Diophantine Frequency Synthesis for Fast-Hopping, High-Resolution Frequency Synthesizers

Paul Peter Sotiriadis, *Member, IEEE*

***Abstract*—The application of the Diophantine Frequency Synthesis (DFS) methodology is presented and certain practical aspects of it are illustrated through the design and frequency planning of two Forward DFS synthesizers each using two Integer-*N* phase-locked loops (PLLs). Both synthesizers achieve frequency resolution about $100\times$ times better that their constituent PLLs without compromising hopping speed performance or spectral purity.**

***Index Terms*—Diophantine equation, frequency hopping, frequency synthesis, frequency synthesizer, number theory, phase-locked loop (PLL), timing systems, wireless.**

I. Introduction

FREQUENCY synthesis is an important part of modern wireless and wired communication systems. The requirements for multiple-band and multiple-standard operation, with minimal power consumption, makes the design of the frequency synthesis blocks challenging.

Frequency synthesizers have a long list of technical specifications; some of them are more essential and usually, mutually conflicting. Frequency resolution and hopping speed are two such ones in general and especially in Integer-*N* phase-locked loops (PLLs) designs where the two quantities are strongly coupled since the hopping time is inversely proportional to the loop bandwidth which is, in a sense, proportional to the frequency step [1].

Fractional-*N* PLLs on the other hand alleviate the hopping-speed versus frequency resolution trade-off, but this is done at the cost of spectral purity, especially near-in.

Direct digital synthesis (DDS) technology offers very high resolution, hopping speed and very wide frequency range. However, typical DDS spectra are much more populated with spurious signals, even near-in, and have higher noise floor compared to Integer-*N* PLLs. Moreover, the frequency range of DDS is limited by the performance of digital-to-analog converters.

Classical multiloop architectures using Integer-*N* PLLs [2] can provide clean output signal and wide frequency range at the cost of complexity, and in some cases, of the frequency hopping speed. Complex, discrete-component implementations of multiloop architectures are typical.

Diophantine Frequency Synthesis (DFS) [3], [4] is a new methodology for designing frequency synthesizers with high resolution, fast frequency hopping, and low spurs, especially near-in. It can be used for any frequency range and it is particularly convenient for higher frequencies where the alternative choices are limited.

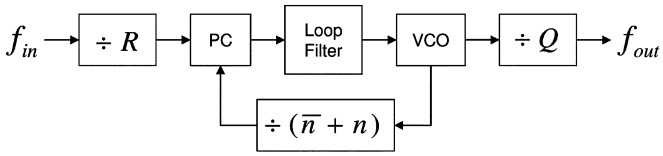

Fig. 1. Basic PLL.

DFS is based on mathematical properties of Diophantine equations [5] and the DFS schemes can be implemented by employing two or more Integer-*N* PLLs that are driven by the same reference and whose output frequencies are added (or subtracted) to give the output frequency of the synthesizer. DFS *distributes* the output frequency resolution among the PLLs.

This paper presents two forward two-PLL DFS architectures, their circuit implementations and spectral measurements of the output signals, in an effort to demonstrate how one can use DFS as a design method.

The focus of this paper is on frequency planning and high-level architectures. To maintain focus and avoid repeating well-known facts about the design of PLLs and mixers, two commercial components were used for the implementations: the triple-PLL CDCE706 by Texas Instruments and analog multiplier AD835 by Analog Devices.

The two architectures have output-frequency ranges 0–10 and 195–205 MHz, respectively, and frequency steps equal to 542 Hz. In both cases the phase-comparator frequencies of the constituent PLLs are 133 and 130 kHz, and, their output-frequency steps are 67 and 43 kHz, respectively,[1] i.e., about $100\times$ larger than the steps of the synthesizers.

Some elements of the DFS theory, used in this work, are summarized in Section II, details of the theory can be found in [3].

