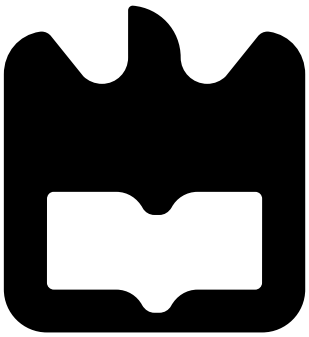

AAD - Assignment 2

107474-Joseane Pereira
109050-Gabriel Costa
Universidade de Aveiro, DETI

December 28, 2024

Contents

1 Introduction

This assignment focuses on the implementation and analysis of different accumulator designs in VHDL, including a single-cycle version, a pipelined version, and versions with shift functionality.

2 Implementations

2.1 Single-Cycle Accumulator

As suggested with the provided starting point, the basic accumulator was implemented in `accumulator_single_cycle.vhd` with the following features:

- Simple write and read ports
- Direct accumulation without pipelining
- Basic addition functionality
- Synchronous reset to initialize the accumulator
- Parameterizable data width for flexibility

The accumulator operates by reading an input value, adding it to the current stored value, and then writing the result back to the storage register in a single clock cycle. This design ensures minimal latency but does not support high-frequency operations due to the lack of pipelining.

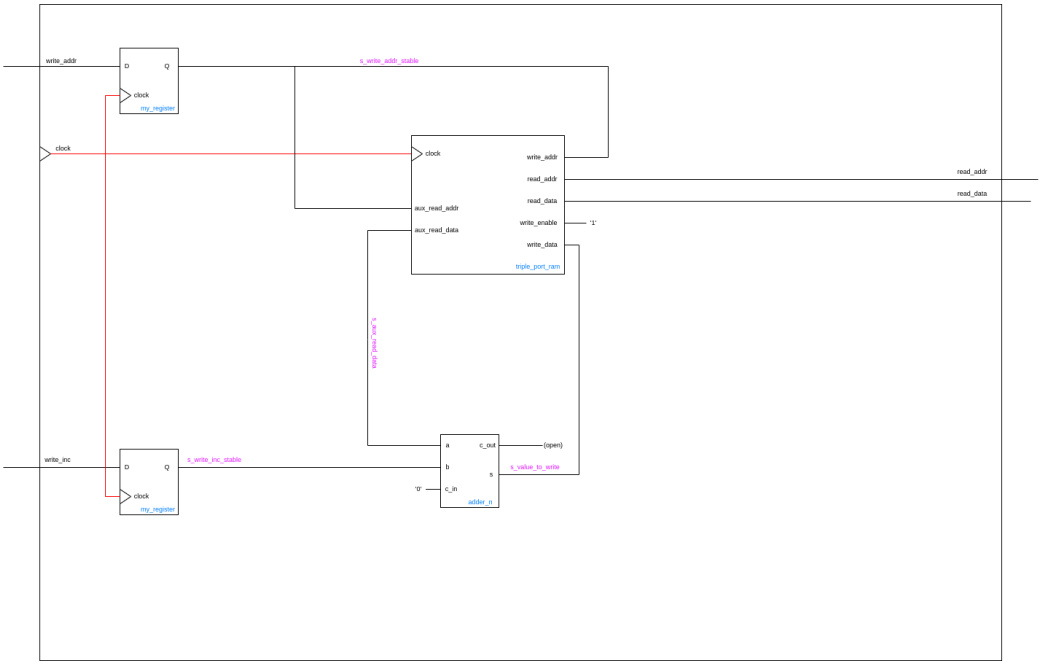

Figure 1: Single-cycle accumulator block diagram

2.2 Pipelined Accumulator

The pipelined version `accumulator_pipeline.vhd` improves upon the basic design by:

- Adding pipeline stages to improve throughput
- Using registers to stabilize signals
- Implementing hazard detection

