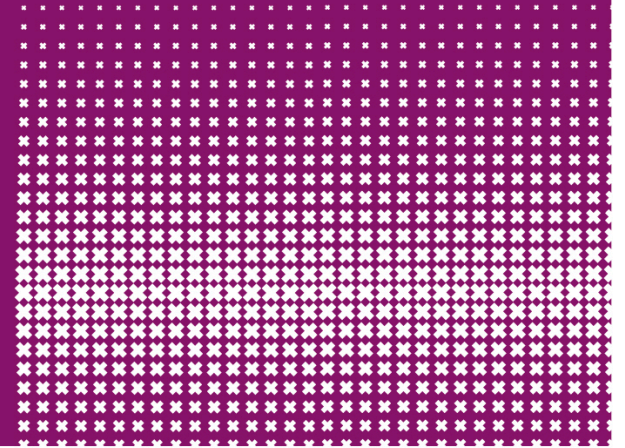




ing. Edwin Steffens

Practical lecturer – Informatics Institute - Faculty of Science



Computer Architecture

Lab 1 – Performance measurement

Computer Architecture course

Lab Experiments

Expectations

Introduction SIM-PL

Kickstart Lab 1

Lab Experiments

Expectations

- Labs are compulsory.
- Lab descriptions and quizzes are on Canvas.
- Lab-notes are handed in via Canvas before each deadline.

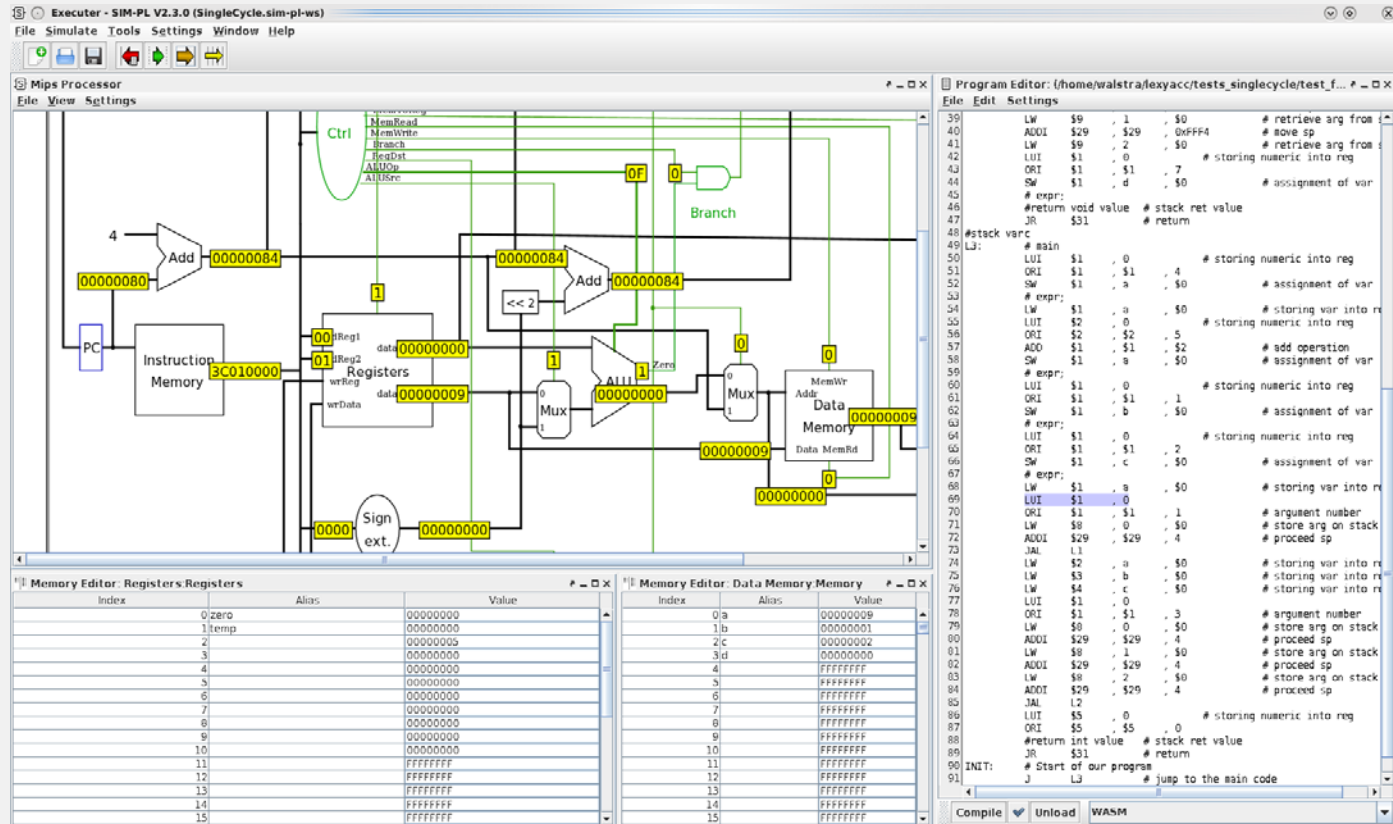
- Dedication, curious, critical and active.
- Respect your fellow students.
- No loud talking, music or gaming!
- Tidiness and don't leave without cleaning up your mess!

