Fenerbahce University Logical System Design

FB-CPU Design

Doğu Şahin , Agah Atay Özbelli , Zeynep Üraz , Onur Kart

Istanbul Türkiye

e-mail: { [dogu.sahin@stu.fbu.edu.tr](mailto:dogu.sahin@stu.fbu.edu.tr) , [agah.ozbelli@stu.fbu.edu.tr](mailto:agah.ozbelli@stu.fbu.edu.tr) , [zeynep.uraz@stu.fbu.edu.tr](mailto:zeynep.uraz@stu.fbu.edu.tr) , [onur.kart@stu.fbu.edu.tr](mailto:onur.kart@stu.fbu.edu.tr) },

ABSTRACT

This project aims to design and implement a processor using Verilog hardware description language in conjunction with FPGA technology. The processor, referred to as fb\_cpu, is complemented by two additional modules: a memory module named blram and a testbench module known as tb\_fb\_cpu.

The fb\_cpu module's primary function involves processing instructions stored within the memory, executing operations according to the defined instruction set. The blram module acts as a simulated Random Access Memory (RAM), responsible for storing both instructions and data accessible by the processor.

Within the project scope, the testbench module, tb\_fb\_cpu, establishes the testing environment. It initializes both the processor and memory, conducts memory checks, and evaluates the processor's functionality using predefined test cases.

Within the project scope, the testbench module, tb\_fb\_cpu, establishes the testing environment. It initializes both the processor and memory, conducts memory checks, and evaluates the processor's functionality using predefined test cases

These test cases encompass a range of scenarios, assessing the processor's capability to execute fundamental operations. Operations include loading data, performing arithmetic operations such as addition, multiplication, and subtraction, as well as evaluating control flow operations

The ultimate objective of this project is to create and validate a fundamental processor architecture on an FPGA platform.

INTRODUCTION

The current project revolves around crafting and deploying a processor using FPGA (Field Programmable Gate Array) technology. In the domain of digital design, FPGAs stand out as versatile platforms enabling users to tailor and configure digital circuits to meet specific needs. This project, executed through Verilog hardware description language, aims to fashion a fully operational processor within the FPGA framework.

At its core, this project unfolds with a dual purpose: firstly, to delve into and grasp the complexities inherent in FPGA-based design, and secondly, to fashion a functional processor capable of executing a defined set of fundamental operations. By leveraging Verilog, the project endeavors to encapsulate the essence of CPU architecture, providing insights into how instructions undergo processing and execution within the adaptable realm of programmable logic.

This initial segment serves as a springboard for a comprehensive exploration of the project's aims, intricately dissecting the architecture, functionalities, and envisioned outcomes of the FPGA-driven processor. Following sections will delve deeper into the system's blueprint, software tools engaged, and the achieved results gleaned from the project's developmental phases and rigorous testing.

SYSTEM ARCHITECTURE

The success of the proposed FPGA-based processor greatly hinges upon the meticulous selection of development tools, simulators, and a precisely defined architecture.

**Development Tools:**

-Verilog HDL (Hardware Description Language): Verilog serves as the primary language, adeptly describing the processor's hardware components with its modular and concurrent nature, enabling efficient digital circuit representation

- FPGA Platform: Hardware implementation heavily relies on an FPGA platform, allowing tailored adjustments to the FPGA model and vendor specifics based on project requirements.

**Architecture Design:**

-Processor Module (fb\_cpu): This core component executes instructions, housing instruction registers, program counters, and an arithmetic logic unit (ALU). Its operation is governed by a finite state machine, delineating operation sequences

-Memory Module (blram): Simulating a simplified RAM, this module stores instructions and data, facilitating smooth data transfer and storage operations interfacing with the processor.

-Testbench Module (tb\_fb\_cpu): Orchestrating the testing process, this module initializes the system, applies stimuli, and verifies processor responses through diverse test cases for comprehensive validation.

**Simulators:**

Verilog Simulators: Dedicated Verilog simulators like ModelSim or VCS play a crucial role in functional verification and debugging. They provide a simulated environment to analyze the processor's behavior pre-physical implementation.

**Development Workflow:**

-Synthesis and Implementation: Following successful simulation, the design undergoes synthesis to map onto the FPGA, transforming high-level descriptions into a netlist. Implementation involves strategic component placement and routing on the FPGA.

-Programming Tools: Specific to the chosen vendor, FPGA programming tools play a pivotal role in loading the synthesized design onto the FPGA.

**Interconnection:**

Wires and Signals: Crucial interconnections between modules rely on wires and signals, efficiently conveying essential information such as addresses, data, and control signals, ensuring seamless communication between processor and memory.

In conclusion, the system architecture encapsulates a cohesive integration of Verilog HDL, FPGA technology, and meticulously designed purpose-driven modules. Leveraging development tools and simulators, a systematic design, simulation, and implementation approach is ensured, guaranteeing the robustness and functionality of the FPGA-based processor.

SOFTWARE USED

For this project, we used the VirtualFPGA platform (https://avionchip.com/virtualfpga/) for FPGA development. The virtual environment offered by this web-based platform facilitated the design, simulation, and testing phases of our Verilog-based processor. Alongside this, we selected Visual Studio Code to serve as both our text editor and integrated development environment (IDE) for efficient coding and project management purposes. The integration of these tools resulted in a cohesive and user-friendly workflow, emphasizing simplicity and ease of operation specifically tailored for FPGA development.

RESULTS

The functionalities supported by the developed processor are defined by the FB-CPU ISA (Instruction Set Architecture) table, outlining the allowable operations. In the Verilog depiction of the FB processor, the ALU (Arithmetic Logic Unit) takes charge of executing arithmetic operations. Specifically, the FB-CPU accommodates three fundamental arithmetic operations: addition, subtraction, and multiplication. Executing each operation relies on the incoming transaction code, and the resulting outputs are stored within the ACC (accumulator) register.

Test cases 1, 2, and 3 were individually tested and rigorously examined using simulation in the AvionChipSimulator. This comprehensive testing process not only validated our project but also contributed significantly to enhancing our algorithmic reasoning abilities and mastery of the Verilog language.

This experience was simulated and performed accordingly with AvionChipSimulator.

PROJECT TEAM

Doğu Şahin

Role: Chip researcher

Agah Atay Özbelli

Role: PowerPoint presenter

Zeynep Üraz

Role: Powerpoint preparative

Onur Kart

Role: Editor

NOTE: We are going to upload our CVs on Github

REFERENCES FILES

The Youtube video link;

<https://youtu.be/5Vk80HxUfDo?si=h-dxaraPLztf42QK>

The Gıthub link;

github.com/sahindogu

github.com/Zeynepuraz

github.com/agahatay

github.com/onurkart

REFERENCES

1-)<http://www.levent.tc/>

2-) IEEE Xplore

3-) Wikipedia.com