# Experience from a New Course on Digital Logic and Computer Fundamentals at NTNU

Lasse Natvig
*Department of Computer and Information Science,*
*Norwegian University of Science and Technology (NTNU),*
*N-7491 Trondheim, NORWAY*
*lasse.natvig@idi.ntnu.no*

Tormod Njølstad
*Department of Physical Electronics,*
*Norwegian University of Science and Technology (NTNU),*
*N-7491 Trondheim, NORWAY*
*tormod.njolstad@fysel.ntnu.no*

## Abstract

*We have developed a compact new course on digital logic design and computer fundamentals, integrated with laboratory assignments using state-of-the-art design tools and a custom designed lab system board. The labs give the students hands-on experience with FPGA design using Xilinx Foundation and 8051 assembler programming. Through five assignments the students design, verify and demonstrate a simple audio processing system. Feedback from students tells that the lab makes the course fun and helps the understanding of the theory. In this paper we outline the objectives and contents of the course, give a brief description of the labs and summarise the experience from running this course for about 2000 students during the last four years.*

**Keywords:** Digital design, computer fundamentals, education, laboratory, FPGA, 8051.

# 1. Introduction

As part of an almost fulfilled process of restructuring and revising the entire set of courses at NTNU, it was decided to merge the courses *Digital Logic Design* and *Computer Fundamentals* and a related lab course into one large new course. More than just do a merge, we regarded this as a golden opportunity to start from scratch and develop a completely new course, by selecting the best available modern text books [1][2], and by developing an integrated set of five modern laboratory assignments [3].

The new course was implemented and first offered in 1998. We will here outline the content of the lectures, the accompanying theory exercises and the set of laboratory assignments. Finally, we will summarise our experiences learned from running this 450 to 600-students/year laboratory-intensive course, after four years of experience.

## 2. Course objectives

The overall course objectives are to provide (1) a general and basic foundation for understanding and designing digital logic systems ranging from simple combinatorial circuits to sequential designs including finite state machines (FSM) and datapaths; and (2) to give an introduction to instruction sets, computer architecture, and processor design methodology.

The goal of the laboratory assignments is to provide the students with hands-on experience with modern design tools, as well as to support and mature the comprehension of the theory.

## 3. Course content

The course is intended for undergraduate students in electrical engineering and computer science, who have completed introductory courses in high-level programming and in electrical circuit theory. The students are expected to spend about 12 hours per week, throughout the 14 weeks semester, with 3 hours per week of lectures and the remaining time consisting of laboratory and theory exercises.

The first half of the lecture sequence is covering digital logic design including: number systems, binary arithmetic, binary codes, Boolean algebra, Boolean functions, digital logic gates, simplification methods (the map and tabulation methods), combinatorial components, sequential logic, finite state machines, introduction to programmable logic, arithmetic circuits and storage components.

The second half covers an overview of the organisation and implementation of computers, hierarchical computer design, algorithmic state machines (ASM), processor architecture and instruction formats, addressing modes, datapath and control unit, microprogramming, RISC, CISC, I/O, buses, interrupt, memory hierarchy, cache, virtual memory, and a short overview of computer history.

The students have six theory exercises that are compulsory and follow the pace of the lectures. About 50% of each exercise must be solved properly to pass, and the works are corrected by teaching assistants. Six teaching assistants are helping the students in group rooms when they work with the exercises. After the deadline of each exercise, it is also given a lecture explaining its most tricky parts.

## 4. Laboratory Assignments

Through the 5 labs the students design, verify and demonstrate a simple audio processing system, implemented in the Xilinx FPGA and controlled by the 8051 μC on the NTNU board, see Figure 1 (system overview) and Figure 4 (picture).

The lab board is a simple stand-alone computer with a 16-key keyboard, a small LCD display, light emitting diodes (LEDs), 8 relays and optocouplers, and a 25 pin parallel port for communication with a host computer (PC). It also contains a stereo audio codec chip and external connections for headphones and microphones. Sitting on top of the lab board is a Xess Board from Xilinx. The Xess Board contains a Field Programmable Gate Array (FPGA) Xilinx XC4010 that implements “programmable hardware” by making it possible to download digital logic to the FPGA chip and execute it as ”hardware”.

The Xess Board contains an 8051 microcontroller that can be used for executing software that controls the downloading, execution and verification of the lab assignments. It also makes it possible to conduct simple experiments in HW/SW codesign.