Introduction SIM-PL

Logic simulator

SIM-PL

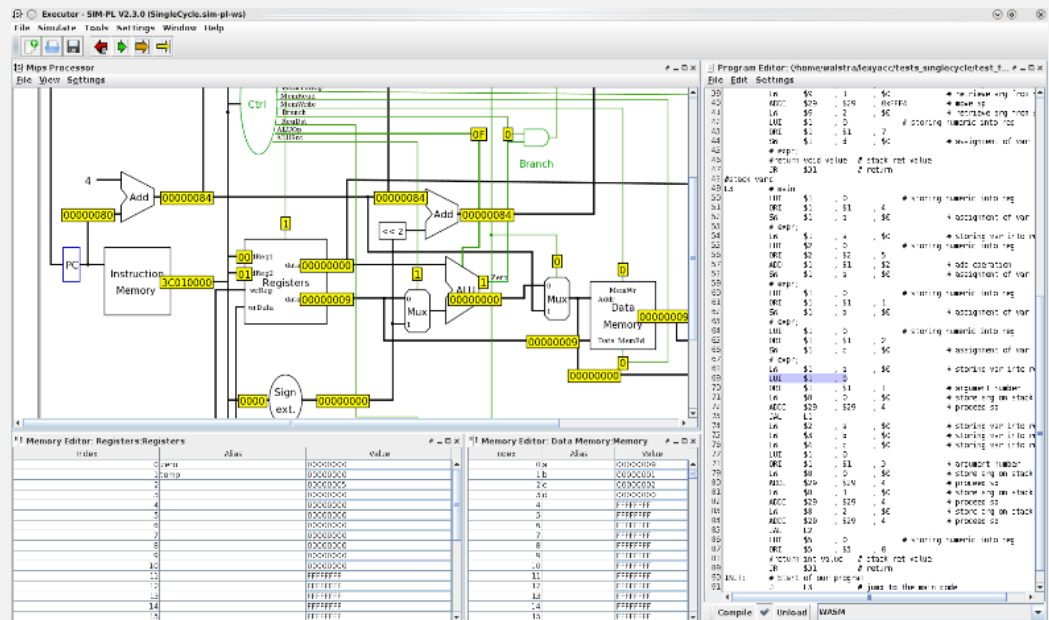
build and simulate computer architectures



SIM-PL

building and simulate computer architectures

- **Editor** application for building (complex) components and architectures.
- **Executer** application for simulating architectures
- Developed in Java



SIM-PL - Executer

The screenshot displays the SIM-PL V2.3.0 MultiCycle ARCH.slm-pl-ws interface. The main window shows a block diagram of a computer with 'CONTROL' and 'DATAPATH' blocks, 'INPUT' and 'OUTPUT' ports, and a clock signal. The 'Program Editor' on the right contains MIPS assembly code for calculating squares. The 'Time Sequence Diagram' at the bottom left shows a clock signal. The 'Memory Editor' at the bottom right shows a table of memory addresses and values.

