

Direct All-Digital Frequency Synthesis Techniques, Spurs Suppression, and Deterministic Jitter Correction

Paul P. Sotiriadis, *Senior Member, IEEE*, and Kostas Galanopoulos, *Student Member, IEEE*

Abstract—Direct all-digital frequency synthesizers are favored by modern nanoscale CMOS technologies but suffer from strong frequency spurs and timing irregularities. To counter these drawbacks various jitter-correction and spurs-suppression techniques have been proposed. This paper presents a comprehensive literature review and a comparative study of such techniques, applied to popular direct all-digital frequency synthesis cores, identifying their strengths and weaknesses.

Index Terms—Clock generation, digital-to-frequency converter, direct digital period synthesis, direct digital synthesis, flying adder, frequency synthesis, jitter, phase accumulator, frequency spurs.

I. INTRODUCTION

EFFORTS TO develop direct all-digital frequency synthesizers (DADFS) can be traced at least three decades back [1]–[3]. With increasing integration density, digital circuits become faster, smaller and more energy efficient. In contrast, many analog circuit blocks became more and more challenging to design, due to lower power supply voltage and the trend to co-integrate them with big digital engines in digital-oriented integrated circuit technologies. Also, digital integrated circuit design is supported (at least partially) by automated design and layout tools, allowing for short specs-to-product time and easy technology migration.

The cores of DADFS are finite state machines (FSM) driven by a clock signal which can be single- or multi-phased. This means that their output pulses begin and end at the rising (and/or falling) edges of the reference clock, f_{clk} , or at those of its phases, if it is multi-phased. This implies that the only *perfect* periodic output waveforms that can be generated have frequencies f_{clk}/N (or $f_{clk} \cdot M/N$, where M is the number of phases), i.e., when the DADFS behaves like an integer frequency divider (single- or multi-phased).

For all other synthesized frequencies (in the sense of time-average rates of pulses) the output waveforms are irregular, i.e., there is a (strong) deterministic timing jitter and the spectra contain (strong) frequency spurs. This inherent imperfection of

DADFS cores creates the need for additional circuitry to suppress the frequency spurs and/or to correct the timing jitter of the output, to the extent possible.

This work presents a comprehensive and comparative review of the techniques that have been proposed to correct the output signals of DADFS cores.

The correction techniques are discussed in conjunction with popular DADFS cores classified into two groups: the digital-to-frequency converters (DFC) and the digital-to-period converters (DPC); where the first ones synthesize signals whose time average (TA) *frequency* is proportional to a programmable number, which is referred to as the frequency control word (FCW), and the second ones synthesize signals whose TA *period* is proportional to a programmable number.

Section II introduces popular DADFS cores on which the discussion of the spurs suppression and jitter correction methods is based. Section III presents the class of *retiming techniques* for reducing the jitter. Section IV discusses the *cleanup-PLL* approach for frequency-domain filtering of the DADFS' output. Section V presents the *dithering methods* which are the only purely digital ones. Section VI provides a comparative discussion of the aforementioned methods.

II. DIRECT ALL-DIGITAL FREQUENCY SYNTHESIZERS

This section discusses some representative DADFS of the DFC and DPC classes, along with their basic concepts, frequency ranges and typical spectra. Many variations and combinations of these cores appear in the literature.

The first DADFS we discuss (and probably the most popular one) is the pulse direct digital synthesizer (PDDS), which is a DFC and is based on a phase accumulator similar to that used in standard direct digital synthesizers (DDS) [1]–[3]. The DPC class is represented here by the flying adder (FA) [5]–[11] and the fractional $N/N + 1$ divider (which has recently attracted attention as an independent DPC type DADFS [39]).

A. Pulse Direct-Digital Synthesizer (PDDS)

The phase accumulator shown in Fig. 1, also called PDDS, [3] is the most commonly used DFC core. It consists of an n -bit adder with overflow output and an n -bit register. The adder increases the value of the register by w (modulo 2^n) at every rising-edge of the clock (assuming a rising-edge triggered register). Parameter w is the frequency control word (FCW) of the PDDS.

At the k -th clock period, the register's value is $x_k = (k \cdot w) \bmod 2^n$ (assuming zero initial value of the register)

Manuscript received September 05, 2011; revised December 08, 2011; accepted March 03, 2012. Date of current version May 09, 2012. This work was supported in part by the Synergy Microwave Corporation, NJ, USA. This paper was recommended by Associate Editor A. Tasic.

The authors are with the Department of Electrical and Computer Engineering, National Technical University of Athens, Athens 157 80, Greece (e-mail: pps@ieee.org; galanopu@ieee.org).

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/TCSI.2012.2191875

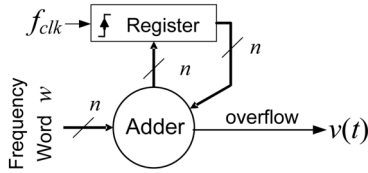


Fig. 1. Pulse DDS (PDDS) with overflow output.

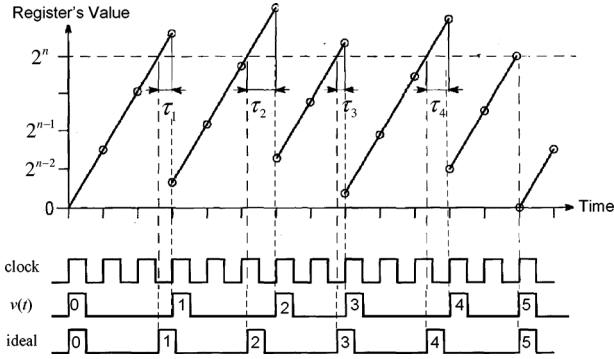


Fig. 2. Typical waveform of PDDS [modified figure from [22]].

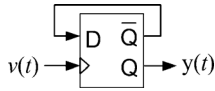


Fig. 3. D-FF converting overflow signal to squarewave.

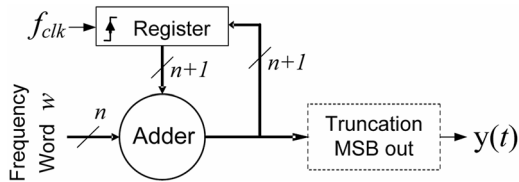


Fig. 4. Pulse DDS (PDDS) with MSB output.

and the output is $v_k = 1$ when there is an overflow (i.e., when $(k \cdot w) \text{div} 2^n > ((k-1) \cdot w) \text{div} 2^n$) and $v_k = 0$ otherwise.

The main difference of PDDS to the standard direct digital synthesizer (DDS) [1] is that it does not use a look up table (LUT) nor a digital-to-analog converter (DAC). Both PDDS and DDS have instantaneous frequency hopping, but PDDS is purely digital, it has significantly lower power consumption, higher maximum operating frequency and it requires much less chip area. On the other hand PDDS suffers from frequency spurs and timing irregularities, which can be minimal in DDS.

A typical output waveform of PDDS, $v(t)$, is shown in Fig. 2 along with the corresponding ideal one (of the same average frequency and duty cycle) and the values of the register. Notice the time offsets $\tau_j \geq 0, j = 1, 2, 3, \dots$, between the rising edges of the pulses of the ideal waveform and those of the output waveform. The first ones appear when the continuous linear

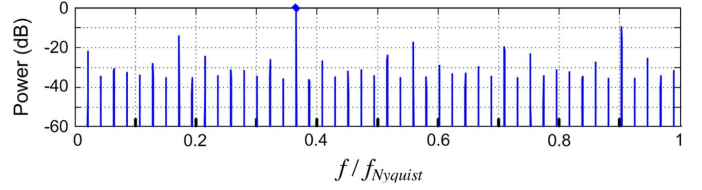


Fig. 5. Typical spectrum of the MSB-output PDDS in Fig. 4 [MATLAB].

phase-segments cross 2^n whereas the second ones appear either at the same time (if $\tau_j = 0$) or at the first rising edge of the clock following.¹

For most values of the FCW, $0 \leq w < 2^n$, the output is *not* a regular periodic pulse sequence, that is, the time-distance between consecutive pulses is not fixed but instead a periodic sequence, of period $2^n \cdot T_{\text{clk}} / \text{gcd}(w, 2^n)$. The TA rate of output pulses is $(w/2^n) \cdot f_{\text{clk}}$. Note that the proportionality of f_{av} to w makes PDDS a DFC.

