SPARTAN–field programmable gate array implementation for analog waveforms generation by direct digital synthesis

Moulai Khatir Ahmed Nassim[1,2], Ziani Zakarya[2,3]
[1]Department of Electrical Engineering and Electronics, Faculty of Technology, University of Tlemcen, Tlemcen, Algeria
[2]Research Unit for Materials and Renewable Energies (URMER), University of Tlemcen, Tlemcen, Algeria
[3]Department of SNV, Institute of Sciences of University Center of Salhi Ahmed Naama, Naama, Algeria

Article Info

Article history:

Received Sep 12, 2024
Revised Apr 5, 2025
Accepted Jun 10, 2025

Keywords:

Direct digital synthesis
Field programmable gate array
VHDL
Vivado
Xilinx

ABSTRACT

In the last thirty years, low power field programmable gate arrays (FPGAs) becoming more commonly used to implement a countless of applications in different electronics industry domains. Due to their flexible design, strong compatibility, parallel computing, and compared to the CPU architecture, FPGA accentuate computing efficiency and con sidered as one of the devices with the lowest application risk and the shortest development cycle among the variety of available programmable circuits families. This article details the design and implementation of a direct digital synthesis (DDS) signal generator using the Spartan-6 FPGA, focusing on high-quality sine wave generation. The system utilizes look-up tables (LUTs) and Block RAM (BRAM) for efficient storage and retrieval of sine wave data, while an 8-bit DAC0808 digital-to-analog converter (DAC) ensures precise waveform output. The FPGA's reconfigurable architecture allows real-time adjustments of frequency and phase, making the design suitable for various signal processing applications and modulation techniques like binary phase shift keying (BPSK).

Corresponding Author:

Moulai Khatir Ahmed Nassim
Department of Electrical Engineering and Electronics, Faculty of Technology, University of Tlemcen
Tlemcen, Algeria
Email: ahmednassim.moulaikhatir@univ-tlemcen.dz

1. INTRODUCTION

In recent years, field pro grammable gate arrays (FPGAs), particularly those in the SPARTAN series, have gained significant traction in digital systems due to their flexibility, performance, and cost-effectiveness, especially in applications like digital signal processing (DSP), communications, and embedded systems. SPARTAN FPGAs, such as the Spartan-7, provide high DSP performance (up to 176 GMAC/s) and are well-suited for real-time applications, where their parallel processing capabilities offer a clear advantage over traditional processors [1], [2]. These devices excel in efficiently handling high-throughput tasks while maintaining low power consumption, making them ideal for embedded systems and communication protocols [1]. Furthermore, the integration of DSP blocks on SPARTAN FPGAs enhances their ability to accelerate computation-heavy processes, such as convolutional neural network (CNN) inference, highlighting their growing importance in modern hardware acceleration [2], [3]. This project emphasizes the FPGA-based generation of analog waveforms using direct digital synthesis (DDS), a technique widely recognized for its ability to produce and delivering precise and stable frequencies across a broad range. DDS allows the generation of various waveforms, such as sine, square, and triangular waves, making it ideal for applications in communication systems, instrumentation, and signal processing [4].

The aim of this project is to implement a signal generator on the SPARTAN FPGA using the DDS technique to create high-quality analog waveforms. By exploiting the parallel processing, reconfigurable architecture, and look-up table (LUT) capabilities of the FPGA, this design provides significant advantages in terms of flexibility, precision, and scalability compared to traditional waveform generators [5]. Additionally, the FPGA's capabilities allow real-time modifications of signal parameters, such as frequency and phase, which are crucial in advanced signal processing applications [6], [7].

To achieve precise control of the output phase during frequency switching transitions and enhance phase noise while improving frequency agility, a non-linear digital design technique is employed [8]. This method generates analog waveforms using stored digital samples from a LUT that contains digital data of a sinusoidal waveform [9], [10]. By sequentially reading these values and sending them to the input of a digital-to-analog converter (DAC), the system synthesizes and outputs a smooth sinusoidal signal [11]. To achieve this, we implemented a very high speed integrated circuit hardware description language (VHDL) program that constructs a read-only memory (ROM) configured as a LUT specifically designed for sinusoidal waveform generation.

In order to maximize efficiency and reduce the usage of the FPGA’s general logic resources, we took advantage of the Xilinx Spartan-6 FPGA’s embedded Block RAM (BRAM). This allowed us to model the ROM in a way that minimized the impact on the device's logical resources, optimizing both memory usage and performance [12]. By using BRAM for the ROM, we were able to create a compact and efficient design that maintains high accuracy in waveform generation without compromising the FPGA’s processing capabilities for other tasks.

