Turk J Elec Engin, VOL.14, NO.3 2006, © TÜBİTAK

Digital Fractional Frequency Synthesizer Based on Counters

Milan STORK
Department of Applied Electronics and Telecommunications Faculty of Electrical Engineering
University of West Bohemia, P.O. Box 314, 30614 Plzen, CZECH REPUBLIC
e-mail: stork@kae.zcu.cz

Abstract

This paper describes the architecture of a new pure digital frequency synthesizer based on pulse generators, counters and a register. The technique described here is much simpler than other methods. The synthesizer presented is suitable for the design of VLSI architectures or for programmable large-scale integration circuits.

Key Words: *Frequency synthesizer, phase locked loop, counter, register*

1. Introduction

A frequency synthesizer can be described as an active electronic device that accepts a reference frequency and then generates one or more new frequencies as defined by a control word. Modern electronic and telecommunication systems demand frequency synthesizers of high resolution, wide bandwidth and fast switching speed. Conventional frequency synthesis techniques in use today may be classified as the following 3 types:

a) Phase-locked loop (PLL) based, or "indirect"

b) Mixer / filter / divide, or "direct analog"

c) Direct digital synthesis (DDS)

Each of these methodologies has advantages and disadvantages. Direct analog synthesis uses the functional elements of multiplication, division and other mathematical manipulation to produce the desired frequency, but this method is a very expensive. DDS uses logic and memory to digitally construct the desired output signal. On the output, a digital-to-analog (D/A) converter is used to convert the digital signal to analog domain. PLL-based frequency synthesis has been widely used in industry. However, one of the major difficulties associated with the PLL-based technique is that a PLL with a wide frequency range cannot be achieved easily. In addition, fast switching is difficult to achieve. Typically, the output frequency step size of this method is the reference frequency. With fractional-N synthesis technique [1], finer frequency control can be achieved; however, these systems typically have very narrow bandwidth.

In this paper, a new simple architecture of digital frequency synthesizers with square wave output is presented. The synthesizer described is the most suitable for the design of VLSI architectures or for programmable large-scale integration. On the other hand, this synthesizer has the disadvantage of low output frequency, but this can be overcome by using this synthesizer together with a phase-locked loop.

The aim of frequency synthesis is to generate an arbitrary frequency, f_X, from a given standard frequency, f_S; in other words, to solve equation (1):

$$f_X = k_X * f_S \tag{1}$$

where k_X in the simplest case is a fraction formed by small, relatively prime integers. That is,

$$k_X = X_1/Y_1 \tag{2}$$

and the synthesizer is reduced merely to a chain of one frequency divider and one multiplier. If X_1 and Y_1 in (2) are products of small prime numbers, the synthesizer may be realized by a chain of frequency multipliers and dividers. However, there are difficulties with hardware solutions, mainly generation of spurious signals and frequent enhancement of the phase noise level.

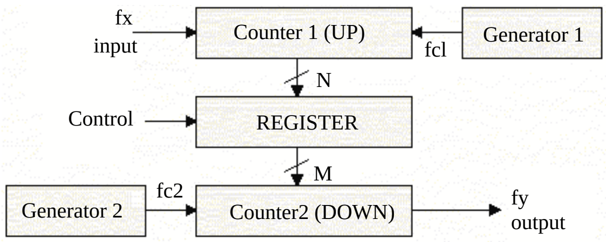

Figure 1. Block diagram of the digital synthesizer.

2. Principles of the New Synthesizer

In Figure 1, there is a block diagram of the digital frequency synthesizer [2]. It consists of Counter 1, which counts up frequency f_{C1} gated by input frequency f_X. Parallel output from Counter 1 is connected to Register input, and Register output is connected to preset inputs of Counter 2, which counts down frequency f_{C2}. On the output of this Counter 2 there is frequency f_Y. It is expected that $f_{C1} > f_X$. Number $C1$, which is stored in Counter 1 during the period of the f_X, is given by (3):

$$C1 = f_{C1}/f_X \tag{3}$$

This number is written in the Register, where this value can be changed by the Control to $C2$:

$$C2 = g(C1) \tag{4}$$

where $g(.)$ denotes some function of $C1$.

