# Computer Systems and -architecture

Project 7: Datapath in Use

*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: **December 20, 2022, 22u00**
- Evaluation and feedback: **December 23, 2022**

## Project

Read section 4.9 of Chapter 4. You can use all Logisim libraries for this assignment.

1. Exceptions are a very important part of a datapath and control. In this exercise, you will add a basic form of exception handling to your datapath: when an exception is detected, your program counter should halt at the instruction that caused the exception. Arithmetic overflow should be detected and supported. Invalid instructions should also result in an exception.

2. During the evaluation, you will have to demonstrate the proper operation of your datapath with **two** assembler programs. Try to use subroutines at least once (don't forget to initialize the stack pointer).

   Use the programs below as inspiration.

   (a) A program that calculates the Fibonacci numbers (`http://en.wikipedia.org/wiki/Fibonacci_numbers`) and stores them in memory. After which number does overflow occur?

(b) A program that finds the smallest element in an array of integers stored in memory. Store the smallest element back in memory.

(c) A program that calculates the greatest common divisor (using the Euclidean algorithm, but without recursion) of two integers read from memory. Store the result back in memory.

(d) A program that writes a 1 to a register of your choice if an array of integers in memory contains duplicates, and writes 0 if it doesn't.

(e) A program that finds an element in a sorted array (with given size). The index is written to a register of your choice, and -1 is written if it does not occur. Use the binary search algorithm (`https://en.wikipedia.org/wiki/Binary_search_algorithm`).

(f) A program that sorts an array of integers in memory. You can use a sort algorithm of your own choice. If you want a challenge, you can try to implement the quick sort algorithm.