Experiment 4

Part 1 (5 points)

Implement two different kinds of Finite Impulse Response (FIR) filters:

a) Fully parallel FIR filter below.
b) Fully serial FIR Filter below.

Verify your design using a comprehensive testbench. Use design a) as the reference design, and keep the top level the same for both.

```
entity fir_filter is
    port(
        clk    : in  std_logic;
        reset  : in  std_logic;
        -- Input signals
        samp_i : in  std_logic;
        data_i : in  std_logic_vector(15 downto 0);
        -- Output signals
        samp_o : out std_logic;
        data_o : out std_logic_vector(15 downto 0));
end entity fir_filter;
```

For each filter determine the following properties:

1. Number of CLB slices
2. Number of embedded multipliers
3. Minimum clock period after synthesis [ns]
4. Maximum clock frequency after synthesis [MHz]
5. Minimum clock period after implementation [ns]
6. Maximum clock frequency after implementation [MHz]
7. Minimum latency after implementation [ns]
8. Maximum throughput after implementation [multiplications/s]
9. Latency * area (minimum latency after implementation * number of CLB slices)
10. Throughput / area (maximum throughput after implementation / number of CLB slices).

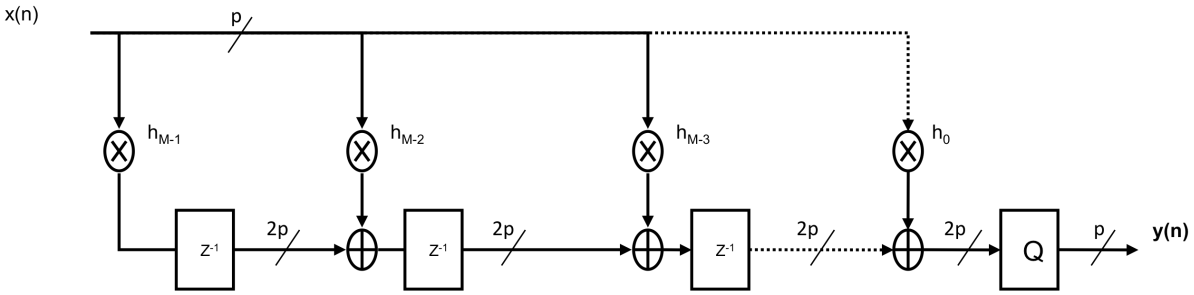

Figure 1 Parallel FIR

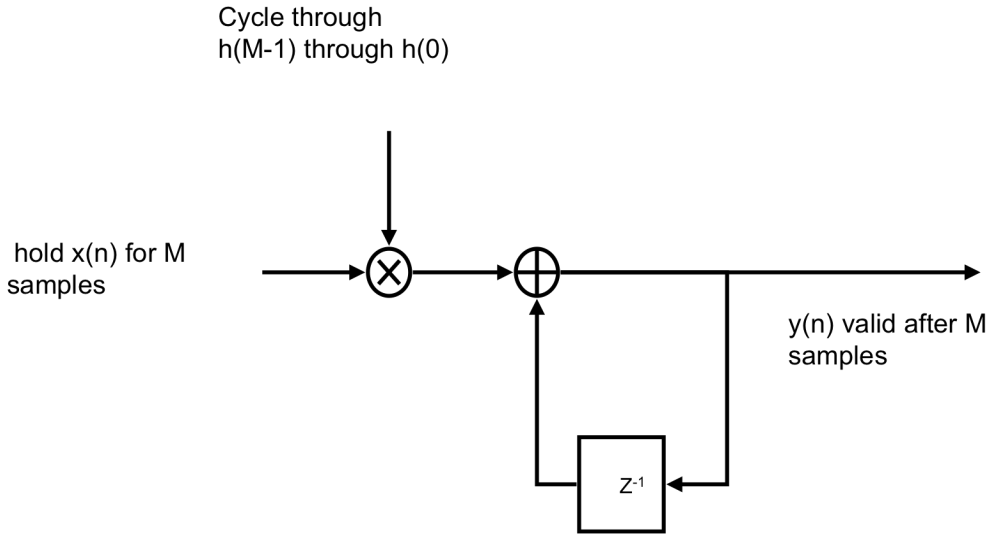

Figure 2 Serial FIR

Part 2 (2 points)

Implement two different kinds of Finite Impulse Response (FIR) filters:

a) Fully parallel FIR filter above. Use an Inferred 18x18 bit multiplier.
b) Fully serial FIR Filter above. Use an inferred 18x18 bit multiplier and inferred block ram.

For all inference rules, reference the following document:

http://www.xilinx.com/itp/xilinx92/books/docs/xst/xst.pdf

Use the same testbench in Part 1 to verify designs.

As a part of the design process:

1. Draw an optimized block diagram of the serial FIR using medium scale logic components, such as multiplexers, one-bit shifters, etc .
2. Translate your block diagram into behavioral VHDL using structural only for the multipliers and the block rams (for the serial FIR)
3. Determine filter taps based on a cut-off frequency of $\pi/4$
   (the equation is $2 \cdot \sin(\pi \cdot n/2) / (\pi \cdot n)$)
4. Write a separate testbench capable of verifying the functionality of both of your filters. The testbench should input an impulse into the filter.
5. Synthesize and implement your design.

In the lab deliverables include:

1. Optimized block diagram of the serial FIR (hand-drawn hardcopy submitted in class and used during an exit quiz, and an electronic copy submitted using Blackboard in the pdf or MS Word format (preferably, the scanned version of the hand-drawn hardcopy).
2. VHDL source codes for:
   a. Parallel FIR Filter
   b. Serial FIR Filter
3. Testbench and waveforms from the functional, post-synthesis, and timing simulation (electronic versions submitted using Blackboard) for both filters.
4. Timing and resource utilization parameters (listed in Part 1) for both filters.

Important Dates

	Monday section	Tuesday section	Wednesday section	Thursday section
Hands-on Session and Introduction to the Experiment	02/22/2010	02/23/2010	02/24/2010	02/25/2010
Demonstration and Deliverables Due	**03/01/2010**	**03/02/2010**	**03/03/2010**	**03/04/2010**