# 4-bit Calculator

*Team E*

16.06.23

**Team members:**

- Amit Chakma
- Md Limon Apu
- *Yashodhan Vishvesh Deshpande*

# 1 Introduction

Building a 4-bit unsigned calculator is the subject of our semester's assignment. Various mathematical operations, such as binary addition, subtraction, multiplication, and division, are to be implemented. In order to do this, we combine many modules, including Adder, Subtractor, Multiplier, and Divider, using sub-modules that use simple gates. Following the design and coding of the functionality in VHDL, the calculator is next constructed by translating the schematics to a PCB layout. We go into great detail at every stage about the software used in the related study. Due to its programmability and capacity for numerous testing, we decided to utilize FPGA for this project. The components are configured using VHDL which is a hardware description language.

# 2 Concept description

1. **Block diagram of our application**

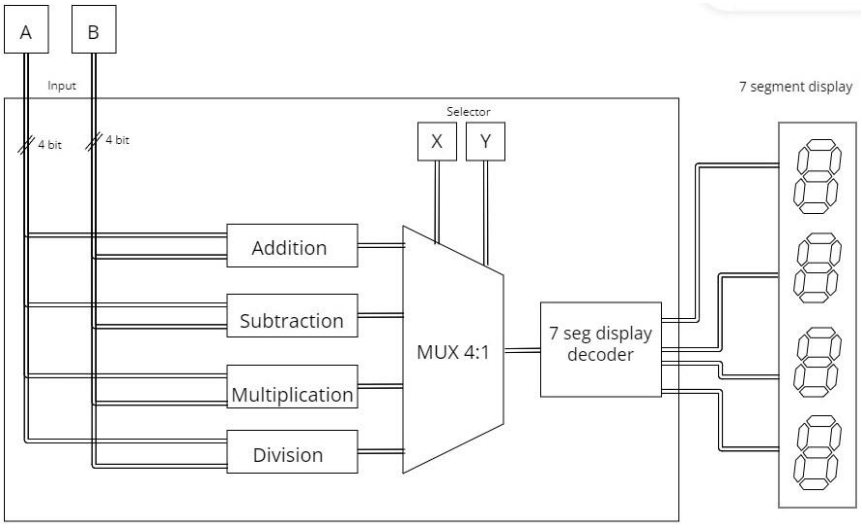


Fig.1: Main function calculator

2. **Application of our project:**

   The 4-bit calculator system consists of input slide switches on the PCB, through which binary values are provided. These inputs are then processed by individual blocks for multiplication, addition, subtraction, and division, generating the corresponding

results. The outputs from these blocks serve as inputs to a 4:1 multiplexer, which selects the appropriate driver line based on 1-bit binary selector inputs, indicating the desired operation. The selected output is then decoded by a decoder, converting it into decimal values for display on the 7-segment displays. The calculator's main purpose is to perform basic arithmetic operations, including addition, subtraction, multiplication, and division.

# 3 Project/Team management

Our project follows an iterative development model, allowing us to work on and test our scenario in a step-by-step manner. This approach enables us to revisit previous steps, make necessary adjustments, and move forward based on our findings. To ensure an organized workflow and effective communication, we divided specific tasks among team members and held weekly meetings to discuss progress and plan future steps. During the implementation phase, we prioritized in-person collaboration whenever possible to work collectively towards achieving our project objectives. Below is a documentation of each group member's progress, highlighting the dates when meetings took place. This systematic approach helped us stay on track and successfully accomplish our project goals.

| | | | | | |
|---|---|---|---|---|---|
| | Date | 25.05.23 | 31.05.23 | 8.06.23 | 16.06.23 |
| | Task | Task-1 | Task-2 | Task-3 | Task-4 |
| Md Limon Apu | To-do | Research on project | Research on VHDL | Implement in Xillix | Connect to FPGA |
| | Status | Done | Done | Done | Done |
| Yashodhan Vishvesh Deshpande | To-do | Research on project | Research on Schamtic | Implement in Kicad | Schamatics |
| | Status | Done | Done | Done | Done |
| Amit Chakma | To-do | Research on project | Research on VHDL | Implement in Modelsim | Simulation |
| | Status | Done | Done | Done | Done |

Fig.2: Workload of group member

# 4 Technologies

- *VHDL*

VHDL (VHSIC Hardware Description Language) is a powerful hardware description language used for modeling the behavior and structure of digital systems. It allows designers to describe and specify complex digital circuits at various levels of abstraction, from system-level designs to individual logic gates. VHDL is commonly used for design entry, documentation, and verification purposes.

During the implementation phase of our project, we utilized ModelSim, a software tool developed by Siemens (previously Mentor Graphics), which is specifically targeted for Intel® FPGAs devices. ModelSim provides a multi-language environment and serves as a valuable tool for developing VHDL code. It offers features such as syntax checking, compilation, and most importantly, simulation capabilities. The user-friendly interface of ModelSim allows for efficient code development and simulation, aiding in the verification and testing of our design.

