International Journal of Business and Technology

Volume 9
Issue 1 *Spring 2021*

Article 24

March 2021

Direct Digital Synthesis Optimization Based on VHDL Code

Marsida Ibro
Aleksander Moisiu University, marsidaibro@uamd.edu.al

Direct Digital Synthesis Optimization Based on VHDL Code

Marsida Ibro [1], Gentiana Blakaj [2], and Galia Marinova [3]

[1] Aleksander Moisiu University, Durres, Albania
marsidaibro@uamd.edu.al
[2] University for Business and Technology (UBT), Pristina, Kosovo
gb46704@ubt-uni.net
[3] Technical University of Sofia, Bulgaria
gim@tu-sofia.bg

Abstract. In this paper, we will describe the synthesis of the Direct Digital Synthesis (DDS) circuit using the VHDL language. Nowadays the DDS is being used widely in the fields of telecommunications, including signal generator circuits. The DDS circuits are used to generate analog signals with the use of digital circuits. This work aims to present the implementation of DDS with the VHDL language, which offers compatibility with FPGA devices. Initially, the VHDL code was generated automatically via Matlab/Simulink model, designed by using HDL Coder components, which are compatible with VHDL.
The VHDL code is modified in order to optimize the automatically generated code and the performance of the DDS circuit, Due to this optimization done on VHDL code about 15% improvement in power consumption and reduction of resource utilization is achieved.

Keywords: DDS, FPGA, VHDL, MATLAB/Simulink, Low Power Consumption

1 Introduction

Direct Digital Synthesis (DDS) is known as a signal generator realized on digital circuit that produces analog signals at the output. DDS implements a different approach compared to the most common frequency synthesis technique, the Phase-Lock loop (PLL). PLLs are usually composed of the phase detector, voltage controlled oscillator (VCO), and loop filter [1]. The use of digital techniques, combined with high-speed logic, makes DDS a powerful technique for generating analog signals, the frequency of which can be easily tuned whenever needed. DDS plays a key role in various microwave/radio frequency designs, which need a signal generator that does not have noises. DDS, similar to a numerically controlled oscillator (NCO), generates sinusoidal signals with maximum clarity and the change of the frequencies can be done easily and quickly. With the growing need for flexibility, precision, and effectiveness, FPGAs started to play an important role in the field of digital circuit design. The big-

gest advantage of using an FPGA is that can be programmed and modified in a very short period compared to Application-Specific Integrated Circuits (ASICs).

The paper is organized in the following sections: in Section 2, Related works provide the literature review of the FPGA-based DDS designs; Section 3 explains how the VHDL optimization is done and implemented in Zedboard (Zynq-7000) device; Experimental results in Section 4 provide the simulations results concerning the performance of the design; and in Conclusions are presented the results of the work.

2 Related works

In this section, we will give a literature review on implementations of FPGA-based DDS. Moreover, DDS circuits can be implemented by using a look-up table or CO-ordinate Rotation Digital Computer (CORDIC) algorithms. In addition to the advantages, DDS also has disadvantages where phenomena such as Spurious Free Dynamic Range (SFDR) need to be improved because it helps to reduce the noise caused by quantization.

The work in paper [2] focuses on a look-up table, multiplier digital filters, noise shaping, and dithering. The use of the high-order integrated comb filter (CIC), without the use of multipliers, when implemented in FPGAs avoids the use of DSP blocks. The design environment used is MATLAB followed by VHDL code and testbench.

The article [3] describes the design and implementation of a new circuit of a Modified Sinusoidal PWM in FPGA. The design was first simulated using MATLAB Simulink to control a single-phase DC inverter in AC and to study the performance in terms of reduction in Total Harmonic Distortion (THD). The sine wave signals were generated based on the Direct Digital Synthesizer (DDS) technique and the software's such as Xilinx Vivado, Xilinx System generator, or DSP Builder Tool are used to execute VHDL code without using IP blocks.