The design was programmed and synthesized within the Xilinx ISE Design Suite, a robust development environment that provides extensive support for the Spartan-6 series. This tool chain facilitated the implementation, simulation, and verification of our design, ensuring that it met performance criteria while maintaining efficient resource usage on the FPGA [13], [14].

To convert the digital samples stored in the LUT into an analog signal, we used an 8-bit monolithic DAC, specifically the DAC0808 model. This DAC operates with ±5 V power supplies, ensuring accurate conversion of digital data into an analog signal [15]. With this configuration, we were able to achieve high-quality waveform generation suitable for a wide range of applications that require stable and precise analog signals. The choice of the DAC0808 was made due to its high performance, particularly in terms of resolution and signal fidelity, which is essential for applications where waveform precision is critical [16].

2. METHOD

The MIMAS V2 is an FPGA development board that integrates the AMD Spartan-6 XC6SLX9 CSG324, providing an efficient platform for prototyping and implementing digital designs. It features 512 Mb DDR SDRAM, allowing for enhanced data storage and faster processing capabilities, which is essential for complex applications. Additionally, the board includes onboard SPI flash used for storing the bitstream, facilitating easy configuration and reprogramming of the FPGA. A downloader cable is typically employed to transfer the bitstream to the board, enabling straightforward programming and testing. This setup supports various applications, including DSP, embedded systems, and educational projects, making it a versatile tool for both professionals and students in electronics and computer engineering [17].

The Figure 1 show the used MIMAS V2 FPGA development board which features multiple GPIO pins that enable versatile interfacing with external devices, enhancing its functionality for various applications. Additionally, it includes header connectors that simplify access to these pins, making prototyping and integration easier for developers [17]. The Spartan-6 MIMAS V2 FPGA integrates embedded internal storage, utilizing DDR SDRAM operating at 166 MHz with a capacity of 512 Mb LPDDR. This type of memory is particularly advantageous for applications requiring fast data access and storage, as it allows for efficient handling of larger data sets. Each block in the DDR SDRAM is designed as simple dual-port (SDP) RAM, which enables simultaneous read and write operations. This architecture significantly enhances performance by allowing the FPGA to process multiple data streams concurrently, making it ideal for high-speed applications such as DSP, real-time data acquisition, and other compute-intensive tasks. This FPGA board features 2K×8-bit port BRAM configured as SDP architecture. This design allows for simultaneous read and write access, enhancing data throughput. The BRAM serves as a LUT for digital samples storage, enabling rapid retrieval of the pre-defined values. Table 1 summarizing the characteristics of single port BRAM in the Spartan-6 FPGA [18].

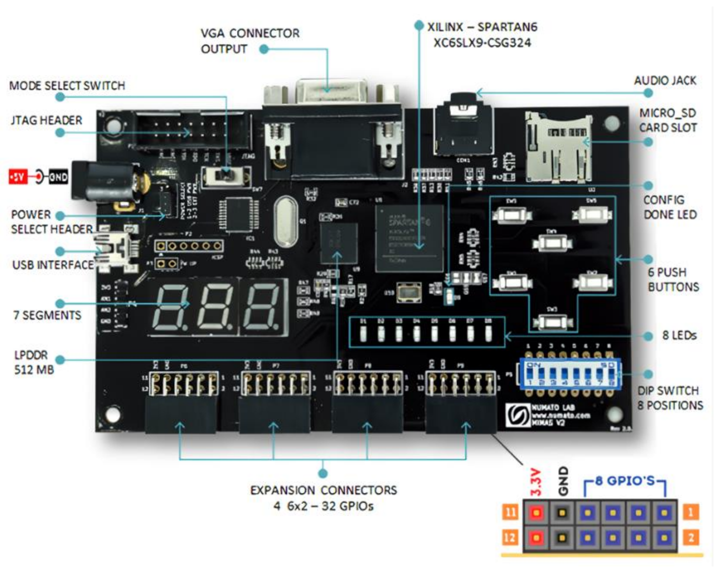


Figure 1. Mimas V2 Spartan-6 FPGA development board [17]

Table 1. Overview of single port BRAM features in Spartan-6 FPGA [9]