By using VHDL and ModelSim in our project, we were able to effectively design, develop, and test our digital system, ensuring its functionality and correctness before proceeding with further stages of implementation.

- *FPGA*

The Xilinx Nexys A7 FPGA board, powered by the Xilinx Artix-7 FPGA, serves as the core hardware component for our calculator project. The Artix-7 FPGA offers a programmable and versatile platform capable of executing a wide range of logical operations. Unlike fixed components like ASICs, the FPGA provides the advantage of reprogram ability, allowing us to update and test our code multiple times during the development process. The specific FPGA used, EP4CE22E22C8N from Intel's Cyclone IV E family, features 22320 cells utilizing 60nm technology and operates at a voltage of 1.2V. With its extensive capabilities and flexibility, the Xilinx Nexys A7 FPGA board enables us to implement and validate our calculator design efficiently on a physical PCB.

- *KiCAD*

KiCAD is a powerful electronic design automation (EDA) software that we employed for our project to design the printed circuit board (PCB). With its comprehensive set of tools, KiCad facilitated seamless integration of the FPGA module, input switches, and 7-segment displays into our PCB design. We followed a two-layer PCB approach, optimizing signal routing and simplifying the overall design. By utilizing KiCAD's features for schematic capture, component placement, and PCB routing, we were able to create an efficient and well-structured PCB layout. The software's extensive library content further aided in the selection and placement of components, ensuring proper electrical connectivity. Overall, KiCad played a crucial role in enabling the successful design and realization of our project's PCB.

- *Vivado*

*In our project, we had utilized Vivado, a software tool provided by Xilinx, which greatly aided us in implementing and programming the FPGA module for our calculator system. Vivado served as a comprehensive and user-friendly development environment, offering a wide range of features that were essential for our FPGA design. We took advantage of its RTL synthesis, simulation, and place-and-route algorithms to optimize our design for optimal performance, power efficiency, and efficient resource utilization. Vivado's intuitive graphical interface allowed us to easily configure the FPGA, assign pins, and generate the necessary programming bitstream. We also greatly appreciated its robust verification and debugging capabilities, which allowed us to thoroughly test and validate our design before deploying it onto the FPGA. Vivado played a pivotal role in the successful development and programming of our FPGA module, ensuring a seamless integration into our calculator project.*

# 5 VHDL and FPGA Implementation

In our project, we embarked on a systematic approach by initially coding and testing essential components, including the Half Adder, Full Adder, and Ripple-Carry Adder. With dedicated testbenches, we ensured their accurate functionality. Progressing further, we developed larger modules such as the 4-bit Adder, Subtractor, Multiplier, Divider, Decoder, and Multiplexer, specifically designed for 4-bit operations. Thoroughly tested and validated, these modules were seamlessly interconnected to form a cohesive calculator system.

### I. Adder/Subtractor:

Notably, the integration of the Adder and Subtractor involved a mode selector and XOR gates for 2's complement conversion of input B. Accurate carry bit counting was also incorporated. Illustrated in the figure, our design showcases the interconnected components, encompassing inputs and outputs, forming the robust foundation of our 4-bit calculator system.

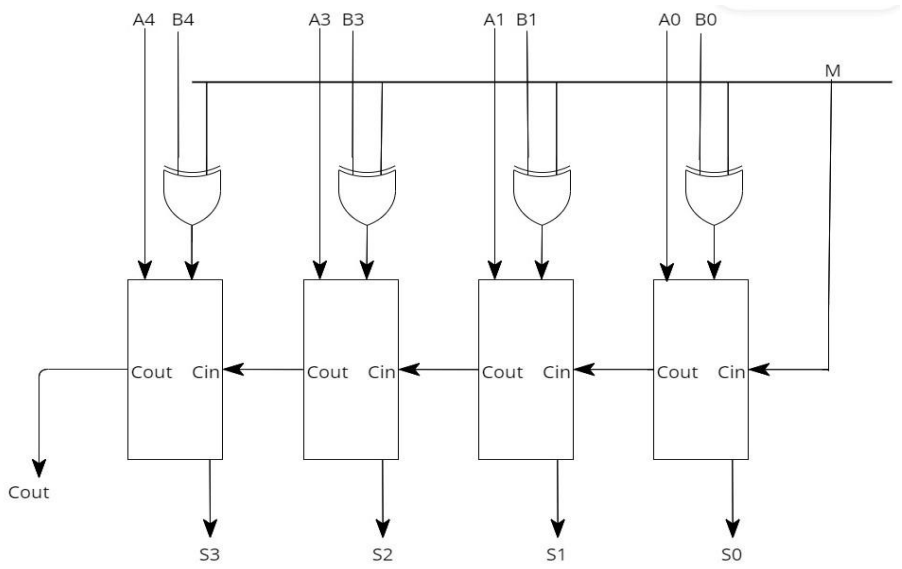


Fig.4: 4-bit adder/subtractor

### II. Multiplier:

### III. Divider:

### IV. State machine:

# 6 PCB Design

# 7 Sources/References

*GitHub: https://github.com/Yashodhandesh/HW-AEE-Group-E*