In the article [4], the authors have built an improved version of the multi-direct digital synthesizer (MDDS) algorithm in FPGA and used it in the synthetic aperture radar (SAR) resolution. The FPGA device is used to generate the desired signals and to improve the phase error value.

Compared to the other articles, which dealt with the LUT-based DDS designs, the article [5] shows how to use the CORDIC algorithm in implementing different digital modulation techniques through MATLAB and VHDL by using the concept of the software-defined radio. The DDS application in this case has high precision, lower costs, and high phase resolution.

The feedback phase-locked loop carrier synchronizer proposed in [6] is suitable for parallel implementation on an FPGA for modulation. The Direct Digital Synthesizer is designed by using the CORDIC algorithm and the VHDL code is created to fulfill the pipelined description model. The simulation results show that with this method, the resource utilization is 2% for the Cyclone IV FPGA and the maximum operating clock is > 100 MHz.

3 VHDL code optimization for the DDS design

In the paper [7], the Matlab/Simulink model generates the VHDL code automatically. This model is composed of four subsystems which are the phase accumulator (subsystem 1), the phase-to-amplitude converter (subsystem 2), the look-up table (LUT) (subsystem 3), and the dither (subsystem 4). The Simulink model helps to generate the VHDL codes separately for each subsystem and the DDS design VHDL code is generated by putting together these codes.

The most important parameter in this design is the frequency code word (FCW or FTW), which determines the frequency of the generated sine wave in the output. In this case, the value of the frequency is chosen 800 MHz so that the maximum operating frequency of Zynq-7000 should not be exceeded. In the optimized VHDL code, the FCW parameter is a 32-bit input and can be changed independently from the design. A new DDS VHDL code is developed as an optimized version of the one published on the EduBlog Multimedia Assistant for Teaching Direct Digital Synthesis [8]. On Fig.1 it is shown how the VHDL code is optimized and how the FCW parameter has been put as an input of the DDS.

```
LIBRARY IEEE;
USE IEEE.std_logic_1164.ALL;
USE IEEE.numeric_std.ALL;

ENTITY Accumulator IS
  PORT( clk  : IN std_logic;
        Reset : IN std_logic;
        clk_enable : IN std_logic;
        i_fcw  : IN std_logic_vector(31 DOWNTO 0) --uint32
        ce_out : OUT std_logic;
        Out1 : OUT std_logic_vector(31 DOWNTO 0) --uint32
        );
END Accumulator;

ARCHITECTURE rtl OF Accumulator IS
  -- Signals
  SIGNAL enb : std_logic;
  SIGNAL alpha32_bit_out1: unsigned(31 DOWNTO 0);--uint32
  SIGNAL fcw : unsigned(31 DOWNTO 0);  -- uint32
  SIGNAL Delay_out1: unsigned(31 DOWNTO 0);  -- uint32
  SIGNAL Out1: unsigned(31 DOWNTO 0);  -- uint32
```

Fig. 1. The optimized VHDL code

Then it is implemented in the FPGA device. Usually, the FPGA design flow is done in 5 steps [9]:

- Functional/Device specification;
- HDL coding;
- Logic Synthesis;
- Mapping (Place & Route), and
- FPGA configuration.

4 Experimental results

In this section, we outline the simulation results of the new optimized VHDL-based DDS design. To implement the VHDL code for the DDS design we have used the software Vivado version 2014.2 and the Zedboard device. The Vivado design tool performs setups to enable the FPGA device's designs, components and perform different simulations after the implementation [9]. The designs can be programmed in a hardware readable language like Verilog or VHDL especially when it is needed to implement algorithms. Furthermore, there is the other option to implement the project via a schematic-based entry. To test the performance of the implementation, the Vivado tool is used to perform Behavioral Simulation, Gate Level Simulation, and Timing Simulation.