Number C2 is given by (5):

$$C2 = f_{C2}/f_Y = g(C1) = g(f_{C1}/f_X) \tag{5}$$

Output frequency f_Y can be expressed from (5) by (6):

$$f_Y = f_{C2}/g(f_{C1}/f_X) \tag{6}$$

When, for example, function $g(.) = 1/k_1$ (which can be simply realized by shifting a binary number in the Register), the output frequency f_Y is given by (7):

$$f_Y = f_{C2} * k_1 * f_X/f_{C1} \tag{7}$$

Equation (7) shows that output frequency f_Y is a product of frequency $f_{C2},, k_1$ and input frequency f_X divided by frequency f_{C1}. All of these parameters can be individually set. The length of the counters and registers must be sufficient to prevent overrun. If the binary counter is expected, then the minimal length L of the Counter 1 [bit] is given by (8):

$$L \Rightarrow Ceil(log_2(f_{C1MAX}/f_{XMIN}))[\text{bit}] \tag{8}$$

where f_{C1MAX} and f_{XMIN} are maximal clock and minimal input frequency and the *Ceil* function converts a numeric value to an integer by returning the smallest integer greater than or equal to its argument. In Figure 2, the synthesizer is shown as a building block.

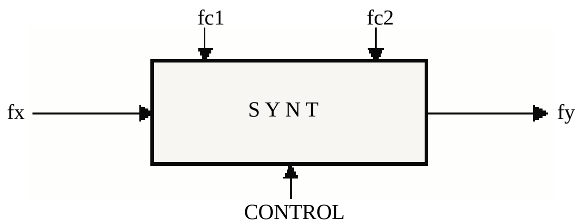

Figure 2. The digital frequency synthesizer as a building block.

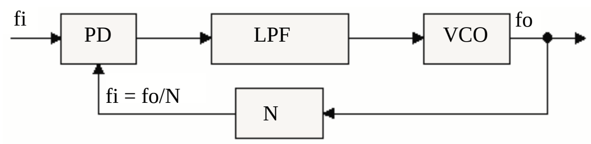

Figure 3. Block diagram of the basic Phase-Locked Loop. PD - phase detector, LPF - low pass filter, VCO -voltage controlled oscillator, N - frequency divider by N (N is integer number).

3. Digital Synthesizer and Phase Locked Loop

The phase-locked loop (PLL) works as a feedback system as shown in Figure 3 [3]. The task of the PLL is to maintain coherence between the input (reference) signal frequency, f_i,, and the respective output frequency, f_O, via the phase detector (PD) [4]. When the PLL locks onto a reference signal the output frequency is given by (9):

$$f_O = N * f_i \tag{9}$$

where N is an integer divide number of the divider.

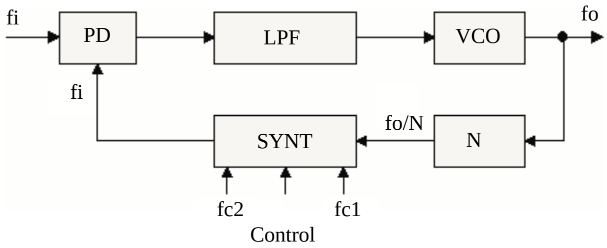

Figure 4. The Phase-Locked-Loop with the digital frequency synthesizer placed in feedback

Normally, frequency dividers can only produce integer divide ratios (N is an integer). Fractional division is accomplished by alternating the instantaneous divide number between N and $N+1$, but this causes phase modulation on the VCO [5]. Therefore a different complicated technique is used for correction of this error [6]. In Figure 4, the SYNT circuit is used in PLL [7]. Frequency on the SYNT input is f_O/N and frequency on the SYNT output is given by (10):

$$f_i = f_{C2} * k_1 * f_O/(f_{C1} * N) \tag{10}$$

From (10) we can derive the frequency of the voltage controlled oscillator, which is shown in (11):

$$f_O = f_{C1} * N * f_i/(f_{C2} * k_1) \tag{11}$$

When number C2 in the register is given by (12) (binary number C1 is multiplied by m_1, e.g., the register is shifted to the left, instead of divided by k_1), the frequency of the voltage controlled oscillator is given by (13):

$$C2 = m_1 * C_1 \tag{12}$$

$$f_O = f_{C1} * m_1 * N * f_i/f_{C2} \tag{13}$$

From (13) we can see that output frequency f_O is a function of integer m_1, N and clock frequencies f_{C1}, f_{C2} [8-11].

