Introduction to Electronic System Design (DAT093) Overview of lab series

Sven Knutsson, Lars Svensson

Draft of Version 1.1, August 10, 2018

1 Purpose

The overall purpose of the DAT093 lab series is *to prepare the students for modern system-driven digital design projects, including those in other courses of the EESD program*—a tall order for a half-time, seven-week course! From this overall purpose, we have derived several partial goals:

VHDL proficiency: Hardware Desciption Languages (HDLs) are the foundation of modern digital hardware design; a competent designer is expected to be familiar with both VHDL and VerilogHDL, in addition to other languages and formalisms. To reduce the scope, we focus on VHDL in this lab series; but just as for software languages, the insights gained will apply elsewhere also.

Task and tool knowledge: Effective and efficient HDL-based design requires familiarity not only with the language, but also with software tools and development hardware, and with the procedures and practices for utilizing all of them well. In the lab series, we will use the simulator QuestaSim from Mentor Graphics[1], the design environment Vivado from Xilinx[2], and the FPGA board Nexys 4 from Digilent[3], plus some peripheral add-ons. The software tools are heavily used also in industry; the particular FPGA board was developed mainly for the educational market, but is perfectly usable for smaller industrial designs.

[1] `www.mentor.com/products/fv/questa/`
[2] `www.xilinx.com/products/design-tools/vivado.html`
[3] `www.digilentinc.com`; search for "nexys 4"

Orientation in digital signal processing hardware: The typical embedded electronic system uses input signals from its environment to carry out its tasks. To be usable, these signals must often be filtered, reformatted, or otherwise manipulated. We use some hardware required for such manipulation as *design examples* in this lab series: again, the skills are useful in and of themselves, but also extend to more general designs.

2 Series overview

The lab series comprises seven labs, where the latter tasks build on their predecessors. Labs 0–3 will be carried out individually; the skills gained and demonstrated there will be used during the remainder of the series.

0. **Tool and task kickstarter:** The initial lab aims to familiarize the student with the tools; introduce basic VHDL concepts such as entities and architectures; and demonstrate simple test bench verification procedures. Simple adder and counter designs are used as examples.

1. **Generics and two's-complement arithmetic:** This lab introduces parameterized designs through generics; extends the adder example to cover subtraction; and shows one way to build a multiplication circuit.

2. **Behavioral VHDL and FIR filters:** The arithmetic blocks from the previous labs are used to build simple forms of filters which are ubiquitous in embedded systems. We introduce fractional numbers and explore how processing speed may be traded for hardware amount.

3. **Test bench design and design synthesis:** To be able to design your own blocks, you need to be able to design your own test benches as well. This lab also introduces synthesis of your design into an FPGA configuration, and shows how to compare synthesis results for different implementations.

Labs 4–6 will be carried out in groups of two students.

4. **Component handling:** A somewhat more complex system is built up hierarchically from simpler blocks which are separately verified. Overall verification is now not only done with simulation: the design is also synthesized and verified on FPGA boards in the course lab.

5. **A/D and D/A systems:** Non-trivial FPGA designs need to interface with external components. Using data converters as an example, we explore how

to extract the necessary information from data sheets and how to implement interface and control logic.

6. **A multirate system:** The final lab session will use the hierarcical, parameterized design style to realize a system that interfaces with two data converters and uses a filter when changing the signal sample rate.

3 General instructions

Some code and data files needed to complete the labs will be available for download from the course home page.

You may want to have access to a VHDL textbook to look up details of syntax etc. We do not specify a VHDL textbook for this course; however, several are available as e-books at the Chalmers library. In addition, these days, Google is very handy for detailed syntax questions.

As for other lab sessions in Chalmers courses, you will be expected to work independently to solve problems that may occur. During the scheduled sessions, teaching assistants will be available if you get stuck; but you will still need to prepare for each session. Each lab PM will contain directions for preparations. *Your lab TA will assume that you have carried out these preparations before coming to the lab.*

Each lab PM will contain many tasks and actions, itemized by bullets and gathered into sections. *Show your progress to your lab TA at the end of each section of the PM;* that makes it possible for him/her to keep track and to steer you in the right direction if necessary. Take advantage of any feedback the TAs give you. Be aware that the labs are *compulsory*; therefore, when you have worked through all tasks and are ready to leave, *make sure that your TA is aware and has checked you off the list.* Before you go, read through the wrap-up section of the PM one more time, and ask your TA about any issues that are still unclear.

Later labs build on the previous labs. Thus, to get the most out of the lab series, it is important not to fall behind—you will then be less well prepared for subsequent sessions. If you do not finish all tasks during a session, *do it as soon as you can.* Make a special effort to finish Labs 0–3 before the start of Lab 4.

This 2018 installment of the lab PMs are based on version 10.7a of QuestaSim and version 2018.2 of Vivado. None of the tasks depends on a certain software version; but if you use other versions, you may find that window panes, menus, libraries, and documentation are arranged somewhat differently.