The overflow pulse sequence $v(t)$ can be converted to a squarewave using a D-flip-flop with negative feedback as shown in Fig. 3.

On every pulse of $v(t)$, the value (0 or 1) of the output $y(t)$ is inverted. This D-flip-flop topology behaves as a frequency divider by 2 with its output having a duty cycle of *about* 50%, resulting in an output TA pulse rate of

$$f_{\text{av}} = \frac{w}{2^{n+1}} \cdot f_{\text{clk}} \text{ with } 0 \leq w \leq 2^n. \quad (1)$$

Alternatively, a larger accumulator (adder and register) of $n+1$ bits can be used (Fig. 4) with the MSB of the register being the output. The TA pulse rate here is also given by (1), where for $w = 2^n$ the PDDS is a divider by 2. Note that the variant of PDDS in Fig. 1 combined with the D-FF in Fig. 3, and PDDS in Fig. 4 have the same output waveforms.

These basic forms of PDDS can generate TA frequencies within $[0, f_{\text{clk}}/2]$. The upper bound of $f_{\text{clk}}/2$ is due to the division by 2 used to generate a $\sim 50\%$ duty cycle squarewave output. At an extra cost of more complex hardware we can double the frequency range to f_{clk} by operating on both rising and falling edges of the clock.

The *fundamental* period T_y of $y(t)$ in Fig. 4 can be expressed as $T_y = 2^{n+1} \cdot T_{\text{clk}} / \text{gcd}(w, 2^{n+1})$, where $T_{\text{clk}} = 1/f_{\text{clk}}$, and it may be *significantly larger* than $1/f_{\text{av}}$. A period T_y contains $w / \text{gcd}(w, 2^{n+1})$ output pulses. Also, the frequency of the dominant output frequency component is f_{av} , which is a harmonic of the fundamental output frequency $f_{\text{fun}} = 1/T_y = \text{gcd}(w, 2^{n+1}) \cdot f_{\text{clk}} / 2^{n+1}$ as every other frequency component of the output is.

The spectrum of the output $y(t)$ may have many strong spurs [54] as is indicated in Fig. 5 (for $w = 376, n = 10$), in agreement with the irregularity of waveform $v(t)$, present for most values of w , e.g., in Fig. 2. We have set $f_{\text{Nyquist}} = f_{\text{clk}}/2$.

There are many variations of the basic PDDS at both the functional and the architectural level. One example is to use a modulo- Q phase accumulator, where Q is not necessarily a power of 2. If in addition Q is programmable, a much larger set of TA frequencies $f_{\text{av}} = (w/Q) \cdot f_{\text{clk}}$ is achievable [4].

¹Resulting in absolute timing jitter less than or equal to half the clock period.

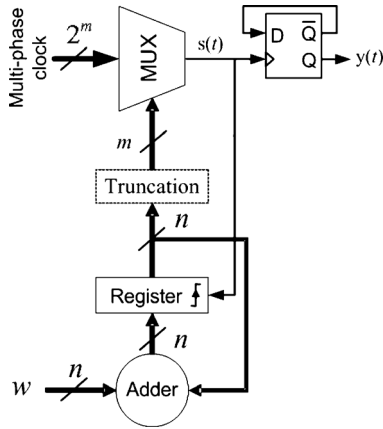


Fig. 6. The basic flying adder is an example of DPC type DADFS.

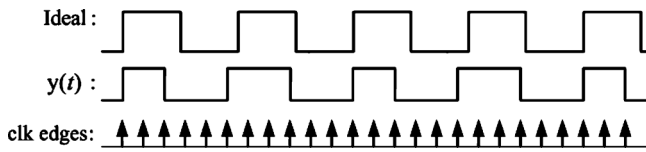


Fig. 7. Ideal output, real output $y(t)$, and combined edges of the multi-phase clock of the flying adder DPC [MATLAB].

B. Flying Adder Synthesizer (FA)

The flying adder (FA) [5]–[11] and similar architectures [42]–[51] represent a popular class of DPC type DADFS (also called direct digital period synthesizers [10]) which is based on multi-phase frequency division. Its basic structure is shown in Fig. 6. Note that although FA's structure includes a phase accumulator core as PDDS does, its operation is quite different to that of PDDS, [8], [9].

The input clock f_{clk} is multi-phased with 2^m equally spaced phases. The m high-order bits of the register control a multiplexer (MUX) which selects one of these phases as its output $s(t)$. It is important to note that the phase accumulator is clocked by $s(t)$, and *not* the clock signal, resulting in a DPC rather than a DFC operation. The squarewave output of the circuit $y(t)$ is generated by $s(t)$ using a D-flip-flop divider modulo-2 as is done in PDDS, Fig. 3.

For every rising edge of $s(t)$ (again, we assume rising edge triggered register), the value of $y(t)$ is inverted and the phase accumulator's value is increased by w modulo 2^n . The practically useful operation of the FA is when $w \geq 2^{n-m}$ in which case, every rising edge of $s(t)$ results in a change in the selection of the clock phase by the MUX. Moreover, the larger the w is the longer the time interval is between consecutive rising edges of $s(t)$.

As seen in Fig. 7, FA suffers from timing irregularity similar to that of PDDS, [9], for most values of w . The maximum absolute timing jitter of this technique is less than or equal to one half of the clock period divided by the number of clock phases (2^m). The spectrum of FA typically has a large number of strong spurs as shown in Fig. 8 similarly to that of PDDS [11].

The TA output frequency is given by expression (2) below for the full range of values of w . The range of values of f_{av} is $[f_{\text{clk}}/2, 2^{m-1}f_{\text{clk}}]$. When $w \geq 2^{n-m}$, f_{av} is the frequency of the dominant frequency component in the spectrum. This may

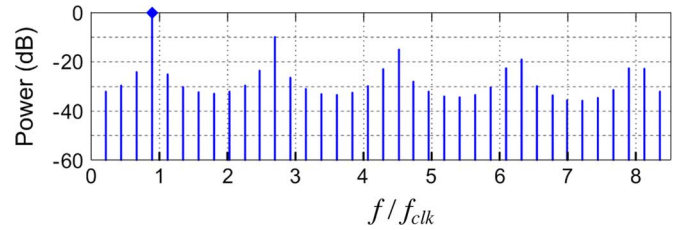


Fig. 8. Typical spectrum of the flying adder (4-phase clock) [MATLAB].

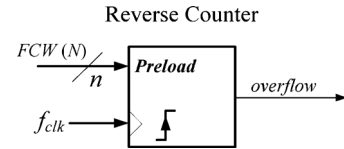


Fig. 9. IND implemented as a reverse binary counter.

not be true for some values of $w < 2^{n-m}$ for which the output has very irregular timing [9].

$$f_{\text{av}} = f_{\text{clk}} \cdot \begin{cases} 1/2, & \text{if } w = 0 \\ \frac{2^{n-1}}{2^n - (2^m - 1)w}, & \text{if } 0 < w < 2^{n-m} \\ 2^{n-1}/w, & \text{if } 2^{n-m} \leq w \end{cases} \quad (2)$$

Note that despite the division by 2 at the output, the FA can generate TA frequencies much higher than f_{clk} assuming it is driven by a multi-phase clock of sufficiently large number of phases. Of course, the requirement of a multi-phase clock is also a disadvantage of the FA since it typically requires a PLL (e.g., with ring oscillator) or a DLL to generate it.

C. Integer N and fractional $N/N + 1$ Dividers

Although integer- N divider (IND) and fractional- N divider (FND) are simple structures typically used as building blocks in frequency synthesizers [28], [31], [33], they can also be used independently as DADFS [39]. We discuss IND first as an introduction to the structure and limitations of FND. Note that when FND is used in the feedback path of fractional- N PLLs, N usually takes large values, e.g., 100 or higher [28], [31]. In contrast, when FND is used as an independent DADFS, N takes small values like 2, 3, etc., [39].

An IND of modulo N generates a "perfect" (50% duty cycle regular) periodic waveform of period N times that of the clock, where N can be programmable and play the role of the FCW, i.e., $f_{\text{IND}} = f_{\text{clk}}/N$.

One of the many ways to implement an IND is as a reverse binary counter with overflow output, shown in Fig. 9. At every rising clock edge the value of the counter decreases by 1. The overflow pulse triggers the counter to reload with the value of the FCW, N . The output pulse rate is f_{clk}/N .