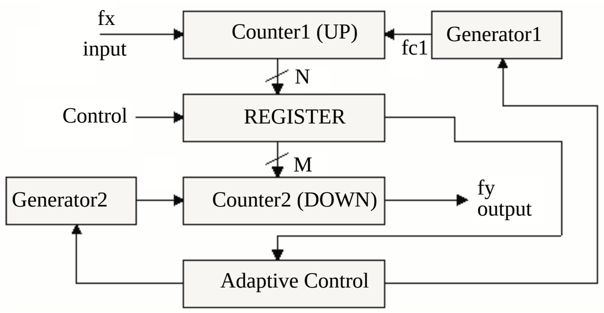

Figure 5. Possibility of the adaptive control using in the digital frequency synthesizer.

4. Error Reduction in Synthesizer

The digital synthesizer shown in Figure 1 has the following disadvantage. When the numbers in counters are small (numbers are integers), the output frequency is not accurate. This error can be reduced by adaptive control as shown in Figure 5.

Figure 5 is the almost the same as Figure 1; only adaptive control is added. The adaptive control block reads the contents of the Register. When the number is too small, the frequency of Generator 1 is multiplied and also the frequency in Generator 2 is multiplied, so that the ratio of $f_{C1}/f_{C2} = \mathit{constant}$. On the other hand, if the number in the Register is too big, the frequencies of both generators are divided by the same number.

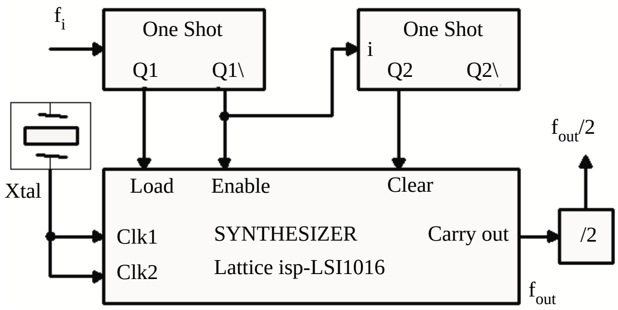

Figure 6. Example of the digital frequency synthesizer realized by using Lattice isp-LSI1016 IC's, oscillator and 2 one shot devices. In this experiment, Clk1 and Clk2 were connected together.

5. The Digital Synthesizer Main Advantages and Disadvantages

The digital synthesizer described (Figure 1) has the following disadvantages:

a) Accuracy depends on integer number in Counters

b) Not suitable for high output frequency

c) Only square wave output

The main synthesizer advantages are:

a) Pure digital architecture

b) Wide range

c) No setting problems

d) Stable

e) Fast response

f) Easily realized by programmable logic array

g) Easily reprogrammable and reconfigurable

h) Microcontroller adaptive control can be simply added for improving quality.

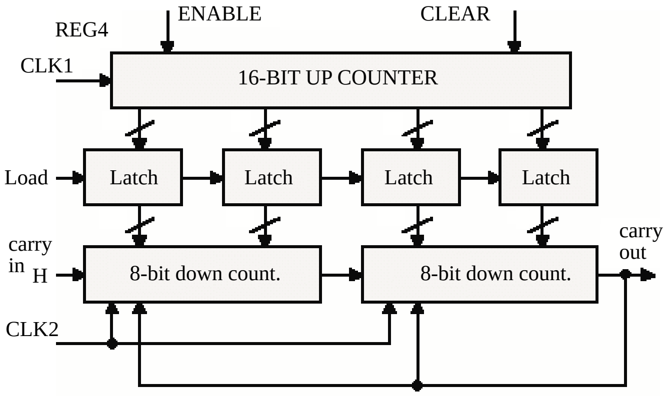

Figure 7. Internal block diagram of the digital frequency synthesizer based on programmable logic Lattice isp-LSI1016.