Faculty and students at the two involved departments designed the lab board. 30 units were built during the summer 1998 at a cost of approximately 150.000 NOK including all components and salary for the students in the “production phase”. In 2000 we built an additional batch of 15 lab boards. Xilinx donated the Xess board and associated software licences through its XUP programme [5].

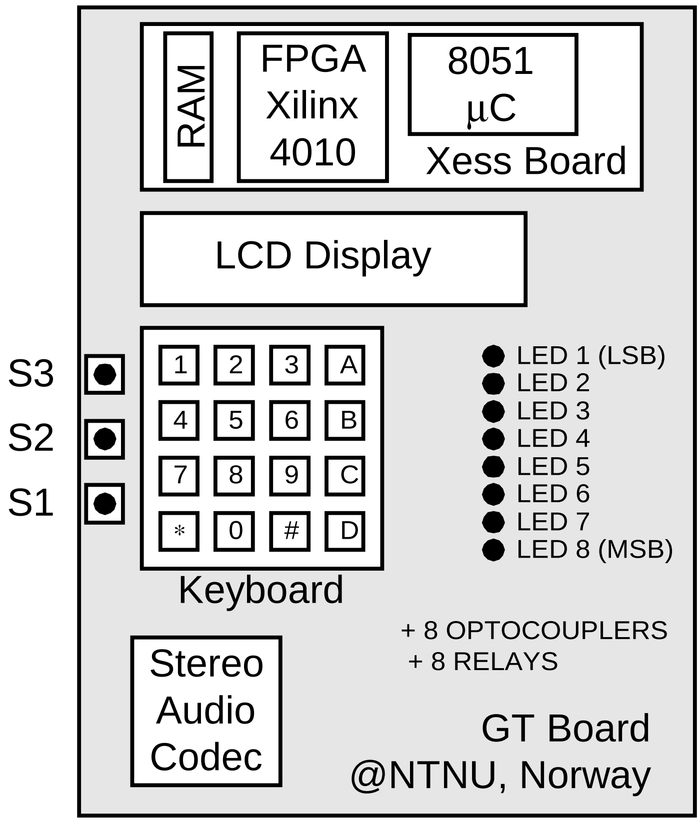


**Figure 1**. The lab system board, developed at NTNU, includes keyboard, display, LEDs, audio codec etc, and an Xess board with FPGA and 8051 µC.

The FPGA block diagram in Fig. 2 shows the testbench including the µC interface, and the students' modules: ABS, RC and MUL. The testbench is an environment of both HW (in the FPGA) and SW (running on 8051) that reduces the students work necessary for testing their designs, and also leads to an uniform main structure on the students’ design making supervision, testing and error correcting easier. To make it possible for the students to develop a simple real-time audio processing system through five lab assignments they are given different partial solutions at the various stages. The students must typically understand the partial solution, design and implement the missing parts, and integrate them into a complete solution tested in the testbench. The realistic complexity and functionality of the final result is motivating, and the development process is realistic.

In the same pace and perspective as the lectures, the labs are following a bottom-up verification scheme and can be summarised as follows:

- **Lab 1 (FPGA)** *Introduction*. Design entry, simulation and implementation using the Xilinx Foundation tools. Use of the NTNU board and our embedded testbench in the FPGA. The students implement a counter by using a predefined module in the Foundation tool. The counter is integrated with the testbench. The output from the counter is shown on the LEDs and on the LCD display.
- **Lab 2 (ABS)** *Combinatorial design*. The students design and test a module called ABS that gets a signed 16 bit number (2's complement) as input and converts it to a 16 bit value containing the absolute value of the input.

