Analysis and Comparison of Pipelined, Parallel, and Combined Pipelined-Parallel FIR Filter Implementations

March 13, 2025

Abstract

This report presents a detailed technical analysis of three hardware implementations of a low-pass FIR filter. The designs are categorized as:

1. **Pipelined Architecture** (single multiplier-accumulator with pipeline stages),
2. **Parallel Architecture** (processing multiple taps per clock cycle), and
3. **Combined Pipelined & Parallel Architecture** (integrating both approaches).

We evaluate each architecture with respect to area, timing performance, and power consumption, and discuss the underlying trade-offs in detail.

1 Introduction and Design Specifications

The FIR filter was designed using MATLAB with the following specifications:

- **Transition Region:** Approximately from 0.2π to 0.23π rad/sample.
- **Stopband Attenuation:** At least 80 dB.
- **Filter Taps:** A 100-tap filter (or larger) with coefficients quantized to 16-bit fixed-point format.
- **Implementation Platform:** FPGA-based hardware implementation using Verilog.

The filter coefficients, derived via MATLAB, are implemented in hardware through three distinct architectures. While all designs fulfill the signal processing requirements, their implementations vary in the following key aspects:

- **Latency:** The number of clock cycles needed to produce an output sample.
- **Throughput:** The maximum number of samples processed per unit time.
- **Area:** Resource utilization in terms of LUTs, flip-flops, and DSP blocks.
- **Power Consumption:** Total on-chip power with emphasis on dynamic versus static components.

2 Hardware Implementation Results and Analysis

2.1 Resource Utilization

Table 1 summarizes the utilization metrics for the three architectures:

Table 1: Resource Utilization Summary

Architecture	LUTs (%)	FFs (%)	DSPs	I/Os (%)
Pipelined	2–6	3–10	1	∼48
Parallel	2–3	3–10	2–3	∼48
Combined	5–6	3–10	2–3	∼48

Analysis:

- The *pipelined architecture* reuses a single DSP block across multiple pipeline stages. This minimizes the DSP footprint but increases the overall processing latency.

- The *parallel architecture* duplicates multiplier-accumulator units to process multiple filter taps per clock cycle, which inherently increases DSP usage (by a factor proportional to the parallelism factor L) and slightly increases the LUT and flip-flop count.
- The *combined architecture* leverages both pipelining and parallelism, achieving a balance where deeper pipelining allows for high clock frequencies, while parallelism reduces the number of cycles required per sample.

2.2 Timing Performance

Table 2 provides the worst negative slack (WNS), worst hold slack (WHS), and the maximum estimated clock frequency for each implementation.

Table 2: Timing Closure Summary

Architecture	**WNS (ns)**	**WHS (ns)**	**Max Freq (MHz)**
Pipelined	0.13–2.18	0.03–0.07	200–350
Parallel	0.13–0.35	0.02–0.12	150–300
Combined	0.13–2.18	0.02–0.12	150–350

Analysis:

- *Pipelining* improves the maximum operating frequency by breaking long combinational paths into shorter segments. The increased pipeline depth results in higher throughput at the cost of increased latency measured in clock cycles.
- The *parallel design* reduces the number of cycles per filter operation but can impose additional routing complexity, leading to slightly tighter timing margins.
- The *combined architecture* benefits from both techniques. The pipelined stages help maintain a high clock frequency while parallelism decreases the effective latency. The timing reports show that while the worst-case negative slack can be high (up to 2.18 ns), it is offset by the overall design margin, ensuring robust operation.

2.3 Power Estimation

Table 3 presents the estimated total on-chip power and the percentage contribution of dynamic power for each design.

Table 3: Power Consumption Summary

Architecture	**Total Power (W)**	**Dynamic Power (%)**
Pipelined	0.11–0.16	19–35
Parallel	0.11–0.16	19–35
Combined	0.13–0.16	20–35

Analysis:

- *Static Power* dominates in FPGA implementations, typically due to leakage currents and always-on circuitry.
- *Dynamic Power* is a function of the clock frequency, capacitive load, and switching activity. The parallel and combined architectures show a marginal increase in dynamic power due to the simultaneous switching of multiple DSP blocks and additional registers.
- Despite the increase, overall power remains low (well below 0.2 W), which is acceptable for modern FPGA applications.

3 Conclusions

3.1 Throughput and Latency Considerations

- **Pipelined Architecture:**

  - *Latency:* The total latency is the product of the number of pipeline stages. Although each clock cycle processes one tap, the deep pipeline ensures that a new sample is accepted every clock cycle after the initial fill.
  - *Throughput:* High due to the continuous streaming nature once the pipeline is full, but the effective throughput in terms of samples processed is limited by the sequential multiplication and accumulation operations.
- **Parallel Architecture:**
  - *Latency:* Fewer cycles are needed per sample because multiple taps are processed concurrently. However, this concurrency can lead to longer combinational paths if not properly pipelined.
  - *Throughput:* Enhanced throughput due to the reduction in cycle count per filter operation, making it ideal for high-speed applications.
- **Combined Architecture:**
  - *Latency and Throughput:* This architecture strategically blends pipelining and parallelism. The design achieves lower latency per sample (comparable to the parallel architecture) while also ensuring that the long combinational paths are broken down, thereby sustaining high clock frequencies.
  - The improved throughput is a direct result of reduced effective cycle count per operation.

3.2 Implications of Timing and Power Results

- The positive timing slack across all designs implies that the implementation is robust against process, voltage, and temperature variations. The pipelined and combined architectures, with their higher maximum frequencies, offer significant headroom for scaling the design.
- The slight increase in dynamic power for the parallel and combined architectures is a natural trade-off for enhanced throughput. Since dynamic power $P_{dyn} \propto f \times C \times V^2 \times \alpha$, the increased switching activity in parallel units is expected.
- Area and power constraints are critical in FPGA-based designs. The measured resource utilization confirms that even the most resource-intensive architecture (combined pipelined-parallel) occupies only a small fraction of the available FPGA resources, leaving ample room for additional functionality or further optimizations.

3.3 Final Conclusions

All three FIR filter implementations satisfy the design specifications with respect to frequency response and quantization accuracy. The key takeaways are:

1. **Timing Robustness:** Positive slack in all designs ensures reliable operation. The pipelined and combined designs are particularly advantageous when targeting higher clock frequencies.
2. **Area Efficiency:** The pipelined design is the most area-efficient in terms of DSP usage. However, the parallel and combined architectures offer superior throughput with only a modest increase in resource usage.
3. **Power Efficiency:** Total on-chip power is low for all implementations. Although parallel processing incurs slightly higher dynamic power due to increased activity, the overall power consumption remains within acceptable limits for low-power applications.
4. **Application-Driven Choice:**
   - Use the **pipelined approach** for applications where minimal area and DSP usage are paramount.
   - Opt for the **parallel approach** when reducing latency is critical.
   - The **combined pipelined-parallel approach** is ideal for applications that demand both high throughput and high operating frequency, with robust timing margins.