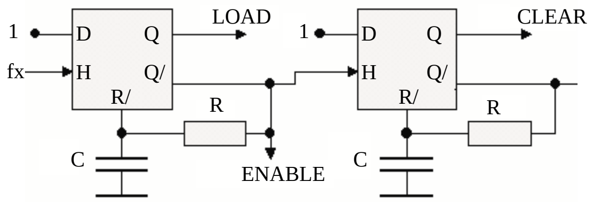

Figure 8. The dual one shots circuit used in experimental 16-bit digital frequency synthesizer.

6. The Experimental Results

The digital synthesizer was designed and built according to the above discussion. The synthesizer requires, e.g., 1 Lattice ispLsi 1016 device (in-system programmable large-scale integration circuit), X-tal oscillator and 2 peripheral one-shot devices (Figure 6). An internal block diagram of the 16-bit synthesizer is shown in Figure 7. The connection of the 2 one-shot devices is shown in Figure 8. For device testing, $f_{C1} = f_{C2}$ = 31.111 MHz and $k_1 = 1$, and so according to relation (7) the ideal output frequency is:

$$f_Y = f_X$$

A photograph of the PC board of the synthesizer is shown in Figure 9. Two, different delays one-shot circuits were tested. For delay of 0.6 μs (load + clear) the input frequency, f_x, and output frequency, f_y, were measured and C1 number in the up-counter and C2 number in the down-counter were computed and the number difference dif = C1 - C2 was also computed. The results are shown in Table 1. It can be seen that the differences are constant and the error in frequency can be easily corrected. For delay of 60 ns (load + clear) input and output frequencies are the same to within 1 MHz. For frequency 1 to 3 MHz, the results are shown in Table 2. For $f_{C1} = f_{C2}$ = 31.111 MHz, the maximal input frequency is approx. 3.5 MHz for good function. Minimal input frequency (to avoid an overflow of the 16-bit counter) is 476 Hz.

The experimental digital synthesizer was also used in PLL feedback to produce a fractional PLL. The output frequency spectrum is shown in Figure 10.

At the present time, the digital synthesizer was also designed in VHDL language. This design was realized and tested on an Altera FPGA development board with an EP20K200E device. The clock signals f_{C1} and f_{C2} are driven by a 33.3 MHz free running oscillator. The whole design consumes only 6% of available logic cells. Despite the low demand of the logic cells, counter 1 is 24 bits long and counter 2 is

32 bits long. This width provides the input frequency range from 2 Hz to 6.2 MHz (for a 33.3 MHz clock). The 8 bit difference between the width of counters allows one to divide or multiple output frequency up to the 8th power of 2. Fine tuning of the output frequency is provided by the combinational logic for adding or subtracting 23 bit integer numbers to the content of the register.

Table 1. Input and output frequencies for 600 ns delay (load + clear), fx - input frequency, fy - output frequency, C1, C2 are the numbers in Counter1 and Counter 2, diff - counters difference (C1 - C2).

fx [Hz]	fy [Hz]	C1	C2	diff
1502	1502	20713	20713	0
2010	2014	15478	15460	18
4008	4016	7762	7745	17
6004	6026	5181	5162	19
10008	10068	3108	3090	18
20004	20258	1555	1535	20
40000	40980	777	759	18
100000	106600	311	291	20

Table 2. Input and output frequencies for 60 ns delay (load + clear).

fx Input	[kHz] frequency
fy Output	[kHz] frequency
2	2
6	6
10	10
100	100
400	400
800	800
1082.1	1083.0
2020	2032
3275	3276