Synthesis is used to convert HDL code to a port-level netlist. The netlist derives all the components, elements, interfaces, and other necessary details such as busy area, operating temperature, etc. Another feature of synthesis is the syntax control of HDL code.

The process of mapping maps the generic logic design (gates, flip flops, modules, and input/output switches) to the logic technology contained inside the chosen FPGA device. Placement & Route is a crucial step in the entire implementation. Placement is responsible for deciding where the components should be placed within the FPGA. The Routing is responsible for the connections between the different components.

On Fig.2 are presented the results from behavioral simulation of the new VHDL code. The sine wave generated is visible at the output.

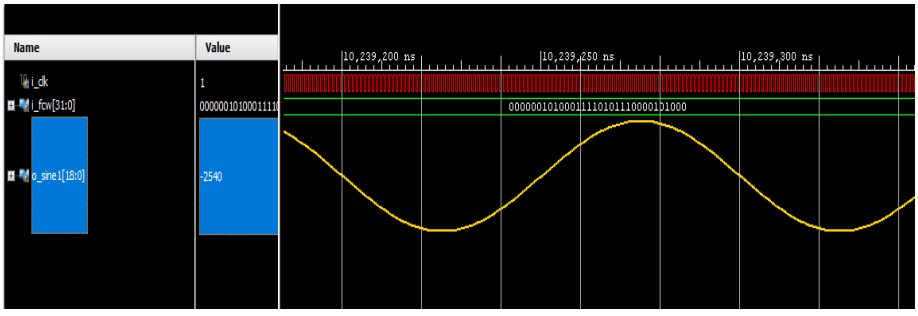

Fig. 2. Behavioral simulation of the new VHDL code of the DDS. The sine wave generated

The implementation on Zynq-7000 of the optimized VHDL code for DDS is shown in Fig. 3. After the synthesis is done without errors, the design is implemented in a

schematic design as shown in Fig. 4. This process maps the design to the Zedboard and the FPGA family used.

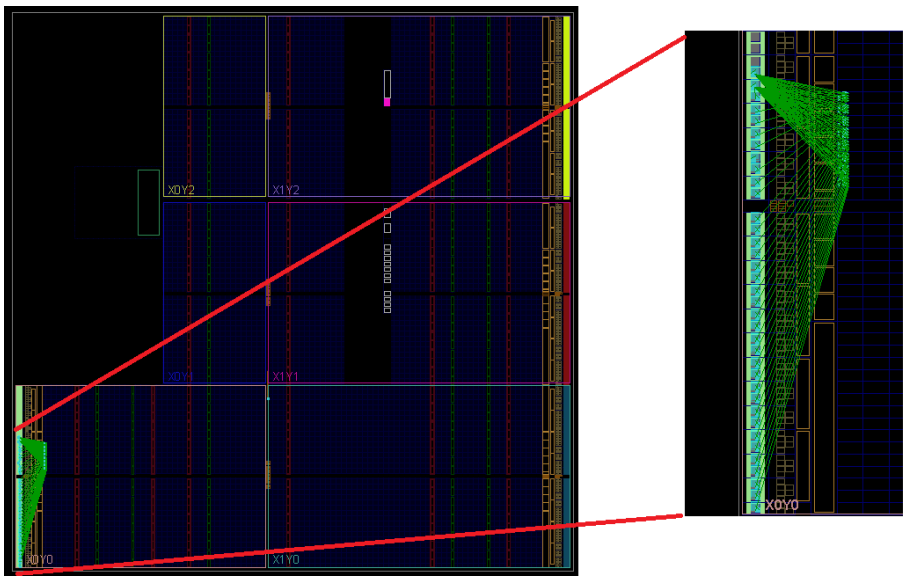

Fig. 3. The implementation of the synthesized design on Zynq-7000 in Vivado floorplanner