Feature	Description
Memory type	Single port BRAM
Memory depth	Up to 18K bits (for 1 block)
Memory width	1, 2, 4, 9, 18, or 36 bits
Total RAM blocks	36 blocks per Spartan-6 device
Configuration	Configured as 1K×18 or 2K×9 in a single port
Access mode	Read or write operations can be performed
Block size	1K, 2K, 4K, or 9K bits
Single clock cycle	Supports single clock cycle read/write
Usage	Ideal for implementing FIFOs, caches, and LUTs

In our case, the LUT presents a set of binary precalculated values recorded in memory locations and obtained using the following relation at several angles.

$$LUT(n) = A * \sin\theta \tag{1}$$

$LUT(n)$: refers to the output value of the LUT at index n which inour case is to 64 as synthesizer resolution. A: represents the amplitude of the sine wave. θ: denotes the angle in radians, which can be related to the input sample index. Using this equation, a LUT can efficiently generate the values of à sine wave, allowing for quick access during DSP tasks. This approach is widely utilized in digital audio, telecommunications, and other applications requiring precise waveform generation.

The change in angle between successive samples in a discrete representation of the sinusoidal wave is represented by the following (2) [19].

$$\Delta\theta = \theta(i) - \theta(i-1) = \frac{2\pi}{L} \tag{2}$$

where: $\Delta\theta$ is the phase difference between two consecutive samples; $\theta(i)$ is the phase angle at the i-th. sample; $\theta(i-1)$ is the phase angle at the previous sample; and L is the total number of samples (or points) in one complete cycle of the waveform.

This relationship shows that the phase increment between samples is determined by dividing a full cycle (2π radians) by the total number of points (L) in the waveform, which is essential for accurately generating smooth and continuous sinusoidal signals in applications such as DSP and waveform synthesis. The variation of L for a fixed frequency in the input clock results a variation of sinusoid and we can have the angle of any index [20].

$$\theta(i) = \Delta\theta * i \tag{3}$$

From (1) and (3), we get:

$$LUT(n) = A * sin\frac{2\pi}{L} * i \tag{4}$$

We can vary the resolution with a change in L. The output wave form amplitude depends on the value of ‘A’ in (1). Following design requirements, we can select and scale the value of ‘A’ to maximize the synthesizer’s output dynamic range. We used DAC0808 as an 8-bit digital to analog converter with an output span in positive side between 0 V to 5 V by modifying (1).

The sample values of the LUT are calculated using (5) and stored in ROM modeled in VHDL. For our DAC: A=255_{10}=FF_H. The calculated value using (5) with L=64 results in the LUT shown in Table 2.

$$LUT(n) = \frac{A}{2} + \frac{A}{2} * sin\frac{2\pi}{L} * i \tag{5}$$

Table 2. Sine LUT with length, L=64

Address (HEX)	Data (BIN)	Address (HEX)	Data (BIN)	Address (HEX)	Data (BIN)
00	10000000	16	11101010	2C	00001010
01	10001011	17	11100010	2D	00000110
02	10011001	18	11011010	2E	00000011
03	10100101	19	11010001	2F	00000001
04	10110001	1A	11000111	30	00000000
05	10111100	1B	10111100	31	00000001
06	11000111	1C	10110001	32	00000011
07	11010001	1D	10100101	33	00000110
08	11011010	1E	10011001	34	00001010
09	11100010	1F	10001011	35	00001111
0A	11101010	20	10000000	36	00010110
0B	11110000	21	01110011	37	00011101
0C	11110110	22	01100111	38	00100110
0D	11111001	23	01011010	39	00101111
0E	11111101	24	01001111	3A	00111001
0F	11111110	25	01000011	3B	01000011
10	11111111	26	00111001	3C	01001111
11	11111110	27	00101111	3D	01011011
12	11111101	28	00100110	3E	01100111
13	11111001	29	00011101	3F	01110011
14	11110110	2A	00010110	-	-
15	11110000	2B	00001111	-	-

Our 64×8 bit ROM is implemented using BRAM on a Xilinx Spartan-6 FPGA. The design stores a sine LUT, enabling efficient retrieval of pre-calculated sine values. The ROM is programmed in VHDL, ensuring hardware-level control and flexibility. A flowchart (Figure 2) is used to represent the design process, including steps such as initialization, address generation, and data retrieval from memory. This approach optimizes resource utilization and performance, leveraging the capabilities of Spartan-6 BRAM for compact and efficient storage [21].