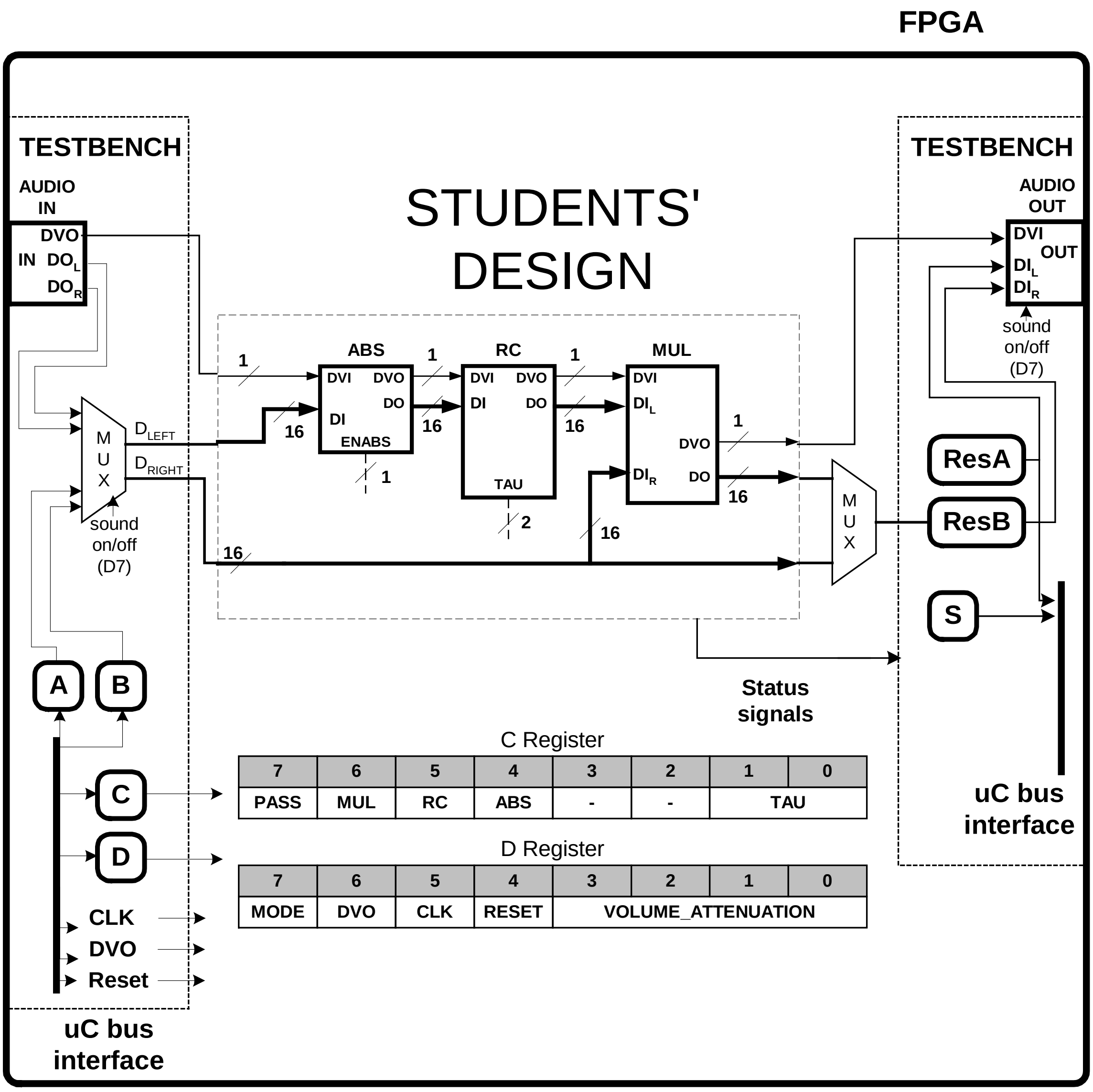


**Figure 2**: The FPGA design with students' modules ABS, RC and MUL, surrounded by our embedded testbench, controlled by the µC bus interface.

- **Lab 3 (RC)** *Sequential design*. The students design and test a programmable digital RC-filter. They start with a partial solution. They draw a finite state machine in the schematic editor, and integrate the completed design in the testbench. The RC module is an example of a simple datapath with control.

- **Lab 4 (MUL)** *Algorithmic state machines (ASM) and introduction to a hardware description language*. The students design a 16 x 16 bits 2's complement multiplier with schematic capture and use of VHDL. They are given an incomplete VHDL program that must be understood and completed.
- **Lab 5 (µC)** *Microcontroller and assembly programming*. The students develop a program for the 8051 microcontroller that reads values from the keyboard on the lab board both with polling and by using interrupt. They are also using the Keil PK51 C compiler from Keil Elektronik GmbH for studying the relationship between C code and assembler code. See Figure 3.

