Modeling and Implementation of FPGA – Based Flexible Voltage-to-Frequency Converter

Eltimir Chavdarov Stoimenov, Ivailo Milanov Pandiev and Marieta Georgieva Kovacheva

Abstract - **In this paper an approach for modeling and implementation of a voltage-to-frequency converter (VFC) through the use of VHDL is described. The proposed converter is based on a charge-balanced conversion method. The VFC circuit consists of integrator, comparator, one-shot timer and two-input multiplexer used as a two-input analog switch that closes the positive feedback. The functional elements of the VFC structure are implemented in FPGA device using the Matlab Simulink® environment and the System Generator (SYSGEN) toolbox (from Xilinx Vivado® design suite). The created device has wide-band frequency response (up to 1M*Hz*) and can operate with single supply voltage at 3.3*V*. Simulation and experimental results show good agreement with theoretical predictions. An 8-bit analog-to-digital converter (ADC) is used to convert the input control voltage to a discrete time digital representation. This allows the usage of the designed VFC in mixed-signal systems.**

Keywords – **Mixed-signal circuits, Voltage-to-frequency converter (VFC), FM modulator, FSK, FPGA, System Generator**

I. INTRODUCTION

The voltage-to-frequency converters (VFCs) or voltage-controlled oscillators (VCOs) are widely used in analog PLLs, analog-to-digital conversion, precision frequency-to-voltage conversion, long-term integration, linear frequency modulation and many other mixed-signal functions [1-4]. The output signal of a VFC is pulse train at a frequency precisely proportional to the applied input voltage.

For the commercial VFCs, based on charge balanced conversion method, such as AD650 (from Analog Devices), LM2907 (from Texas Instruments) and LM331 (from Texas Instruments) the transfer function and the related electrical parameters are formed by the internal active building blocks and a group of external passive components. Furthermore, the accuracy of the electrical parameters is largely determined by the manufacturing tolerances and the temperature drift of the values of the external passive components.

A variety of VCO prototype circuits for mixed-signal processing, are available in the literature, moreover the trends are towards the development of specialized integrated circuits operating at higher frequencies ($>1GHz$) [5-7]. The majority of the published circuits of VCOs are with fixed sensitivity without demonstration of possible modification of this parameter. In many cases the voltage-to-frequency conversion of small signals (such as bio-signals) requires modification of the sensitivity according to the variation of the amplitude, the bandwidth or the noise level. To solve this problem, a field programmable gate array (FPGA) devices can be used. The FPGA are programmable CMOS integrated circuits, which proved to be 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 [8-10]. The so-called “All programmable SoC” contains standard FPGA logic and a hardwired CPU cores (e.g. dual core ARM Cortex-A9) on the same IC.

The aim of this work is to develop a VFC with programmable parameters that operates up to 1M*Hz* through the use of charge balanced conversion method. The proposed electronic circuit is based on FPGA Xilinx Zynq^{TM} -7020 programmable SoC [11] and by using PmodAD1^{TM} analog-to-digital converter (ADC) module.

II. PRINCIPLE OF CHARGE BALANCED VFC OPERATION

The VFCs with digital output are an important link between the analog and the digital domains. A literature review in the area shows that there are two types of VFCs or VCOs: (1) multivibrator and (2) charge balanced [3, 12, 13]. The charge balance VFC is more accurate than the multivibrator type and its output is a pulse train, not a unity mark-space square wave.

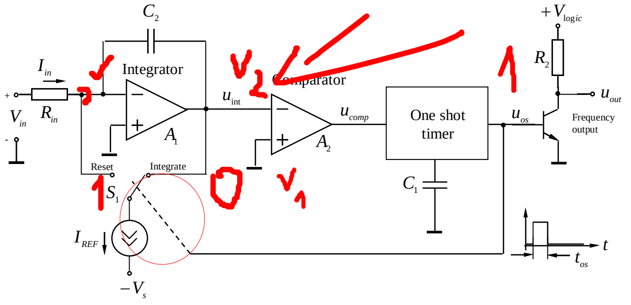


Fig. 1. Charge balanced VFC architecture [12].

