

# Establishing a New Course in Reconfigurable Logic System Design

Jim Torresen, Jorgen Norendal and Kyrre Glette  
Department of Informatics, University of Oslo  
P.O. Box 1080 Blindern, N-0316 Oslo, Norway  
Phone: +47 2285 2454  
Fax: +47 2285 2401  
E-mail: jimtoer@ifi.uio.no  
Web: <http://www.ifi.uio.no/~jimtoer>

**Abstract**—Reconfigurable computing has grown to become important in hardware design. In autumn 2005, we taught for the first time a new course in digital system design with its main focus on FPGA technology and design using VHDL. This paper reports about the various issues dealt with including what topics to cover, text book selection, lab exercises etc. A summary of the students feedback is also included.

## I. INTRODUCTION

Digital system design is important in industry today. Our department has a course in traditional digital design but this only slightly touches into hardware design languages (HDL) and reconfigurable computing (RC). Thus, as a part of restructuring and renewing our courses, we decided to build a new course on the more advanced topics of digital systems design. There is not one single best way to organize such a course and this paper should be taken as one example of how it can be done. A course will always have room for improvement and comments are welcome.

The main steps for preparing and carrying through the course are summarized in Fig. 1. A number of issues were discussed and evaluated for the last year. These are described in the following sections.

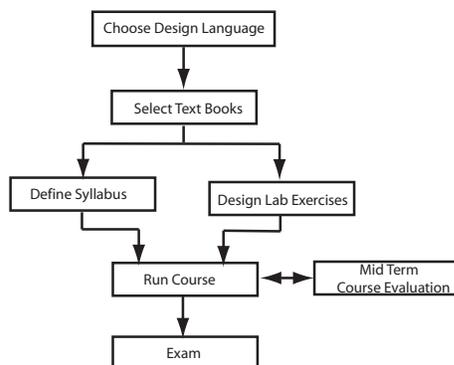


Fig. 1. The main tasks for carrying out our new course.

## II. DESIGN LANGUAGE

There are a number of languages to select from when designing digital systems. They can mostly be categorized into two families: Traditional low level hardware description

languages – including VHDL (Very High Speed Integrated Circuits (VHSIC) HDL) and Verilog, and high-level languages including SystemC and SystemVerilog. After discussing with several people in industry and reviewing the status of several languages, we concluded early that VHDL is the language of choice for RC designs. This was also beneficial for applying the design tools provided for free by the Field Programmable Gate Arrays (FPGA) vendor. We got confirmed from our investigations that in Europe Verilog is mainly used for ASIC (Application Specific Integrated Circuit) design and was for that reason not considered appropriate for our course that would be based on FPGA technology.

High level languages are improving and people designing hardware systems have got interest in evaluating them. However, they are so far mostly used for system *modelling* rather than system design and synthesis. Thus, we decided to include a small part in the course about the status of high-level languages but have students learn and use VHDL for the design work. It can be argued that at the university you should teach possible languages for the future rather than what is presently being used in industry. We felt that having lab exercises were an important part of learning which also would give the students an insight into advanced system-on-chip design. This would probably have required more effort (or been expensive) both for us and also for the students if selecting any other tool than the one provided by the FPGA vendor.

## III. TEXT BOOKS

A large number of text books were evaluated. We had a number of criteria used in the evaluation. Several books were not relevant due to lack of updates for the latest technology or coding standards. We early saw that we needed two books, one about VHDL and another about reconfigurable logic technology. A large number of books covering VHDL is available and many were evaluated. They either contain both digital logic basics in addition to VHDL or VHDL only. The former kind of books were not of interest in this course since digital design is taught in an earlier course. This is since VHDL is often in these books *integrated* with the digital design basics. Thus, it would not be very practical to extract the VHDL part.

Few books on FPGA and System-on-Chip (SoC) are available. Those about SoC are often focused at Application-Specific Integrated Circuit (ASIC) design. The main selection criteria relevant to reconfigurable logic were:

- Independence of a specific reconfigurable logic family and vendor.
- Include description of the new features in FPGAs (soft/hard cores, Intellectual Property (IP) cores, high speed I/O etc.).
- Cover difference and migration between FPGA and ASIC.

