CS&A: Lab Sessions

Project: FSM

Ruben Van den Bossche

1BA INF - 2010-2011

1 Time Schedule

Projects are solved in pairs of two students. Projects build on each other, to converge into a unified whole at the end of the semester. During the semester, you will be evaluated three times. At these evaluation moments, you will present your solution of the past projects by giving a demo and answering some questions. You will immediately receive feedback, which you can use to improve your solution for the following evaluations.

For every project, you submit a small report of the project you made by filling in `verslag.html` completely. A report typically consists of 1000 words and a number of drawings/screenshots. Put all your files in a tgz archive, as explained on the course's website, and submit your report to the exercises on Blackboard.

- Report deadline: **November, 28 2010, 23u55**
- Evaluation and feedback: **November, 30 2010**

2 Project

Read sections C.7, C.8 and C.10 of Appendix C.

1. Build an **8-bit register** using D flip-flops. Inputs are 8-bit D (that denotes the input data) and C (the clock signal). 8-bit Q (that denotes the output data) is the only Output.

2. Build a **counter** using your own 8-bit carry lookahead adder and 8-bit register. Inputs are C (the clock) and D (an 8-bit number to which the counter counts), the output is the current 8-bit value of the register. At every clock tick, the counter adds 1 to the number in the register. When the register value is greater than D, the value is reset to zero. You can use the Logisim *Comparator*.

3. Build a **clock divider**. A clock divider is used to create a slow "daughter" clock from a faster "parent" clock. Inputs are C (the clock) and N (8-bit number). The clock divider generates an output clock signal with a frequency that is N times lower than the input clock signal. Use components of your own as much as possible.

4. Build a **finite-state machine** that implements a traffic light system on a cross section. Finite-state machines use memory and a clock. Since finite-state machines are *synchronous*, a new state is computed every clock cycle. A 2 Hz clock has a full clock cycle of 1 second. Use your counter and clock divider to advance through the states and make sure your state transitions happen at the right time. The two traffic lights behave like the following figure: