Practical 7 / *Prakties 7*

Hierdie opdrag moet gedemonstreer word vir die dosent of 'n demi voor die einde van jou groep se sessie, en 'n punt van 0 tot 5 sal toegeken word. U CONTROLLER_FSM.vhd leêr moet ook op SUNLearn gelaai word voor 17h00 14 Mei 2015.

This assignment must be demonstrated to the lecturer or a demi before the end of your group's session, and a mark of 0 to 5 will be awarded. Your CONTROLLER_FSM.vhd file must also be uploaded onto SUNLearn before 17h00 14 May 2015.

Komponente - *Components*

Elke student moet 'n DE0 bordjie uitteken en hou tot die einde van die semester. As die bordjie aan die einde van die semester nie teruggegee word in dieselfde toestand as waarin dit uitgeteken is nie, sal daar geen punte vir hierdie module vir die betrokke student toegeken word nie.

Werk op jou eie. Elke student moet sy/haar eie kode skryf en op SUNLearn oplaai.

Every student must sign out a DE0 board and keep it until the end of the semester. If the board is not returned at the end of the semester in the same condition as it was signed out, no marks will be given to the student who signed it out.

Work on your own. Each student must write his/her own code and upload it onto SUNLearn.

Om te Begin - *Getting Started*

Laai die projek templaat ("DE0_processor_basis.zip") van SUNLearn af en unzip dit. Maak die projek oop deur op die "DE0.qpf" leêr te dubbel-kliek. Laai die toets programme ("TestProgs.zip") van SUNLearn af en unzip dit. Die Verwerker het twee gedeeltes nodig om iets te kan doen: 'n Hardeware beskrywing (VHDL kode) en instruksies om uit te voer (masjien kode). Begin met 'n oefen-lopie:

1. **VHDL kode:** Laai die "processor_demo.jic" leêr wat binne in "DE0_processor_basis" is op die DE0 bordjie (Kyk die video op SUNLearn). Hierdie leêr bevat 'n hardeware beskrywing van 'n verwerker waarvan die CONTROLLER_FSM volledig is.
2. **Masjien kode:** Laai een van die toets

Download and unzip the project template ("DE0_processor_basis.zip") from SUNLearn. Open the project by double clicking on the "DE0.qpf" file. Download and unzip the test programs ("TestProgs.zip") from SUNLearn. The processor needs two parts to be able to do something: A hardware description (VHDL code) and instructions to execute (machine code). Start with a practice run:

1. ***VHDL code:*** *Upload the "processor_demo.jic" file that is inside the "DE0_processor_basis" folder onto the DE0 board (watch the video on SUNLearn). This file contains a hardware description of a processor with a completed CONTROLLER_FSM.*
2. ***Machine code:*** *Upload one of the test programs onto the flash mem-*

programme wat in "TestProgs" is op na die Flash geheue op die DE0 bordjie deur van die "DE0_ControlPanel.exe" gebruik te maak (kyk die video op SUNLearn). Die toets programme bevat instruksies wat deur die verwerker uitgevoer kan word.

Maak seker jy kan die verwerker aan die gang kan kry en dat dit die regte gedrag toon volgens die instruksies in die toets program (Kyk na die toets program se *.txt weergawe, of vra 'n demi as jy onseker is).

Die VHDL kode in "DE0_processor_basis.zip" is volledig, behalwe vir die eindige-toestandmasjien (die CONTROLLER_FSM komponent). Die hoofdoel van die taak vir prakties 7 is om die eindige-toestandmasjien in die bestaande argitektuur van CONTROLLER_FSM te implementeer. Daar is toetsprogramme in "TestProgs.zip" wat verskillende instruksies toets. Sooswat daar vordering gemaak word met die CONTROLLER_FSM se argitektuur, kan die gedrag van die toets programme vergelyk word met die volledige verwerker ("processor_demo.jic") om seker te maak dat joune reg werk.

LET OP: Die kloksein in "DE0_processor_basis" is gestel na 1 Hz sodat dit makliker is om te sien wat aangaan, terwyl die "processor_demo.jic" s'n na 20 Hz gestel is. LEDG(9) is verbind aan die klok, LEDG(8) is verbind aan INCR_INSTR_NUMBER_NE, LEDG(7 DOWNTO 0) is verbind aan die 8 bisse van die instruksie nommer. Die "processor_demo.jic" kan uitgeken word aan die vier desimale punte wat aan is op die 7-segment vertooneenhede.

ory of the DE0 board by using the "DE0_ControlPanel.exe" program (watch the video on SUNLearn). The test programs contain instructions that can be executed by the processor.