A basic structure of a charge-balanced VFC is shown on Fig. 1. The integrator’s output voltage u_{int} is proportional to the charge stored in the capacitor C_2. The input voltage V_{in} forms an input current of $I_{in} = V_{in} / R_{\mathrm{int}}$. This current

E. Stoimenov is with the Department of Electronics and Electronics Technologies, Faculty of Electronic Engineering and Technologies, Technical University - Sofia, 8 Kliment Ohridski blvd., 1000 Sofia, Bulgaria, e-mail: e_stoimenov@tu-sofia.bg

I. Pandiev is with the Department of Electronics and Electronics Technologies, Faculty of Electronic Engineering and Technologies, Technical University - Sofia, 8 Kliment Ohridski blvd., 1000 Sofia, Bulgaria, e-mail: ipandiev@tu-sofia.bg

M. Kovacheva is with the Department of Electronics and Electronics Technologies, Faculty of Electronic Engineering and Technologies, Technical University - Sofia, 8 Kliment Ohridski blvd., 1000 Sofia, Bulgaria, e-mail: m_kovacheva@tu-sofia.bg

charges the capacitor C_2 and causes the integrator output voltage to ramp When V_{int} reach 0*V*, the comparator flips, triggering the one-shot timer. This connects internal current source I_{REF}, to the integrator input during the one-shot period t_{os}. This current causes the integrator output to ramp positively until the one-shot period ends. Then the operating cycle starts again. The oscillation is regulated by the balance of current (or charge) between the input current and the time averaged reference current. The equation of current balance is given by

$$I_{in} = I_{REF}\, f_{out} t_{os} \tag{1}$$

or

$$f_{out} = I_{in} / (I_{REF} t_{os})\,, \tag{2}$$

where f_{out} is the output oscillation frequency.

The analysis of formula (2) shows that the linear variation of the output frequency can be achieved by changing the input current. Moreover, the duty cycle of the output square wave can be changed by modifying the one-shot impulse duration – t_{os}.

The charge balanced VFCs have several important practical applications. For example, these types of VFCs are used in realization of the electronic measurement systems for sending analog signals over long distances without loss of accuracy, linear PLLs, frequency synchronizers and etc. The objective of this paper is to demonstrate a method for modeling and implementation of the well known charge-balanced VFC using FPGA devices.

III. Design Flow for Implementation of FPGA-based Electronic Circuits

The proposed design flow is based on the methods for mixed-signal circuits design and the techniques for FPGA synthesis and optimization, given in [9, 14, 15].

The design flow diagram is shown in Fig. 2. The diagram is a framework, which includes basic steps, intermediate products and relations between them, necessary for modeling, design and implementation of a given electronic circuit. The practical approach presented here is based on a Top-Down analysis approach. First, the designer should define the technical specification and create a block diagram of the system. Then it should be determined which blocks could be implemented in System Generator (SYSGEN) and which need to be created in some other way, e.g. by using VHDL, IP core and etc. Then a new project in Vivado® needs to be set up and a System generator (DSP design source) block is added to the project. Next, a functional model is created and simulated in the Matlab Simulink® environment. If the simulation results are satisfactory then the whole system could be build. If not, the designer should improve the simulation model and re-simulate it until good agreement is achieved.

When the whole system, including the Simulink® model plus the external blocks, is prepared a FPGA implementation and physical testing take place. Again, verification of the test results and system improvement is done iteratively until satisfactory results are achieved. Finally the system documentation is created.

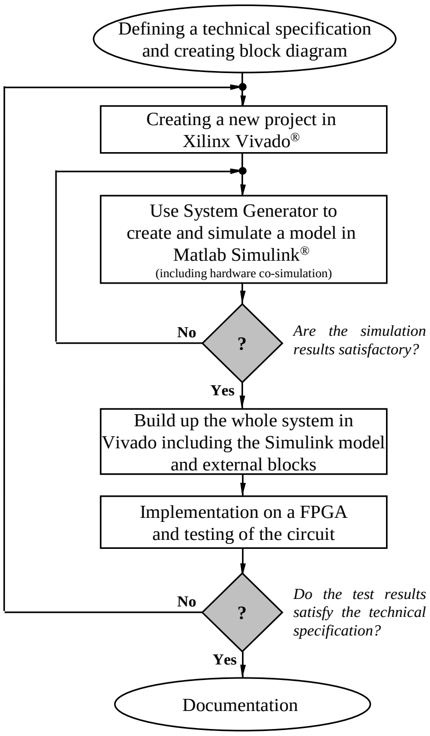

Defining a technical specification and creating block diagram
Creating a new project in Xilinx Vivado®
Use System Generator to create and simulate a model in Matlab Simulink®
(including hardware co-simulation)
No
?
Are the simulation results satisfactory?
Yes
Build up the whole system in Vivado including the Simulink model and external blocks
Implementation on a FPGA and testing of the circuit
No
?
Do the test results satisfy the technical specification?
Yes
Documentation

Fig. 2. Design flow for implementation of FPGA-based electronic circuits using System Generator.

