Documentation

Pi-Sense

26-05-2023

Important Notes:

- ***The descriptions in italics in this document (except for some section headings) are exemplary and explanatory and must be removed from the completed report.***
- ***Identify which section of this report was created by which team member***
- ***Your documentation should have ca. 8 pages (content! Without cover sheet, references, appendix etc.).***

1 Team members

1. Arsany Girgis
2. Emirkan Sali
3. Lochana Abhayawardana

2 Introduction

Digital system design encompasses a wide range of applications, from simple control circuits to complex computing systems. The ability to efficiently design and implement such systems is crucial in various domains.
FPGAs are programmable integrated circuits that offer significant advantages in the design and development of digital systems. Unlike fixed-function ASICs (Application-Specific Integrated Circuits), FPGAs can be reconfigured to implement custom logic circuits. This flexibility allows for rapid prototyping, iterative refinement, and customization of designs based on specific application requirements. VHDL enables efficient design representation, simulation, synthesis, and verification. VHDL serves as an essential tool for capturing the intended functionality of digital systems, facilitating their implementation on FPGAs.

The 4-bit calculator project serves as a concrete example to demonstrate the significance of FPGAs and VHDL in digital system design. FPGAs offer a highly configurable platform that can accommodate the calculator's logic elements, arithmetic circuits, and interconnections. The reconfigurable nature of FPGAs allows for iterative development, testing, and refinement of the calculator design. Additionally, VHDL provides a robust and expressive language to describe the calculator's behavior and structure. By leveraging VHDL, we can efficiently capture the complex arithmetic operations, control logic, and interface requirements of the calculator, facilitating its implementation on the FPGA.

The objective of our project is to design and implement a 4-bit calculator using FPGAs and VHDL. The calculator supports basic arithmetic operations, including addition, subtraction, multiplication, and division. We aim to explore the capabilities of FPGAs and the power of VHDL in efficiently representing the calculator's functionality. Furthermore, we intend to demonstrate the practical application of FPGAs and VHDL in the digital system design domain, showcasing their importance in implementing complex arithmetic computations.

3 Concept description

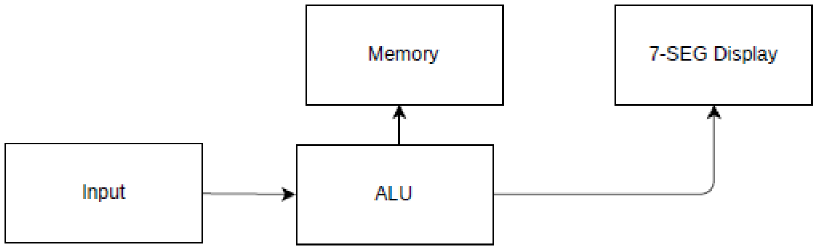

Figure 1

The concept of a 4-bit calculator on an FPGA (Field-Programmable Gate Array) involves designing and implementing a digital circuit that performs basic arithmetic operations on 4-bit binary numbers. The FPGA serves as a programmable hardware platform that allows us to create and configure custom digital circuits to perform specific functions.

- *Designing the Calculator:*

The calculator design starts with defining the input and output requirements. In this case, the inputs are two 4-bit binary numbers (a and b), an operation select signal (op), and the outputs are the result (result), carry-out (carry_out), and overflow (overflow) signals.
The desired operations (addition, subtraction, multiplication, and division) are implemented using appropriate arithmetic and logical circuits within the FPGA design.

- *VHDL Implementation:*

The calculator's functionality is described using a hardware description language like VHDL or Verilog.
The operations are typically implemented using standard arithmetic and logical operators provided by the language, such as addition (+), subtraction (-), multiplication (*), division (/), and remainder (rem).
Additional logic is implemented to handle special cases, such as division by zero.

- *Synthesis and Place & Route:*

Once the VHDL/Verilog code is written, it is synthesized using a synthesis tool that converts the code into a gate-level representation.
The synthesized design is then subjected to a place and route (P&R) process, where the FPGA resources (logic elements, flip-flops, interconnects, etc.) are allocated and connected to implement the desired functionality.

4 Project/Team management

We used agile as a project method where we had weekly sprints
All members participated in all tasks.