Proceedings of the 6th Nordic Signal Processing Symposium - NORSIG 2004
June 9 - 11, 2004
Espoo, Finland

INTEGRATING THE COURSES OF DIGITAL ELECTRONICS AND SIGNAL PROCESSING BY MEDIAN FILTERS

Janne Koljonen and Jarmo T. Alander

Department of Electrical Engineering and Industrial Management
University of Vaasa, PO Box 700, FI-65101 Vaasa, Finland
e-mail: *FirstName.LastName*@uwasa.fi

ABSTRACT

Rapidly developing technology enables engineers to create more and more complex methods and systems. This rapid development and complexity of systems is a great challenge to contemporary technical education. In this paper we will describe how we have tried to integrate a couple of engineering courses so that the courses can benefit from each other and the students get an idea how different engineering topics are interrelated. It is supposed that the integration of several courses helps students to cope with the complexity of modern engineering. The integrating element between the courses is the design, simulation and synthesis of a median filter, which is simple enough that it can be implemented as a student project work during an elementary digital electronics course while also being a very useful example in a basic signal processing course. In addition, it provides further challenges for more advanced courses, in our case the advanced digital electronics course.

Keywords: digital electronics, digital signal processing, engineering education, median filters, VHDL language

1. INTRODUCTION

Modern technology develops fast producing new engineering methods and systems in an increasing speed. This causes a major challenge to contemporary engineering education system. When and how to teach all important background, theory and practice to students within a very limited time while keeping the students' motivation high.

Our proposal is to loosely integrate several courses so that the students get some idea how different topics can be utilised and merged into modern engineering products. This paper describes how we have tried to combine the course of basic digital electronics, which gives an idea of how all digital systems function even at the lowest levels, and the basic course of digital signal processing, which tries to give a general overview of the basic theory and methods of modern signal processing. The integrative element between the courses is a joint project work to implement a median filter using first VHDL language modeling and simulation [IEE87] and later an FPGA synthesis on the advanced digital electronics course.

All our automation (TAU) courses are "standardised" as follows: 30h lectures + 20h exercises + 10h project work guidance by the teacher, while the student is expected to spend in addition about 60h to study the course. Each course is of 3cp (weeks) including the project work which is estimated to be 1cp. The intention of the project work is help the students to acquire some necessary practical engineering design and working skills in addition to the theory learned during lectures and exercises.

The project works are done in small groups (1-4 person). It's intended that the students carry out the project as independently as possible. They solve the problem, design the needed structure and units, program and test them and finally report the results. However, this could be too demanding if no guidance would be available. That's why we have two-hour guidance sessions approximately every third week, where the students represent their results so far and ask advice for the problems occurred. We found it useful to give subgoals for the next guidance session in order to keep all groups roughly on the same pace although it is not forbidden to move on quicker. The fastest groups cope with the project work with two or three guidance sessions, the average groups need five to six times.

2. MEDIAN FILTERS

To find an interesting but not too complex project for a primary digital electronics course is somewhat difficult. A median filter is quite ideal because its structure is clearly modular and it does not contain much arithmetics, in particular it does not require multiplication operations, which are quite difficult to implement at the gate level. For implementing a simple median filter we need a set of comparison and swap modules (fig. 3) and some control circuits to start with.

In practise we have subdivided the project into three main phases (fig. 1):

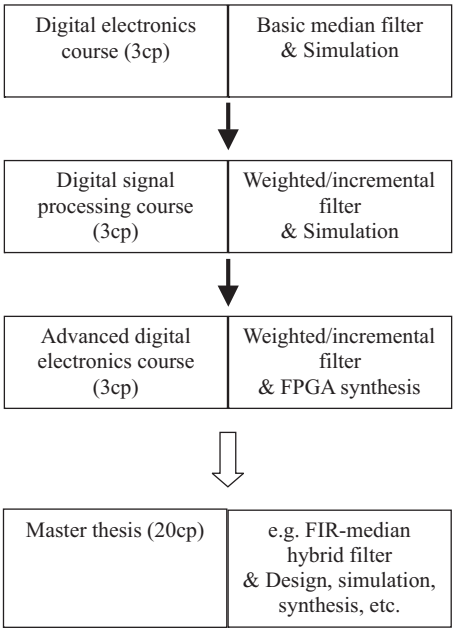

Fig. 1. Subprojects assigned to courses

- the basic median filter as a subproject for the digital electronics course,
- "advanced" median filter for the signal processing course, and
- the implementation of the above filters in hardware (FPGA) for the advanced digital electronics course.

In all phases the implementation is done using the well known VHDL digital circuit modeling and simulation language. In the final phase the advanced median filter is implemented as an FPGA (Altera) hardware.

