Computer Systems and -architecture

Project 2: Adders

1 Ba INF 2023-2024

Kasper Engelen
`kasper.engelen@uantwerpen.be`

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` or `zip` archive, as explained on the course's website, and submit your report to the exercises on Blackboard.

- Report deadline: **Monday November 6, 2023, 22u00**
- Evaluation and feedback: **Thursday November 16, 2023**

Project

Read sections C.3, C.5 and C.6 of Appendix C. 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).
   (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 a 12-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 12-bit adder, that adds two 12-bit wide inputs.

   (c) 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. Add an output bit denoting overflow to the circuit containing your 12-bit two's complement adder.

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

   (a) Build a circuit for a 4-bit adder block. This block has inputs 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) Optimise this 4-bit adder block to decrease the circuit latency (maximum number of ports traversed). Do this by expressing your block as a set of Boolean equations (for every output pin, in terms of input pins), and using Boolean algebra or a truth table to optimise these equations. What is always the maximum latency for combinatorial circuits? Build your optimised 4-bit adder block.

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

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

4. Verify that your 12-bit two's complement carry lookahead adder is correct.

   - Do this by connecting two 12-bit inputs to both the carry lookahead 12-bit adder and the ripple carry 12-bit adders, and comparing the outputs.
   - To compare the carry lookahead 12-bit adder and the ripple carry 12-bit adders, count the latency of both blocks.

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