However, the overflow pulses are only one clock pulse long. To convert them to 50% duty cycle pulses we need to include a D-flip-flop divider by 2 as in Fig. 3, or alternatively, use a $n + 1$ bits counter with its MSB as the final output (FCW remains n bits wide). Either way the generated frequency is divided by 2. Again to fix this we can trigger the circuit on both the rising and the falling edges of the clock.

The spectrum of an IND divider is shown in Fig. 10. Graph (a) shows that of the 50% duty cycle output, while graph (b) shows that of the overflow. Note that the short length of the

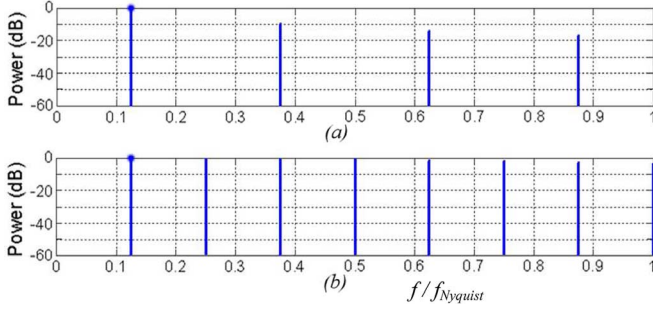


Fig. 10. Spectrum of: (a) Regular periodic 50% duty cycle squarewave. (b) Overflow-pulse sequence repeating the pattern [1, 0, 0, 0, 0, 0, 0, 0], [MATLAB].

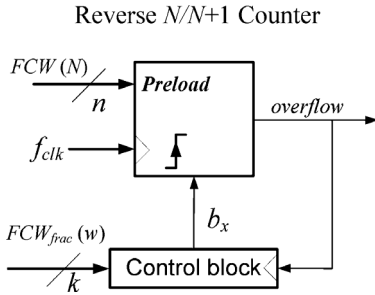


Fig. 11. Basic structure of a FND.

overflow pulses results in a spectrum with strong harmonics at frequencies $(k/N) \cdot f_{clk}$, $k = 1, 2, 3, \dots$ which can be used in places where *comb-like* spectra are required.

In the case of FND, Fig. 11, the input frequency f_{clk} is divided by $N + b_x$, where $b_x \in \{0, 1\}$, [33], [39]. For every overflow pulse (generated every N or $N + 1$ clock pulses) the control block generates the next value of b_x so that the TA rate of the overflow pulses is given by (3) where w is the k -bit long fractional FCW.

$$f_{av} = f_{CLK}/(N + w/2^k) \text{ with } 0 \leq w < 2^k. \quad (3)$$

As in PDDS and IND, the output can be converted to about $\sim 50\%$ duty cycle squarewave via a D-flip-flop counter modulo 2 at the output and the original frequency range can be recovered by operating the circuits at both the rising and the falling edges of the clock.

Consider the following example of a dual-edge triggered FND (with 50% duty circle converted output) with $N = 3$ and b_x being the signal formed by repeating the pattern [0 0 0 1 1 0 1 0]. The TA value of b_x is $3/8$ so the division ratio is $3 + 3/8 = 27/8 = 3.375$ resulting in frequency $f_{clk}/3.375$.

The output spectrum is shown in Fig. 12 and has many strong spurs due to the timing irregularities of the output signal. The absolute jitter is less than or equal to one half of the clock cycle. In general, the spurs depend on the way the b_x signal is generated and so the selection of the algorithm of the control unit is important. Section V-B illustrates how random-dithering methods can be used to suppress the spurs.

Finally, generalizing the basic FND, b_x can be a signed multi-bit value so that the control block can select from a larger set of possible division ratios.

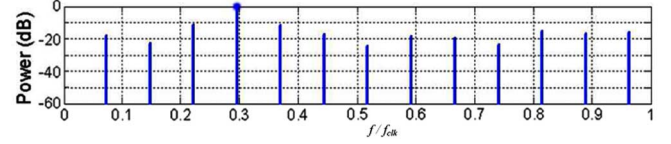


Fig. 12. FND's spectrum of TA frequency $f_{clk}/3.375$ [MATLAB].

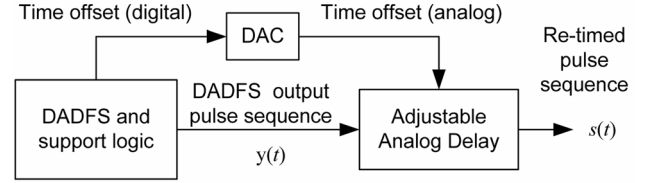


Fig. 13. Typical pulse retiming using analog adjustable-delay element(s).

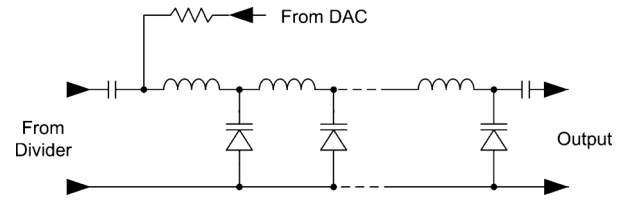


Fig. 14. Transmission line approximation circuit for implementing the delay element for pulse retiming [12], [13]. [Modified figure from [12]].

III. RETIMING TECHNIQUES FOR SPUR SUPPRESSION AND JITTER CORRECTION

To reduce the timing irregularities and suppress the spurious spectral components of DADFS one can delay the output pulses, each one by a certain amount of time (retiming), so that the resulting waveform is an ideal periodic squarewave. To achieve a (theoretically) perfect pulse retiming, one has to use analog circuit blocs. Of course, it is desirable to keep analog blocks minimal otherwise the advantage of simple DADFS architectures is lost.

The following sections present a collection of retiming techniques. Note that although most of them have been proposed for PDDS or FA cores, in principle they can be used with any other types of DADFS cores as well.

A. Retiming Using an Analog Delay Element

Consider the output $v(t)$ of the PDDS shown in Fig. 2 and suppose that it is passed through an adjustable-delay element so that pulse $j = 1, 2, 3, \dots$ is delayed by $T_{clk} - \tau_j$, where τ_j is the time elapsed between the rising edge of the j -th pulse of the ideal signal (appearing first) and that of the j -th pulse of the output as shown in Fig. 2. In the resulting pulse sequence, the j -th pulse overlaps with the $(j + 1)$ -th pulse of the ideal signal.

A high level architecture using analog adjustable-delay element(s) and achieving this retiming is shown in Fig. 13. Note that the offset time $T_{clk} - \tau_j$ can be easily calculated by an additional digital (logic) block.

One way to implement the adjustable delay element is by using an adjustable transmission line, or more realistically, a lumped approximation of it as shown in Fig. 14, [12], [13].

The main difficulty of this approach is to map the desirable delay values to the delay control voltages generated by the DAC.

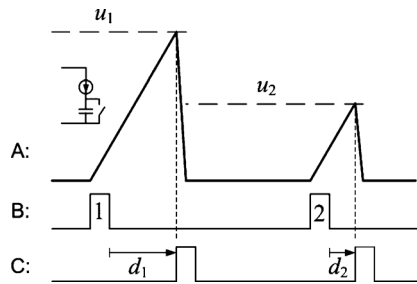


Fig. 15. (A) Capacitor voltage. (B) DADFS output. (C) Delayed pulses of (B).

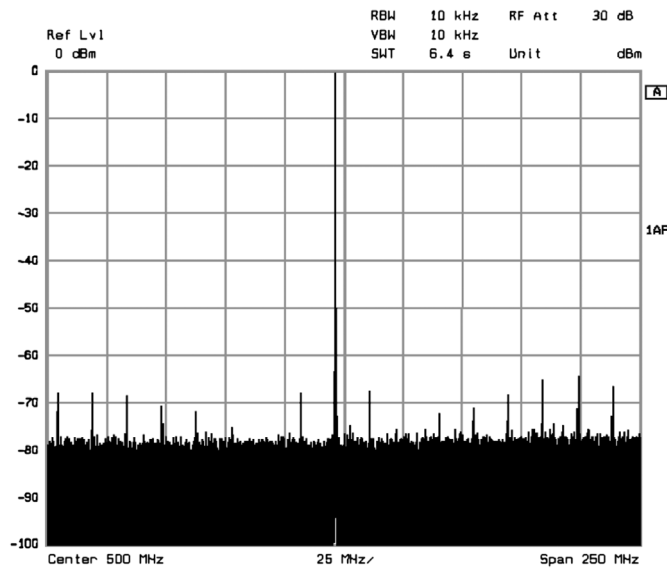


Fig. 16. Spectrum of retiming using analog delay elements technique [65-nm CMOS integrated circuit implementation] [18].