II. Elements of DFS

DFS uses two or more PLLs like that in Fig. 1[2] having a fixed *prescaler* divider R. The size of the *feedback* divider is $\bar{n} + n$, where integer $\bar{n}$ is fixed and greater than integer R, and, n is an integer variable ranging from $-R$ to $+R$. An output divider Q may be present. The output frequency is

$$f_{\text{out}} = \frac{\bar{n} + n}{Q \times R} f_{\text{in}}. \tag{1}$$

[1]Output frequency dividers, $C_1 = 2$ and $C_2 = 3$, available in CDCE706, are used (Fig. 3). They are not necessary for implementing DFS in general.

[2]Instead of a phase comparator (PC) the PLL may have a phase-frequency comparator or phase detector etc. An output divider (Q) may be present.

Manuscript received August 14, 2007; revised December 4, 2007. This paper was recommended by Guest Editor A. Tasic.

The author is with the Johns Hopkins University, Baltimore, MD 21201 USA (e-mail: pps@ieee.org).

Digital Object Identifier 10.1109/TCSII.2008.919499

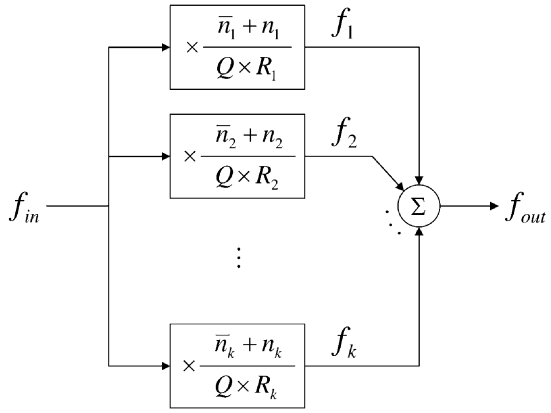

Fig. 2. Abstract, high-level, k-PLL DFS scheme.

The general, abstract k-PLL DFS architecture is shown in Fig. 2. The PLLs, with parameters $R_i, n_i, \bar{n}_i$, for $i = 1, 2, \ldots, k$ and Q, are driven by the same signal and their output frequencies are added (or subtracted) to provide the output frequency of the synthesizer

$$f_{\text{out}} = \left(\frac{n_1}{R_1} + \frac{n_2}{R_2} + \cdots + \frac{n_k}{R_k}\right)\frac{f_{\text{in}}}{Q} + \bar{f}_{\text{out}} \tag{2}$$

where the fixed-frequency term $\bar{f}_{\text{out}}$ is given by

$$\bar{f}_{\text{out}} = \left(\frac{\bar{n}_1}{R_1} + \frac{\bar{n}_2}{R_2} + \cdots + \frac{\bar{n}_k}{R_k}\right)\frac{f_{\text{in}}}{Q}. \tag{3}$$

Now we focus on the first term in the RHS of (2), which is variable. DFS' main Theorem follows [3].

Theorem 2.1: [3]: If $R_1, R_2, \ldots, R_k$ are pairwise relatively prime positive integers (i.e., no pair of them has common divider other than ± 1) then, for every integer n, such that $-R_1R_2\ldots R_k \leq n \leq R_1R_2\ldots R_k$ we can find integers $n_1, n_2, \ldots, n_k$ with $-R_i \leq n_i \leq R_i$, for all $i = 1, 2, \ldots, k$, for which

$$\frac{n_1}{R_1} + \frac{n_2}{R_2} + \cdots + \frac{n_k}{R_k} = \frac{n}{R_1R_2\ldots R_k}. \tag{4}$$

Rephrasing Theorem 2.1 and using (2) we conclude that by appropriately programming the (sizes of the) feedback dividers $n_i,\ i = 1, 2, \ldots, k$, the output frequency f_{out} takes all values within the range

$$\left[\bar{f}_{\text{out}} - \frac{f_{\text{in}}}{Q}, \bar{f}_{\text{out}} + \frac{f_{\text{in}}}{Q}\right] \tag{5}$$

with *uniform* frequency step equal to

$$f_{\text{step}} = \frac{f_{\text{in}}}{QR_1R_2\ldots R_k}. \tag{6}$$

Now suppose that some of the PLL frequencies, $f_1, f_2, \ldots, f_k$ were subtracted, instead of being added to the general sum, i.e., for some values of index i, n_i/R_i and $\bar{n}_i/R_i$ were replaced by $-n_i/R_i$ and $-\bar{n}_i/R_i$, respectively, in (2), (3), and (4). Then, the above statements would still be valid and the expressions of the range and the step would remain the same—this is because the sets of values of $n_1, n_2, \ldots, n_k$ are symmetric with respect to 0.