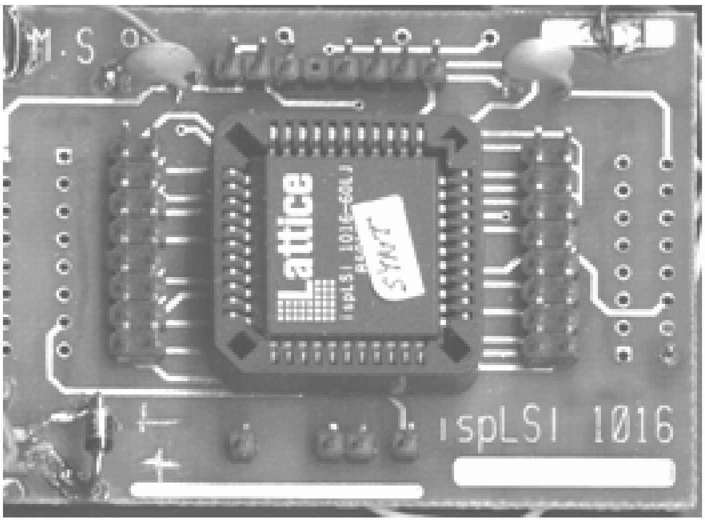

Figure 9. PC board of experimental digital synthesizer. Based on one Lattice ispLsi 1016 device.

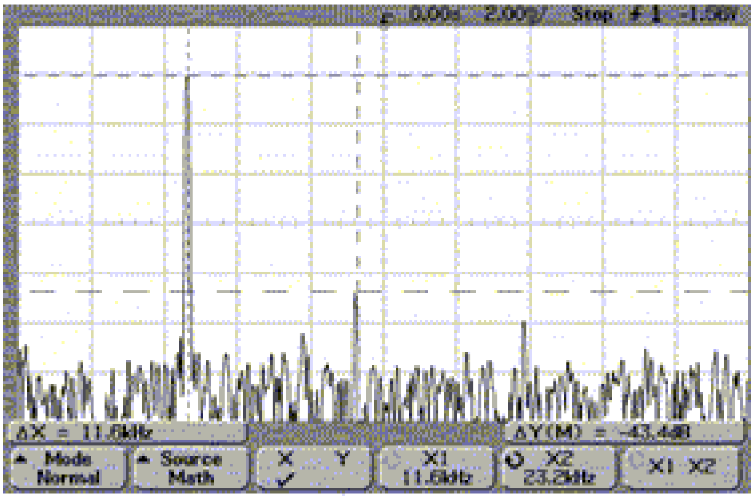

Figure 10. The output signal frequency spectrum of the fractional PLL, based on digital frequency synthesizer.

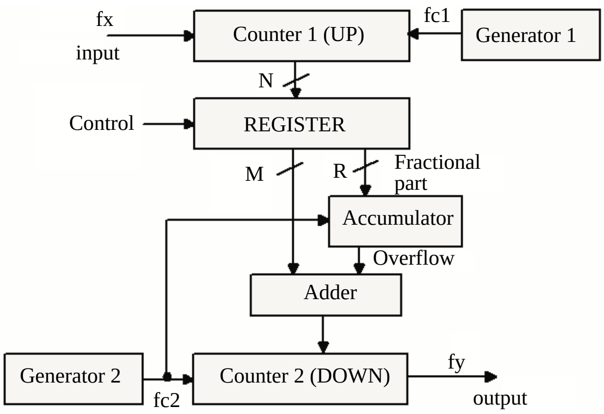

Figure 11. The fractional synthesizer with frequency error correction.

7. Frequency Error Correction

The error in the frequency synthesizer described (higher output frequency than ideal frequency value) is caused by removing the "fractional part" after the arithmetic operation on number C1, because number C2 can only be an integer (according to equation (4): $C2 = g(C1)$). This error can be corrected similarly, like in PLL fractional synthesizers. The sigma-delta modulation (accumulator and overflow output) is used in fractional PLL synthesizers. The digital accumulator and adder for frequency error correction is used for the synthesizer presented. The synthesizer block diagram with correction is shown in Figure 11. The architecture described in Figure 11 was simulated in Matlab. Examples of simulation results for input frequency multiplies by 11 and 5.7 are shown in Figure 12, 13 and 14. From this simulation it can be seen that a simple digital error correction system can be added for better synthesizer performance.

