



2DT901 – Lab assignment 1: LEGv8

Goal for this lab:

- Learn to write simple programs in LEGv8 Assembly language, using an emulator.
- Manually translate short machine code programs from binary form to LEGv8 Assembly language.
- Gain some understanding of the different parts of a CPU and how these parts are used in a simple Assembly program.
- Learn to use the flag registers to implement selection and iteration in assembly code.

Presentation rules

- 1) You have to submit a report for assignment, with the LNU template.
- 2) You have to submit a file, either in *.docx* or *.pdf* format.
- 3) You are allowed to use this file to make your report, or you can use a new one. In the case you use a new file, make sure to refer to the exercises you are answering. In the case of a new file, you still have to use the LNU template.
- 4) The file you submit **must be renamed** as follows:
<2DT901_group_name_assignment1 >. Example:
2DT901_group2_assignment1.pdf.
- 5) Deadline is **24 April 2022**.
- 6) To pass the assignment, you must pass all the 7 tasks.

Tasks

Task 1:

Download the LEGv8 emulator from the following Github link:

<https://github.com/arm-university/Graphical-Micro-Architecture-Simulator>

Download the simulator as a .zip file and unzip it.

Open the file LEGv8_Simulator.html, located in the folder: LEGv8_Simulator\war\

Write the following code in the simulator and run it:

```
MOVZ    x0, #5
MOVZ    x1, #10
ADDI    x1, x1, #2
ADD     x2, x0, x1
```

What number is stored in register x2 after you run the program?

Task 2:

Translate the following machine code instructions to LEGv8 Assembly code:

110100101000000000001000000000010

110100101000000000001110011100100

110010110000001000000000010000101

D360 0CA5

Task 3:

Create a LEGv8 Assembly program to calculate the value of the following expression:

$$4 \cdot 5 + 16 \cdot 11 + 25$$

When finished, the result shall be stored in register x0.

Note: The Graphical-Micro-Architecture simulator for LEVv8 does not implement the instruction MUL for multiplication, so you have to do the multiplications in another way!

Task 4:

Write a LEGv8 Assembly program to calculate the sum $1\,893\,423 + 443\,924$. The numbers are decimal integers.

You will probably encounter a problem to load these large numbers into registers, so you will have to find a way to solve this problem!

Task 5:

Write a LEGv8 Assembly program to calculate the sum

$$1 + 3 + 5 + \dots + 99.$$

When finished, the sum shall be stored in register x1.

Task 6:

Run the following program in the LEGv8 simulator:

```
//Set up base memory address
MOVZ      x7, #0x1000, LSL #16

//Store the numbers 15 and 25 in data memory
MOVZ      x1, #15
STUR      x1, [x7, #0]
MOVZ      x1, #25
STUR      x1, [x7, #8]

//Load the stored from memory
LDUR      x1, [x7, #0]
LDUR      x2, [x7, #8]
```

Look at the parts of the simulated CPU in the simulator. Answer the following questions for each type of instructions:

1. Which parts of the CPU are involved when the instruction is executed?
2. Explain why these parts are involved.

Task 7:

```
//Set up base memory address
MOVZ      x7, #0x1000, LSL #16

//Store the numbers 1, 4, 1, 5, 9, 2 in dynamic memory
MOVZ      x1, #1
STUR      x1, [x7, #0]
MOVZ      x1, #4
STUR      x1, [x7, #8]
MOVZ      x1, #1
STUR      x1, [x7, #16]
MOVZ      x1, #5
STUR      x1, [x7, #24]
MOVZ      x1, #9
STUR      x1, [x7, #32]
MOVZ      x1, #2
STUR      x1, [x7, #40]
```

Write a loop to add all the numbers stored in memory. When finished, the result shall be stored in register x0.