Computer Systems and -architecture

Project 2: Adders

1 Ba INF 2022-2023

Brent van Bladel
brent.vanbladel@uantwerpen.be

Don't hesitate to contact the teaching assistant of this course. You can reach him in room M.G.305 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 one tgz archive, as explained on the course's website, and submit your report to the exercises on Blackboard.

- Report deadline: **October 28, 2022, 22u00**
- Evaluation and feedback: **November 8, 2022**

Project

Read sections B.3, B.5 and B.6 of Appendix B. You can only use the following Logisim libraries for this assignment: Base, Wiring, Gates, Input/Output.

1. Build a 1-bit full adder (with carry in and carry out).
2. Build a circuit of a 16-bit two's complement ripple carry adder.
   (a) Create a new circuit in the same logisim file where you created your 1-bit adder. You can do this by choosing from the menu 'Project' - 'Add Circuit'. You can then choose which circuit you want to edit by double clicking it in the library menu on the left side. You can use a self made circuit as a building block in another circuit, in the same way as you use other blocks/gates in your circuit. The interface of your block is determined by the input and output ports you created in its circuit.
   (b) Use 1-bit adders to create a 16-bit adder, that adds two 16-bit wide inputs.
   (c) What is the number of gates you used to build your 16-bit ripple carry adder?
   (d) What is the latency (number of AND/OR ports traversed) when calculating the sum using a 16-bit ripple carry adder?

(e) Think about a way how overflow can be determined from carry outs. Overflow happens for example in these cases: 32767 + 1 = -32768 or -32768 + (-1) = 32767. Add an output bit denoting overflow to the circuit containing your 16-bit two's complement adder.

3. Build a circuit of a 16-bit two's complement carry lookahead adder using four 4-bit adder blocks.

   (a) 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}.

   (b) Build a circuit of a 16-bit two's complement carry lookahead adder by creating a "carry lookahead unit" that uses four of your own 4-bit adder blocks.

   (c) What is the number of gates you used to build your 16-bit carry lookahead adder?

   (d) What is the latency (number of AND/OR ports traversed) when calculating the sum using a 16-bit carry lookahead adder?

   (e) On this 16-bit adder circuit, create an extra output bit, denoting overflow.

4. Verify that your 16-bit two's complement carry lookahead adder is correct.

   - Do this by connecting two 16-bit inputs to both the carry lookahead 16-bit adder and the ripple carry 16-bit adders, and comparing the outputs.
   - Compare the carry lookahead 16-bit adder and the ripple carry 16-bit adders by counting the latency of both adders.

5. To prepare for the next lab session, read sections B.5 and B.6 of Appendix B.