WWW.IJRPR.COM

International Journal of Research Publication and Reviews

Journal homepage: www.ijrpr.com ISSN 2582-7421

Signal Processing on FPGA Using Hardware Co-Simulation

Amulya Kale[a], Harsh Khatri[b], Harsh Thakar[c]

[a]*Department of Electronics and Telecommunication, Pune Institute of Computer Technology, Pune, India*
[b]*Department of Electronics and Telecommunication, Pune Institute of Computer Technology, Pune, India*
[c]*Department of Electronics and Telecommunication, Pune Institute of Computer Technology, Pune, India*

ABSTRACT--

Signal processing is a crucial aspect of various engineering applications, ranging from wireless communication systems to image and audio processing. Field-Programmable Gate Arrays (FPGAs) offer flexible and reconfigurable hardware platforms for implementing signal processing algorithms. However, designing and optimizing signal processing systems on FPGAs can be challenging due to the complex interplay between hardware and software components. This paper proposes a novel approach for signal processing on FPGAs using hardware co-simulation. The objective is to leverage the advantages of both hardware acceleration and software simulation to enhance the design and optimization process. Hardware co-simulation allows for the simultaneous execution of the FPGA hardware design and the software simulation model, enabling real-time interaction between the two domains. The proposed methodology integrates the high-level description of the signal processing algorithm using hardware description languages (HDL) with the software simulation environment. The HDL code is synthesized and implemented on the FPGA, while the software simulation model runs concurrently. By establishing a seamless interface between the FPGA and the software simulation, the co-simulation enables efficient testing, debugging, and performance evaluation of the signal processing system. Furthermore, the hardware co-simulation approach provides valuable insights into the interaction between the FPGA hardware and the algorithm being processed. It allows designers to analyze and optimize critical factors such as latency, throughput, power consumption, and resource utilization. By iteratively refining the design based on the co-simulation results, the overall signal processing system can be effectively optimized for improved performance and efficiency. The effectiveness of the proposed approach is demonstrated through case studies involving various signal processing algorithms, including digital filters, transforms, and modulation schemes. The results highlight the benefits of hardware co-simulation, such as accelerated development cycles, reduced time-to-market, and enhanced system performance.

Keywords:Hardware Co-Simulation, MATLAB, FPGA

INTRODUCTION

Signal processing is an electrical engineering subfield that focuses on analyzing, modifying, and synthesizing signals, such as sound, images, and scientific measurements. Signal processing techniques are used to optimize transmissions, digital storage efficiency, correcting distorted signals, subjective video quality and to also detect or pinpoint components of interest in a measured signal. Field Programmable Gate Arrays (FPGAs) are highly configurable digital circuits that provide the flexibility and performance required for implementing complex digital signal processing algorithms. To ensure the proper functioning of these algorithms, designers must simulate their circuits before implementation. Hardware co-simulation is a technique that allows designers to simulate their digital circuits on a host computer using specialized hardware co-simulation tools. This paper presents a technical overview of signal processing on FPGA using hardware co-simulation. We describe the advantages of using FPGAs for digital signal processing and the benefits of hardware co-simulation for FPGA design. We also discussed several key design considerations for implementing digital signal processing algorithms on FPGAs using hardware co-simulation.FPGAs consist of an array of configurable logic blocks (CLBs) that can be interconnected to implement complex digital circuits. The flexibility and programmability of FPGAs make them an ideal platform for implementing digital signal processing algorithms. The ability to perform parallel operations and custom instruction sets makes FPGAs faster than traditional digital signal processors (DSPs) and general-purpose processors (GPPs). FPGA architecture provides a high level of customization, allowing designers to allocate resources efficiently and optimize the digital signal processing algorithm to meet specific performance requirements.

Hardware co-simulation is a technique that allows designers to simulate their digital circuits on a host computer using specialized hardware co-simulation tools. This technique involves connecting the FPGA to a host computer via a high-speed interface and running the simulation on the host computer. The FPGA implements the digital circuit, while the host computer runs the simulation, providing inputs to the FPGA and capturing the outputs. Hardware co-simulation provides a more detailed view of the digital circuit behavior, making it easier to identify and debug design issues. It also reduces design time and minimizes design errors, leading to faster development cycles and time-to-market.