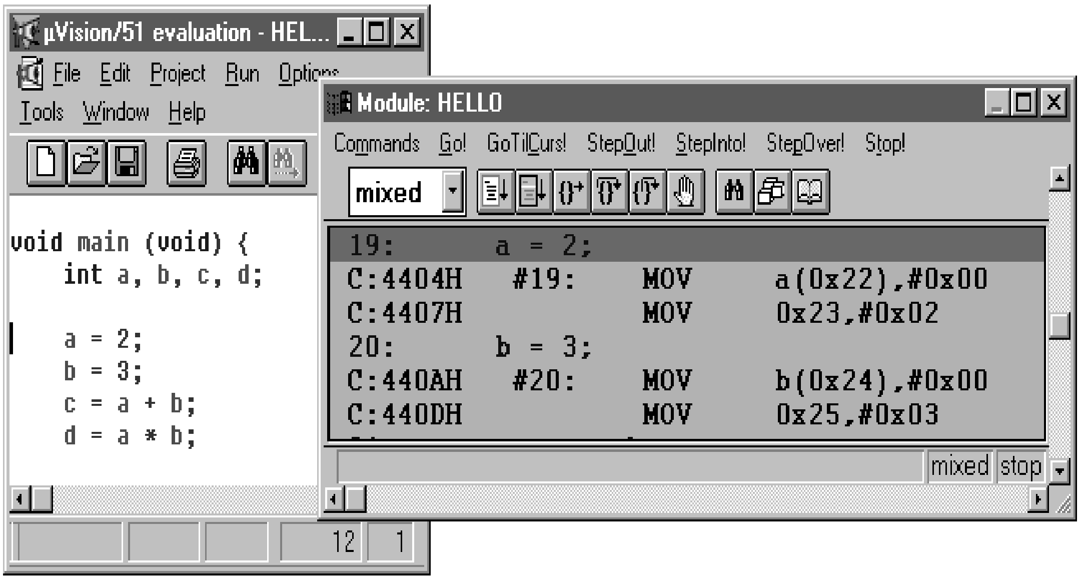


**Figure 3**. User interface of the Keil C compiler showing C code compiled to assembly code running on the 8051 microcontroller.

More details about the lab assignments can be found in the lab compendium [4]. The students typically prepare for the lab by reading the assignment text and doing preparations before they enter the lab. There they have to complete the lab assignment within four hours. This includes demonstrating the result and their understanding to a teaching assistant.

It is given a lab preparation lecture ahead of each lab and also lab preparation advice in group rooms. The lab boards (picture in Figure 4) are stored in the lab and are administrated by the teaching assistants. About 25 to 30 pairs of students are doing the lab simultaneously, with help of 2 – 3 teaching assistants. A new lab assign-ment every second week, and 5 or more 4-hour lab-batches during a week makes it possible for 600 students to complete the lab during the semester.

## 5. Experience

In a course with 450 to 600 students, there are challenges in addition to the normal pedagogical challenges. The logistics of leading such a large number of students through 6 compulsory theory exercises, 6 non-trivial lab assignments, and approval of their lab report at a moderate cost had not been possible without our small organisation of about 25 assistants. Most of these are teaching assistants recruited among older students, and they participate 120 hours each in the supervision and approval of the theory exercises and lab assignments. Doctoral or senior students administrate and supervise the teaching assistants, give lectures about theory and lab assignments, and maintain the 45 NTNU laboratory boards (Fig. 1), the FPGA designs (Fig. 2), the design tools, and related WEB pages. Complete and continuously updated information on WEB are extensively used by the students and the course staff [4]. This includes cgi scripts making possible semiautomatic registering of lab-partners (including individual choice of lab-schedule) in less than 2 working days.

Since the course is new and substitutes three older and different courses it is difficult to evaluate its effect on how much the students learn. However, NTNU is systematically collecting feedback from students. Each semester we get 300-400 feedback evaluation forms, and we are having two meetings with student representatives during the semester. Among the 600 students currently taking the course both previous knowledge, preparedness and motivation is very different. Most of them are studying electronics, telecommunication, computer science or industrial economics.

Our impression is that many of the students feel the laboratory assignments requires a lot of work, but that it is motivating to use modern powerful tools and to work through the semester towards a final goal – a simple audio processing system. It is certainly much simpler and cheaper to teach courses without such a lab, but we think hands-on-experience motivates students both for the reading in this course. Also, in the longer term, it might give interest for further studies in digital design, hardware, and combined HW/SW systems. Many of the students get excited and decide to choose later courses in the same direction.