By the advanced median filter we mean here weighted and incremental median filters. The weighted median filter is a straightforward extension of the basic median filter, which can be implemented *e.g.* by having also the weights routed in parallel to the signal samples and adding the weights until the weight mid point has been reached.

The incremental median filter is a bit more challenging. The idea of the incremental filter is to save processing time, in our hardware implementation case also hardware complexity, by just correcting the result of the previous output by taking into account the change in the window location. In a one dimensional case, only one new sample enters the window while another leaves it. The situation for higher dimensional median filters is not much more complex.

The incremental median filter is a nice example of how the implementation on hardware can provide new efficiency possibilities compared to the general purpose programming approach for signal processing. Namely, the hardware of FPGA gives us the opportunity to massive parallel processing. In the incremental median filter case we can compare the new sample against every other sample in the window in one step and thus find its ranking within the window (fig. 5). This is not possible when using an ordinary single processor CPU. The FPGA environment makes the incremental median filter attractive because of their spatially linear scaling by window size.

For more information on median filters see *e.g.* [YHAN91, YYGN96].

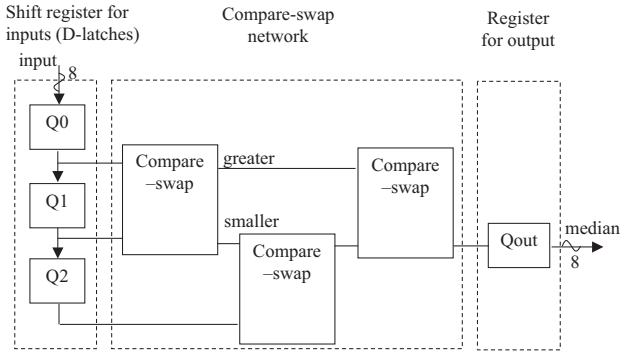

Fig. 2. Basic median filter for window size = 3 and word length = 8

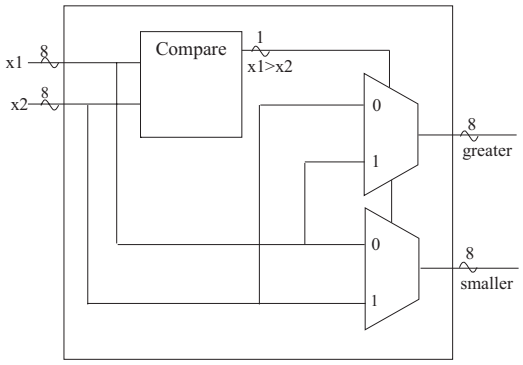

Fig. 3. Compare and swap unit

2.1. Examples

Here we will show some parts of the implementation as an example of the modules created.

Because median filter consists of separate modules, which in turn can be subdivided to even smaller units, it is easy to split the project in a student group so that every one can design, program and test separate units independently. Then the modules can be joint and tested together. In this way it is not always necessary to find common time, which can be a limiting factor in group-projects.

In figures 2 and 3 there is an example how median filter can be split into separate modules. The splitting process

```
library IEEE;
use IEEE.std_logic_1164.all;

entity compare_swap is
        port(x1,x2 : in std_logic_vector(7 downto 0);
        greater,smaller : out std_logic_vector(7 downto 0));
end compare_swap;

architecture behaverioral of compare_swap is

component compare_unit is
        port(num1,num2 : in std_logic_vector(7 downto 0);
        isgreater : out std_logic);
end component;

component MUX2_1 is
        port(a0 : in std_logic;
        num1,num2 : in std_logic_vector(7 downto 0);
        numout : out std_logic_vector(7 downto 0));
end component;

signal great : std_logic;

begin

comp: compare_unit port map(x1,x2,great);
mux1: MUX2_1 port map(great,x2,x1,greater);
mux2: MUX2_1 port map(great,x1,x2,smaller);

end behaverioral;
```