*Make sure that you can get the processor to work and that it functions correctly according to the instructions in the test program (Look at the *.txt version of the test program or ask a demi if you are unsure).*

The VHDL code in "DE0_processor_basis.zip" is complete, except for the finite state machine (the CONTROLLER_FSM component). The main goal of the practical 7 is to implement the finite state machine in the existing architecture of CONTROLLER_FSM. There are test programs in "TestProgs.zip" that test different instructions. As progress is made with the CONTROLLER_FSM architecture, the behaviour of the test programs on your processor can be compared to the completed processor ("processor_demo.jic") in order to check that your processor is working.

NOTE: *The clock signal in "DE0_processor_basis" is set to 1 Hz so that it is easier to see what is happening, while the clock signal for "processor_demo.jic" is set to 20 Hz. LEDG(9) is connected to the clock, LEDG(8) is connected to INCR_INSTR_NUMBER_NE, LEDG(7 DOWNTO 0) are connected to the 8-bits of the instruction number. The "processor_demo.jic" can be identified by the four decimal points on the 7-segment displays that are turned on.*

Opdragte - *Tasks*

1. Gebruik die inligting in die lesings oor die verwerker (vanaf Lesing 26) om 'n toestandtabel vir elke instruksie af te lei (Doen elke instruksie apart anders raak die tabel baie groot). Kyk spesifiek na die "Instruction Set" tabel, die "do nothing" waardes vir die beheerseine en die gedeeltes oor "FSM Design". Kyk ook na die kommentaar in die VHDL kode vir elke komponent.

2. Teken 'n volledige toestanddiagram wat al die toestande vir elke instruksie wys.

3. Gebruik die toestanddiagram en die toestandtabelle om VHDL kode te skryf wat die eindige-toestandmasjien implementeer in die bestaande argitektuur van die CONTROLLER_FSM komponent. Die eindige-toestandmasjien kan in dele geïmplementeer word:

   (a) Begin met die WR, D0 en D1 instruksies, "TestProg1.bin" toets net hierdie instruksies.

   (b) Voeg die CP, NO en JU instruksies by, "TestProg2.bin" toets net die WR, CP, D0, D1, NO en JU instruksies.

   (c) Voeg die AD instruksie by, "TestProg3.bin" toets net die WR, D0, D1, NO, JU en AD instruksies.

   (d) Voeg die EQ en CJ instruksies by, "TestProg4.bin" toets net die WR, D0, D1, NO, CJ en EQ instruksies.

4. Die "DemoProg.bin" program vertoon 4 getalle op 'n keer van 'n studente nommer, 15407756, en dan 'n teller wat geïnkrementeer word elke keer wat die studente nommer gewys word. Gebruik die Hexedit program (of enige ander program wat 'n binêre leêr kan verander) en verander die "DemoProg.bin" program om jou

1. *Use the information in the lectures about the processor (starting from lecture 26) to derive state tables for each instruction (Do each instruction separately, othewise the table gets very big). Specifically look at the "Instruction Set" table, the "do nothing" control signal values and the parts on "FSM Design". Also look at the comments in the VHDL code of each component.*

2. *Draw a complete state diagram showing all the states for all the instructions.*

3. *Use the state diagram and state tables to write VHDL code that implements the finite state machine in the existing architecture of the CONTROLLER_FSM component. The finite state machine can be imlemented in parts:*

   (a) *Start with the WR, D0 and D1 instructions, "TestProg1.bin" tests only these instructions.*

   (b) *Add the CP, NO and JU instructions, "TestProg2.bin" tests only the WR, CP, D0, D1, NO and JU instructions.*

   (c) *Add the AD instruction, "TestProg3.bin" tests only the WR, D0, D1, NO, JU and AD instructions.*

   (d) *Add the EQ and CJ instructions, "TestProg4.bin" tests only the WR, D0, D1, NO, CJ and EQ instructions.*

4. *The "DemoProg.bin" program displays 4 digits at a time of a student number, 15407756, and then a counter that is incremented each time the student number is displayed. Use the Hexedit program (or any other program that can edit binary files) to edit the "DemoProg.bin" file so that it will display your student number. Tip: Compare the first couple of instructions of the*

studente nommer te wys. Tip: Vergelyk die eerste paar instruksies van die program met wat vertoon word op die 7-segment vertooneenhede.

5. Laai jou verwerker se hardeware beskrywing op die DE0 bordjie (genereer 'n .jic leêr en gebruik dit), laai jou "Demo-Prog.bin" in die Flash geheue wat jou studente nommer wys, en demontreer vir 'n demi.

Onthou om jou kode (slegs die "CONTROLLER_FSM.vhd" leêr) op SUNLearn op te laai voor 17h00 14 Mei 2015!

program with what is displayed on the 7-segment display units.

5. *Upload your hardware description of the processor onto the DE0 board (generate a .jic and use it), upload your "Demo-Prog.bin" onto the flash memory that will display your student number, and demonstrate to a demi.*

Remember to upload your code (only the "CONTROLLER_FSM.vhd" file) to SUNLearn before 17h00 14 May 2015!