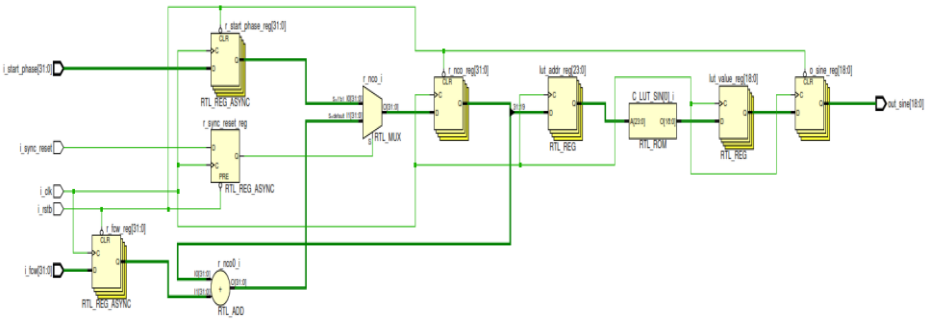

Fig. 4. Schematic of the implemented design of the VHDL code

After the synthesis and implementation in Zedboard and Vivado software [10] are successfully done, we tested the power consumption and resource utilization parameter are shown in Fig. 5 and Fig. 6.

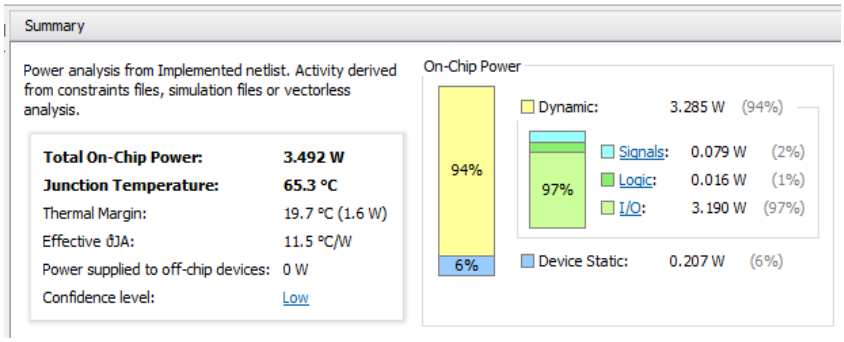

Fig. 5. Power consumption of the optimized VHDL DDS design

Resource	Utilization	Available	Utilization %
Slice LUTs	8	53200	0.02
Slice Registers	27	106400	0.03
IO	36	202	17.82
Clocking	1	32	3.12

Fig. 6. Resource utilization of the optimized VHDL DDS design

These reports give a detailed overview of the key factors that affect the power consumption and the allocation of the components for the Zynq-7000 device. As it is shown in Fig. 5, the power consumption is dominated by the dynamic power consumption by 94% compared to the 6% of the static power consumption. The overall power consumption level is 3.492 W and the junction temperature is 65.3°C. Figure 6 shows the device utilization summary for the design and the generated report comprises the number of Slice LUTs, number of Slice Registers, I/O, and clock. The domination in resource allocation by 17.82% is from the I/O parameter as it is dominating the power consumption. The detailed results of the power consumption and resource utilization optimization are given in Table 1 and Table 2. The power consumption reduction in % in the new optimized VHDL code for the DDS sine generator is calculated in Table 1.

Table 1. Comparison of the results for power consumption

Power consumption	VHDL code from [7]	New optimized VHDL code	% of reduction of the power consumption
Signals	0.093 W	0.079 W	15 %
Logic	0.024W	0.016 W	33.3 %
I/O	3.783 W	3.190 W	15.6 %
Dynamic power consumption	3.899 W	3.285 W	15.7 %
Static power consumption	0.244 W	0.207 W	15.1 %
Total	4.144 W	3.492 W	15.7 %

DOI: 10.33107/ijbte.2021.6.3.22

Table 2. Comparison of the results for resource utilization

Resource utilization	Slice LUTs	Slice Registers	I/O	Clocking
VHDL code from [7]	16 (0.03%)	32 (0.03%)	36 (17.82%)	1 (3.12%)
New optimized VHDL code	8 (0.02%)	27 (0.03%)	36 (17.82%)	1 (3.12%)
Available	53200	106400	202	32

5 Conclusions

This paper gives a brief description of the DDS designs based on FPGAs and the VHDL language. The optimized VHDL code used in this paper is done for teaching purposes and does not imply the optimization of any model. It is added to the MATLAB design of DDS at the Multimedia Assistant for Teaching Direct Digital Synthesis educational blog at https://allaboutdds.edublogs.org/home/.