IV. FPGA Configuration of VFC

The block diagram of the proposed VFC realized in SYSGEN (System Generator toolbox from Xilinx Vivado® design suite) is given on Fig. 3. It contains the following major blocks: the two *Gateway in* blocks – **input_voltage and reference_voltage** transfer the input voltages – V_{in} and V_{ref} into the VCO. The digital values for V_{in} and V_{ref} are obtained by two external 12-bit ADCs sampling with approximately 139 kSPS. In order to restore the values of the input and the reference voltages the **ADC1 ULSB** and **ADC2 ULSB** blocks multiply the incoming ADC code with the U_{LSB} of the ADCs. As the $V_{ref,ADC}$ for these ADCs is equal to 3.3V for U_{LSB} the following is obtained

$$U_{LSB} = V_{ref,ADC} / (2^{12} - 1)\,. \tag{3}$$

The **mux_sw** is realized with one multiplexer with two inputs and one output. The input voltage V_{in} is applied to the first input – *d0* and the difference between the input voltage and the reference voltage is applied to the second input – *d1*. The state of the **mux_sw** output is given by

$$output = \begin{cases} d0, when\ sel = \text{'0'} \\ d1, when\ sel = \text{'1'} \end{cases} \tag{4}$$

The output signal of **mux_sw** is applied to the integrator input. The integration is realized with standard accumulator block with the following transfer function

$$q(n) = \begin{cases} 0,\ if\ \ rst = \text{'1'} \\ q(n-1) + b(n-1),\ \ otherwise. \end{cases} \tag{5}$$

The arithmetic relational operator acts as a comparator which tracks the integrator output signal. If this signal drops below zero the **comparator** output is set to ‘1’. The

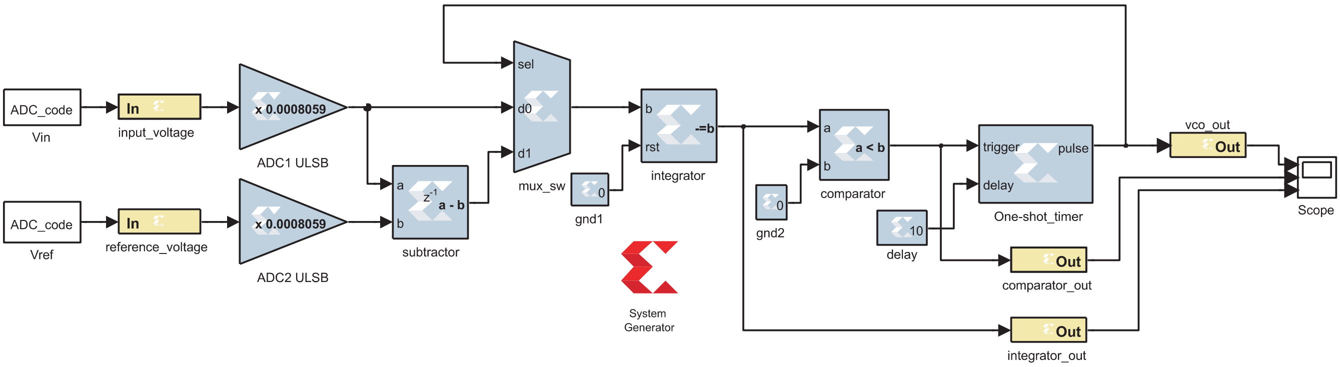


Fig. 3. Scheme diagram of a digital VFC employing SYSGEN blocks.

rising edge from the **comparator** activates the one-shot timer which produces a pulse with duration determined by the value in the **coefficient** block.

The **one-shot_timer** block is implemented by a black box and custom VHDL code. Fig. 4 shows the behavioral VHDL description of the **one-shot_timer**. The library clause and the use clause make all declarations in the packages electrical_systems, `std_logic_1164`, `ieee.numeric_std` and `UNISIM.Vcomponents` visible in the model. The VHDL model has four input ports (`CLK`, `CE`, `trigger` and `delay`) and one output port (`pulse`). All signals are of `std_logic` type, defined in package `std_logic_1164`. The `CLK` input is the master clock signal which is mandatory for the proper operation of the timer. The `CE` input port is not used, but it is required by the system generator when a VHDL code is imported in a black box. The proposed one-shot timer model is composed by an *entity* and an *architecture*, where bold text indicates reserved words and upper-case text indicates predefined concepts. The *entity* declares the generic model parameters, as well as specifies interface terminals of nature `std_logic` type. The generic parameters and constants, used in the simultaneous statements, are not given with their concrete numerical values in the model description. The *architecture* contains the implementation of the model. It is coded by combining structural and behavioural elements.