Temperature, power supply voltage, and process variations require compensation.

The analog adjustable transmission line can be replaced by a digital delay line *with analog delay control*, like a chain of current-starved inverters [52]. This reduces the analog elements in the circuit but suffers more by temperature, process and other variations.

Another way for converting digital value $T_{clk} - \tau_j$ to delay is to start charging a capacitor from zero (or other fixed) initial voltage, using a fixed current, when the rising edge of the j -th pulse arrives; and generate an output pulse when the capacitor's voltage reaches a certain threshold corresponding to $T_{clk} - \tau_j$.

The approach is illustrated in Fig. 15 where the capacitor starts charging up when the *first* pulse of DADFS arrives; the threshold voltage u_1 is chosen so that it takes time $d_1 = T_{clk} - \tau_1$ for the capacitor's voltage to reach it; when the threshold is crossed, a mono-stable circuit is triggered and the capacitor is discharged, and so on.

Instead of adjusting the threshold voltage to get the desirable delay one can adjust the charging current (current switching circuits, popular in DACs, can be used). This concept has been used in [14]–[18]. The output spectrum of an implementation of this technique is shown in Fig. 16, [18].

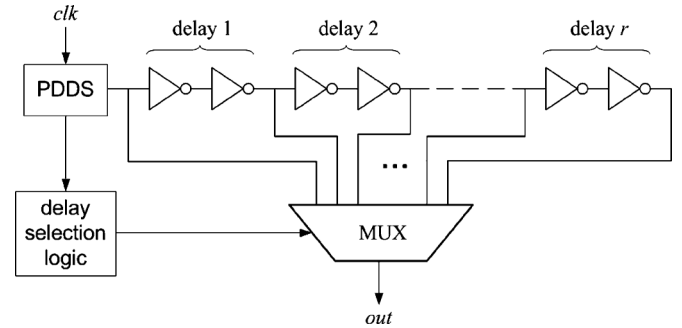


Fig. 17. Digital delay-line for pulse retiming (no phase-locking).

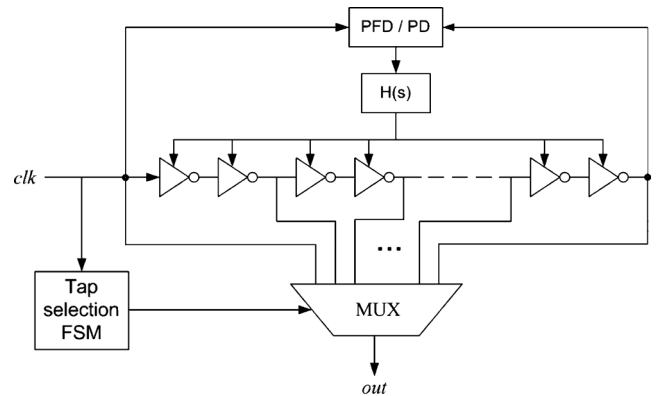


Fig. 18. Phase-locked digital delay-line for pulse retiming, [21]–[27].

B. Retiming Using a Digital Delay Line

One can replace the adjustable analog delay element in Fig. 13 along with the DAC, with a digital delay line using fixed-delay elements, as in Fig. 17, [19], [20].

Since the delay line has only a finite number of taps, certain quantization of the delay values is imposed (it depends on the frequency synthesized w.r.t. the reference one). Delay quantization implies that this approach results in time-resolution enhancement rather than a complete retiming of the output signal. Therefore, although the timing irregularities are reduced, the spectrum quality may or may not improve.

Tuning the delay gates to achieve a specific delay value per tap may not be easy to do over a wide temperature range (and voltage and process variation in an integrated circuit design). Finally, the accumulated *random* jitter of the delay gates (typically series of inverters) may cause some additional degradation of the signal.

C. Time Resolution Enhancement Using a DLL

To minimize the issues of delay variation over process, voltage and temperature variation, as well as the random jitter, one can phase-lock the last tap of the delay in Fig. 17 to the reference clock as shown in Fig. 18. The concept was proposed in [21]–[23] for spurs suppression.

The use of DLL was also suggested in [27] as a coarse timing correction, supplemented by the free running digital delay line approach in Section III-B. These two methods decrease the output jitter by a factor M where M is the number of delay

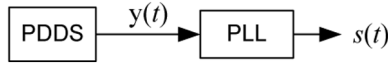


Fig. 19. PDDS followed by a cleanup PLL, [29].

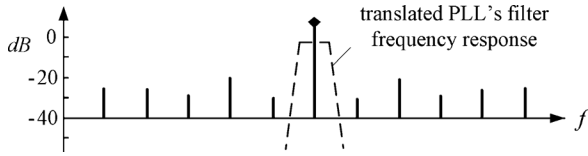


Fig. 20. The cleanup PLL acting as a narrow-band filter centered at the desirable (typically the average-) frequency component of PDDS' output.

taps. An integrated circuit implementation of this technique results in the spectrum of Fig. 32(a) [37].

IV. CLEANUP PLL TECHNIQUES FOR SPURS SUPPRESSION AND JITTER CORRECTION

Cleaning up the unwanted spurs of a DADFS' spectrum using passive analog filters can be done only in a few cases; e.g., a low pass filter can attenuate the harmonics and a sufficiently narrowband filter can possibly select the dominant frequency component when the nearby spurs are at a certain distance and of relatively low power. Making the filters tunable and stable over temperature (and process in integrated circuits) variation is usually challenging or impractical.

A most practical approach is to use a PLL that locks to the dominant (or possibly other desirable) frequency component of the DADFS and acts as a frequency-translated filter. The following sections illustrate the technique.

A. Cleanup PLL Technique

The approach of using a PLL to clean the spectrum of a signal from frequency spurs has been used successfully for decades [28]. In [29] it was proposed to do so for PDDS, as shown in Fig. 19.

The concept is based on the assumption that the PLL (which should *not* have a prescaler divider) has a linear phase-frequency detector (PFD) (or phase detector) and the PLL's loop filter (following the PFD) has a cutoff frequency that is smaller than the offset frequency of the most near-in spur, as shown in Fig. 20.

In this case, the output of the PFD contains only a dc component, corresponding to the phase difference, and frequency components beyond the filter's bandwidth.² This results in a clean PLL output.

A similar concept has been used in [27] but with the PDDS used as a feedback divider inside a PLL.

B. PLL Loop With Analog Phase-Error Correction

In Section IV.A our assumption was that PLL's filter was narrow and steep enough to remove the spurious frequency components which are down-converted to baseband by the linear PFD. The requirements for the filter and the PLL can be relaxed if the architecture in Fig. 21 is used instead.

²Attention must be paid to the harmonics of the dominant component as they can shift the phase-difference value of the PFD.

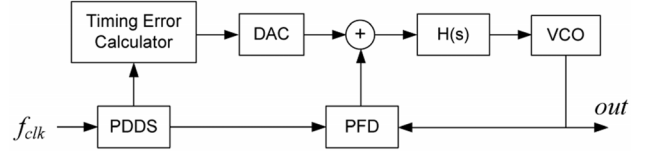


Fig. 21. Cleanup PLL with timing error compensation in the PFD.

Here the instantaneous nonzero output of the PFD, due to the timing irregularities of PDDS, is (partially) canceled by the DAC which is driven by the timing error calculator.

The concept is very similar to classical error compensation in Fractional-N PLL architectures [30], [31]. The architecture in Fig. 21 has been used in [12], [32]. While this method produces a cleaner output, the overhead of analog components is significant and comparable to the hardware needed for a typical standalone Fractional-N PLL.

V. DITHERING TECHNIQUES FOR SPURS SUPPRESSION

The spurs-suppression and jitter-correction techniques discussed in Sections III and IV use analog blocks which may be considered a drawback as explained before.

Digital random-dithering is a classical, purely digital alternative approach for spurs suppression that has been used extensively and very efficiently in standard DDS, Fractional-N PLLs and other similar architectures [28], [31].

