Computer Systems and -architecture

Project 2

1 Ba INF 2012-2013

Bart Meyers
bart.meyers@ua.ac.be

Don't hesitate to contact the teaching assistants of this course. You can reach them in room M.G.3.17 or by e-mail.

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 500 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, 4 2012, 23u55**
- Evaluation and feedback: **November, 6 2012**

Project

1. Build a decoder for a 2-bit input.
   (a) Only use AND-, OR-, and NOT-ports.
2. Build a multiplexer for four 2-bit inputs.
   (a) Only use AND-, OR-, and NOT-ports.
   (b) What will be the size of your "select" input.
   (c) Use your decoder of the previous exercise.
   (d) In order to assess the performance of your multiplexer, find its *latency* by calculating the maximal path. This is the total number of gates (do not count NOT-gates, their latency time can be ignored) that are maximally passed.
3. 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.

4. Build a circuit of an 12-bit two's complement adder.

   (a) Use 1-bit adders to create an 12-bit adder, that adds two 12-bit wide inputs.

   (b) Think about a way how overflow can be determined from carry outs. Overflow happens for example in these cases: 2047 + 1 = -2048 or -2048 + (-1) = 2047. Build a circuit of an 12-bit two's complement adder that has an output bit denoting overflow.

5. Build a circuit of an 12-bit two's complement carry lookahead adder using 3 4-bit adder blocks.

   (a) What are the "super propagates" and the "super generates", C1, C2 and C3 values for the addition of numbers 1100 1011 0011 and 0111 1100 0100 (see Appendix C page C-44)? Calculate the carry out of the most significant bit (i.e. c_12).

   (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 12-bit two's complement carry lookahead adder by creating a "carry lookahead unit" that uses 3 of your own 4-bit adder blocks.

   (d) On this 12-bit adder circuit, create an extra output bit, denoting overflow.

   (e) To compare the carry lookahead 12-bit adder and the ripple carry 12-bit adders, count the latency of both blocks.