Fig. 4. The VHDL code for the compare and swap unit

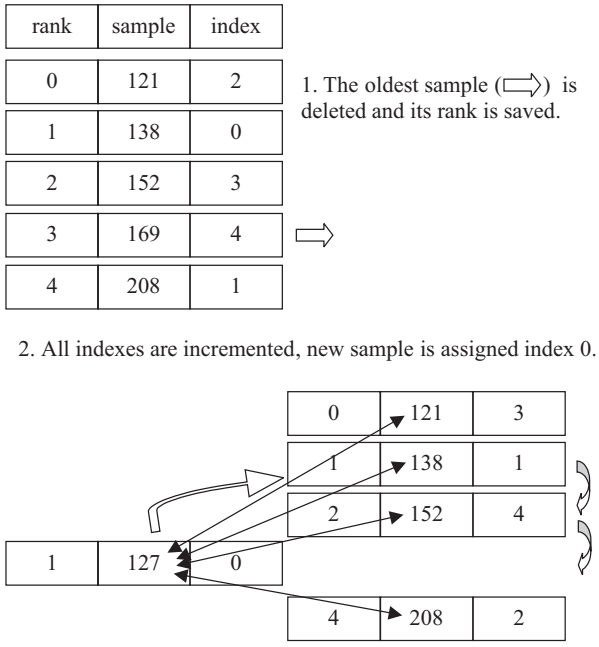

Fig. 5. The principle of incremental median filter

could be continued until all blocks are expressed at the gate level. We suggest that different parts of the median filter are programmed at different levels so that the possibilities of VHDL design become well illustrated. To make each project unique the design levels of the blocks can be permuted.

3. DISCUSSION

The motivation behind this small pedagogical experiment was the worry that the engineering studies are becoming more and more complex and thus easily less motivating to the new students. The pressure is high to bring more complex examples even to the basic courses. It was supposed by the authors that some kind of loose integration of the basic courses could bring some help with this respect. The small size of our department makes it quite easy to make this kind of experiments while the teachers are lecturing several quite loosely related courses making *e.g.* practical arrangements like timing, and coordination straightforward.

3.1. Benefits

The benefits of the proposed approach includes

- the integration of the otherwise not so much related courses,
- median filter has a set of natural parameters (weighted, incremental, window size, word length *etc.*), which makes it possible to assign each student project group a unique problem,
- concentrating to one problem type makes the guidance of the student groups easier (as compared to totally different projects); makes it also possible for the students to learn from other group's experiences,
- makes it possible for the students to do rather large projects as a set of smaller interrelated projects, and
- motivates teachers to develop new methods to motive also students.

3.2. Drawbacks

Correspondingly the drawbacks of the proposed approach include

- only one main topic, the median filter, may be somewhat limiting
- not all students are able to participate due to timetable and course order constraints
- the applications of median filters are somewhat limited and at their best with image processing, which is not so easy to demonstrate with the basic FPGA kit used.

4. CONCLUSIONS AND FUTURE

The proposed integration of courses by project works in common have several benefits that are supposed to aid the students to cope with the modern complex engineering environment. Based on the preliminary results of this project new similar integrative project works are planned for other signal processing and automation courses at our university. One natural course would be embedded system where the median filter could be implemented in a micro controller *e.g.* for sound processing experiments.

One way to proceed could be to use the median filters as object to signal processing software testing [MA01, MA03].

Median filters are quite ideal for basic course projects. Similar topics could be stack filters in general and morphological filters, which are both reasonable to implement as low level digital circuits.

Acknowledgements

The students of the TAU.103 Introduction to Digital Electronics, TAU.108 Digital Signal Processing, and TAU.303 Advanced Digital Electronics courses are warmfully acknowledged, without their enthusiasm this project would never have been fulfilled.

5. REFERENCES

[IEE87] IEEE. *IEEE VHDL Language Reference Manual*, 1987. IEEE 1076-1987.

[MA01] Timo Mantere and Jarmo T. Alander. Automatic test image generation by genetic algorithms for testing halftoning methods – comparing results using wavelet filtering. In Jarno Tanskanen and Jarno Martikainen, editors, *Proceedings of the 2001 Finnish Signal Processing Symposium, FINSIG'01*, pages 55–58, Espoo (Finland), 5. June 2001. Helsinki University of Terchnology, Faculty of Electrical Engineering.

[MA03] Timo Mantere and Jarmo T. Alander. Testing digital halftoning filters by generating test images and filters coevolutionarily. In David P. Casasent and Ernest L. Hall, editors, *Intelligent Robots and Computer Vision XXI: Algorithms, Techniques, and Active Vision*, volume SPIE-5267, pages 257–268, Providence, Rhode Island, 28. -29. October 2003. The International Society for Optical Engineering, Bellingham, WA.

[YHAN91] Olli Yli-Harja, Jaakko Astola, and Yrjö Neuvo. Analysis of the properties of median and weighted median filters using threshold logic and stack filter representation. *IEEE Transactions on Signal Processing*, 39(2):395–410, February 1991.

[YYGN96] Lin Yin, Ruikang Yang, Moncef Gabbouj, and Yrjö Neuvo. Weighted median filters: A tutorial. *IEEE Transactions on Circuits and Systems II, Analog and Digital Signal Processing*, 43(3):157–192, March 1996.