The VHDL code is generated automatically by using the Matlab/Simulink model and we have optimized it by putting the FCW as an input, independently from the circuit design. The VHDL code is implemented on Zedboard and the Vivado software. From the results, we see that the power consumption is improved by more than 15% compared to the first VHDL code. Another improved parameter is the resource allocation, which has been optimized mostly in the Slice LUTS and Slice Registers.

Achnowledgement

This study is realized and partly supported by the CEEPUS network CIII-BG-1103-06-2122, especially in the part Joint doctoral program within CEEPUS.

References

1. A. Mann, A. Karalkar, L. He and M. Jones, "The design of a low-power low-noise phase lock loop," 2010 11th International Symposium on Quality Electronic Design (ISQED), 2010, pp. 528-531, DOI: 10.1109/ISQED.2010.5450522.
2. S. Jahn and H. -G. Brachtendorf, "DDS with Noise Reduction by Multiplier-Less Filter Methods," 2021 31st International Conference Radioelektronika (RADIOELEKTRONIKA), 2021, pp. 1-4, DOI: 10.1109/RADIOELEKTRONIKA52220.2021.9420194.
3. A. Al-Safi, A. Al-Khayyat, and Z. Aklah, "FPGA-Based Implementation of MSPWM Utilizing 6-Input LUT for Reference Signal Generation," 2021 IEEE 11th Annual Computing and Communication Workshop and Conference (CCWC), 2021, pp. 0597-0602, DOI: 10.1109/CCWC51732.2021.9375926.

4. K. -R. Kim and J. -H. Kim, "Wideband Waveform Generation and Measurement for High-Resolution X-Band UAV-SAR," IGARSS 2020 - 2020 IEEE International Geoscience and Remote Sensing Symposium, 2020, pp. 6479-6482, DOI: 10.1109/IGARSS39084.2020.9324252.
5. P. J. Katkar and Y. S. Angal, "Realization of CORDIC algorithm in DDS: Novel Approach towards Digital Modulators in MATLAB and VHDL," 2015 International Conference on Information Processing (ICIP), 2015, pp. 355-359, DOI: 10.1109/INFOP.2015.7489407.
6. P. Fiala and R. Linhart, "High efficient carrier phase synchronization for SDR using CORDIC implemented on an FPGA," 2015 23rd Telecommunications Forum Telfor (TELFOR), 2015, pp. 512-515, DOI: 10.1109/TELFOR.2015.7377519.
7. M. Ibro and G. Marinova, " Multi-tool environment with an online assistant for teaching Direct Digital Synthesis", 28th International Conference on Systems, Signals and Image Processing, IWSSIP 2021, Bratislava, Slovakia, 2021
8. Edublog Homepage: Multimedia Assistant For Teaching Direct Digital Synthesis, created by Marsida Ibro and Assoc.Prof.Dr. Galia Marinova, https://allaboutdds.edublogs.org/home/
9. Design Flows Overview, Vivado Design Suite User Guide, Xilinx Inc., 2020, web page. Available: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2020_1/ug892-vivado-design-flows-overview.pdf
10. Vivado Design Suite User Guide: Power Analysis and Optimization, Xilinx Inc 2020, web page.Available: https://www.xilinx.com/support/documentation/sw_manuals/xilinx2020_1/ug907-vivado-power-analysis-optimization.pdf