *Phase-Locked Loops: Design, Simulation, and Application*. New York: McGraw-Hill, 2003.
[2] A. Yamagishi, M. Ishikawa, T. Tsukahara, and S. Date, "A 2-V, 2-GHz low-power direct digital frequency synthesizer chip-set for wireless communication," *IEEE J. Solid-State Circuits*, vol. 33, no. 2, pp. 210–217, Feb. 1998.
[3] J. Vankka and K. Halonen, *Direct Digital Synthesizers: Theory, Design and Applications*. New York: Springer-Verlag, 2006.
[4] A. M. Sodagar and G. R. Lahihi, "A novel architecture for ROM-less sin-output direct digital frequency synthesizers by using the 2nd-order parabolic approximation," in *Proc. IEEE/EIA Int. Freq. Control Symp. Exhib.*, Jun. 2000, pp. 284–289.

[5] A. Ashrafi and R. Adhami, "A 13-bit resolution ROM-less direct digital frequency synthesizer based on a trigonometric quadruple angle formula," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 9, pp. 1096–1098, Sep. 2005.
[6] H. Jafari, A. Ayatollahi, and S. Mirzakuchaki, "A low power, high SFDR, ROM-less direct digital frequency synthesizer," in *Proc. IEEE Conf. Electron Devices Solid-State Circuits*, Dec. 2005, pp. 829–832.
[7] C.-C. Wang, J.-M. Huang, Y.-L. Tseng, W.-J. Lin, and R. Hu, "Phase-adjustable pipelining ROM-less direct digital frequency synthesizer with a 41.66-MHz output frequency," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 53, no. 10, pp. 1143–1147, Oct. 2006.
[8] A. Ashrafi and R. Adhami, "An optimized direct digital frequency synthesizer based on even fourth order polynomial interpolation," in *Proc. 3eight IEEE Southeastern Symp. Syst. Theory*, Mar. 2006, pp. 109–113.
[9] D. De Caro and A. G. M. Strollo, "High-performance direct digital frequency synthesizers in 0.25 μm CMOS using dual-slope approximation," *IEEE J. Solid-State Circuits*, vol. 40, no. 11, pp. 2220–2227, Nov. 2005.
[10] B. Izouggaghen, A. Khouas, and Y. Savaria, "Spurs modeling in direct digital period synthesizers related to phase accumulator truncation," in *Proc. IEEE ISCAS*, May 2004, vol. 3, pp. 389–392.
[11] B. Pontikakis, H.-T. Bui, F.-R. Boyer, and Y. Savaria, "Precise free-running period synthesizer (FRPS) with process and temperature compensation," in *Proc. IEEE MWSCAS*, Aug. 2007, pp. 1118–1121.
[12] P. Sotiriadis, "Timing and spectral properties of the flying adder frequency synthesizers," in *Proc. IEEE Int. Freq. Control Symp.*, Jul. 2009, pp. 788–792.
[13] H. Mair and L. Xiu, "An architecture of high-performance frequency and phase synthesis," *IEEE J. Solid-State Circuits*, vol. 35, no. 6, pp. 835–846, Jun. 2000.
[14] L. Xiu and Z. You, "A flying-adder architecture of frequency and phase synthesis with scalability," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 10, no. 5, pp. 637–649, Oct. 2002.
[15] L. Xiu and Z. You, "New frequency synthesis method based on flying-adder architecture," *IEEE Trans. Circuits Syst. II, Analog Digit. Signal Process.*, vol. 50, no. 3, pp. 130–134, Mar. 2003.
[16] P. Sotiriadis, "Intrinsic jitter of flying-adder frequency synthesizers," in *Proc. IEEE Int. Sarnoff Symp.*, 2009, pp. 1–4.
[17] L. Xiu, "A 'flying-adder' on-chip frequency generator for complex SoC environment," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 54, no. 12, pp. 1067–1071, Dec. 2007.
[18] Y.-M. Chung and C.-L. Wei, "An all-digital phase-locked loop for digital power management integrated chips," in *Proc. IEEE Int. Circuits Syst. Symp.*, May 2009, pp. 2413–2416.
[19] 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 Syst. II, Analog Digit. Signal Process.*, vol. 46, no. 7, pp. 945–950, Jul. 1999.
[20] T. Olsson and P. Nilsson, "An all-digital PLL clock multiplier," in *Proc. IEEE Asia-Pacific Conf. ASICs*, Taipei, Taiwan, 2002, pp. 275–278.
[21] T. Olsson and P. Nilsson, "A digitally controlled PLL for SoC applications," *IEEE J. Solid-State Circuits*, vol. 39, no. 5, pp. 751–760, May 2004.
[22] C.-C. Wang, C.-C. Huang, and S.-L. Tseng, "A low-power ADPLL using feedback DCO quarterly disabled in time domain," *Microelectron. J.*, vol. 39, no. 5, pp. 832–840, May 2008.
[23] C.-C. Chung 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.
[24] L. Xiu, "The concept of time-average-frequency and mathematical analysis of flying-adder frequency synthesis architecture," *IEEE Circuits Syst. Mag.*, vol. 8, no. 3, pp. 27–51, Sep. 2008.
[25] K.-J. Lee, S.-H. Jung, Y.-J. Kim, C. Kim, and S. Kim, "A digitally controlled oscillator for low jitter all digital phase locked loops," in *Proc. IEEE Asian Solid-State Circuits Conf.*, Nov. 2005, pp. 365–368.
[26] H.-J. Hsu, C.-C. Tu, and S.-Y. Huang, "A high-resolution all-digital phase-locked loop with its application to built-in speed grading for memory," in *Proc. Int Symp. VLSI Des., Autom. Test*, Apr. 2008, pp. 267–270.
[27] H.-J. Hsu and S.-Y. Huang, "A low-jitter all-digital phase-locked loop using a suppressive digital loop filter," in *Proc. VLSI-DAT*, Apr. 2009, pp. 158–161.