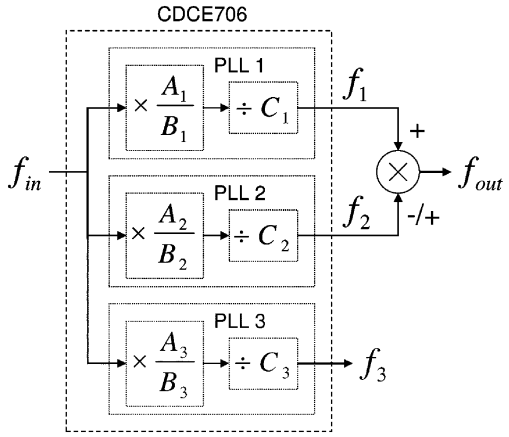

Fig. 3. The high-level architectures of the forward two-PLL DFS synthesizers. Case 1: $f_{\text{out}} = f_1 - f_2$. Case 2: $f_{\text{out}} = f_1 + f_2$.

TABLE I
RANGES OF FREQUENCY DIVIDERS

Divider	Min	Max
A_i	1	4095
B_i	1	511
C_i	1	127

Therefore, Theorem 2.1 is also valid for all combinations of additions/subtractions of the frequencies f_i, i.e., all combinations of $\pm$ in

$$f_{\text{out}} = \sum_{i=1}^{k} \pm\left(\frac{n_i}{R_i} + \frac{\bar{n}_i}{R_i}\right)\frac{f_{\text{in}}}{Q}.$$

with the understanding that: i) when the signs are chosen they remain *fixed*, and ii) all related expressions are modified accordingly.

A significant advantage of DFS, implied by (5) and (6), is that with small values of $k, R_1, R_2, \ldots, R_k$ and Q, the frequency step, f_{step}, can be made very small, while at the same time, the phase-comparator frequencies of the PLLS, $f_{\text{in}}/R_i, i = 1, 2, \ldots, k$, and the output range $2f_{\text{in}}/Q$ can be very large.

III. TWO FORWARD TWO-PLL DFS SYNTHESIZERS

The goal of the paper is to demonstrate DFS through examples without repeating well-known facts about PLL design. To this end, Texas Instruments' programmable triple-PLL device CDCE706 was used which automatically adjusts its loop-filters' parameters and voltage-controlled oscillator (VCO)'s tuning ranges. It is noted however that the convenience it offers is paid by its low spurious-free dynamic range (SFDR), ~ 55 dB (worst-case specification), at the output of the PLLs. Two of the three PLLs in CDCE706 were employed in both of the DFS schemes. The high-level architectures are shown in Fig. 3 corresponding to the "$-$" and "$+$" choices in the mixer, respectively.

The operating ranges of the prescaler, feedback and output dividers are shown in Table I. The frequency range of the VCOs is 80–300 MHz implying (see Fig. 3)

$$80 \text{ MHz} \lessapprox \frac{A_i}{B_i} f_{\text{in}} \lessapprox 300 \text{ MHz}. \tag{7}$$

A. DFS Case 1 : $f_{\text{out}} = f_1 - f_2$

The architecture in Fig. 3 with frequency substraction, "−", in the mixer is considered first. The goal is to achieve output frequency range 0–10 MHz with fine frequency resolution and high phase-comparator frequencies in the constituent PLLs. The input reference frequency is f_{in} =64 MHz. From Fig. 3 we have