Random dithering applied to DADFS tends to *break* the unwanted periodic patterns present in the output sequences and to spread the power of the corresponding frequency spurs over wide range of frequencies (ideally frequency continua).

In some sense, major part of the spurs' power is converted into wideband noise raising the noise floor. This can be conceived as the drawback of the dithering techniques; however, incorporating noise shaping techniques into the dithering mechanism can alleviate this weakness of the techniques [33], [34], [38].

For presentation clarity the concept of dithering is discussed for PDDS and Fractional-N divider types of DADFS. It can be directly extended to FA-type and other DADFS architectures involving phase accumulation.

A. Frequency and Phase Dithering in PDDS

There are two ways to apply dithering to a PDDS; by perturbing the FCW, w , (*frequency dithering*) and by perturbing the output of the accumulator (*phase dithering*), [35].

Frequency dithering is illustrated in Fig. 22 where a random number sequence is added directly to the FCW. It is preferable that the output sequence of the random number generator has a zero mean so that the TA frequency of the PDDS remains unaltered.

The phase accumulator acts like a low pass filter $1 + Z^{-1} \pmod{2^n}$ to the dithering sequence $\{r_k\}_{k=0,1,2,\dots}$. This implies a relatively narrow-band spreading of the power of frequency spurs. To achieve a more wide-band spreading one has to use high-frequency shaped random dither which in practice results in much cleaner output spectra. The easiest way to do so is to pass the output of the random number generator through a simple high-pass filter, e.g., $1 - Z^{-1}$, which has also the desired

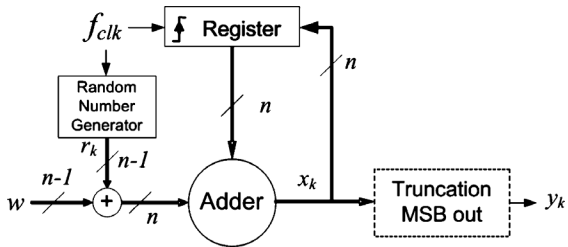


Fig. 22. PDDS with random frequency dithering.

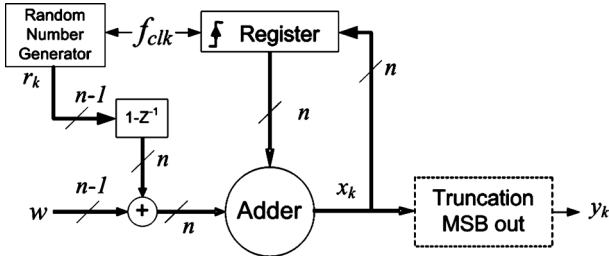


Fig. 23. PDDS with random high-pass shaped frequency dithering.

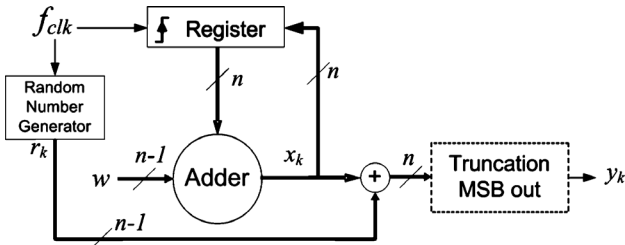


Fig. 24. PDDS with random phase dithering.

property of producing zero mean output to maintain the TA frequency. The modified frequency dithering topology is shown in Fig. 23.

Note that for every clock cycle, k , the value of the register in Fig. 23 is $x_k = [x_{k-1} + (w + r_k - r_{k-1}) \bmod 2^n] \bmod 2^n$, or equivalently, $x_k = (x_{k-1} + w + r_k - r_{k-1}) \bmod 2^n$. If we assume for simplicity that $x_0 = r_0 = 0$ and sum over k (modulo 2^n) we get $x_k = (k \cdot w + r_k) \bmod 2^n$. Moreover, the output can be expressed as $y_k = x_k \text{div } 2^{n-1}$ and so

$$y_k = [(k \cdot w + r_k) \bmod 2^n] \text{div } 2^{n-1}. \quad (4)$$

Phase dithering is another way of applying dithering for spurs suppression, shown for PDDS in Fig. 24. The random sequence is added, modulo 2^n , to the value of the register and the MSB of the sum is used as the output. This creates a random dither on the phase of the signal produced by the PDDS. Note that the mean value of the sequence $\{r_k\}_{k=0,1,2,\dots}$ is irrelevant unless the phase information is important.

Assuming for simplicity that $x_0 = 0$, it is $x_k = (k \cdot w) \bmod 2^n$ and the output is $y_k = \{[(k \cdot w) \bmod 2^n + r_k] \bmod 2^n\} \text{div } 2^{n-1}$ which simplifies to $y_k = [(k \cdot w + r_k) \bmod 2^n] \text{div } 2^{n-1}$. The expression is identical to (4) implying the equivalence of the two dithering techniques. However, phase dithering is easier to implement, as it requires simpler hardware.

The spectra of a PDDS without and with dithering are shown in Fig. 25(a) and (b) respectively. The dithering sequence used was I.I.D. and uniformly distributed in $\{0, 1, 2, \dots, 2^{n-1}\}$, a

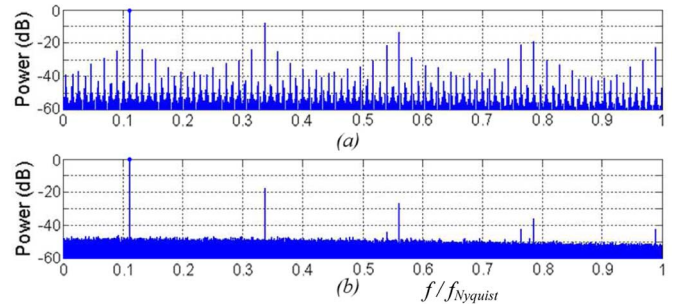
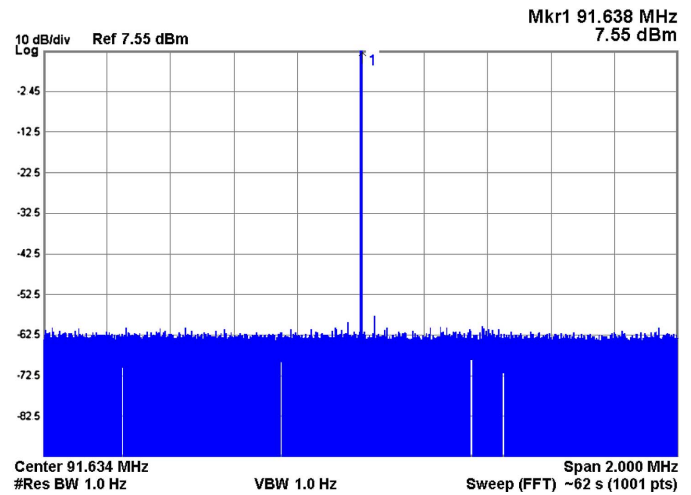
Fig. 25. Output of the PDDS: (a) Without dithering. (b) With random phase dithering or $(1 - Z^{-1})$ shaped frequency dithering [MATLAB].

Fig. 26. Spectrum of the output of phase dithered PDDS, [Measurements based on Xilinx Spartan 3e FPGA implementation in our lab].

typical choice when only the MSB of an n -bit accumulator is outputted.

As seen there is a dramatic improvement in the clarity of the spectrum and the SFDR (ignoring the harmonics). The noise floor however has been raised. Stronger dithering levels may be used to suppress nonharmonic and harmonic spurs even further but at the cost of even higher noise floor.

The noise floor level depends also on the operating frequency (clock). Higher clock frequencies result in lower noise floors. An (FPGA) implementation of a phase dithered PDDS in our lab resulted in a -70 dBc/Hz noise floor (using a 200 Mhz input clock) (Fig. 26). In terms of spurs performance the implementation verifies the theoretical simulation. The only differences are two small (-65 dbc) spurs appearing symmetrically to the carrier most probably caused by an interfering signal modulating the FPGA's output.

Note that when PDDS is phase or frequency dithered with strong dither levels, as in the example above, the spectrum may improve but the output waveform does not resemble the ideal squarewave anymore. Instead it may be very random and it cannot be used for clocking any synchronous digital circuit.

