FPGA Implementation of Digital PLL-based Frequency Synthesizer with Programmable Frequency Dividers

Marieta Kovacheva[1] , Eltimir Stoumenov[2] and Ivailo Pandiev[3]

Abstract – **In this paper an approach for modelling and implementation of digital PLL (DPLL)-based frequency synthesizer with programmable division factors through the use of a VHDL is described. The functional elements of the DPLL structures are implemented in FPGA devices using the Matlab Simulink environment and the System Generator toolbox (from Xilinx Vivado® design suite). The experimental test that has been used for validating the FPGA configuration is based on the ZedBoard development board, which is built around the Xilinx Zynq-7020 SoC. The proposed device has wide-band frequency response (up to 1M*Hz*) and can operate with single supply voltage at 3.3V. Simulation and experimental results show good agreement with theoretical predictions.**

Keywords – **Mixed-signal circuits, Digital PLL, Numerically controlled oscillator (NCO), Frequency synthesizer, VHDL, FPGA.**

I. INTRODUCTION

The digital phase-locked loops – PLLs (DPLLs) are mixed-signal systems with feedback that include basically a voltage-controlled oscillator (VCO) or numerically controlled oscillator (NCO), phase detector (PD), and low-pass filter (LPF) within its loop. For the DPLL the frequency of the generated clock signal has to be synchronized (or locked) with the frequency of an external input digital signal [1, 2]. The DPLL systems are widely used in the measurement systems and telecommunications and are used for building of frequency synthesizers and synchronizers for grid connected inverters, FM demodulators, decoders, etc [1, 3, 4, 5, 6, 7, 8, 9]. Therefore of particular interest is the study of methods and techniques for creating digital PLLs suitable for various mixed-signal devices and systems.

While some approaches imply the use of ASICs for digital PLL, which requires considerable design time and costs, FPGAs have been recently raised as flexible, fast-prototyping and comparatively economical solutions for design of complex mixed-signal processing systems. In the recent years a new class of FPGA devices appeared on the market. The so-called "All Programmable SoC" contains standard FPGA logic and a hardwired CPU cores (dual core ARM Cortex-A9) on the same IC. This allows the user to take advantage of the highly developed ARM technology and the unprecedented computing power of the FPGA technology. The two biggest FPAGs manufacturers – Altera and Xilinx, offers such devices on the market. There are two major differences between the Xilinx and Altera devices [10]. On the one hand, for the device boot-up Xilinx uses a processor centric approach. This means that by executing a hard coded program called BootROM, the processor configures the processor system (PS) and the programmable logic (PL) sections of the Zynq-7000 devices [11]. In other words the FPGA section cannot be configured without the interaction of the PS. On the other hand, the Altera devices allow three different ways for system configuration on boot-up: (1) the processor configures the PS and the PL section (identical to Xilinx approach) [12]; (2) the FPGA section configures the processor system; (3) both systems could be configured independently. This is a major advantage for safety-critical systems. All types of Spartan -3 and -6 FPGAs [13, 14] have built PLL blocks with a working frequency range from few tens of kilohertz to several hundred of megahertz. The electronic circuits implemented with them are optimized mainly for realization of frequency synthesizers.

The aim of this work is to develop a frequency synthesizer with programmable division factors that operates up to 1M*Hz* through the use of digital PLLs. The proposed electronic circuit is based on FPGA Xilinx Zynq-7020 programmable SoC and by using PmodAD1™ analog-to-digital converter (ADC) module.

II. PRINCIPAL OF DPLL OPERATION

In general the DPLLs are mixed-signal electronic systems that produce a clock output signal which is locked or synchronized with the external input signal. It is possible to have a constant value of the phase shift ϕ_0 between the input signal and the clock output signal, but when locked, the frequencies must exactly match, i.e.:

$$\phi_{out}(t) = \phi_{in}(t) + \phi_0 \text{ and} \quad (1)$$

$$\omega_{out}(t) = \omega_{in}(t). \quad (2)$$

The PD of the DPLL circuits compares the phase at each input and generates a signal U_ϕ proportional to the phase difference between the clock output signal and the external input signal.

$$U_\phi(t) = K_D[\phi_{out}(t) - \phi_{in}(t)]. \quad (3)$$

[1]Marieta Kovacheva is with the Faculty of Electronic Engineering and Technologies at Technical University of Sofia, 8 Kl. Ohridski Blvd, Sofia 1000, Bulgaria, E-mail: m_kovacheva@tu-sofia.bg.