To map the ROM addresses from 00H=000000_2 to 3FH=111111_2, a 6-bit counter can be used to sequentially generate these addresses with each increment. For that we declared the variable cnt as integer range 0 to 63 to be able to call the LUT values from a table of 63 integers, the counter incorporates an asynchronous RESET feature, which ensures that the RESET test takes priority over the clock (CLK) signal. This means that whenever the RESET signal is activated, it can immediately set the counter back to its initial state, regardless of the clock's status. Such priority for the RESET signal is crucial for ensuring system stability and preventing erroneous counting during power-up sequences or unexpected conditions. Implementing asynchronous resets is a common practice in digital design to enhance reliability and control over counter behavior [22]. As the counter progresses, it effectively feeds the stored data from the ROM to the DAC0808's input. The data delivery speed is directly proportional to both the counter's clock speed and the ROM's output capability, to avoid distortion at the DAC output, it's essential to limit the clock speed to prevent exceeding the slew rate of the DAC. The slew rate is the maximum rate at which the output of the DAC can change, and exceeding this rate can lead to errors in signal reproduction. If the clock speed is too high, the DAC may not be able to track the rapidly changing input signals, resulting in distortion or inaccurate output waveforms [23], [24].

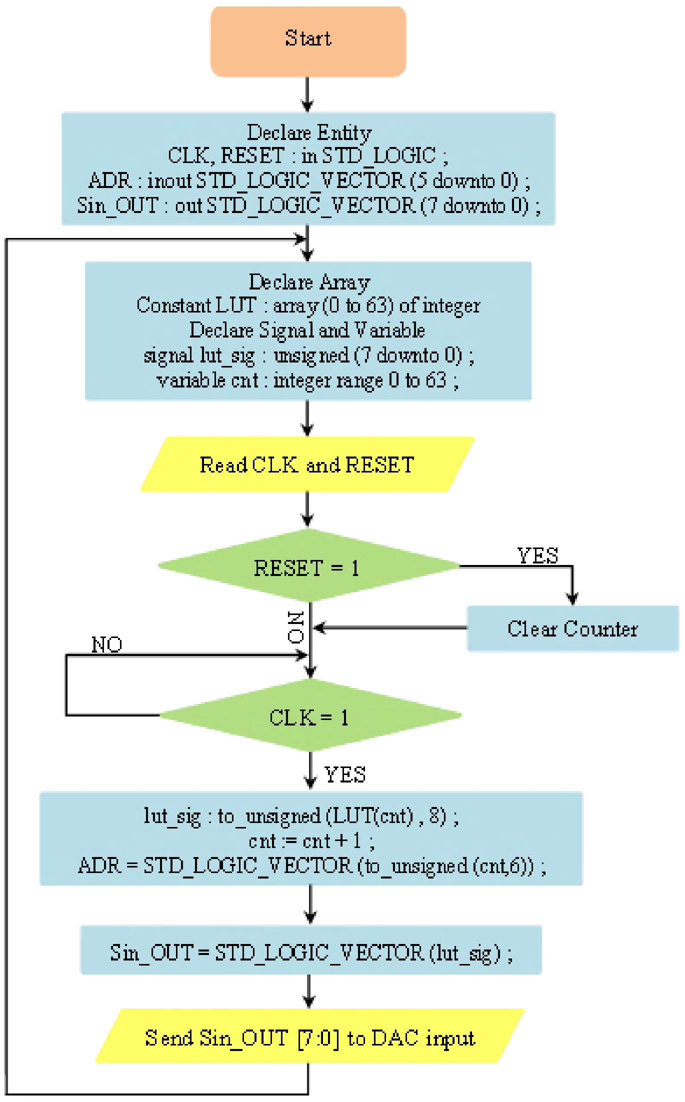
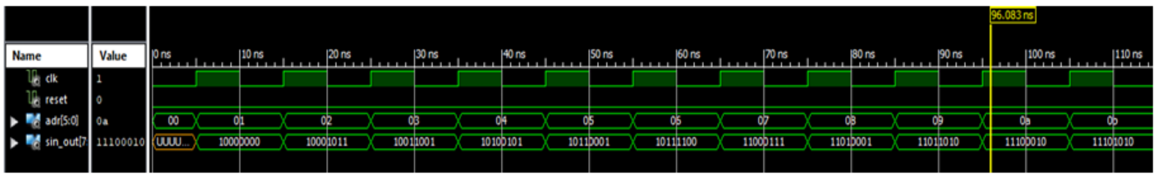

