Pipelined FIR Filter Implementation

March 6, 2025

1. Code Structure

- **Parameters & Ports:**
  - Parameterized number of taps (N = 211), input data, coefficient, multiplier, and accumulator widths.
  - Defined input ports (`clk`, `rst_n`, `data_in`, `data_in_valid`) and output ports (`data_out`, `data_out_valid`).
- **Coefficient Array:**
  - A local parameter array of signed 16-bit coefficients is defined to represent the FIR filter taps.
- **Sample Storage:**
  - A shift register array stores the latest N input samples, updating with every valid input.
- **FSM and Pipelining:**
  - Four states (ST_IDLE, ST_MULT, ST_ADD, ST_DONE) control the pipeline.
  - In ST_IDLE, a new sample is shifted in.
  - ST_MULT performs the multiplication of the current sample and its corresponding coefficient.
  - ST_ADD accumulates the product.
  - ST_DONE outputs the final sum when all taps have been processed.

2. Pipelined FIR Filter Architecture

- **Pipelining Approach:**
  - A two-cycle pipeline per tap: one cycle for multiplication (using the DSP block capabilities) and the next for accumulation.
  - The pipelined design improves throughput by processing each tap in sequential stages while overlapping operations.
- **Finite State Machine (FSM):**
  - Manages the pipelined stages with clear state transitions.
  - The FSM ensures that once a new sample is available, the shift register is updated, and multiplication-accumulation for all taps occurs sequentially.
- **Testbench and Constraints:**
  - A dedicated testbench verifies the design by providing clock (100 MHz with 10 ns period) and reset stimuli, and then driving an impulse and zero input to observe the filter output.
  - Timing constraints ensure that the design meets the 100 MHz clock requirement.

Conclusion: The design's modularity, through parameterization and FSM-driven pipelining, provides a robust and efficient FIR filter implementation suitable for high-speed DSP applications. The testbench confirms correct functionality under the imposed timing constraints.