The high level on the `trigger` port activates the one-shot timer output – `pulse`. The duration of the output signal impulse could be determined by the following equation

$$t_{impulse} = Delay \times T_{clk}\,, \qquad (6)$$

where *Delay* is a numeric value set by the user on the same named input port of the **one-shot_timer** block. T_{clk} is the period of the master clock signal of the system. The $t_{impulse}$ corresponds to the t_{os} (Fig. 1).

For the proposed VFC there are four getaways out blocks which serve as outputs. The **one-shot_out** and **comparator_out** getaways are connected to the FPGA digital I/O pins and could be observed directly. The **integrator_out** signal is first applied to a 8-bit DAC so it can be observed after digital-to-analog conversion.

The transfer function of the VFC shown on Fig. 3 is given by

$$f_{out} = V_{in} / (V_{ref}\, t_{impulse})\,. \qquad (7)$$

According to equation (7), it can be concluded that the sensitivity of the proposed VFC is obtained by the following equation $K_f = 1/t_{impulse}V_{ref}$. The variation of the sensitivity can be achieved by changing the value of the coefficient or by changing the T_{clk} of the system.

```
library ieee;
use ieee.std_logic_1164.ALL;
use ieee.numeric_std.ALL;
-- synopsys translate_off
library UNISIM;
use UNISIM.Vcomponents.ALL;
-- synopsys translate_on

entity oneshot is
port ( clk : in STD_LOGIC;
ce : in STD_LOGIC;
trigger : in STD_LOGIC;
delay : in STD_LOGIC _VECTOR (7 downto 0);
pulse : out STD_LOGIC :='0');
end oneshot;

architecture BEHAVIORAL of oneshot is
signal count: INTEGER range 0 to 255; -- count variable
signal flag : STD_LOGIC := '0'; -- count variable
begin
process (flag,clk,delay)
begin
-- wait for trigger leading edge
if trigger = '1' then
count <= to_integer(unsigned(delay));
elsif rising_edge(clk) then
if count > 0 then
pulse <= '1';
count <= count - 1;
else
pulse <= '0';
--flag <='0';
end if;
end if;
end process;
end BEHAVIORAL;
```

Fig. 4. One-shot timer behavioral VHDL model.

V. Simulation and Experimental Results

The workability of the proposed VFC (Fig. 3) is proved by simulation results obtained by the simulator built in Matlab Simulink® and also through experimental studies conducted on a FPGA development 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 dynamic range of the input and the output signal varies from 0*V* to 3.3*V*.

The oscillation frequency f_{out} of the prototype VFC against the input voltage is plotted on Fig. 5. The value of the coefficient equals 10 and V_{ref} is 2.5*V*. The system master clock frequency equals 1.562M*Hz*. When the input voltage changes from 10mV to 1.5*V* the oscillation frequency varies from 606.5*Hz* to 93.561k*Hz*, the linearity error δ is not higher than 1% at input voltage higher than 11m*V*. Thus, a figure of merit (FOM = sensitivity (M*Hz*) / error (%)) is equal to 12.5. The result is in a good agreement with the calculated by (5).

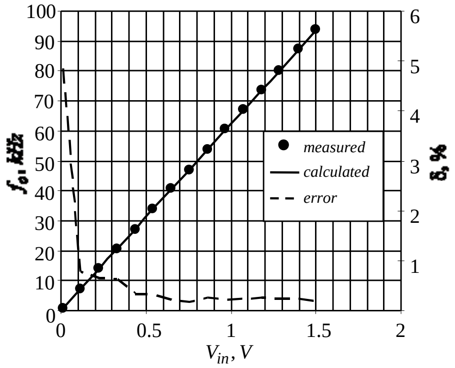
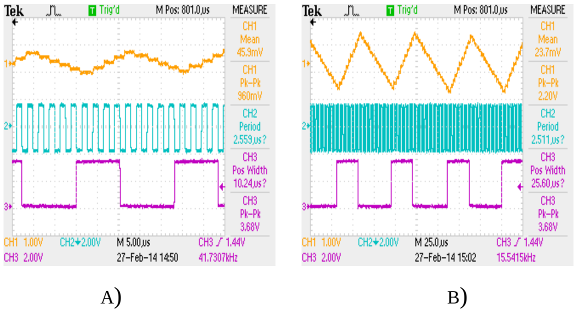


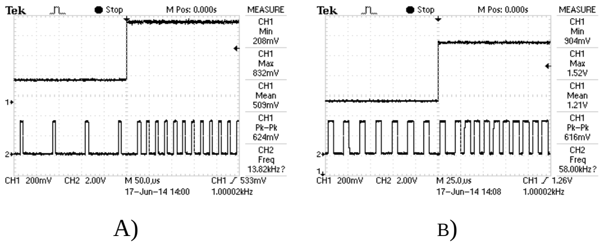
Fig. 5. The oscillation frequency versus the input voltage at sensitivity 62.5k*Hz*/*V* ($t_{impulse} = 6.4\mu s$).