LITERATURE SURVEY

IEEE Signal Processing Magazine [1] by Andres Kwasinski and Min Wu. This journal addresses a variety of issues, such as "After half a century of progress, some claim that signal processing is now mature in terms of theories and techniques, and likely would not have a major research breakthrough."
Real Time Hardware Co-simulation for image processing algorithms using Xilinx system generator [2] by Mohammed AbdulhalimAlareqi, Rachid Elgouri, Mateur Khalid. The paper describes the system architecture and presents the performance analysis of the system using hardware co-simulation.
M. Janarthanan et al. presents a survey of hardware co-simulation techniques for FPGA-based signal processing systems. The paper reviews several hardware co-simulation techniques such as HDL co-simulation and Transaction-Level Modeling (TLM) co-simulation and their applicability in FPGA-based signal processing systems.
Vivado Design Suite User Guide: (UG897) [3] by AMD Xilinx. This user guide contains comprehensive instructions and step-by-step guidance for applying and running projects in the Xilinx Vivado.
Vitis Model Composer user guide [4] by AMD Xilinx. This user guide covers developing PL IP blocks for the hardware platform, as well as PL kernels, functional simulation, and analyzing time, resource utilization, and power closure. It also entails creating the hardware platform for system integration. This document's topics that pertain to this design process include Hardware Design using the HDL Library, designing a Model Composer, HDL Library Design Compilation Types Integration and validation of the system Integrating and verifying system functional performance, such as timeliness, resource use, and power closure. This article contains the following topics that are relevant to the design process that is performing Analysis in Model composer and Application of Hardware Co-Simulation.
Installing Vitis, Vivado Board File [5] by Diligent Reference. This guide walks through the process of installing and configuring the Vivado and Vitis development environments. These applications are used to develop projects to run on Digilent FPGA Development Boards. In addition to the installation, Digilent's Board Support Files will be installed, which are used to make the process of creating a new project significantly faster. In addition, the board files make it significantly easier to add a variety of peripherals (such as DDR memory), as well as a Zynq processor (for Zynq boards) to a project..

PROPOSED METHODOLOGY

The Vitis model Composer is configured to work with the proper FPGA board. In this case, any suitable FPGA board, such the Spartan7 XC7S50-CSGA324, may be utilised. The model for JTAG hardware co-simulation has been used, and I/O and clock planning have been finished. The Vitis model composer's parameters are selected and created. During compilation, a drawing of the model and a programming file in Verilog HDL are produced and may be examined using Xilinx ISE. The behavioural grammar of the module is confirmed when it is synthesised, implemented, and tested on an FPGA. To test an architecture, the VITIS Model Composer may produce test benches, test vectors, and user constraints files (UCF). VITIS Model Composer was first created to cope with complex Digital signal processing (DSP) applications, even if it has other applications connected to this topic, such signal processing. To create an FPGA bit file that can be used as input, bitstream compilation is crucial.

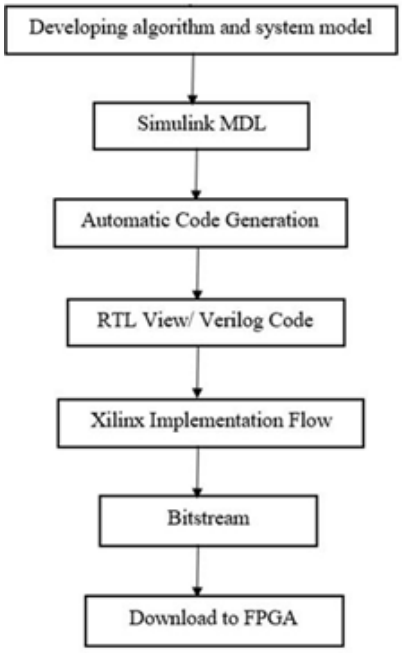

Fig. 1. Design flow for Signal processing with Vitis Model Compose

Design Considerations for Implementing Digital Signal Processing Algorithms on FPGAs using Hardware Co-Simulation:

To guarantee the appropriate operation of the digital circuit, designers must take into account a number of critical design concerns while implementing digital signal processing algorithms on FPGAs utilising hardware co-simulation:

1. Algorithm optimisation: To make use of the parallelism offered by FPGAs, digital signal processing algorithms must be optimised for parallel execution.
2. Resource Allocation: Resources such as CLBs, memory, and interconnects are scarce in FPGAs. To make sure that the FPGA can implement the digital signal processing method, designers must carefully arrange these resources.
3. Clocking: To guarantee optimal operation, FPGAs need precise clocking. To achieve the FPGA's clocking requirements, designers must carefully plan their digital circuits.
4. Simulation Accuracy: To ensure that the digital circuit works as intended, hardware co-simulation accuracy is essential. The gear that designers use must be reliable. The digital circuit is precisely simulated by co-simulation tools.

SYSTEM ARCHITECTURE

The goal of this project is to develop real-time hardware that performs better in terms of size and speed. It focuses on the implementation of an effective architecture for signal processing algorithms like segmentation (threshold) and audio or video signal processing by using the fewest possible system generator blocks for DSP tool. Hardware Co-Simulation relieves the user of the textual HDL programming. Hardware co-simulation mainly entails using Xilinx's System generator to translate the MATLAB Model into FPGA readable code. Now that the bit file created by the system generator has been used to code the FPGA, the ADC and FPGA will be used to filter the noisy real-world data,to the measurement instrument, and DAC. The design and implementation of signal processing algorithms employing application-specific VLSI architecture, including programmable digital signal processors and specialised signal processors implemented with VLSI, is the focus of the field of very large scale integrated (VLSI) signal processing.

