Task 4: Digital FIR Filter Design

This report presents the design and simulation of a digital FIR (Finite Impulse Response) filter. The design uses a low-pass filter as an example, with 16 taps and a Hamming window for coefficient generation. The implementation is demonstrated using Verilog code along with performance analysis.

FIR Filter Basics

An FIR filter is defined by the equation:

y[n] = h0*x[n] + h1*x[n-1] + ... + hM*x[n-M]

where h0..hM are the filter coefficients.

Key characteristics of FIR filters include:

- Linear phase response (when coefficients are symmetric)
- Always stable (finite impulse response)
- Can be implemented efficiently with shift registers and MAC units.

Filter Specifications

Example Specifications:

- Filter Type: Low-Pass FIR
- Number of Taps: 16
- Cutoff Frequency (Fc): 0.25 * Fs
- Window: Hamming
- Sampling Frequency (Fs): 50 MHz (example)

Verilog Code Snippet

```
module fir_filter (
    input clk,
    input reset,
    input signed [15:0] x,
    output reg signed [31:0] y
);
    parameter N = 16;
    reg signed [15:0] shift_reg [0:N-1];
    wire signed [15:0] coeff [0:N-1] = '{16, 32, 48, 64, 64, 48, 32, 16,
                                         16, 32, 48, 64, 64, 48, 32, 16};
    integer i;
    always @(posedge clk) begin
        if (reset) begin
            for (i=0; i<N; i=i+1) shift_reg[i] <= 0;
            y <= 0;
        end else begin
            shift_reg[0] <= x;
            for (i=1; i<N; i=i+1)
                shift_reg[i] <= shift_reg[i-1];
            y <= 0;
            for (i=0; i<N; i=i+1)
                y <= y + shift_reg[i] * coeff[i];
        end
    end
endmodule
```

Performance Analysis

The filter processes one input sample per clock cycle after initial pipeline fill. The latency is equal to the number of taps (16 cycles). With proper pipelining, the design can operate at high clock frequencies (e.g., >100 MHz on FPGA).