[2]Eltimir Stoimenov is with the Faculty of Electronic Engineering and Technologies at Technical University of Sofia, 8 Kl. Ohridski Blvd, Sofia 1000, Bulgaria, E-mail: e_stoimenov@tu-sofia.bg.

[3]Ivailo Pandiev is with the Faculty of Electronic Engineering and Technologies at Technical University of Sofia, 8 Kl. Ohridski Blvd, Sofia 1000, Bulgaria, E-mail: ipandiev@tu-sofia.bg.

where K_D is the PD gain in *V/radians*.

There are two basic types of PDs, a XOR gate and a phase frequency detector (PFD) [3]. The XOR PD is simply an exclusive-OR gate. The basic advantages of XOR-based DPLLs are the good noise rejection and wider working bandwidth [1, 3]. The gain of the XOR PD may be written as $K_D = V_{DD}/\pi$.

The output voltage U_ϕ of the PD is filtered by LPF, removing the high frequency harmonics. Actually, the LPF determines the average value of the U_ϕ. To ensure sufficient stability of the DPLL most commonly first-order LPF with one pole-zero pair is used [1, 3]. Thus, for high frequencies the slope of the complete phase transfer function is reduced to $-20dB/dec$, while the phase margin becomes more than 45°.

The frequency of the VCO in the DPLLs can be varied by means of the control voltage U_f, obtained by the LPF, according to the following general formula

$$f_{out} = f_0 + K_0 U_f. \quad (4)$$

where K_0 is the VCO gain in Hz/V and f_0 is the VCO centre frequency.

Numerically controlled oscillators or digital controlled oscillators (DCO) can easily be obtained by VCOs, which are controlled by digital-to-analog converter (DAC) [1, 9].

The DPLLs allow a fast lock time to be achieved and are attractive for clock generation on high performance microprocessors. An object of analysis and modeling in this paper is the DPLL, employing XOR PD. A basic block diagram of a DPLL using digital PD is shown on Fig. 1. In locked state for low frequencies the small-signal transfer function is [2, 3]

$$T(p) = \frac{N}{M}\frac{K_0 K_D K_N T_{LP}(p)}{p + K_0 K_D K_N T_{LP}(p)} = \frac{N}{M}\frac{K T_{LP}(p)}{p + K T_{LP}(p)}, \quad (5)$$

where *N* and *M* are the feedback loop division factors of the frequency dividers, $K = K_0 K_D K_N$ is the gain of the loop network, $T_{LP}(p) = \frac{1 + p\tau_2}{1 + p(\tau_1 + \tau_2)}$ ($\tau_1 = R_1 C$ and $\tau_2 = R_2 C$) is the transfer function of the LPF with one pole-zero pair.

The parameter K_N of the divider is equal to $1/N$.

After substituting the formula for the transfer function of the LPF into (5) for the phase transfer function of the DPLL is found

$$T(p) = \frac{N}{M}\frac{K(1 + p\tau_2)/(\tau_1 + \tau_2)}{p^2 + p(1 + K\tau_2)/(\tau_1 + \tau_2) + K/(\tau_1 + \tau_2)} =$$

$$= \frac{p(2\xi\omega_n - \omega_n^2/K) + \omega_n^2}{p^2 + 2\xi\omega_n p + \omega_n^2}. \quad (6)$$

After comparison of the left and right sides of the above equation the following formula was obtained: $\omega_n = \sqrt{K/(\tau_1 + \tau_2)}$ – natural frequency and $\xi \approx \omega_n \tau_2 / 2$ – damping factor.

For the impulse VCO output signal the frequency lock range of the DPLL is given by [9]

$$\Delta\omega_L \approx \pm 2 K_D K_0 U_{im} U_{om} / \pi. \quad (7)$$

where U_{im} and U_{om} are the amplitudes of the input and output signals.

Therefore, the capture range for simple RC group is $\Delta\omega_C \approx \sqrt{\omega_L / \tau}$ ($\tau = RC$ is a time constant of the filter). Lower time constant τ determined wider capture range.