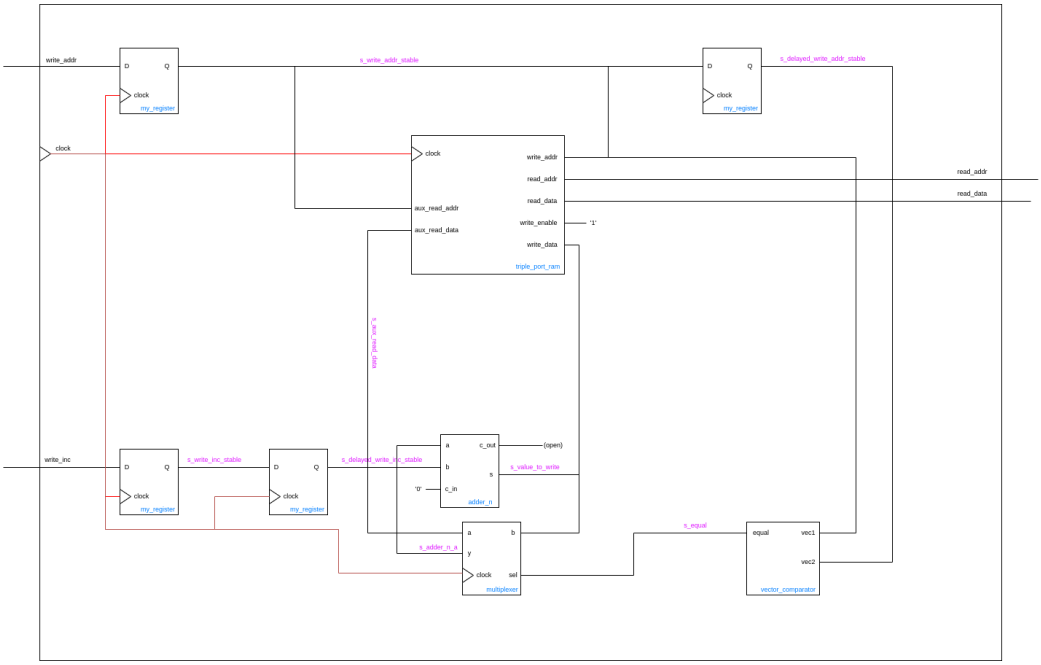

Figure 2: Pipelined accumulator block diagram

2.3 Shift Accumulator

We implemented a version of the accumulator using a barrel shifter for both the pipelined and single cycle versions (`shift_accumulator_pipeline.vhd` and `shift_accumulator.vhd`), both of which improve upon the basic design by:

- Dynamic shift operations on input data
- Configurable shift amounts
- Integration with the accumulation logic

The shift accumulator design leverages a barrel shifter to perform dynamic shift operations on the input data. This allows for efficient manipulation of data bits, which is crucial for various digital signal processing tasks. The configurable shift amounts enable the design to be flexible and adaptable to different requirements, making it suitable for a wide range of applications.

In the single cycle version, the shift and accumulation operations are performed within a single clock cycle. This design is optimized for scenarios where speed is critical, and the simplicity of the control logic is a priority.

The block diagram in Figure 3 illustrates the architecture of the single cycle shift accumulator.

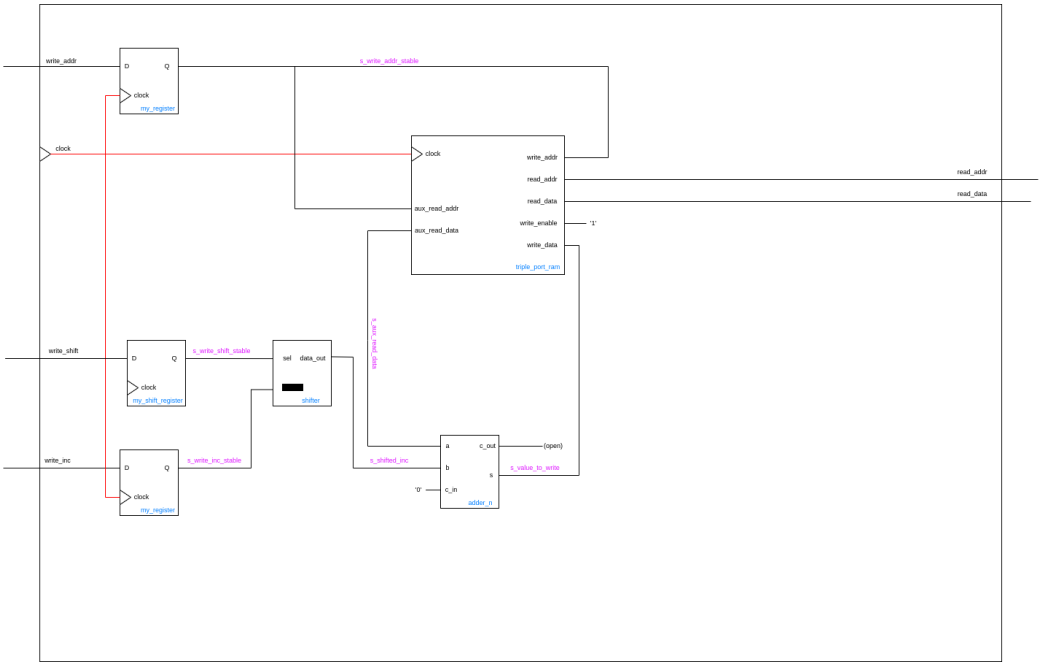

Figure 3: Shift accumulator single cycle block diagram

For the pipelined version, the design is divided into multiple stages, allowing for higher clock frequencies and improved throughput. Each stage of the pipeline performs a portion of the shift and accumulation operations, enabling the design to handle higher data rates. This is particularly beneficial in applications where large volumes of data need to be processed efficiently. The block diagram in Figure 4 shows the architecture of the pipelined shift accumulator.