Start
Declare Entity
CLK, RESET : in STD_LOGIC ;
ADR : inout STD_LOGIC_VECTOR (5 downto 0) ;
Sin_OUT : out STD_LOGIC_VECTOR (7 downto 0) ;
Declare Array
Constant LUT : array (0 to 63) of integer
Declare Signal and Variable
signal lut_sig : unsigned (7 downto 0) ;
variable cnt : integer range 0 to 63 ;
Read CLK and RESET
RESET = 1
YES
Clear Counter
NO
NO
CLK = 1
YES
lut_sig : to_unsigned (LUT(cnt) , 8) ;
cnt := cnt + 1 ;
ADR = STD_LOGIC_VECTOR (to_unsigned (cnt,6)) ;
Sin_OUT = STD_LOGIC_VECTOR (lut_sig) ;
Send Sin_OUT [7:0] to DAC input

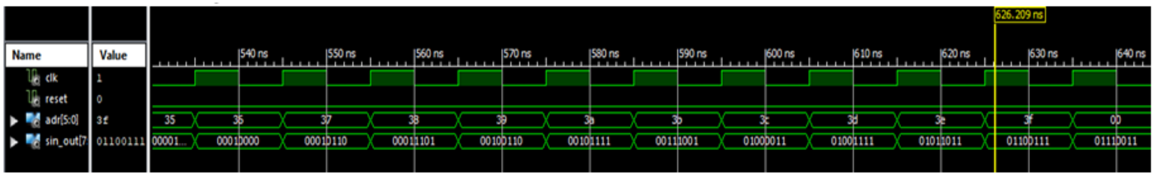
Figure 2. Flowchart of VHDL description

ADR is declared in the entity of our program as an inout STD_LOGIC_VECTOR (5 downto 0) to serve as a bidirectional port for providing addresses to the ROM. This configuration is essential for address management in memory architecture, enabling both reading from and writing to memory locations. Implementing a high-state asynchronous RESET is crucial for ensuring that the counter operates reliably. The initial condition checks whether reset=1. If true, it clears all flip-flops in the counter, ensuring a known starting state. If reset is not activated, the design proceeds to check for a rising edge on the CLK signal. Upon detecting a rising edge, the design accesses the corresponding LUT value located at the index of the cnt array. This value is then converted to an unsigned 8-bit format and assigned to the lut_sig signal [25].

At the end of each process execution of our VHDL behavioral description, we must assign the resulting value of lut_sig, converted to STD_LOGIC_VECTOR, to the Sin_OUT signal, which is declared as an output in our entity. This step is crucial for generating the corresponding analog signal from digital representation. The flowchart in Figure 2 summarizes the design steps involved, while Figure 3 illustrates waveform simulations with ROM addresses (ADR) ranging from 00 to 3F divided into two figures Figure 3(a) from 00 to 0B and Figure 3(b) from 35 to 3F, conducted using a test bench in Xilinx ISE, which is a simulation environment used to verify the functionality of digital designs. It contains the instantiation of the design under test (DUT) along with stimulus signals and input vectors to exercise the DUT. The test bench generates input and monitors outputs, allowing designers to analyze the behavior of their design before implementation on hardware. This process helps identify errors and ensures that the design meets specifications. It does not require physical hardware and runs entirely within the simulation environment [26].


Name
Value
clk
reset
adr[5:0]
sin_out[7
96.083 ns
00 01 02 03 04 05 06 07 08 09 0a 0b
UUUU... 10000000 10001011 10011001 10100101 10110001 10111100 11000111 11010001 11011010 11100010 11101010

(a)

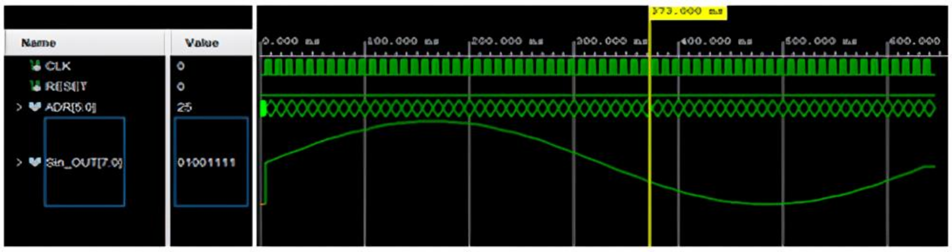


(b)