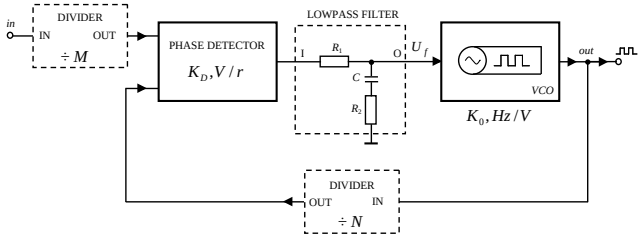


Fig. 1. Block diagram of a digital PLL-based frequency synthesizer

A particularly important application of the digital PLL is the frequency synthesizing. A frequency divider with *N* and *M* division factors is connected to each of the two inputs of the PD, as shown on Fig. 1. In locked state for the frequency of the VCO is obtained

$$f_{out} = (N/M) f_{in}. \quad (8)$$

III. FPGA CONFIGURATION OF DPLL

The synthesis of the digital PLL circuits is strongly dependent by the incoming signal features, such as amplitude, frequency and noise level. This means that the implementation of the circuits should allow changing the gain to cope with a variation in the amplitude of the signal, and it should be able to change the pole frequency of the LPF and time constant of the integrators to different noise levels. These requirements are easily achieved using a FPGA to synthesize those mixed-signal devices.

A. A NCO implementation

The block diagram of the created NCO employing SYSGEN (System Generator toolbox from Xilinx Vivado® design suite) blocks is given on Fig. 2. Here the input quantity is represented by a numerical value N_{in} and the reference value is represented by N_r. The N_{in} and its negative value are applied to the inputs of the multiplexer mux_sw. This block acts as a controllable switch which commutates the N_{in} and $-N_{in}$ to the integrator input. The integrator is realized with a standard accumulator block configured in summing mode.

The signal from the integrator then is applied to a comparator with hysteresis – comp_hys block. This block is realized with standard relational element ($a \geq b$), SPDT type of multiplexer ($2 \rightarrow 1$) and one delay block (z^{-1}). The comparator reference values (N_r and $-N_r$) are applied to the multiplexer inputs. For symmetrical output signal with duty cycle equal to 50% the absolute value of the two reference values should be equal. For proper operation N_r should be much greater than input quantity N_{in}.

The output of the comp_hys block controls the address input of the mux_sw block and serves as a system output.

The output signal frequency of the NCO can be calculated by using the following equation

$$f_{out} = N_{in} / 4T_S N_r , \quad (9)$$

where T_S is the period of the NCO clock signal ($T_S = 1/CLK_{NCO}$).

After building and simulation phase, the test of the NCO within Matlab Simulink, is exported in the Xilinx Vivado® programming system. In this programming system logical synthesis and configuration of the FPGA IC are performed. In the process of the logical synthesis the VHDL description is transformed into a set of logic gates and flip-flops.

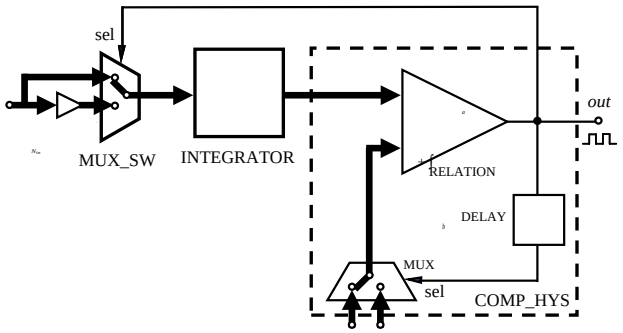


Fig. 2. Block diagram of a digital NCO employing SYSGEN blocks

B. A frequency divider implementation

The frequency synthesizer is realized by adding a frequency divider in the DPLL feedback. The divider is implemented by using black box, which is a standard block from the Xilinx blockset in Matlab Simulink. The black box allows realization of any desired function by using VHDL description.

The divider has a programmable coefficient N' and its output frequency is given by

$$f_{out} = f_{in} / 2N' . \quad (10)$$

As it can be seen, the divider could be multiple of the input, only even one. The duty cycle of the output signal is 50 %, which ensures maximal performance of the DPLL.

The divider realization is based on the classical approach with a counter working on the rising edge on the input signal. When the value of the counter reaches the coefficient N' the output signal flips. The divider is realized and simulated in the Xilinx Vivado® software environment.

IV. Experimental Results and Discussion

The workability of the proposed Digital PLL – based frequency synthesizer on Fig. 1 is proven through the simulation results using the simulator built in Matlab Simulink environment and also through the experimental test from the circuits configured on prototype board. The experimental test that has been used for validating the FPGA configurations is based on the ZedBoard development board, which is built around the Xilinx Zynq-7020 SoC device. The supply voltage is equal to 3.3V.