$$f_{\text{out}} = f_1 - f_2 \tag{8}$$

where

$$f_i = \frac{A_i}{B_i C_i} f_{\text{in}} \tag{9}$$

for PLLs $i = 1, 2$. Moreover, the VCO's frequencies are

$$f_{\text{VCO}_i} = \frac{A_i}{B_i} f_{\text{in}} \tag{10}$$

and the phase-comparator frequencies are

$$f_{\text{PC}_i} = \frac{f_{\text{in}}}{B_i}. \tag{11}$$

It is convenient to set $R_i = B_i C_i / Q, i = 1, 2$ where

$$Q = \gcd(B_1 C_1, B_2 C_2) \tag{12}$$

and express the denominators of f_i in (9) as

$$B_i C_i = R_i Q \tag{13}$$

$i = 1, 2$. Note that $\gcd(R_1, R_2) = 1$, i.e., R_1, R_2 are relatively prime as it is required by Theorem 2.1.

We decompose the numerators in (9) as $A_i = n_i + \bar{n}_i$, like we did in (1), and impose the constraints

$$-R_i \le n_i \le R_i \tag{14}$$

used in Theorem 2.1. Then, (9) becomes

$$f_i = \frac{n_i + \bar{n}_i}{R_i} \left(\frac{f_{\text{in}}}{Q} \right). \tag{15}$$

From equations (4), (8), and (15), and Theorem 2.1 (as well as the comments on the addition/subtraction of the frequencies f_i following Theorem 2.1) we have that

$$f_{\text{out}} = \left(\frac{\bar{n}_1}{R_1} - \frac{\bar{n}_2}{R_2} \right) \frac{f_{\text{in}}}{Q} + \frac{n}{R_1 R_2} \frac{f_{\text{in}}}{Q} \tag{16}$$

where n can take every integer value from $-R_1 R_2$ to $R_1 R_2$ by appropriately programming n_1 and n_2 within their pre-set ranges (14) using the DFS algorithm [3]. Moreover, it is $\bar{f}_{\text{out}} = \bar{f}_1 - \bar{f}_2$, where

$$\bar{f}_i = \frac{\bar{n}_i}{R_i} \frac{f_{\text{in}}}{Q}, \qquad i = 1, 2 \tag{17}$$

and the range of f_{out} is given in (5).

The desirable output range is 0–10 MHz implying $\bar{f}_{\text{out}} - f_{\text{in}}/Q = 0$ and $\bar{f}_{\text{out}} + f_{\text{in}}/Q = 10$ MHz which give $\bar{f}_{\text{out}} = 5$ MHz and $f_{\text{in}}/Q = 5$ MHz. We relax the last one as

$$\frac{f_{\text{in}}}{Q} \gtrapprox 5 \text{ MHz}. \tag{18}$$

Input (reference) frequency $f_{\text{in}} = 64$ MHz was used. From (18) this implies $f_{\text{in}}/5 \text{ MHz} = 12.8 \gtrapprox Q$ and so the maximum value of Q resulting in full coverage of the desirable output range 0–10 MHz is

$$Q = 12. \tag{19}$$

The minimization of mixing spurs involves the mixer's intermodulation behavior, the set of all frequency pairs (f_1, f_2), the harmonic contents and amplitudes of the mixed signals and the filters before and after the mixer.

Based on the characteristics of the multiplier AD835 that is used here as a mixer, the ∼10-MHz range of f_1 and f_2 centered at $\bar{f}_1$ and $\bar{f}_2$, respectively, (this results from (15) and the range of n_i, (14)), and the 0–10-MHz output range, we make the conservative choice to set $\bar{f}_1, \bar{f}_2$ close to 100 MHz. Given the frequency limits of the VCOs (7), this choice implies $C_1, C_2 \in \{1, 2, 3\}$.

To simplify the selection process for B_1, B_2, C_1, and C_2 we make the assumption that C_1 and C_2 are factors of R_1 and R_2 respectively, and set $X = R_1/C_1$ and $Y = R_2/C_2$ which combined with (13) and (19) imply $B_1 = 12X$ and $B_2 = 12Y$. From (16), the frequency step of the synthesizer is $f_{\text{in}}/(R_1 R_2 Q)$ where

$$R_1 R_2 Q = 12 X Y C_1 C_2. \tag{20}$$

Moreover, from Table I we have that $B_i \le 511$ and so

$$X, Y \le 511/12 = 42.583\ldots. \tag{21}$$

Equations (12) and (19) along with $B_1 = 12X$ and $B_2 = 12Y$ impose the requirement

$$\gcd(XC_1, YC_2) = 1 \tag{22}$$

Note that $\gcd(XC_1, YC_2) > 1$ would lead to unnecessary reduction of the output frequency resolution anyway.