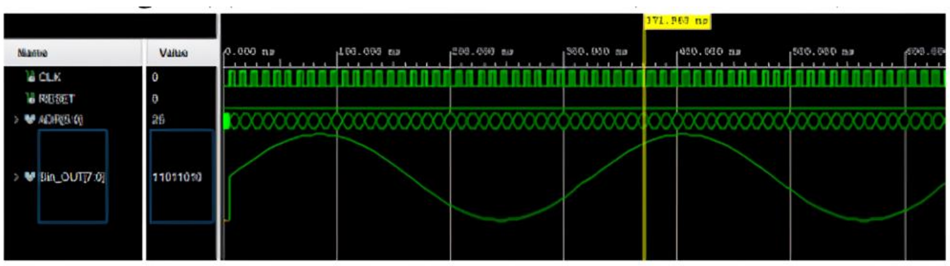
Figure 3. Waveform simulation with ROM ADR from (a) 00 to 0B and (b) 35 to 3F

3. RESULTS AND DISCUSSION

To visualize our experimental results of analog waveforms obtained at the output of the DAC shown in Figure 4, we use Vivado design suite which provides a comprehensive set of tools for hardware description, synthesis, simulation, and programming of Xilinx devices. Vivado is an advanced IDE by Xilinx, designed for creating complex FPGA designs, especially for newer families like the 7-series, UltraScale, and Versal. These FPGA families offer high performance and low power consumption, with Vivado optimizing their capabilities. It supports HDLs like VHDL and Verilog, as well as high-level synthesis (HLS) for C, C++, and OpenCL. Vivado enables synthesis, simulation, and optimization for high-speed applications and includes tools for debugging and real-time simulation. It also supports IP Integrator for integrating pre-built blocks, making it ideal for embedded systems and high-performance computing [27]. As shown in both Figures 4(a) and 4(b) for L=32 and L=64, with the increase of L, memory demand rises, which leads to a greater number of digital samples and extends the time required to access the LUTs. By adjusting the value of L, the system can be configured to generate various sinusoidal frequencies, making it suitable for creating different types of modulated signals, such as binary phase shift keying (BPSK) and amplitude shift keying (ASK). These modulation techniques rely on altering the phase or amplitude of a carrier signal and by controlling L, the system can efficiently switch between different frequencies to achieve the desired modulation format [28].



(a)



(b)

Figure 4. Sinusoid with (a) L=64 (f=2.5 KHz) and (b) L=32 (f=5 KHz)

4. FIELD PROGRAMMABLE GATE ARRAY IMPLEMENTATION

To download The VHDL program on Mimas V2 Spartan-6 FPGA we need to generate programming file and run it. This will create a .bit file, which is the binary file used to load the VHDL design onto the FPGA. To interact with Xilinx's implementation tools we used an user constraint file (UCF) file which is a case sensitive ASCII file generated by the user to defines how the logical design in the VHDL code maps to the physical pins and resources on the FPGA [29].

In a UCF file, both # and // are used for comments, allowing designers to annotate the file and explain the constraints. Each command or statement is terminated by a semicolon (;), which is standard syntax. Most of the content consists of mapping signals to specific pins using the following key commands:

- NET: refers to the logical signal in the design.
- LOC: specifies the physical location (pin) on the FPGA to which the signal is mapped.

Specifically, the UCF file allows constraining the following aspects of the FPGA design:

- Pin Assignments: maps logical signals to specific physical pins on the FPGA.
  Example: `NET "clk" LOC = "P85"`; (assigns the "clk" signal to pin P85).
- I/O standards: defines the voltage standard for each signal (e.g, LVCMOS33, SSTL).
  Example: `NET "clk" LOC = "P85" | IOSTANDARD = LVCMOS33;`

5. CONCLUSION

This project successfully implemented a DDS signal generator on a Spartan-6 FPGA. The use of LUTs in BRAM enables precise generation of analog waveforms, such as sine and square signals. The integration of the DAC0808 DAC ensures accurate conversion of digital samples. Thanks to the FPGA's reconfigurable architecture, signal parameters like frequency and phase can be adjusted in real time. This design is particularly suitable for applications in DSP and modulation techniques such as BPSK and ASK. Overall, this project demonstrates the advantages of FPGA technology for high-precision and scalable signal generation. For future research, it would be valuable to explore the integration of higher-resolution DACs and investigate the potential for real-time signal synthesis across a broader frequency range, which could enhance the versatility and performance of DDS systems in more complex applications.

ACKNOWLEDGMENTS

The authors would like to thank the Ministry of higher education and scientific research of the Algerian government and the Faculty of Technology, Tlemcen University for providing the funding for this research.