B. Dithering of the Fractional $N/N+1$ Divider

The FND includes the control block (see Fig. 11) generating sequence $\{b_x\}$ which is responsible for the interpolation between division ratios N and $N + 1$. The generation of sequence $\{b_x\}$ can incorporate dithering for spurs suppression [33]. Such

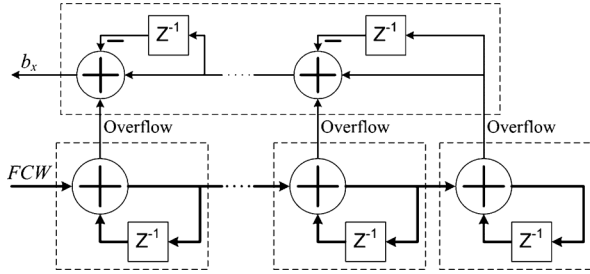


Fig. 27. A typical block diagram of a high-order MASH topology.

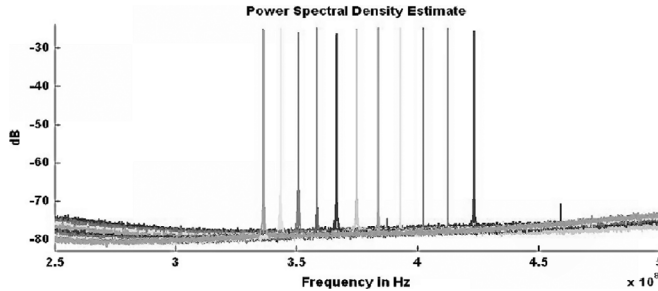


Fig. 28. Dithered FND output using MASH for various division ratios. [The sequence is numerically calculated then loaded into a digital signal generator to produce the output that is measured] [39].

a control block is usually realized as a multi-stage noise shaping (MASH) structure, Fig. 27, a form of digital delta-sigma modulator [34]. It generates the dithering sequence b_x which perturbs the *period* of the generated output signal. Therefore we can view this type of dithering as *period dithering*.

Using a simple first-order MASH as the control block (which is identical to PDDS with overflow output, Fig. 1) still results in spurious output. For this reason, higher order MASH is preferred which is built by combining multiple first-order ones as in Fig. 27, [33], [34]. More complex types of MASH can produce multi-bit values for the dithering signal [40], [41].

The choice of the MASH architecture and its parameters result in phase or frequency dithers of particularly shaped noise spectra (typically high-frequency ones), which can result in high SFDR output (Fig. 28) [39].

C. Dithering Techniques for the Flying Adder

In the core of the FA, Fig. 6, only the m of the $n > m$ bits of the phase accumulator are passed to the phase-selection MUX. Therefore, a sequence of I.I.D. random variables, uniformly distributed in $\{0, 1, 2, \dots, 2^{n-m} - 1\}$ is a meaningful choice for phase dithering to spread the power of the frequency spurs, as it was done in Section V-A.

Fig. 29 shows the spectra of a FA without and with phase dithering, the results are similar to those achieved with dithered PDDS. Again, the SFDR improvement (at least near-in) is impressive; however, the noise floor has been raised and the output signal is very random without any resemblance to a periodic squarewave.

Another architecture that can also be categorized as dithered FA is presented in Fig. 30, [36]. Firstly the Phase Accumulator of the FA is split into two parts (the fractional and the integer part). The circuit topology of the Fractional Accumulator is identical to that of a first order MASH, so we can also

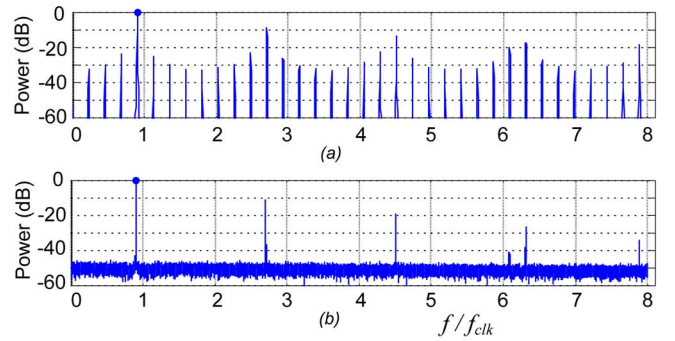


Fig. 29. Output of the FA: (a) without dithering; (b) with random phase dithering or $1 - Z^{-1}$ shaped frequency dithering [MATLAB].

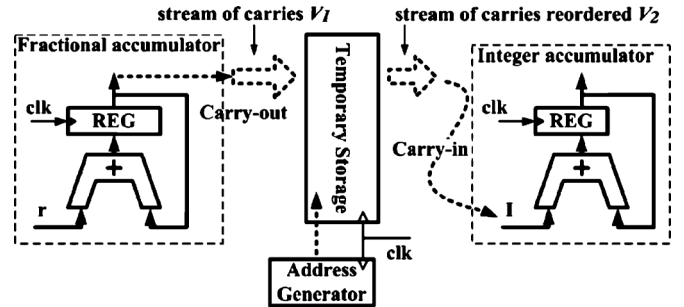


Fig. 30. Carry reorder dithering technique for the FA.

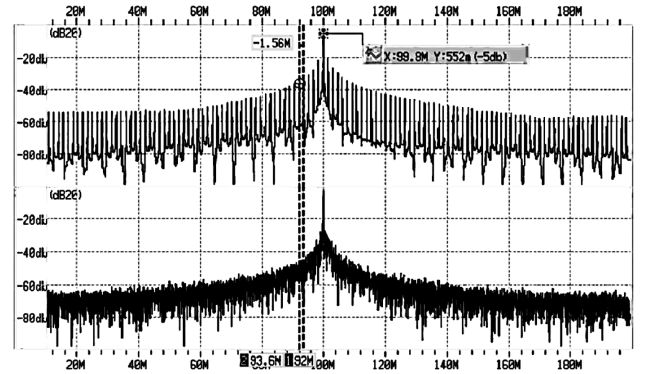


Fig. 31. Period dithering of FA with carry reorder generated by a first order MASH: (a) Without dithering; (b) With dithering [SPICE simulation based on a $0.11 \mu\text{m}$ process] [36].

state that the Integer Accumulator is now period-dithered by the MASH output just like in Section V-B (dithered FND).

Again, using a first-order MASH still results in spurious output so a higher order MASH is preferred. Alternatively (Fig. 30) the output sequence of a first order MASH can be (pseudo) randomly reordered before going into the carry input of the integer accumulator. An instance of the output spectrum of this technique is shown in Fig. 31.

Small, with respect to w , frequency dithering of the FA can maintain the “periodic squarewave” form of the output. In this case the output can be used for clocking digital circuits and in similar applications where the time-domain properties are important, while the dithering allows for certain spurs control or spectral broadening [7], [47], [49].

D. Combining Dithering With Retiming Techniques

The retiming techniques in Sections III-B and III-C, which are based on selecting among a finite number of discrete delay

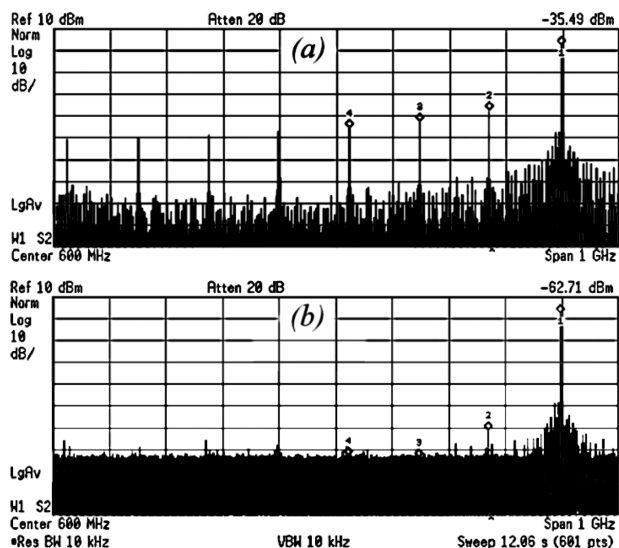


Fig. 32. Combining dithering with retiming. (a) Retiming only. (b) Retiming with random phase dithering, [90 nm CMOS ASIC implementation measurements, using a 5-bit DTC (Variation of Fig. 18 topology)]. [37].

values, provide a time-resolution enhancement of the output. However the improvement in the spectral domain is limited as shown in Fig. 32(a).

To deal with this problem, frequency or phase dithering can be used in addition. As expected, this reduces the remaining spurs further but also raises the noise floor as a side effect. The results are on the same level as the other dithered DADFS and are shown in Fig. 32(b), [21], [22], [37].

