VLSI Lab 4

Nicky Advokaat - 0740567 - n.advokaat@student.tue.nl
Marcel Moreaux - 0499480 - m.l.moreaux@student.tue.nl

4$^{\text{rd}}$ quartile, 2014

Abstract

This report contains solutions for the problems described in Assignment L4 for the course VLSI Programming. We will create an upscaler filter.

Contents

1 Problem Specification and Requirements

2 Solution

In this section we describe the key ideas behind our design, and the decisions we made during the design process.

3 Results

4 Appendix A: Answers to inline questions

4.1 Question 1

$$y[n] = z[Mn]$$

$$z[n] = \sum_{0 \leq j < 4L} h[j] \cdot q[n-j]$$

$$q[n] = \begin{cases} x[n \text{ div } L] & \text{if } n \text{ mod } L = 0 \\ 0 & \text{otherwise} \end{cases}$$

$$y[n] = \sum_{0 \leq j < 4L} h[j] \cdot q[nM-j]$$

$$y[n] = \begin{cases} \sum_{0 \leq j < 4L} h[j] \cdot q[(nM-j) \text{ div } L] & \text{if } (nM-j) \text{ mod } L = 0 \\ \sum_{0 \leq j < 4L} h[j] \cdot 0 & \text{otherwise} \end{cases}$$

The *otherwise* case is always 0 so doesn't contribute to the sum. We can continue with just:

$$y[n] = \sum_{0 \leq j < 4L} h[j] \cdot q[(nM-j) \text{ div } L]) \qquad \text{if } (nM-j) \text{ mod } L = 0$$

$(nM - j) \text{ mod } L = 0$ happens at:

- $j = nM \text{ mod } L$
- $j = nM \text{ mod } L + L$
- $j = nM \text{ mod } L + 2L$
- $j = nM \text{ mod } L + 3L$

So four times in every summation. If we let j run from 0 to 4 (excl), we get $h[nM \text{ mod } L + jL]$.

$$y[n] = \sum_{0 \leq j < 4} h[nM \text{ mod } L + jL] \cdot q[(nM - j \cdot L) \text{ div } L]$$

$$y[n] = \sum_{0 \leq j < 4} h[nM \text{ mod } L + jL] \cdot q[(nM) \text{ div } L - j]$$

Which is the equation from the assignment.

4.2 Question 2

4.3 Question 3

5 Appendix B: Verilog source code

This appendix includes Verilog source code for the `filter.v` file in the ISE project.