The oscillation frequency f_{out} of the NCO prototype versus the input quantity is plotted on Fig. 3. The value of the N_r is set equal to 1000. The master clock frequency of the system is equal to 100M*Hz*. When the N_{in} varies from 1 to 20 the oscillation frequency changes from 25k*Hz* to 500k*Hz*. The error δ is not higher than 1.5% in the whole range of the input quantity.

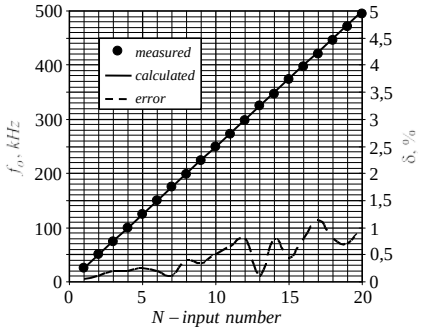
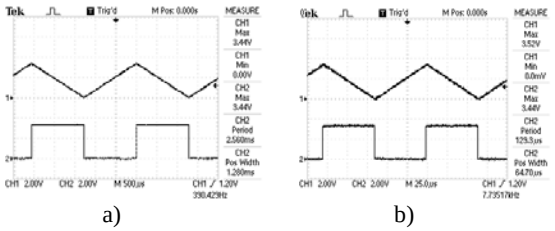


Fig. 3. The oscillation frequency versus the input voltage at 25k*Hz*/*bit* conversion sensitivity (duty cycle – 50%)

Fig. 4a and Fig. 4b show the NCO output signal and the integrator output signal. The integrator output signal is applied to a 8-bit DAC first. As the sampling frequency of the DAC is limited the NCO clock frequency is reduced by a factor of 8. The master clock frequency is equal to 100M*Hz* / 8 or equal to 12.5M*Hz*. In this way the output signal from the integrator block could be properly observed with oscilloscope. Channel 1 (CH1) on Fig. 4a shows the integrator output signal (normalized to 0-3.3*V*) and channel 2 (CH2) shows the NCO output signal. The measurement is taken for $N_{in} = 1$. Fig. 4b is the same as Fig. 1, but with $N_{in} = 20$.



a) b)

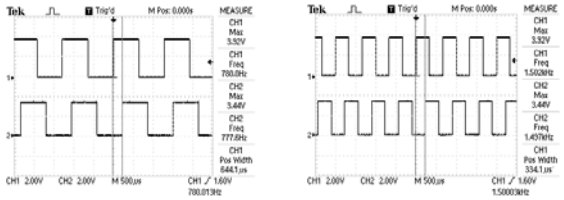
Fig. 4. The output waveforms of the NCO at value of the input digital number equal to: a) 1 and b) 20.

Below an experimental study which proves the proper operation of the DPLL system is described. The DPLL circuit is consisted by a XOR component acting as a PD, analog LPF

and a NCO. The master clock signal of the system is chosen $f_{CLK} = 390kHz$ (derived by the onboard 100M*Hz* quartz generator divided by a factor of 256). The PD gain is given by $K_d = 3{,}3V/\pi$. In order to obtain maximum locking range of the PLL the input signals of the PD should be with 50% duty cycle. The LPF is purely analog and it is composed by a simple RC group with $R = 15.8k\Omega$ and $C = 100nF$ at corner frequency equals 100*Hz*. For $f_{CLK} = 390kHz$ the NCO gain is equal to $98Hz/bit$, which corresponds to $640Hz/V$.

The LPF averages the PD output signal and forms the NCO control quantity. In order to obtain the digital value of the LPF output signal a PmodAD1™ 12-bit ADC is used. The ADC output code is normalized in a way that NCO output frequency equals f_0 when the input voltage is equal to $0{,}5V_{CC}$. The normalization is achieved by multiplication with constant block.

Fig. 5a and Fig. 5b shows the input signal (CH1) and output signal (CH2) of the DPLL. As it can be seen on Fig. 5a for the central frequency f_0 of the NCO, the phase shift between the input signal and the output signal is maintained at a constant value, indicating that the DPLL is in mode of synchronization. Furthermore, the frequencies of the input and output signals are approximately equal (the error is less than 0.5%). Fig. 5b shows the operation of the DPLL for input signal with frequency 1.5 *kHz* which is very close to the maximum of the frequency lock range. The lock and the capture range of the DPLL are measured $\Delta f_L = 1.7kHz$ and $\Delta f_C = 750Hz$, respectively.



a) b)

Fig. 5. The input and output signals: a) 780*Hz* and b) 1.5k*Hz*.

