Course Analysis IL2203 HT 2022, 9 ECTS

The course is currently divided into the modules

- TEN1 Design and validation theory using VHDL as design language 3 ECTS
- SEM1 SystemVerilog language and exercises 1.5 ECTS
- LAB1 VHDL Design and Validation labs 3 ECTS
- LAB2 SystemVerilog Validation Labs 1.5 ECTS

Meeting with Student Representative

a) Lab instructions need to be improved.
b) Canvas web page organisation can be improved using modules – This will be investigated to next year.
c) Changes in Canvas pages should be announced at "anslag". – This will be investigated until next year.

Suggested improvements:

Weed out System Verilog as separate 3hp/6hp course. For Embedded Systems program this is irrelevant since both courses would be mandatory for Electronics and Platform students. For bachelor students taking IL1331, making SystemVerilog as a 3hp extension course would be good if they select to continue on the Master program.

An alternative way would be to add more VHDL material to IL2203 to make the VHDL content 7.5 ECTS. Then we could have both bachelor and master students study the same master course (IL2xxx 7.5 ECTS) instead of having two courses as we do at the moment with the same content (IL2203 VHDL-part 6 ECTS and IL1331 VHDL Design 7.5 ECTS).

These alternatives mean that SystemVerilog must be moved to its own course, but we don't have enough teachers or space in the program to give it as a separate course at the moment.

The other option is making all courses 6 ECTS in the Embedded Systems and Bachelor program. This would add one course to every 30 ECTS block (5 courses instead of 4), but this is not possible given the current program structures and available teacher resources.

Lecture Content

Had long discussions with PhD students on lecture content. We agreed that some simple exercises on basic concepts, like the meaning of a sensitivity list, how to infer clocks etc, would be useful. We can change the current Ö1-Ö3 into video tutorials, and do simple exercises instead, similar to the SystemVerilog exercises in the Seminars.

Also, splitting the construction of the Microcode ROM into a separate lab, teaching how to build small symbolic statemachines (add0, add1, add2, add3) etc to slowly build up all statemachines needed for the instructions in current lab 3.

Lab 3a) FSMs for the instructions

Lab 3b) Connecting the registers in the CPU

Lab 3c) Testing using given testbench

alternative design for lab3, focusing on RTL design

Lab 3a) Expand the description of the ALU into an RTL statemachine, which performs the instructions. Keep the RegFile as is.
Let the students draw the implementation of the RTL code and compare it with the Datapath in Lab2.

Lab 3b) Testing using given testbench.

Analysis of Moments

TEN1 – 46/51 = 90% Pass

SEM1 – 45/50 pass = 90% Pass

LAB1 – 14/42 = 33% ready in time before exam

LAB2 – 9/42 = 21% ready in time before exam