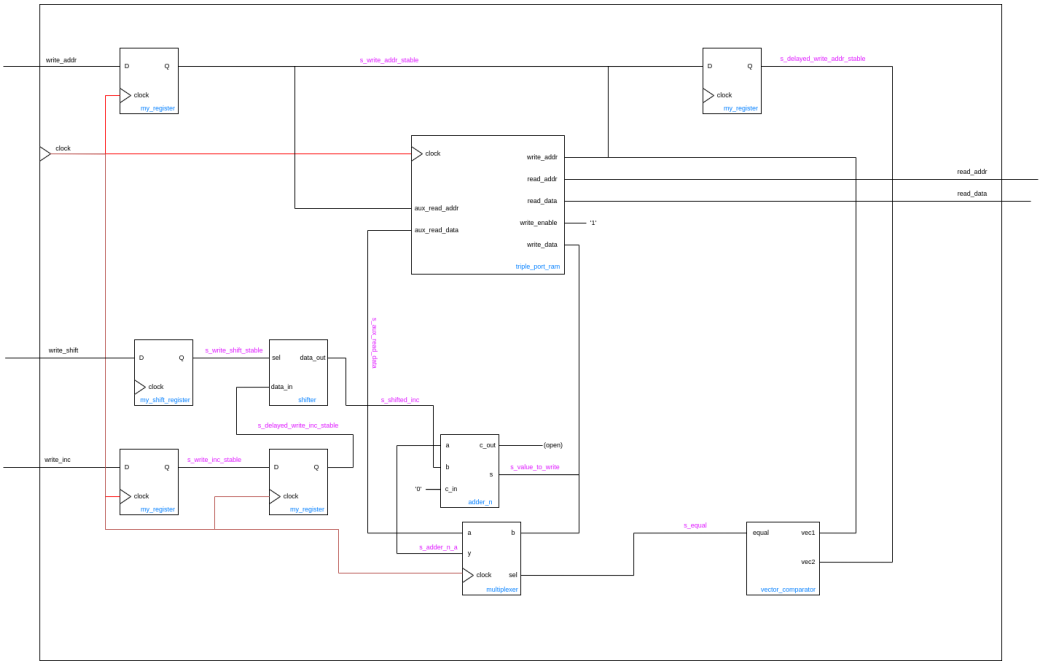

Figure 4: Shift accumulator pipeline block diagram

Overall, the shift accumulator designs provide a robust and flexible solution for data manipulation and accumulation tasks, with the pipelined version offering enhanced performance for high-speed applications.

3 Components

Some of the Components were provided by the professor, and some were implemented by us. Key components used in the designs include:

- Triple Port RAM for memory storage
- Registers for signal stabilization
- Adder N for arithmetic operations
- Shifter for shift operations
- Vector Comparator for address comparison

4 Testing

Testing was performed using:

- GHDL simulator
- VCD waveform generation
- A provisded testbench for the single-cycle accumulator adapted for the pipelined and shift versions

5 Results and Analysis

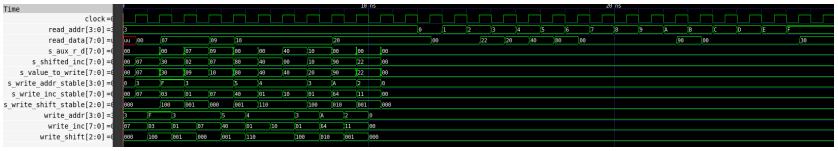

Figure 5: Waveform of the single-cycle accumulator testbench

5 shows the waveform generated by the testbench for the single-cycle accumulator. The testbench was adapted for the pipelined and shift versions, and the results were similar. The shift accumulator was tested with different shift amounts, and the results were as expected.

6 Conclusions

This project demonstrated the design and analysis of various accumulator architectures in VHDL: single-cycle, pipelined, and shift-based. The single-cycle design offered simplicity and minimal latency but was limited in scalability for high-speed operations. The pipelined accumulator addressed this by improving throughput and enabling higher clock frequencies. The shift accumulator added flexibility with dynamic shift operations, making it suitable for data-intensive applications.

Testing validated the functionality of all designs, with each meeting expected performance criteria. Overall, the project highlighted the trade-offs between simplicity, speed, and flexibility in digital system design. Future improvements could focus on optimizing power and area efficiency while exploring additional features.