A validation check of the proposed DPLL-based frequency synthesizer is performed by comparison analysis between simulation results and experimental test of a frequency synthesizer circuit implemented in Xilinx Zynq-7020 SoC device. The programmable frequency divider connected within negative feedback network of the DPLL is with variable division factor *N* from 10 to 30, which produce the frequencies within the lock range. Moreover the input signals are with amplitude 5*V*, frequency 50*Hz* and a duty cycle equal to 50%. Waveforms of the input and output signals are analyzed with digital oscilloscope type TDS1012B. For measuring the frequency of the input and output signals is used frequency counter, model GFC-8010H. The maximum error between simulation results and experimental test is not higher than 1%.

V. Conclusion

In this paper a VHDL-based description of DPLL system has been presented. The created system is implemented as a hierarchical structure of blocks representing the basic elements of the DPLLs. For modeling and implementation of DPLL-based frequency synthesizers a VHDL model of a programmable frequency divider is developed. As well as a numerically controlled oscillator employing SYSGEN blocks is created.

The workability of the proposed system was proved by comparison of the simulation results, with the results of the experimental study of the breadboard circuits with the ZedBoard development board, built around the Xilinx Zynq-7020 SoC. The created DPLL system can be useful for design and modeling of various frequency synthesizers and synchronizers for control systems of PWM rectifiers, matrix converters and parallel power active filters, valid up to 1M*Hz*.

Acknowledgement

This paper is a part of a project under contract № 132pd0001-03/2013, which is funded by the research program of the Technical University of Sofia, Bulgaria.

References

[1] V. Tietze, Ch. Schenk, *Electronic circuits*. 2nd ed., Springer-Verlag: Berlin, Heidelberg, New York, 2008, pp. 1114-1125.

[2] D. Banerjee, D. *PLL Performance, simulation and design* handbook, 4th ed., National Semiconductors, 2006, pp. 6-18.

[3] R. Baker, *CMOS circuit design, layout, and simulation*, 3rd ed., Wiley-IEEE Press: New York, 2010, pp. 551-612.

[4] J. Segundo, L, Quintanilla, J. Arias, L. Enrıquez, J. Hernandez, J. Vicente, "A PLL-based synthesizer for tunable digital clock generation in acontinuous-time ΣΔ A/D converter," Integration, the VLSI Journal, vol. 42, pp. 24-33, 2009.

[5] H. Danneels, F. Piette, G. Gielen, "A novel PPL-based sensor interface for resistive pressure sensors," Procedia Engineering, vol. 5, pp. 62-65, 2010.

[6] J. Sell, A. Niedermayer, B. Jakoby, "A digital PLL circuit for resonator sensors," Sensors and Actuators A, vol. 172, pp. 69-74, 2011.

[7] J. Yuan, Q. He, Y. Hu, H. Li, L. Wang, W. Liu, Y. Xie, "Optimization analysis on the bandwidth algorithm for the phase-locked loop (PLL) in OUT," Optik, vol. 123, pp. 1681-1685, 2012.

[8] M. Antchev, I. Pandiev, M. Petkova, E. Stoimenov, A. Tomova, H. Antchev, "PLL for single phase grid connected inverters," International Journal of Electrical Engineering and Technology, vol. 4, No 5, pp. 56-77, 2013.

[9] M. Seifart, *Analoge Schaltungen*. 6 Auflage. Verlag Technik: Berlin, 2003; pp. 496-503.

[10] R. Vincke, A. Messiaen, J. Boydens, "Hybrid FPGA/Multi-core CPUs for Industrial Applications," Annual journal of electronics, Bulgaria, vol. 7, pp. 84-86, 2013.

[11] Xilinx, "Zynq-7000 All Programmable SoC – Technical Reference Manual," vol. 585, 2013.

[12] Altera, "AlteraSoC: Booting and Configuration," vol. 3, pp. 1-24, 2012.

[13] Using Digital Clock Managers (DCMs) in Spartan-3 FPGAs, Xilinx, Application Note: XAPP462 (v1.1), January 5, 2006.

[14] Spartan-3 Generation FPGA User Guide, Xilinx, Application Note: UG331 (v1.8), June 13, 2011.

[15] Zynq-7000 All Programmable SoC Overview, Xilinx, Application Note: DS190 (v1.6), December 2, 2013.