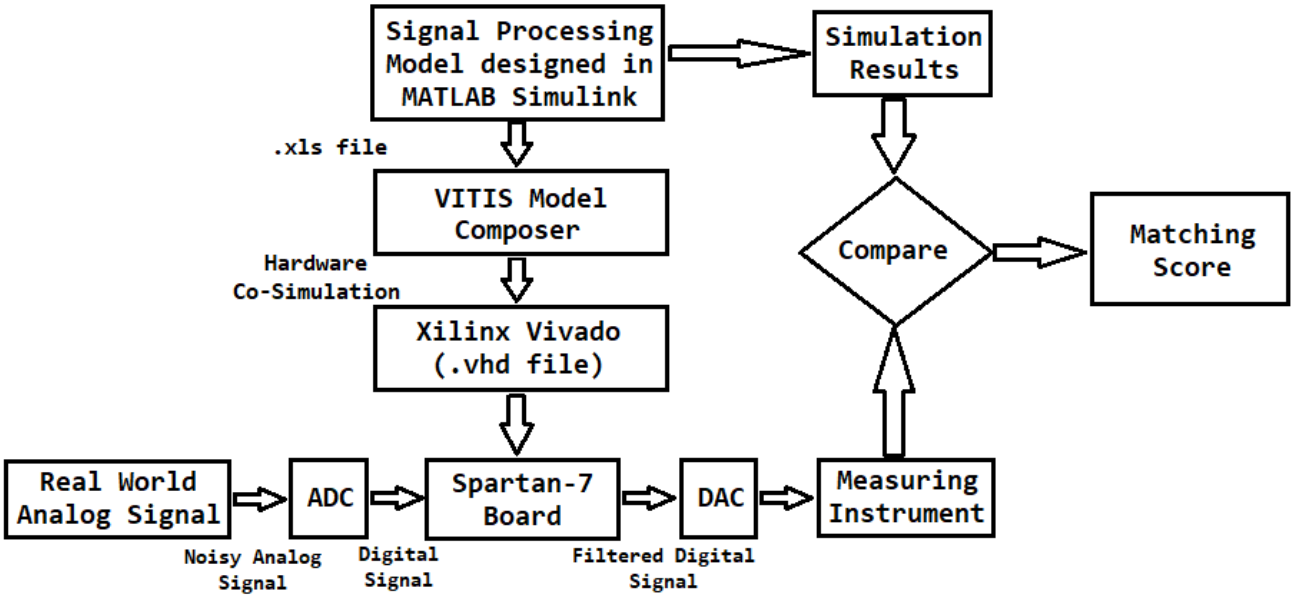

Fig.2.Signal Processing on FPGA Using Hardware Co-Simulation

FUTURE PROSPRCTS

Signal processing is a new technological advancement. Many young people are concerned about how technology can damage their future work prospects. The primary goal is to create the VHDL programme and construct the filter model utilising hardware co-simulation. The HDL compiler in the system will generate VHDL code in MATLAB. The FPGA kit is then programmed using the produced VHDL code. The noise in authentic analogue signals will then be filtered using the FPGA equipment. To do this, we will first use the Analogue to Digital Converter to provide the noisy analogue signal, and then we will apply the digital output to the Filter. To create a signal free of noise, this filtered output will also be sent via a digital to analogue converter.

CONCLUSION

Using hardware co-simulation, we have examined the approaches for signal processing on FPGA that have been up to this point offered by various authors. Additionally, we looked at several review articles regarding hardware co-simulation techniques. Numerous articles offered numerous methods for carrying out Hardware Co-simulation. We may infer from this research that the Xilinx Vitis Model composer is a versatile tool for both software- and hardware-based signal processing applications. With the least amount of time and resources, it provides fast solutions to execute challenging signal processing algorithms on hardware. This makes the Hardware implementation process clear-cut and easy.

ACKNOWLEDGMENT

We would especially like to thank our guide, Mr. Harsh Thakar, for helping us choose the topic, providing us with the excellent opportunity to work on this project of "Signal Processing on FPGA using hardware co-simulation" and guiding us throughout the project. We are very grateful to him for teaching us new terminology and approaches. We thank our guide and reviewers for their support, advice, and helpful critique, all of which have greatly aided the development of our project-related ideas. A special thanks to our H.O.D, Dr. M. V. Munot for their direct and indirect contribution in this project.

REFERENCES

[1] AndresKwasinskiand Min Wu "IEEESignalProcessingMagazine".
[2] Mohammed AbdulhalimAlareqi, Rachid Elgouri, Mateur Khalid "Real Time Hardware Co-simulation for image processing algorithms using Xilinx system generator." International Journal on Electrical Engineering and Informatics 7(4):711-723, December 2015.
[3] Vivado Design Suite User Guide: (UG897), Nov 2022.
[4] Vitis Model Composer User Guide (UG1483), Aug 2022.
[5] "Installing Vitis, Vivado Board File" by Diligent Reference.