The larger C_i is, the smaller the frequency step of PLL_i is. Since it must be $C_1, C_2 \in \{1, 2, 3\}$ and $C_1 = C_2 = 3$ do not satisfy (22), we choose

$$C_1 = 2, \quad C_2 = 3. \tag{23}$$

Setting X and Y to their maximum values, i.e., $X = 41$ and $Y = 42$, or, $X = 42$ and $Y = 41$ leads to $\gcd(XC_1, YC_2) = 2$ and $\gcd(XC_1, YC_2) = 3$, respectively, which also contradict (22). Instead we choose $X = 40$ and $Y = 41$, i.e., slightly smaller values, giving $\gcd(XC_1, YC_2) = 1, R_1 = 80, R_2 = 123, B_1 = 480, B_2 = 492$, and $R_1 R_2 Q = 118\,080$.

The following set of equations summarizes our choices and derivations up to now:

$$\begin{array}{lll} B_1 = 480 & C_1 = 2 & R_1 = 80 \\ B_2 = 492 & C_2 = 3 & R_2 = 123 \end{array} \quad Q = 12.$$

Following the discussion on $\bar{f}_1$ and $\bar{f}_2$ we choose

$$\bar{n}_1 = 1485 \quad \text{and} \quad \bar{n}_2 = 2168 \tag{24}$$

which along with (17) give $\bar{f}_1 = 99.000\,000$ MHz and $\bar{f}_2 = 94.005\,420$ MHz. Therefore, $\bar{f}_{\text{out}} \cong 4.995$ MHz and, ac-

TABLE II
FREQUENCY RANGES AND RESOLUTIONS OF SIGNALS (HZ)-DFS 1

	Min	Central	Max	Step
f_{in}	-	64 000 000	-	-
f_1 (PLL 1)	93 666 667	99 000 000	104 333 333	66 667
f_2 (PLL 2)	88 672 087	94 005 420	99 338 753	43 360
f_{out} (DFS)	-338 753	4 994 580	10 327 913	542

cording to (5), the output frequency range covers the desirable 0–10-MHz band.

Finally, following Theorem 2.1 and the definitions of A_i's, the required ranges of dividers A_1 and A_2 are

$$1485 - 80 \le A_1 \le 1485 + 80$$
$$2168 - 123 \le A_2 \le 2168 + 123. \quad (25)$$

The above choices of prescalers and output dividers, and, the ranges of the feedback dividers result in a DFS scheme with frequencies given in Table II.

The output frequency step is about two orders of magnitude smaller than those of the constituent PLLs.

B. Mixer

The high-level schematic of the mixer is shown in Fig. 4. The mixer is based on the four-quadrant multiplier AD835 chosen for its linearity and appropriate bandwidth.[3] Two buffers, LMH6559, are used to isolate the PLLs from the multiplier and reduce coupling between the PLLs.

The two low-pass filters LPF1 and LPF2 are seventh-order Chebyshev Type-I with 1-dB ripple in the passband and passive ladder implementations. They remove the harmonics of the square-wave signals coming from the PLLs and feed clean sinusoidals to the multiplier. The passbands of LPF1 and LPF2 are 0–112 and 0–123 MHz, respectively.

The third filter, LPF3 following the multiplier removes the high frequency product of the multiplication, at around 200 MHz. It is a fifth-order low-pass Chebyshev Type-I with1-dB ripple in the passband 0–10 MHz and passive ladder implementation.

C. Results of DFS Case 1

Programming the two PLLs with the DFS parameters derived in Section III.A and using the DFS algorithm in [3], based on (16) we can generate all predicted frequencies

$$f_{\text{out}} = 4\,994\,580 \text{ Hz} + n \cdot 542 \text{ Hz}$$
$$n = -9840 \ldots 9840. \quad (26)$$

Given a desirable value of n, parameters n_1 and n_2 are derived using the DFS algorithm in [3]. The frequency ranges of the PLLs and that of the output signal, along with their resolutions (steps), are shown in Table II.