REFERENCES

[1] Xilinx, "Spartan-7 FPGAs: meeting the cost-sensitive market requirements," WP483, Dec. 7, 2020.
[2] D. Wang, K. Xu, J. Guo and S. Ghiasi, "DSP-Efficient Hardware Acceleration of Convolutional Neural Network Inference on FPGAs," *IEEE Transactions on Computer-Aided Design of Integrated Circuits and Systems*, vol. 39, no. 12, pp. 4867-4880, Dec. 2020, doi: 10.1109/TCAD.2020.2968023.
[3] A. K. T. Sulthana, "Simulation and Implementation of BPSK Modulator and Demodulator System on Spartan-3E FPGA," *2019 International Conference on Smart Systems and Inventive Technology (ICSSIT)*, Tirunelveli, India, 2019, pp. 126-128, doi: 10.1109/ICSSIT46314.2019.8987788.
[4] Y. Du, W. Li, Y. Ge, H. Li, K. Deng, and Z. Lu, "Note: A high-frequency signal generator based on direct digital synthesizer and field-programmable gate array," *Review of Scientific Instruments*, vol. 88, no. 9, Sep. 2017, doi: 10.1063/1.5001489.
[5] L.-M. Ren, X. Xue, and Y.-B. Zheng, "The design of high precision arbitrary waveform generator based on DDS technology and FPGA," *Journal of Physics: Conference Series*, vol. 1820, art. 012010, Mar. 2021, doi: 10.1088/1742-6596/1820/1/012010.
[6] W. Wolf, "FPGA-based Signal Processing," *FPGA-Based System Design*, 1st ed. Upper Saddle River, NJ, USA: Prentice Hall, 2004, pp. 247-284.
[7] A. A. Alsharef, M. A. Mohd. Ali, and H. Sanusi, "Direct digital frequency synthesizer simulation and design by means of Quartus-ModelSim," *Journal of Applied Sciences*, vol. 12, pp. 2172–2177, 2012, doi: 10.3923/jas.2012.2172.2177.
[8] X. Han, P. Wang, S. S. Li, and X. Zhao, "Investigation of phase-shifting and frequency conversion sinusoidal signal generator based on DDS and FPGA," *Applied Mechanics and Materials*, vol. 556–562, pp. 1580–1583, May 2014, doi: 10.4028/www.scientific.net/AMM.556-562.1580.
[9] N. B. Ameur, "A Low-Phase Noise ADPLL Based on a PRBS-Dithered DDS Enhancement Circuit," *Journal of Circuits, Systems and Computers*, vol. 26, no. 5, 2017, doi: 10.1142/S0218126617500761.
[10] Y. Hou, C. Li, and S. Tang, "An accurate DDS method using compound frequency tuning word and its FPGA implementation," *Electronics*, vol. 7, no. 11, Nov. 2018, doi: 10.3390/electronics7110330.
[11] M. Pomponio, A. Hati and C. Nelson, "Ultra-low phase noise frequency division with array of direct digital synthesizers," *IEEE Transactions on Instrumentation and Measurement*, vol. 73, pp. 1-10, 2024, doi: 10.1109/TIM.2023.3346538.
[12] V. Thamizharasan and N. Kasthuri, "FPGA implementation of high performance digital FIR filter design using a hybrid adder and multiplier," *International Journal of Electronics*, vol. 11, no. 4, pp. 587-607, 2023, doi: 10.1080/00207217.2022.2098387.

