Design and Implementation of an FIR Filter with a Configurable Pipelined Multiplier

Seyed Sadra Ghavami

1 Introduction

This project presents the design and implementation of a Finite Impulse Response (FIR) filter in Verilog, focusing on an efficient, hardware-optimized approach suitable for FPGA applications. A primary goal is to achieve high performance by reducing the critical path of the filter's datapath. The critical feature of this design is a configurable pipelined multiplier with a single critical path, which allows high-speed operation by optimizing data propagation. The FIR filter is designed to handle signed numbers, with sign extension integrated within the multiplier structure.

2 Design Approach

The FIR filter's hardware design includes two major components: the datapath and the control unit. The datapath incorporates a pipelined multiplier and accumulator that manage multiplication and accumulation of input data with filter coefficients. Figure 1 illustrates the initial data path design, which was iteratively optimized to reduce delay.

The design constraints specified in this project require that the FIR filter handle high-frequency data with minimal delay, achievable by a pipelined multiplier with an optimized critical path.

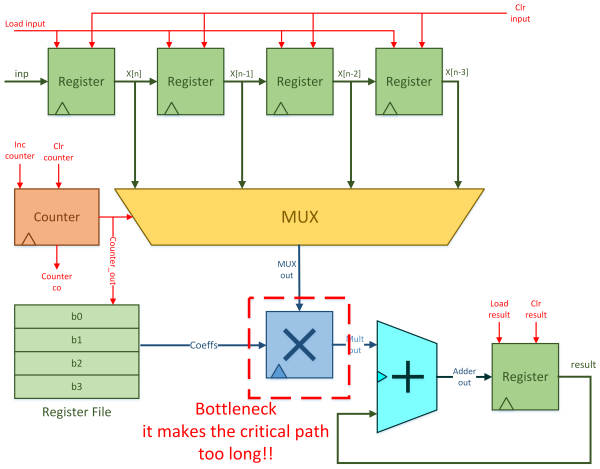

Figure 1: Initial data path design for FIR filter.

In the initial design, the critical path remained a bottleneck, as delay was compounded by full adder stages. Therefore, an optimized multiplier structure was adopted to achieve a critical path of one, reducing propagation delay between the multiplier and the accumulator.

3 Configurable Pipelined Multiplier

The pipelined multiplier is implemented as an array multiplier with distinct stages, each containing multiple full adders. This structure minimizes the critical path by limiting the number of combinational logic elements in each pipeline stage. Figure 2 shows the architecture of the pipelined array multiplier.

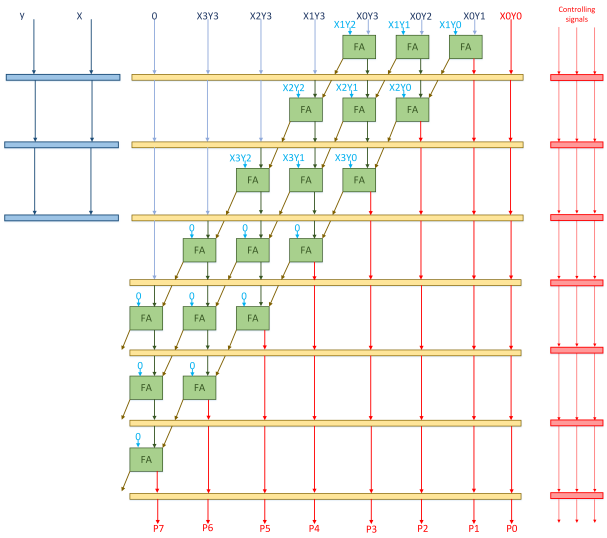

Figure 2: Array multiplier with optimized pipeline stages.

The multiplier stages are configured to achieve a critical path of one by careful placement of full adders and registers. Each stage computes partial products and propagates them forward in a manner that maximizes computation speed.

4 Final Data Path Design

The optimized multiplier was integrated into the final data path of the FIR filter. The data path includes multiple registers that sequentially shift input data and filter coefficients through the pipeline stages. In each clock cycle, partial products are accumulated to calculate the filter output. Figure 3 illustrates the final data path structure.

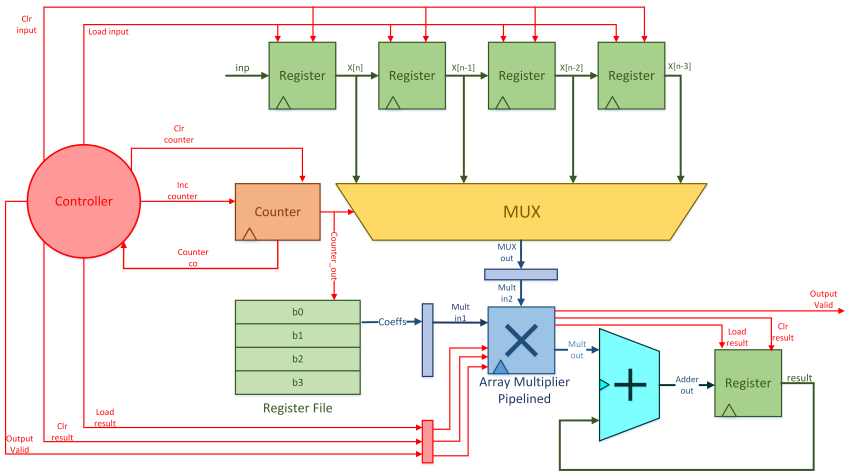

Figure 3: Final data path design with the integrated pipelined multiplier.

The data path operates as follows: 1. **Data Loading**: Input samples are loaded into the first register, while previous samples are shifted to subsequent registers. 2. **Coefficient Loading**: Filter coefficients are loaded into the pipeline at each clock cycle. 3. **Accumulation**: Each stage performs partial multiplication, and results are accumulated in the final output register, producing the FIR output.