These issues made the selection fairly easy since only one book was found to satisfy these criteria [1]. The book – written by C. Maxfield, was new and quite updated on the latest technology. Still it was emphasizing on presenting alternatives from different vendors or often having the presentation vendor-independent. For selection of the VHDL book, it was important to find a book which included testbenches to be able to thoroughly teach design testing (see Section V). The best book found was one by M. Zwolinski [2]. To make the selection of the two books, we also made a number other and related considerations:

- Does a text book need to contain a CD with design tools (as quite a number of books do)? We realised that with (the same) design tools available for free on the web – see Section V-A, this was not important. Further, the software on CD would sooner or later become outdated.
- How much teacher material is available to reduce work for lecture preparation? Little was available for [2]. For [1], all the figures in the book were available electronically.
- Read the *Table of contents* and *Preface* of relevant books. This gave a good overview of the content of a book to see if it satisfied our requirements regarding content.
- Are the books applied in courses elsewhere? Both books were published in 2004 and when we planned the course early in 2005, we found few other courses based on these books.
- Quality of layout and writing. Several issues were to be considered:
  - What edition is the book? First edition may include more errors than a second or later edition. [2] is second edition while [1] is first edition. Thus, a few errors were found in [1] during the semester.
  - Are there any review of the books available? We found few official reviews for these kind of books, something we regret.
  - How well is the book organised? Is the layout inspiring for the reader or too “glossy”?
- How expensive are the books? With students buying the books for their own money, the price should be kept in mind. We have experienced that it is possible to bargain the selling price at the university book shop by referring to the prices on the web (e.g. Amazon).

During book evaluation, we searched the web to see what was used in other universities for similar courses. Some used

their own text while other used older text books that did not seem updated on the latest technology (e.g. did not include SoC design). Thus, at that time we were unable to find any academic course that we felt could be an appropriate example for us. Some universities keep their course material unavailable for Internet. Thus, there may have been updated courses that we did not have access to.

Below, we would like to give a short “review” of the two books based on our experience in using them for our course.

#### A. Review of “*The Design Warrior’s Guide to FPGAs*” by C. Maxfield

This book was selected since it covered an impressive range of relevant topics (including FPGA architectures and programming, embedded processors, high-speed transceivers, verification, FPGA versus ASIC, high-level design languages etc). The possible weakness may then be the shallowness. However, the book is mostly self-explanatory and thus, lack of details are not often a problem for the understanding.

The style of writing is very informal but mostly this makes reading more fun and does not reduce the impression of quality much. We feel the *organization* of the book could be improved as seen from the lecture plan in the Section IV. We often deviated from the chapter structure of the book. E.g. we found it more pedagogical teaching the architecture of an FPGA before going into the details of the programming technologies (EPROM, SRAM, Anti-fuse). Some topics are covered in several chapters. Thus, we often covered several chapters for each lecture topic. The book contains 26 chapters and it would probably have helped on the organization with a smaller number of main chapters and having more subsections within each chapter. To help students learn the text, some problems in the end of each chapter could have been helpful too.

#### B. Review of “*Digital System Design with VHDL*” by M. Zwolinski

We wanted a book which covered the RTL (Register Transfer Language) level of VHDL and a methodology of writing testbenches. Zwolinski’s book includes design topics that one expects a book on this level to contain but in addition it covers writing of testbenches in a very satisfying way. The reader gets an introduction to testbenches early in the book and each design topic is followed up with a companion test bench. Zwolinski also gives the reader an introduction to behavioural synthesis through a very instructive example of a digital filter design. There are numerous of examples throughout the book, and most of them can be downloaded from internet and simulated. Many of the examples can also be synthesised and implemented on our lab board. The experience from the first semester is positive. The index is probably the main weakness of the book. In the next semester we will include a quick reference to the VHDL language in addition to the book.

## IV. SYLLABUS

We early found out that it was no problem filling the course with relevant topics. Thus, the task was more to select what we

regarded as most important leaving out less important topics. As mentioned in Section III, there were few other courses we could compare with regarding advanced reconfigurable technology and design. We ended up with the lecture plan seen in Table I. Each row in the table represents 2x45 minutes of lectures. This is the second course in digital design for our students. Thus, knowledge about digital design using gates is expected.

Topics from the two text books were taught in parallel. Further, the lectures are to a large degree planned to make the students learn the necessary theory in advance of the lab exercises. The schedule of the four lab exercises are shown in the rightmost column in the table. See Section V for more details about the exercises. The first lectures are focused on the basics needed for getting started with practical work. The teaching of the design process focussed on good design practice too. This included Algorithmic State Machine (ASM) diagrams and testbench design.

As seen from the lecture plan, we included a number of new features becoming available including Intellectual Property (IP) cores, high speed I/O and run-time reconfigurable computing. These will probably expand their applicability in the future. The course also tried to teach the students to be able to make technological choices. This is relevant for e.g. IP cores; there are both benefits and drawbacks of applying available IP cores. It can substantially speed up the design work but it may be expensive, and quality of documentation is important if modifications are needed. Another issue is migration and selecting between ASIC and FPGA which involves a number of critical considerations.

