# Week 4 Lab Automation

## Objectives

Develop understanding and experience of:

1. How components we have built in previous weeks could be combined to automate running of instructions.
2. Little Man Computer as a model for understanding the fetch-execute cycle.

The third part of the lab will be used to look at the Logisim part of the coursework.

### Part 1 Automation

Using Logisim, we have started from basic logic gates and built a variety of components including registers, a counter, an ALU and addressable memory.

In this session, we connect components together to form a simple computer that automatically carries out a set of instructions. This version of the computer is very limited, but demonstrates how automation can be built from fairly simple components.

The process for creating the automation was covered in the lectorial session (with images) and there are written instructions below. The tutor will lead a brief recap of the components and guidance on creating the automation.

You may download the Logisim circuit file from the Week 4 lab files folder on Moodle as a starter. This contains a completed ALU from week 2, including the reinforcement exercise. The ALU has three inputs, two 8-bit data inputs to carry out the operation on and a 2-bit input to indicate which operation to do. The ALU has one 8-bit output, the result of the calculation (or logical operation).

The operations are as follows:

|  |  |  |
| --- | --- | --- |
| ALU OP value | Short format | Output |
| 0 | ADD | The result of adding A and B together |
| 1 | AND | The result of a bitwise A AND B |
| 2 | SUB | The result of subtracting 1 from A |
| 3 | LEFT | The result of shifting the bits in A left by one bit so that the rightmost bit will be filled with a zero. |

The principle of this task is to create a simple computer that will store in ROM a list of operations for the ALU to carry out automatically. This will need a counter that increments to move through the list of operations. The inputs to the ALU will come from its previous output and from a pin. You should use Logisim built-in components for the register, ROM and counter.

Building the computer:

1. Create a new circuit for this computer (in the same project as the complete ALU)
2. Add the following components to the computer
   * An ALU
   * A Register – with data bits set to 8
   * An input pin – with data bits set to 8
   * A clock
   * A ROM – with address bit width of 4 and data bit width of 2 – this will store the instructions
   * A counter – with data bits set to 4 – this will be the program counter to keep track of which instruction the device is on
3. Now we can start to connect the components together
   * The output from the counter needs to connect to the Address input of the ROM
   * The Data output needs to connect to the ALU operation (bottom) input of the ALU
   * The pin needs to connect to the second input of the ALU
   * The output of the ALU needs to connect to the register.
   * The output of the register will connect back to the top input of the ALU. This means the calculations will always build on the previous value. That register is called an accumulator.
   * The accumulator register will need a separate constant 1 input to its Enable input.
   * The clock signal needs to be connected to the two components that require it.
4. Now are ready to test the simulation.
   * The instructions will go in the ROM in the order that we want to do them. Remember that the ROM only stores 2 bits at each address, so each entry in the ROM has to be a number between 0 and 3 (for the 4 operations of the ALU).
   * There needs to be a starter value in the input pin. Note that the pin shows binary values (rather than hex).

We need to consider how to test that it is working correctly by carrying out some calculations by hand. We will load a program into the ROM and step through that program by clicking the clock input (with the hand icon highlighted).

I suggest the following situation. This is a different example to that given in the lectorial, but you should take the same approach of working out the expected results in a table. Use a conversion website to convert between decimal, binary and hex if necessary.

* Put the value of decimal 90 in the input pin (convert to binary to do that).
* Have the following instructions – use the table above for the numeric values
  + ADD
  + SUB
  + ADD
  + AND
  + LEFT
  + SUB
* Work out in binary what the result in the accumulator should be at each stage and then convert to hex so you can check against the value in the accumulator.

The tutor will give you some time to test your computer before leading a discussion into the testing and limitations of this computer. There are some extensions to this computer for the reinforcement exercises.

### Part 2 Little Man Computer

The tutor will give some background to the Little Man Computer as a model for the fetch execute cycle.

There are Little Man Computer simulations available online, <https://www.101computing.net/lmc-simulator/> (and click to launch simulator) and <http://www.peterhigginson.co.uk/LMC/>

Both simulators have some programs available as demos, including a version of the program to add two numbers that the tutor talked through. You should try the following exercises.

1. Run your chosen simulator with the add two numbers program. Compare it to the description given by the tutor. Compare the two simulators and the programs provided in pairs if possible.
2. Load the program to find the maximum of two numbers. Read the program using the LMC instruction set to try to work out how it works.
3. The instruction set does not have an operation for multiply. Work out a method for multiplying a number by itself (finding the square). Look at the sample programs to see how this was done. In this case the samples are slightly different, one multiplies two numbers, the other finds the square.

The tutor will lead a discussion into your findings.

### Part 3 Coursework support

This part of the lab is for coursework support. The tutor will look at some aspects of the coursework with the class. You may ask questions about the assignment. The tutor may decline to answer some questions and instead encourage you to think about your approach yourselves or refer you back to previous exercises on this unit.

Remember that it is an individual coursework. You may discuss your approach for the practical work with other students, but that must be done by discussion in words (spoken or written) only, not by actually seeing anyone else’s work that forms part of the assessment. You must not have any access to anyone’s answers to the written discussion questions in any format. You should refer back to any of the lab or reinforcement exercises which can be discussed and shared in full.