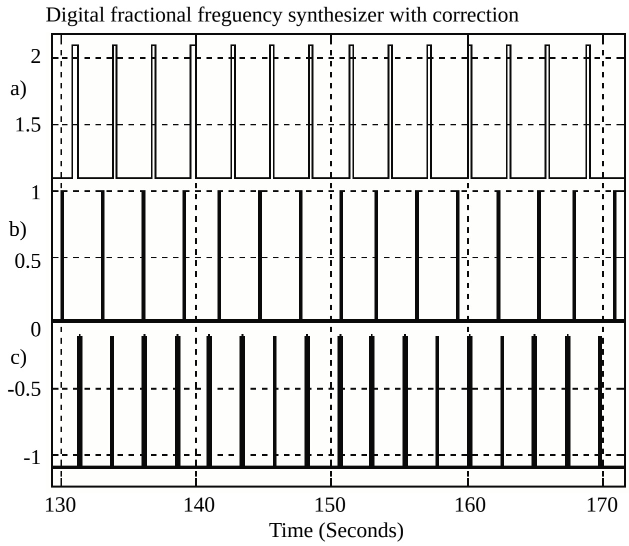

Figure 12. The synthesizer with frequency error correction example for input frequency multiplies by 11. a) pulses for ideal output frequency, b) output pulses of synthesizer with error correction, c) output pulses of synthesizer without error correction (the output frequency is higher than ideal).

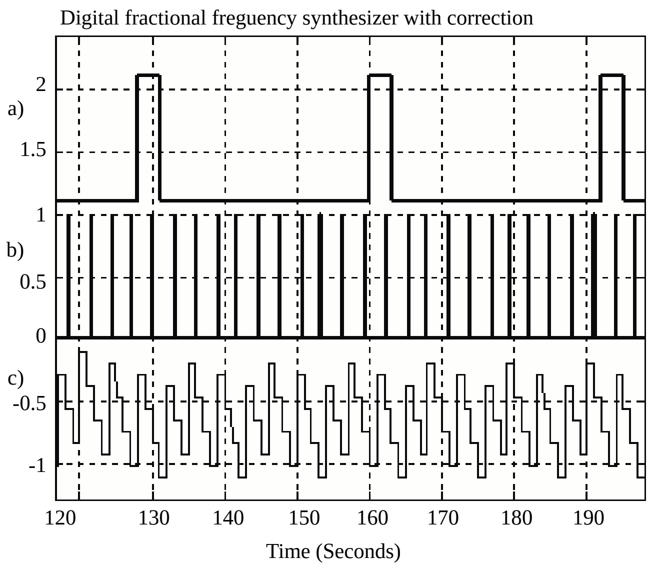

Figure 13. The synthesizer with frequency error correction example for input frequency multiplies by 11. a) pulses of input signal, b) output pulses of synthesizer with error correction, c) signal in digital accumulator for error correction.

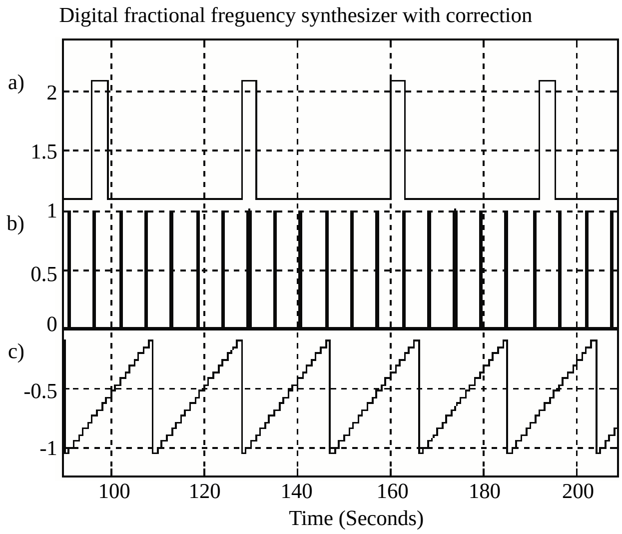

Figure 14. The synthesizer with frequency error correction example for input frequency multiplies by 5.7. a) pulses of input signal, b) output pulses of synthesizer with error correction, c) signal in digital accumulator for error correction.