This two-PLL DFS scheme achieves frequency resolution of 542 Hz with phase-comparator frequencies of the constituent PLLs equal to f_{PC_1} =133 333 Hz and f_{PC_2} =130 081 Hz.

The high phase-comparator frequencies, compared to the frequency resolution, allow for easy suppression of the phase-comparator spurs and for fast frequency hopping. The end result depends of course on the particular design and implementation of the circuit.

[3]Better noise level performance can be achieved by other types of mixers.

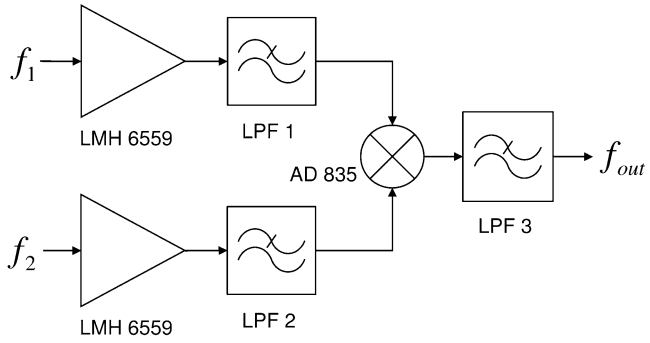

Fig. 4. Mixer, low-pass filters, and buffers.

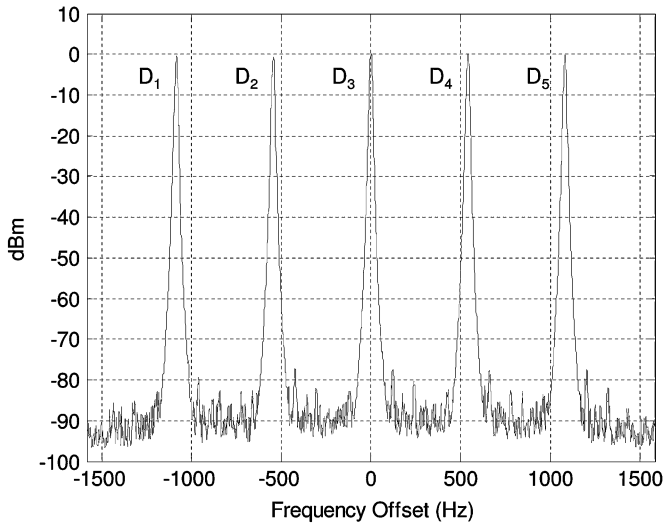

Fig. 5. Five consecutive synthesized frequencies, D_1-D_5 corresponding to values of n_1, n_2, and n shown in Table III. Measurement: $\text{RBW} = 10$ Hz, $\text{VBW} = 10$ Hz and "max" function. The frequency step is 542 Hz while the frequency steps of the two PLLs are about 67 and 43 kHz. PLLs' phase-comparator frequencies are 133 and 130 kHz, respectively.

Using the CDCE707 and AD835 devices, SFDR of ∼77 dB (recall the SFDR specs of CDCE706) and settling time of approximately ∼1.1 ms, to reach 0.1% frequency error, was achieved. The settling time is essentially that of the PLLs in CDCE706 plus some delay due to the digital interface. The delay contributed by the filters (and so the mixer) is negligible since their time responses are orders of magnitude smaller than 1 ms.

To avoid parasitic coupling between the PLLs, which may introduce additional spurious signals, one has to use proper signal isolation and buffering. It is also preferable that the ratio of the two phase-comparator frequencies is sufficiently far from any ratio of small integers.

Fig. 5 shows the spectrum of the (output) synthesized signal for five consecutive frequencies. The achieved frequency step is 542 Hz.

The parameters of the synthesizer, resulted in the spectra of Fig. 5, are listed in Table III.

