TD1: Introduction to Logisim, counter by 4 and ALU

For this course, you will have to build a simple – but working – computer called the β-machine in the LOGISIM simulation environment. This first practical session will introduce you to the Logisim software. You will build and simulate your first circuits: a counter by 4 and a ALU (arithmetic–logic unit). These components will then be useful for your future β-machine.

1. First, install the original version of Logisim from eCampus (Software / Logisim). You might need to install Java in order for the software to run. Familiarize yourself with the software. You can find the `documentation on the website`. You may also want to watch the short tutorial podcast available on eCampus (Theoretical sessions / 99 - Logisim).

2. In Logisim, build a binary counter with the following properties:

   • The counter must hold a 32-bit value and count by steps of 4 at every clock cycle. You can use every building block available in Logisim you might need

   • Your circuit must include a single bit input signal to reset the counter, setting back its value to 0 when enabled, and a 32-bit output reflecting the current value of the counter

   • The counter must count up to the last 4 digits of your student number (e.g. up to 8765 for student ID 20198765), then come back to 0. If your student number is not a multiple of 4, you can round at the next multiple of 4 (8768 in our example).

   Simulate your circuit to check it works properly.

3. Build an ALU with the following properties:

   • It should have two 32-bit inputs A and B, a 32-bit output and a 4-bit input ALUFN to select the operation.

   • It should support the same operations as the ALU presented in the theoretical course.

   • You may use plexers, adders, shifters, comparator, and all other logic and arithmetic boxes available in Logisim with any bit width. It would basically look like on slide 4-27. On slide 4-27, only the comparison box is not a Logisim library object (but it is made of a very few Logisim library objects). Use a subcircuit (menu Project, add circuit) for this comparison box. Notice that the ALU operations are $<$, $\leq$ and $=$, unlike the comparison box ($<$,$=$,$>$).

   Simulate your circuit and make sure that every operation works properly and gives correct answers, e.g. with A=6 and B=3.

   Really test! In many cases, mistakes in the ALU are responsible for loosing points in the project.

Good luck and have fun building your first computer *from scratch* !

"If you wish to make an apple pie from scratch, you must first invent the universe."

— Carl Sagan, Cosmos