8. Conclusion

The frequency synthesizer's form, which is the basis of most radio system designs, and their performance are often key to the overall operation. They are also an important building block in almost all digital and mixed signal integrated circuits as a clock multiplier, apart from the usual integer-N PLL implementation of the clock multiplier, where a voltage controlled oscillator is locked to a clean reference clock [9, 10]. The architectures based on a Delay-Locked Loop (DLL) have been successfully used recently as clock multipliers. The main disadvantage of conventional DLLs, however, is their limited phase capture range.

A new design technique for a frequency synthesizer is presented in this paper. The presented digital frequency synthesizer was patented in the Czech Republic. Schemes for direct and indirect synthesizers are shown and basic equations and a block diagram are also described. The digital frequency synthesizer was realized as a 16 bit device, by using Lattice ispLsi 1016 ICs (Counter max. frequency 80 MHz), and experimental results were introduced. It is important to note that delay caused LOAD and CLEAR can be easily corrected. The synthesizer can be best of all realized simply by using FPGAs or other types of programmable logic. The synthesizer is suitable for fractional frequency multiply, divide or for another frequency processing. The main advantage is that the synthesizer has a fully digital structure and also there are no stability problems. Moreover, the possibilities of wide range input frequency are important. The digital synthesizer can be used with a phase-locked loop for simple production of the fractional PLL. The frequency error correction system was also designed and simulations of system with error correction are presented in this paper.

Acknowledgment

This research work was supported by the Department of Applied Electronics and Telecommunication, University of West Bohemia, Plzen, Czech Republic.

References

[1] V.F. Kroupa, Direct Digital Frequency Synthesizers, New York, IEEE Reprint Press Book, 1998.

[2] M. Stork, A Digital Frequency Synthesizer, Czech Republic Patent, AO 256872, October 30, 1989.

[3] V.F. Kroupa, Theory of Phase-Locked Loops and Their Applications in Electronics (in Czech language), Academia Praha 1995.

[4] M. Curtin, P. O'Brien, Phase-Locked Loops for High-Frequency Receivers and Transmitters - Part 1, Analog Dialogue 33-3, 1999, pp. 1-4, Part 2, Analog Dialogue 33-5, 1999, pp. 1-5, Part 3, Analog Dialogue 33-7, 1999, pp. 1-5, Analog Devices, http://www.analog.com/

[5] D.D. Danielson, S.E. Froseth, A Synthesized Signal Source with Function Generator Capabilities, Hewlett-Packard Journal, Vol. 30, no. 1, January 1979, pp. 18-26.

[6] J. Chodora, A Digitally Corrected Fractional-N Synthesizer, Hewlett-Packard Journal, Vol. 44, no. 2, April 1993, pp. 44.

[7] J. Surber, L. McHugh, Single-Chip Direct Digital Synthesis vs. the Analog PLL, Analog Dialogue, Vol. 30, no. 3, 1996, pp.12-13, Analog Devices, http://www.analog.com/

[8] M. Stork, New Fractional Phase-Locked Loop Frequency Synthesizer Using a Sigma-Delta Modulator, 14th International Conference on Digital Signal Processing, DSP 2002, Santorini - Greece, July 2002, ISBN 0-7803-7503-3, Volume 1, pp. 367 - 370.

[9] M. Stork, Voltage to Frequency Converter, 12th IMEKO TC4 International Symposium, Electrical Measurements and Instrumentation, Zagreb, Croatia, September 2002, ISBN 953-96093-8-0, Proceedings Part 2, pp. 464 - 467.

[10] M. Stork, Frequency Synthesizer Based on Counters, 4th International Conference on Measurement, Smolenice, Slovak Republic, 2003, ISBN 80-967402-6-1, pp. 439-442.

[11] M. Stork, Counter Based Frequency Synthesizer, XVII IMEKO World Congress, Dubrovnik, Croatia, 2003, ISBN 953-7 124-00-2, pp. 933-937.