TABLE III
PROGRAMMING PARAMETERS AND FREQUENCIES CORRESPONDING TO THE GRAPH IN FIG. 5

	n_1	n_2	n	f_1	f_2	$f_1 - f_2$
D_1	-2	27	-2406	98 866 666	95 176 151	3 690 514
D_2	-15	7	-2405	98 000 000	94 308 943	3 691 056
D_3	52	110	-2404	102 466 666	98 775 067	3 691 598
D_4	39	90	-2403	101 600 000	97 907 859	3 692 140
D_5	26	70	-2402	100 733 333	97 040 650	3 692 682

D. DFS Case 2 : $f_{\text{out}} = f_1 + f_2$

The low-pass filter in the mixer, Fig. 4, is replaced by an eighth-order 3-dB ripple Elliptic bandpass one that removes the off-band spurs from the mixing and the harmonics of the PLLs leaking through the mixer. The rest of the hardware, including the low-pass filters LPF1 and LPF2, remains the same.

From Section II, we know that the ranges of n_is are symmetric with respect to zero. Also, the widths of the frequency ranges of DFS schemes 1 and 2 are the same. So, we can use the same parameters B_1, B_2, C_1, C_2 (R_1, R_2, Q) as before to program the pre-scalers and output dividers for DFS scheme 2 where $f_{\text{out}} = f_1 + f_2$. We only need to select $\bar{n}_i$s to set the central output frequency $\bar{f}_{\text{out}}$.

The choice of $\bar{n}_1 = 1762$ and $\bar{n}_2 = 1903$ implies central output frequency $\bar{f}_{\text{out}} = 199\ 981\ 572$ Hz, $\bar{f}_1 = 117\ 466\ 667$ Hz, and $\bar{f}_2 = 82\ 514\ 905$ Hz (note from (3) that for every $m = 0, \pm 1, \pm 2, \ldots$, the choice $\bar{n}_1 = 1762 + mR_1$ and $\bar{n}_2 = 1903 - mR_2$ leads to the same $\bar{f}_{\text{out}}$).

Equation (16) gives the values of f_{out} synthesized using the above choices

$$f_{\text{out}} = 199\ 981\ 572\ \text{Hz} + n \cdot 542\ \text{Hz}$$
$$n = -9840 \ldots 9840.$$

The frequency ranges of the PLLs and that of the output signal, along with their resolutions (steps), are summarized in Table IV. The steps are the same as in DFS scheme 1 and the ranges are shifted but not scaled. The SFDR is $\sim$68 dB (recall the SFDR specs of CDCE706). Settling time is the same as before. Again, the settling delay due to the filters is negligible.

TABLE IV
FREQUENCY RANGES & RESOLUTIONS OF SIGNALS (HZ)-DFS 2

	Min	Central	Max	Step
f_{in}	-	64.000 000	-	-
f_1 (PLL 1)	112 133 333	117 466 667	122 800 000	66 667
f_2 (PLL 2)	77 181 572	82 514 905	87 848 238	43 360
f_{out} (DFS)	194 648 238	199 981 572	205 314 905	542

IV. CONCLUSION

The application of DFS methodology has been presented through the high-level design and frequency planning of two forward two-PLL DFS synthesizers. The two synthesizers achieved frequency resolution about 100 times better that their constituent PLLs without compromising hopping speed performance or spectral purity.

REFERENCES

[1] W. F. Egan, *Frequency Synthesis by Phase Lock*, 2nd ed. New York: Wiley, 1999.
[2] V. Manassewitsch, *Frequency Synthesizers*, 3rd ed. New York: Wiley, 1987.
[3] P. P. Sotiriadis, "Diophantine Frequency Synthesis," *IEEE Trans. Ultrason., Ferroelectr., Freq. Contr.*, vol. 53, no. 11, pp. 1988–1998, Nov. 2006.
[4] P. P. Sotiriadis, "Diophantine Frequency Synthesis; The Mathematical Principles," in *Proc. IEEE Int. Symp. Circuits Syst. 2007*, pp. 3107–3110.
[5] D. E. Flath, *Introduction to Number Theory*. New York: Wiley, 1989.