[13] S. M. Qasim, S. A. Abbasi, and B. Almashary, “A review of FPGA-based design methodology and optimization techniques for efficient hardware realization of computation intensive algorithms,” *2009 International Multimedia, Signal Processing and Communication Technologies*, Aligarh, India, 2009, pp. 313-316, doi: 10.1109/MSPCT.2009.5164238.
[14] K. Chen, Q. Liang, and J. Chen “High precision low jitter pulse generator implemented with FPGA transceiver,” *Measurement*, vol. 231, May 2024, doi: 10.1016/j.measurement.2024.114657.
[15] D. R. Zhou, Y. R. Zhou, J. C. Gong, and Z. F. Mao, “Signal generator design based on the FPGA,” *Applied Mechanics and Materials*, vols. 333–335, pp. 2407–2411, Jul. 2013, doi: 10.4028/www.scientific.net/AMM.333-335.2407.
[16] A. Rodríguez, J. Portilla, E. de la Torre and T. Riesgo, “Teaching hybrid HW/SW embedded system design using FPGA-based devices,” in *2016 Conference on Design of Circuits and Integrated Systems (DCIS)*, Granada, Spain, 2016, pp. 1-5, doi: 10.1109/DCIS.2016.7845372.
[17] A. Agarwal, “Design and FPGA implementation of DDR SDRAM controller,” *International Journal for Research in Applied Science & Engineering Technology (IJRASET)*, vol. 5, no. 4, pp. 1258–1263, Apr. 2017, doi: 10.22214/ijraset.2017.4224.
[18] P. Butler, “An almost pure DDS sine wave tone generator,” Analog, 2019.
[19] S. Li, “Design and implementation of DDS signal generator based on FPGA,” *Academic Journal of Science and Technology*, vol. 9, no. 1, pp. 145–149, Jan. 2024, doi: 10.54097/xdhh4c13.
[20] A. Z. Jidin, I. N. Mahzan, A. S. R. A. Subki, and W. H. W. Hassan, “Improve performance of the digital sinusoidal generator in FPGA by memory usage optimization,” *International Journal of Electrical and Computer Engineering*, vol. 9, no. 3, pp. 1742–1749, Jun. 2019, doi: 10.11591/ijece.v9i3.pp1742-1749.
[21] Z. Amirzadeh and M. Gholami, “Asynchronous counter in QCA technology using novel D flip-flop,” *The European Physical Journal Plus*, vol. 139, 2024, doi: 10.1140/epjp/s13360-024-05141-y.
[22] A. Barkalov, L. Titarenko, and K. Krzywicki, “Reducing LUT count for FPGA-based mealy FSMs,” *Applied Sciences*, vol. 10, no. 15, Jul. 2020, doi: 10.3390/app10155115.
[23] A. Mardari, Z. Jelčicová and J. Sparsø, “Design and FPGA-implementation of Asynchronous Circuits Using Two-Phase Handshaking,” in *2019 25th IEEE International Symposium on Asynchronous Circuits and Systems (ASYNC)*, Hirosaki, Japan, 2019, pp. 9-18, doi: 10.1109/ASYNC.2019.00010.
[24] X. Jiang, J. Wang, Y. Lin and Z. Wang, “FPGA-accelerated maze routing kernel for VLSI designs,” in *2022 27th Asia and South Pacific Design Automation Conference (ASP-DAC)*, Taipei, Taiwan, 2022, pp. 592-597, doi: 10.1109/ASP-DAC52403.2022.9712533.
[25] X.-T. Chen, W.-K. Huang, N. Park, F. J. Meyer and F. Lombardi, “Design verification of FPGA implementations,” in *IEEE Design & Test of Computers*, vol. 16, no. 2, pp. 66-73, Apr.-Jun. 1999, doi: 10.1109/54.765205.
[26] H. Lee, K. Kim, Y. Kwon, and E. Hong, “Real-time particle swarm optimization on FPGA for the optimal message-chain structure,” *Electronics*, vol. 7, no. 11, Nov. 2018, doi: 10.3390/electronics7110274.
[27] F. Quadri and A. D. Tete, "FPGA implementation of digital modulation techniques," in *2013 International Conference on Communication and Signal Processing, Melmaruvathur*, India, 2013, pp. 913-917, doi: 10.1109/iccsp.2013.6577189.
[28] S. Brown and Z. Vranesic, *Fundamentals of Digital Logic with VHDL Design*, 3rd ed., McGraw-Hill, 2009, pp. 526-528.
[29] S. Kilts, *Advanced FPGA design: architecture, implementation, and optimization*, Wiley-Interscience, 2007, pp. 89-91.

BIOGRAPHIES OF AUTHORS

Moulai Khatir Ahmed Nassim received his ingenuity degree in Electronics at Faculty of Technology, University of Tlemcen, Algeria, and his Magister and doctorate in MicroElectronics at Faculty of Technology, University of Tlemcen. Full-time professor of Advanced Digital Electronics (FPGA and VHDL) and electronics graduated program, Department of Electrical Engineering and Electronics, Faculty of Technology, University of Tlemcen, Algeria and member of the Research Unit for Materials and Renewable Energies (URMER), University of Tlemcen, BP-119, Tlemcen 13000, Algeria. He can be contacted at email: ahmednassim.moulaikhatir@univ-tlemcen.dz.

Ziani Zakarya received his ingenuity degree in Physics at Faculty of Science, University of Tlemcen, Algeria, and his Magister and Doctorate in Energy Physics and Materials at Faculty of Science, University of Tlemcen. Full-time professor in Department of SNV, Institute of Sciences, University Center of Salhi Ahmed Naama, BP-66, Naama 45000, Algeria. Member of the Laboratory for the Sustainable Management of Natural Resources in Arid and Semi-Arid Zones, University Center Salhi Ahmed, BP-66, Naama 45000, Algeria. He can be contacted at email: ziani@cuniv-naama.dz.