Fig. 6a and Fig. 6b are presenting the measured output waveform at *delay* equals 4 and 10, respectively. The clock frequency equals $390kHz$ and the input voltage is 1V. The output frequencies at this input voltages, are 41.73kHz and 15.54kHz, respectively. Notably, the output frequency varies proportionally with the coefficient value, as the rise and fall times of produced pulse are unaffected. The error is less than 0.5%.



A) B)

Fig. 6. The output waveform of the VFC at value of the delay coefficient equals to: a) 4 and b) 10.



A) B)

Fig. 7. A frequency manipulated output at square wave input signal with: a) $V_{mean} = 0.5V$ and b) $V_{mean} = 1.2V$.

Fig. 7a and Fig. 7b show the output frequency manipulated signal when a square wave input signal with amplitude 0.6$V_{p\text{-}p}$ and mean values of 500m*V* and 1.2*V*, are applied. The output frequency changes between two values which correspond to the current level of the input voltage. On Fig. 7a is shown that $V_{in,\max} = 1{,}5V$ corresponds to $f_{out} = 90kHz$ and $V_{in,\min} = 900mV$ corresponds to $f_{out} = 55{,}5kHz$. Fig. 7b shows that for $V_{in,\max} = 800mV$ the corresponding output frequency is $f_{out} = 49kHz$ and for $V_{in,\min} = 200mV$ is $f_{out} = 13kHz$.

VI. Conclusion

In this paper a FPGA-based implementation of a charge balanced VFC has been presented. The sensitivity of the created system can be varied by the functional blocks parameters without changing the structure or without adding of external passive components in comparison with other low frequency (up to 1M*Hz*) monolithic VFCs. The system is implemented as a hierarchical structure of standard and custom Simulink blocks and represents the basic elements of the VFCs.

The workability of the proposed system was proved by comparison of the simulation results, obtained in Simulink® and an experimental study results. The experiment is conducted on ZedBoard development board (based on Xilinx Zynq™ -7020 SoC).

The created VFC is intended for development of mixed-signal systems such as PLLs employing analog low-pass RC filters.

References

[1] M. Seifart. *Analoge Schaltungen*. 6 Auflage. Verlag Technik Berlin, 2003 (in German).

[2] W. Jung. *Op Amp Applications Handbook*, Newnes, 2005.

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

[4] Sh.-Ku Kao, Yi-Fu Tang, H.-Chi Cheng. *A VCO-based phase-expanding conversion designed for time-domain data converters*, Microelectronics Journal, Vol. 45, pp. 835-841, 2014.

[5] J. Xie, M. K. Vamshi, M. A. Do, Ch. Ch. Boon, K. S. Yeo. *A low power low phase noise dual-band multiphase VCO*, Microelectronics Journal, Vol. 43, pp. 1016-1022, 2012.

[6] T. A. Al-Maaita, A. H. Tahboub, Kh. AbuGharbieh. *A 10 GHz wideband VCO with low KVCO variation*, Microelectronics Journal, Vol. 45, pp. 103-118, 2013.

[7] M.-T. Hsu, T.-H. Han, Y.-Y. Lee. *Design of Sub-1 mW CMOS LC VCO based on current reused topology with Q-enhancement and body-biased technique*, Microelectronics Journal, Vol. 45, pp. 627-633, 2014.

[8] 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.

[9] Xilinx. Zynq-7000 All Programmable SoC – Technical Reference Manual, Vol. 585, 2013.

[10] Altera. AlteraSoC: Booting and Configuration, Vol. 3, pp. 1-24, 2012.

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

[12] J. Bryant. *Voltage-to-frequency converters* (AN-361). Analog Dev., 1989.

[13] M. Paiva. *Voltage-to-frequency/frequency-to-voltage converter* (AN795). Microchip, 2002.

[14] J. Siegl. *Schaltungstechnik – Analog und gemischt analog/digital*. 2. Auflage. Springer-Verlag, 2005 (in German).

[15] P. Manoilov. *VLSI system design using VHDL*. Sofia, TU-Sofia, 2010 (in Bulgarian).