Grading of the final examination within the time and cost limits given by NTNU is a challenge. The nature of the questions given has tremendous effect on the time used for evaluation. We have to some extent used multiple-choice questions in addition to the more standard questions where the student is asked to explain a central concept, and constructive problems where the student is asked to develop a small part of a digital circuit or computer system. In the future, we hope to assess the accuracy of the multiple-choice part of the examination by correlating it to the results from the other parts.

## Acknowledgments

We are thankful to Odd Tjore, Gunnar Tufte and many other people for their great contribution in developing the lab board and the accomplishment of the course as a whole. The generous donations from Xilinx through their University Programme have also been crucial in the implementation of the course.

## References

[1] Daniel D. Gajski: *Principles of Digital Design*, Prentice Hall 1997, Chapters 1-7.

[2] M. Morris Mano, Charles R. Kime*: Logic and Computer Design Fundamentals*. Prentice Hall 2000, Chapters 7-12.

[3] T. Njølstad, L. Natvig et al.: *SIE4005 Laboratorium. Digitalteknikk og datamaskiner.* NTNU, Norway, 2001 (140 pages, in Norwegian).
[4] The course web site (in Norwegian): http://www.fysel.ntnu.no/Courses/SIE4005
[5] XUP - Xilinx University Programme, http://www.xilinx.com/univ/index.htm

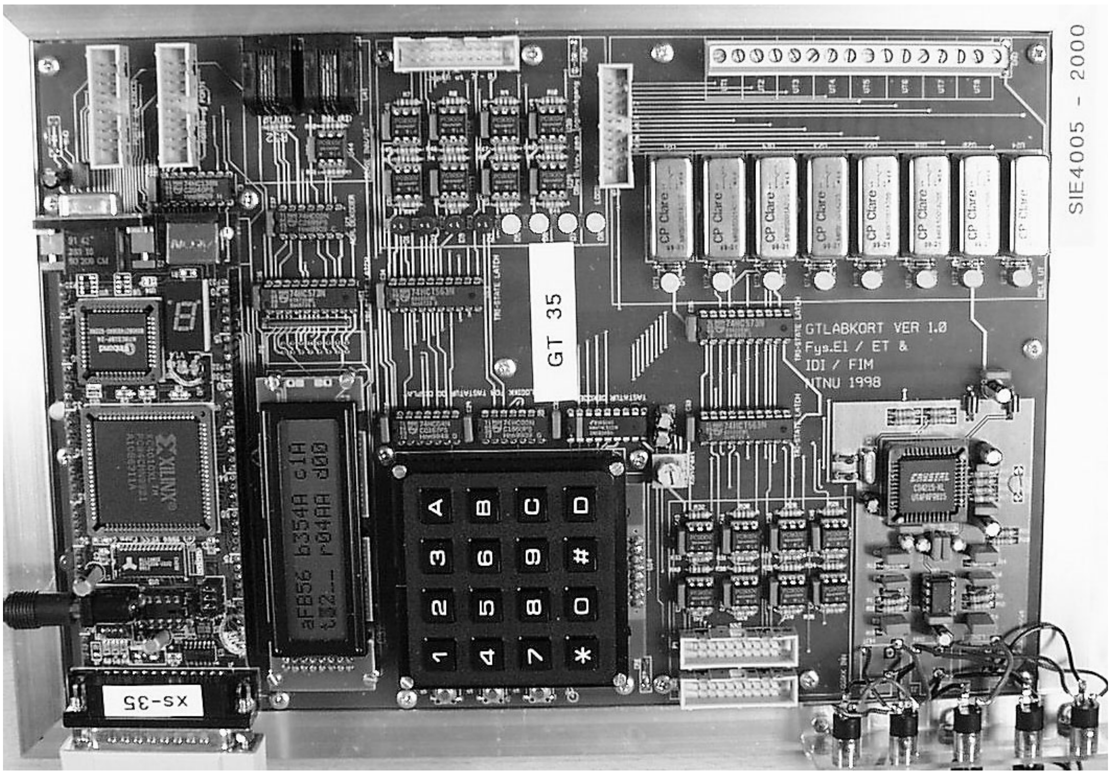


**Figure 4**. A picture of the lab board developed for use in the course *SIE4005 digital design and computer fundamentals* at NTNU.