CS&A: Lab Sessions

Project: Adders

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, 7 2010, 23u55**
- Evaluation and feedback: **November, 9 2010**

2 Project

1. Build a 1-bit full adder (with carry in and carry out).
   (a) Determine the inputs and outputs of a 1-bit full adder and build a truth table.
   (b) Convert the truth table to Boolean algebra, and optimize the Boolean expression.
   (c) Implement the Boolean expression as a circuit called "1-Bit Adder" in Logisim.
2. Build a circuit of an 8-bit adder.
   (a) Use 1-bit adders to create an 8-bit adder, that adds two 8-bit wide inputs.
3. Build a circuit of an 8-bit two's complement adder.
   (a) Think about a way how overflow can be determined from carry outs. Overflow happens for example in these cases: 127 + 1 = -128 or -128 + (-1) = 127.
   (b) Build a circuit of an 8-bit two's complement adder that has an extra output bit, denoting overflow.

4. Build a circuit of an 8-bit two's complement carry lookahead adder using 2 4-bit adder blocks.

   (a) What are the "super propagates" and the "super generates", C1 and C2 values for the addition of numbers 11001011 and 01111100 (see Appendix C page C-44)? Calculate the carry out of the most significant bit (i.e. c_8).

   (b) Build a circuit for a 4-bit adder block. This block has input carryIn, a_0, a_1, a_2, a_3, b_0, b_1, b_2, b_3 and outputs s_0, s_1, s_2, s_3, P_0, G_0. Note that there is no output for carryOut, as a carry lookahead adder doesn't use c_{i-1}.

   (c) Build a circuit of the 8-bit two's complement carry lookahead adder by creating a "carry lookahead unit" that uses 2 of your own 4-bit adder blocks.

   (d) On this 8-bit adder circuit, create an extra output bit, denoting overflow.

   (e) To compare the carry lookahead 8-bit adder and the ripple carry 8-bit adders, count the maximum number of gate delays, i.e. the maximum number of AND and OR gates a signal passes in both adders.