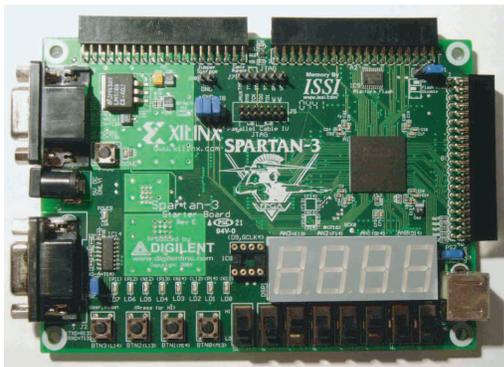


Fig. 2. The Spartan-3 Starter Kit board used in the lab exercises.

## V. LAB EXERCISES

All earlier experience told us that lab exercises were required in the course. It was also clear that they ought to be compulsory since they would be important to get hands-on experience from design. One of the first issues to decide was what FPGA platform to apply. Since Xilinx is the main vendor at the moment, we wanted a system with a Xilinx FPGA. First, we evaluated several *advanced* boards since we wanted to include an exercise with processor-on-chip design. However, after also looking at smaller boards, we found the Spartan-3 Starter Kit board – see Fig. 2, very interesting due to the following features:

- The board allows for design with a MicroBlaze softcore processor.
- It has low price (\$99) and therefore results in a low investment for us (some boards could be obtained as a donation as well) and those students interested could buy a board for their personal use.
- A number of plug-on-boards are available at low cost from Digilent.
- The board has been sold in high volume so it should be well tested.

We ordered a sufficient number of these boards, to have students work in pairs in the lab. Software consisting of Xilinx ISE and ModelSim were used for all the exercises. A last exercise on System-on-Chip design also included Xilinx Embedded Development Kit (EDK).

It was hard to schedule the exercises for several reasons. The deadlines should preferably not be the same as for courses many of the students took in parallel. Further, the first lab exercise could not start before a certain amount of teaching had been undertaken. The last would have to be finished before the semester ended to meet the deadline for reporting.

We wanted the students to get experience with both combinational and sequential design in FPGAs. This also involved inclusion of pre-designed modules. Further, we wanted them to experience how a processor core can be incorporation into an FPGA design. We spent some time to look for existing lab exercises but found few that also included on-chip processor. Thus, we ended up with designing our own exercises. With a maximum of four exercises during the semester – due to the time available and how much lab work we can require from the students, we ended up with the following exercises:

- 1) Design flow tutorial. The tutorial introduces the use of the design tool Xilinx ISE and simulation with Modelsim. Further, a task consisting of making a simple modification of an existing VHDL code and then designing a small decoder are to be undertaken.
- 2) Design of a 7-segment LED (Light Emitting Diode) controller (combinational logic design).
- 3) Electronic lock. Apply the 7-segment LED controller from exercise #2 together with a pre-designed keyboard controller to connect a standard PC keyboard to the 7-segment LED display – see Fig. 3. Add a state machine to realize a code lock for entering 5 numbers to be compared with a pre-stored number code.
- 4) System-on-Chip design. EDK tutorial for MicroBlaze (soft processor core) design. Design a custom MicroBlaze system with a keyboard interface (modify the design in exercise #3) and a VGA monitor interface (add a predefined IP core). Implement software – in C language, for a simple “computer game” based on keyboard input and monitor output.

For exercise #1 we designed our own compact tutorial, while for exercise #4 we applied an EDK tutorial from Xilinx. To be able to make advanced design, we had the system designed in one lab exercise reused (often slightly modified) in the next one. For exercise 1 to 3 we required that the students also

TABLE I  
TEACHING PLAN FOR THE COURSE. MAX REFERS TO [1] AND ZWO REFERS TO [2]

Week	Syllabus	Topic	Content	Lab Ex.
1	Max ch.3	Introduction and Technologies	Overview of PAL, (C)PLD, FPGA and ASIC	
1	Spartan 3 doc.	HW used in lab exercises	Presentation of Spartan 3 lab. board	
2	Max ch. 2, 4 and 5	Technologies cont.	FPGA architectures and programming techn.	
2	Own text + Max ch. 9	Design flow (intro to lab exercises)	Design flow for typical FPGA design	
3	Own text + Max ch. 9	Design flow cont.		
3	Zwo ch. 1, 2 and 3	VHDL intro	Intro to digital design and VHDL	
4	Zwo ch. 4 and 6	VHDL building blocks and testbenches	Comb. and seq. building blocks, testbenches	1
4	Max ch. 2, 4 and 5	Technologies cont.	Alternative FPGA architectures	1
5	Zwo ch. 5 and 7	State machines in VHDL	Sequential logic and testbenches	1
5	Own text	Optimizing FPGA design	Effective utilization of FPGA technology	1
6		No lectures		2
7		No lectures		2
8	Zwo ch. 5 and 7	State machines in VHDL	Complex sequential logic and testbenches	3
8	Max ch. 19	Simulation, synthesis and verification		3
9	Max ch. 13 and 17	Advanced building blocks	Intellectual Properties and processor cores	3
9	Own text + std. libr.	Design libraries	Functions and procedures, standard IEEE libr.	3
10	Own text	System-On-Chip	Intro to the SoC design tool EDK	4
10	Zwo ch. 9 + own text	Synthesis		4
11	Zwo ch. 8 + own text	Simulation methodology	Examples of advanced testbenches	4
11	Max ch. 11	High level design	System-C and other high level languages	4
12	Zwo ch. 10 and 11	Test	Testing and design for testability	
12	Max ch. 7 and 18	Design with ASIC and FPGA	Migration between FPGA and ASIC	
13	Max ch. 21 and 22	New FPGA features	High speed I/O, run-time reconfigurable comp.	
13		Summary and questions		

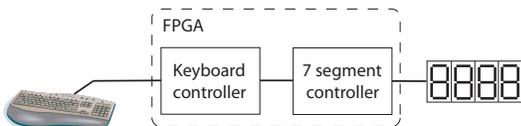


Fig. 3. Block diagram of lab exercise on connecting PC keyboard to a four 7-segment LED display.

implemented corresponding testbenches to test the behaviour of their design. Testbenches are written in VHDL and provide inputs to the design as well monitors the outputs from it. Running the testbench would then verify if the design would give correct response according to what has been specified in the test bench. This could often result in more effective verification than using a waveform generator for test stimuli generation.

Since the lab exercises were given for the first time, it was hard to think about all possible problems that may occur when the students solved the exercises. For most of the exercises, a person who had not been involved in preparing the exercises tried to solve them. This was important for exercise text improvement before the students were given the text. Further, we had teaching assistants in the lab at fixed times during the week.

#### A. Design Tools

We applied the VHDL simulator Modelsim version 6.0a from Mentor Graphics. Further, ISE v7.1i and EDK v7.1i from Xilinx were used. These are full versions of the tools and were obtained for free by donation from Xilinx through their university program. For the students it was possible to download free versions of both Modelsim and ISE from the Xilinx web page [3]. The free version of Modelsim allows to simulate 500 code lines at full speed. If the number of code lines extends 500, the simulation is going substantially slower. The free version of Xilinx ISE is named ISE Webpack 7.1i and contains most of the features of the full version. There has later been several upgrades of the software.

## VI. STUDENTS FEEDBACK

Our local student association organised anonymous questionnaires among the students at the course two times during the semester. The students reported that they found the course very motivating although some of the lab exercises were time consuming. They seem to learn more from the exercises than from the lectures. A possible improvement is to start the first exercise a week or two earlier or make the duration time for it shorter. This would allow for additional time for the last and more time consuming exercises.

Students were positive to quiz questions during the lectures. This was typical some short questions in the end of the lecture to summarize the main points in the lecture. This was also positive to increase the interaction with the students.

We were satisfied with their exam results. The exam (one exam in the end of the semester with no books allowed) was based on a combination of theory questions and a design task (coffee machine controller). All the students passed the exam and we believe the compulsory lab exercises have been important to make them have their own design experience early in the semester.

## VII. SUMMARY

A summary of our experience with establishing a new course has been reported in this paper. There were a number of topics to consider and much work for planning both lectures as well as lab exercises. After the course was held for the first time autumn 2005, student surveys indicated that we could be satisfied with our decisions. However, there will always be things to be improved when the course is given the next time.

## REFERENCES

- [1] C. Maxfield. *The Design Warrior's Guide to FPGAs*. Elsevier, 2004. ISBN: 0-7506-7604-3.
- [2] M. Zwolinski. *Digital System Design with VHDL. 2nd edition*. Prentice Hall, 2004. ISBN: 0-13-039985-X.
- [3] [http://www.xilinx.com/ise/logic\\_design\\_prod/webpack.htm](http://www.xilinx.com/ise/logic_design_prod/webpack.htm).