5 Controller Design

The control unit orchestrates data flow through the FIR filter, managing input loading, coefficient application, and output validation. Figure 4 illustrates the controller state machine, which operates based

on the input validity signal and ensures that each stage in the pipeline is correctly synchronized.

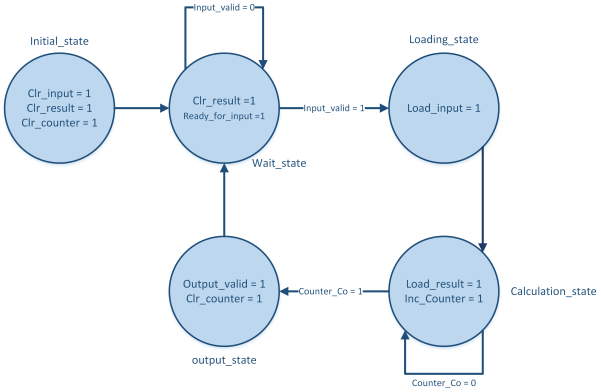

Figure 4: Controller state machine for FIR filter operation.

Controller states: - **Initial State**: All registers are reset to zero to prepare the FIR filter for new data. - **Wait State**: The controller waits for a valid input signal (`input_valid`). If the signal is high, the controller transitions to the loading state. - **Loading State**: Data samples and coefficients are loaded into the pipeline, shifting previous data through the stages. - **Calculation State**: The data and coefficients propagate through the pipeline, and partial products are computed. - **Output State**: Once the calculation completes, the controller activates the `output_valid` signal, indicating that the filter output is ready.

6 Verification and Testing

A testbench was developed in Verilog to verify the FIR filter's accuracy. The testbench provides input samples and compares the filter output to expected values, confirming functionality through waveforms and assertions. Simulation is conducted to validate the filter's behavior under various input conditions, ensuring that all pipeline stages operate as expected. Figure 5 shows a sample simulation result.

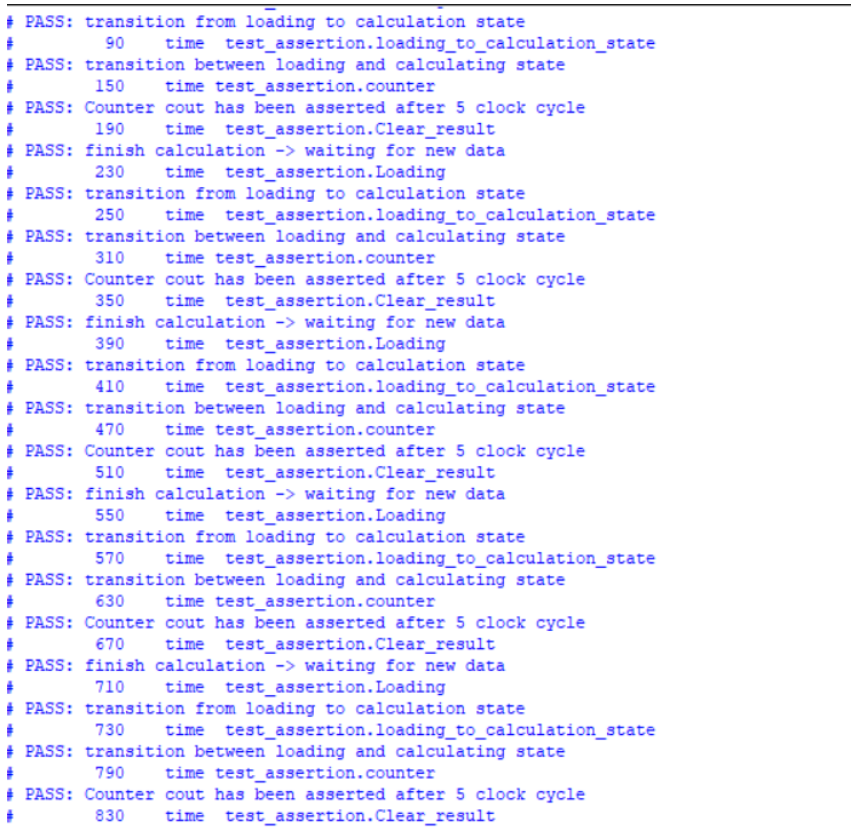

Figure 5: Testbench simulation output with signal validation.

Testing procedures using assertion: - **Input Validation**: Each input sample is loaded, processed, and accumulated correctly within the pipeline. - **Output Assertions**: Assertions check the filter's output at each clock cycle to confirm accuracy. - **Timing Analysis**: Waveforms are examined to ensure correct timing of data propagation and signal validation, with `output_valid` indicating the completion of processing.

7 Synthesis and Performance Analysis

The FIR filter was synthesized for FPGA implementation using the Cyclone II device family. Resource usage and operating frequency were evaluated for different configurations of filter length and data width. The pipelined multiplier enables the design to achieve high frequencies with minimal logic utilization.

Filter Length	Data Width	Total Logic Elements	Max Frequency
50	8	1383	233 MHz
50	16	4041	168 MHz
100	8	2022	185 MHz
100	16	5386	171 MHz

Table 1: Resource utilization and maximum operating frequency for FIR filter configurations.

8 Conclusion

The FIR filter with a configurable pipelined multiplier demonstrates high efficiency, achieving low latency and high-speed performance. The optimized pipeline and control logic make this design suitable for real-time applications requiring high throughput. Future improvements could include support for variable filter coefficients and higher-precision multiplier configurations for enhanced filtering accuracy.