VI. COMPARISON AND CONCLUSIONS

Comparing the DADFS cores in Section II, PDDS is probably the simplest one, after the basic IND. PDDS is a DFC and has a large frequency range, upper bounded³ by $f_{\text{clk}}/2$. The FND is about two times the size of PDDS, it is a DPC with the same upper frequency bound, lower frequency bound determined by the size of the register and period resolution determined by the dithering control block. FA is also a DPC with a simple core but it needs a multi-phase clock that usually requires an additional DLL or PLL to generate it. FA achieves much higher frequency, only limited by the number of clock phases. FA's period resolution can be very high.

In comparing the jitter correction and spurs suppression techniques in Sections III and IV, we should first recall that the advantage of DADFS is in their architectural simplicity, low power consumption, and most importantly the lack or minimal use of analog elements, which makes them easy to design in integrated circuit form, port them from one technology to the next and co-integrate them with digital engines in standard CMOS technologies. Therefore any correction technique extensively using analog blocks or being significantly complex contradicts the purpose of using DADFS.

The techniques in Section III-A promise, in principle, an output signal free of timing irregularities and frequency spurs. However to achieve this, the analog delay element needs to be

calibrated under all process, temperature and voltage variation conditions, which is not trivial to do.

The cleanup PLL approach in Section IV can provide exceptionally clean spectrum and jitter free signal but has a heavy analog hardware and power overhead.

The technique in Section III-B is purely digital in principle but in reality, the delay of the elements in the delay line must be monitored in, or characterized under, all operating conditions; which is not a trivial overhead. Also, the jitter correction it provides is only partial. The variation of the technique in Section III-C requires a DLL and therefore some analog blocks. It also provides only partial correction.

The dithering techniques in Section V are based on all-digital topologies and can dramatically reduce the spurs in the output spectrum, but at the cost of raising the noise floor. Digital noise shaping blocks can be added to alleviate the later potential problem. Also, unless light (frequency) dither is used, the output signal is not squarewave-like and therefore it is not appropriate for clocking digital circuits. Dithering techniques offer the only pure digital solution.

Depending on the target application, one can select the most convenient DADFS along with a jitter-correction/spurs-suppression architecture. One extreme is clocking digital circuits and data synchronization applications which can typically tolerate deterministic jitter of the order of a fraction of the period, here, time resolution enhancement methods can be satisfactory, if needed at all, i.e., a FA without jitter correction may be sufficient.

The other extreme involving highly clean spectra generation can be served by a clean-up PLL following any DADFS type with spurs sufficiently sparse and conveniently located to be removed by a sufficiently narrow PLL bandwidth; in this case the phase noise can be excellent, the noise floor can be below -130 dBc and the spurs can be practically eliminated. There is a trade-off of course between the settling time (and FM modulation bandwidth), spurs suppression and frequency resolution (complexity and power consumption are also involved).

REFERENCES

- [1] J. Tierney, C. M. Radar, and B. Gold, "A digital frequency synthesizer," *IEEE Trans. Audio Electroacoust.*, vol. AC-19, pp. 48–57, Mar. 1971.
- [2] C. E. Wheatley, III and D. E. Phillips, "Spurious suppression in direct digital synthesizers," in *Proc. 35th Freq. Control Symp.*, May 1981, pp. 428–435.
- [3] V. S. Reinhardt, "Direct digital synthesizers," Space and Communications Group, Hughes Aircraft Co., Los Angeles, CA, Tech. Rep., Dec. 1985.
- [4] E. McCune, "Direct digital frequency synthesizer with designable step-size," in *Proc. IEEE Radio Wirel. Symp. (RWS)*, Jan. 2010.
- [5] 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.
- [6] H. Mair, L. Xiu, and S. A. Fahrenbruch, "Precision frequency and phase synthesis," U.S. Patent 6 329 850 B1, Dec. 2001, (filed Dec. 27, 1999).
- [7] 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, 3rd Quart., 2008.
- [8] P. Sotiriadis, "Theory of flying-adder frequency synthesizers part I: Modeling, signals periods and output average frequency," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 8, pp. 1935–1948, Aug. 2010.
- [9] P. Sotiriadis, "Theory of flying-adder frequency synthesizers part II: Time and frequency domain properties of the output signal," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 57, no. 8, pp. 1949–1963, Aug. 2010.

³ . . . unless it is clocked at both the rising and falling edges of the clock.

- [10] D. E. Calbaza and Y. Savaria, "A direct digital periodic synthesis circuit," *IEEE J. Solid-State Circuits*, vol. 37, no. 8, pp. 1039–1045, Aug. 2002.
- [11] P. Sotiriadis, "Exact spectrum and time-domain output of flying-adder frequency synthesizers," *IEEE Trans. Ultrason., Ferroelectr., Freq. Control*, vol. 57, no. 9, pp. 1926–1935, Sept. 2010.
- [12] E. R. Harrison, Jr. and J. , "Means for reducing spurious frequencies in a direct frequency synthesizer," U.S. Patent 4 185 247, filed: Jan. 3, 1978, assignee: U.S. Air Force, Washington, DC.
- [13] V. N. Kochemasov and A. N. Fadeev, "Digital-computer synthesizers of two-level signals with phase-error compensation," *Telecommun. Radio Eng.*, vol. 36/37, pp. 55–59, Oct. 1982.
- [14] L. Dartois, A. Roullet, and R. Riboni, "High frequency digital synthesizer with aperiodic correction optimizing the spectral purity," U.S. Patent 4 792 914, filed: Dec. 22, 1986, assignee: Thomson-csf, Paris, France.
- [15] H. Nosaka, T. Nakagawa, and A. Yamagishi, "A phase interpolation direct digital synthesizer with a digitally controlled delay generator," in *Proc. VLSI Circuit Symp. Dig.*, Jun. 1997, pp. 75–76.
- [16] A. Yamagishi, H. Nosaka, M. Muraguchi, and T. Tsukahara, "A phase-interpolation direct digital synthesizer with an adaptive integrator," *IEEE Trans. Microw. Theory Tech.*, vol. 48, no. 6, pp. 905–909, Jun. 2000.
- [17] H. Nosaka, Y. Yamaguchi, A. Yamagishi, H. Fukuyama, and M. Muraguchi, "A low-power direct digital synthesizer using a self-adjusting phase-interpolation technique," *IEEE J. Solid-State Circuits*, vol. 36, no. 8, Aug. 2001.
- [18] T. Finateu, I. Miro-Panades, F. Boissières, J. B. Bégueret, Y. Deval, D. Belot, and F. Badets, "A 500-MHz $\Sigma\Delta$ phase-interpolation direct digital synthesizer," in *Proc. IEEE Asian Solid-State Circuits Conf.*, Jeju, Korea, Nov. 12–14, 2007.
- [19] H. Tucholski, "Direct digital synthesizer with output signal jitter reduction," U.S. Patent 7 103 622, filed: Oct. 8, 2002, assignee: Analog Devices, Inc., Norwood, MA.
- [20] E. McCune, "Time-filtered squarewave output from direct digital synthesis," in *Proc. Microw. Symp. Dig. (MTT)*, May 2010.
- [21] P. Nuytkens and P. Van Broekhoven, "Digital frequency synthesizer," U.S. Patent 4 933 890, filed: Jun. 13, 1989, assignee: The Charles Stark Draper Laboratory, Inc., Cambridge, MA.
- [22] T. Gradishar and R. Stengel, "Method and apparatus for noise shaping in direct digital synthesis," U.S. Patent 7 143 125, filed: Apr. 16, 2003, assignee: Motorola, Inc., IL.
- [23] E. McCune, "Digital frequency synthesizer and method with Vernier interpolation," U.S. Patent 5 247 469, Sep. 21, 1993.
- [24] D. E. Bockelman and J.-K. Juan, "Time interpolating direct digital synthesizer," U.S. Patent 6 353 649, filed: Jun 2, 2000, Motorola, Inc.
- [25] F. L. Martin, R. E. Stengel, and J.-K. Juan, "Method and apparatus for digital frequency synthesis," U.S. Patent 6 891 420, filed: Dec 21, 2001, Motorola, Inc.
- [26] J.-K. Juan, R. E. Stengel, F. J. Martin, and D. E. Bockelman, "Cascaded delay locked loop circuit," U.S. Patent 7 154 978, filed: Nov 2, 2001, Motorola, Inc.
- [27] S. Agarwal and X. Chen, "Phase error correction circuit for a high speed frequency synthesizer," U.S. Patent 7 205 798, filed: Jan. 28, 2005, assignee: Intersil Americas Inc., Milpitas, CA.
- [28] W. F. Egan, *Frequency Synthesis by Phase Lock*, 2nd ed. New York: Wiley, 1999.
- [29] R. P. Gilmore, "Direct digital synthesizer driven phase lock loop frequency synthesizer with clean up phase lock loop," U.S. Patent 5 757 239, filed: Jan 8, 1997, assignee: Qualcomm Inc., San Diego, CA.
- [30] R. G. Cox, "Frequency synthesizer," U.S. Patent 3 976 945, Aug. 24, 1976.
- [31] U.L. Rohde, *Microwave and Wireless Synthesizers: Theory and Design*, 1st ed. Singapore: Wiley-Interscience, 1997.
- [32] R. L. Van Der Valk, R. J. Dequesnoy, J. H. A. De Rijk, and M. T. Spijker, "Frequency synthesizer," U.S. Patent 5 905 388, filed: Sept. 26, 1997, assignee: X Integrated Circuits B.V., Rotterdam, Netherlands.
- [33] T. A. Riley, M. Copeland, and T. Kwasniewski, "Delta-sigma modulation in fractional-N frequency synthesis," *IEEE J. Solid-State Circuits*, vol. 28, no. 5, pp. 553–559, May 1993.
- [34] Hosseini and M. P. Kennedy, "Maximum sequence length MASH digital delta—sigma modulators," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 54, no. 12, pp. 2628–2638, Dec. 2007.
- [35] C. E. Wheatley, III, "Digital frequency synthesizer with random jittering for reducing discrete spectral spurs," U.S. Patent 4 410 954, Oct. 18, 1983.
- [36] L. Xiu, M. Ling, and H. Jiang, "A storage based carry randomization techniques for spurs reduction in flying-adder digital-to-frequency converter," *IEEE Trans. Circuit Syst. II, Exp. Briefs*, vol. 58, no. 6, pp. 326–330, Jun. 2011.
- [37] S. Talwalkar, T. Gradishar, B. Stengel, G. Cafaro, and G. Nagaraj, "Controlled dither in 90 nm digital to time conversion based direct digital synthesizer for spur mitigation," in *Proc. IEEE Symp. Radio Freq. Circuits*, May 2010, pp. 549–552.
- [38] T. Gradishar and B. Stengel, "System and method for introducing dither for reducing spurs in digital-to-time converter direct digital synthesis," U.S. Patent 7 421 464, 2008.
- [39] J. Rode, A. Swaminathan, I. Galton, and P. M. Asbeck, "Fractional-N direct digital frequency synthesis with a 1-bit output," in *IEEE MTT-S Int. Microw. Symp. Dig.*, Jun. 2006, pp. 415–418.
- [40] H. Wang, P. Brennan, and D. Jiang, "A comparison of Sigma-Delta modulator techniques for fractional-N frequency synthesis," in *Proc. 49th Midwest Symp. Circuits Syst. (MWSCAS)*, San Juan, Puerto Rico, Aug. 6–9, 2006, pp. 659–663.
- [41] X. Mao, H. Yang, and H. Wang, "Comparison of Sigma-Delta modulator for fractional-N PLL frequency synthesizer," *J. Electron. (China)*, vol. 24, no. 3, pp. 374–379, May 2007.
- [42] D. Calbaza and Y. Savaria, "A direct digitally delay generator," in *Proc. Int. Semicond. Conf. (CAS)*, Sinaia, Romania, Oct. 2000, vol. 1, pp. 87–90.
- [43] 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, pp. 637–649, Oct. 2002.
- [44] L. Xiu and Z. You, "A new frequency synthesis method based on flying-adder architecture," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 50, pp. 130–134, Mar. 2003.
- [45] L. Xiu, W. Li, J. Meiners, and R. Padakanti, "A novel all digital phase lock loop with software adaptive filter," *IEEE J. Solid-State Circuit*, vol. 39, no. 3, pp. 476–483, Mar. 2004.
- [46] L. Xiu and Z. You, "A "flying-adder" frequency synthesis architecture of reducing VCO stages," *IEEE Trans. Very Large Scale Integr. (VLSI) Syst.*, vol. 13, no. 2, pp. 201–210, Feb. 2005.
- [47] L. Xiu, "A novel DCXO module for clock synchronization in MPEG2 transport system," *IEEE Trans. Circuits Syst. I, Reg. Papers*, vol. 55, pp. 2226–2237, Sep. 2008.
- [48] L. Xiu, "A flying-adder based on-chip frequency generator for complex SoC," *IEEE Trans. Circuits Syst. II, Exp. Briefs*, vol. 54, pp. 1067–1071, Dec. 2007.
- [49] L. Xiu, "A flying-adder PLL technique enabling novel approaches for video/graphic applications," *IEEE Trans. Consum. Electron.*, vol. 54, no. 2, pp. 591–599, May 2008.
- [50] C.-W. Huang, P. Gui, and L. Xiu, "A wide-tuning-range and reduced-fractional-spurs synthesizer combining Σ - Δ fractional-N and integer flying-adder techniques," in *Proc. IEEE Int. Symp. Circuits Syst. 2009*, pp. 1377–1380.
- [51] L. Xiu, C.-W. Huang, and P. Gui, "Simulation study of time-average-frequency based clock signal driving systems with embedded digital-to-analog converters," in *Proc. IEEE Int. Symp. Circuits Syst. 2009*, pp. 465–468.
- [52] J. M. Rabaey, A. Chandrakasan, and B. Nikolic, *Digital Integrated Circuits*, 2nd ed. Upper Saddle River, NJ: Prentice-Hall, 2003.
- [53] P. Sotiriadis, "Spurs suppression and deterministic jitter correction in all-digital frequency synthesizers, current state and future directions," in *Proc. IEEE Symp. Circuits Syst.*, 2011, pp. 422–425.
- [54] H. T. Nicholas and H. Samuelli, "An analysis of the output spectrum of direct digital frequency synthesizers in the presence of phase-accumulator truncation," in *Proc. 41st Annu. Symp. Freq. Control 1987*, pp. 495–502.



Paul P. Sotiriadis (S'99–M'02–SM'09) received the Diploma degree in electrical and computer engineering from the National Technical University of Athens, Greece, in 1994, the M.S. degree in electrical engineering from Stanford University, Stanford, CA, in 1996, and the Ph.D. degree in electrical engineering and computer science from the Massachusetts Institute of Technology, Cambridge, in 2002.

In 2002, he joined the Johns Hopkins University as Assistant Professor of Electrical and Computer Engineering. In 2007, he joined Apex/Eclipse INC as the Chief Technology Officer

and shortly after that he started Sotekco Electronics LLC, an electronics research company in Baltimore, MD. In 2012, he joined the faculty of the Electrical and Computer Engineering Department of the National Technical University of Athens, Greece.

He has authored and coauthored more than 80 technical papers in IEEE journals and conferences, holds one patent, has several patents pending, and has contributed chapters to technical books. His research interests include design, optimization, and mathematical modeling of analog and mixed-signal circuits, RF and microwave circuits, advanced frequency synthesis, biomedical instrumentation, and interconnect networks in deep-submicrometer technologies. He has led several projects in these fields funded by U.S. organizations and has collaborations with industry and national labs.

Dr. Sotiriadis served as an Associate Editor of the IEEE TRANSACTIONS ON CIRCUITS AND SYSTEMS—PART II: EXPRESS BRIEFS from 2005 to 2010 and has been a member of technical committees of several conferences. He regularly reviews for many IEEE transactions and conferences. He also serves on proposal review panels at the National Science Foundation.



Kostas Galanopoulos (S'11) received the Diploma degree in computer engineering and informatics from the University of Patras, Greece in 2009. He is currently working toward the Ph.D. degree in electrical and computer engineering at the National Technical University of Athens, Greece.

He has coauthored 6 technical papers in IEEE journals and conferences. His research interests include design and optimization of mixed-signal, digital and microprocessor data-path circuits, low power optimization, and all-digital frequency synthesis techniques. He regularly serves as a reviewer for IEEE transactions and conferences.