Program Editor: (home/waltra/simpl/Components/aa_pract/Mul...)

```

1 # Author: Wouter Koolen-Wijkstra
2 # Description: MIPS program
3 # works on single- and multicyle architecture
4 # calculates a list of squares of consecutive integers
5 # This program takes 120 cycles to fully execute on the Pipelined MIPS pro
6 # Import the MIPS assembler definitions (syntax + semantics)
7 @include "MultiCycle.wasm"
8
9 .data MyRegisters : REGISTERS
10 0x00 : WORD 0x00 # put $0 to zero (default assumption)
11
12 .code MyCode: MIPS, MyRegisters
13 WORD result
14 # the starting number to square
15 LI $1, 0x0
16 # the one-after-last number to square
17 LI $2, 0x10
18 # the index to write the result in memory
19 LI $3, 0x4
20 # a (case sensitive) label delimits the loop
21 Loop:
22     #calculate the square
23     MUL $4, $1, $1
24     NOP
25     ADDI $1, $1, 0x1 # to ensure $4 is available on the next cycle
26     SW $4, result($3) # store the square
27     ADDI $3, $3, 0x4 # next memory location
28     BNE $1, $2, loop # compare to maximum integer to square
29     NOP
30     NOP
31
32
33
34
35
36
37
38
39
40
41
42
43
44
45
46
47
48
49
50
51
52
53
54

```

Memory Editor: Regi...

Index	Alias	Value	Index	Alias	Value
0		00000000	1	0x00	FFFF
1		00000001	2		4001
2		00000010	3		4002
3		00000004	4	Loop	0021
4		00000000	5		000C
5		FFFFFFFF	6		2921
6		FFFFFFFF	7		8464
7		FFFFFFFF	8		2963
8		FFFFFFFF	9		C422
9		FFFFFFFF	10		000C
10		FFFFFFFF	11		000C
11		FFFFFFFF	12		FFFF
12		FFFFFFFF	13		FFFF
13		FFFFFFFF	14		FFFF
14		FFFFFFFF	15		FFFF
15		FFFFFFFF			

SIM-PL - Executer

Executer is started at the command prompt

```
java -jar Executer.jar
```

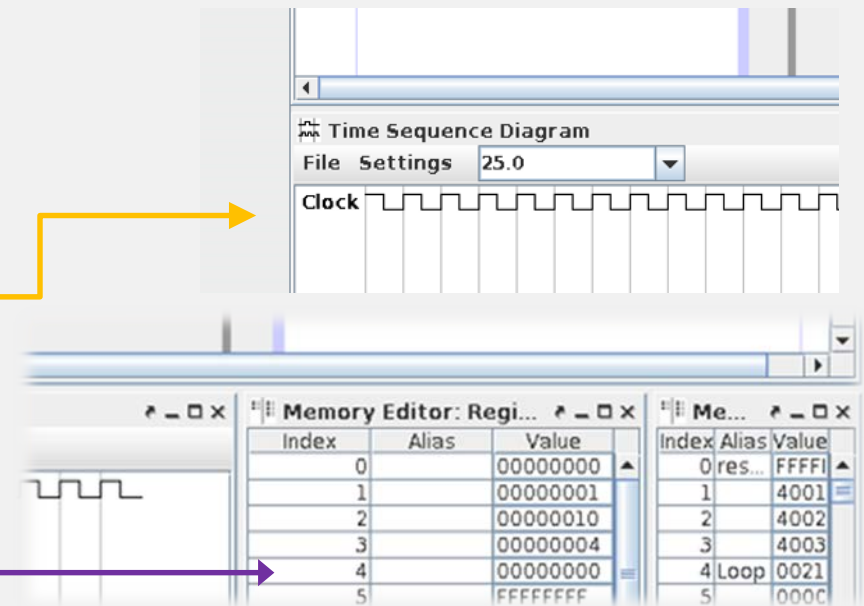
Architectures are loaded in the **Component view**.

Programs are loaded in the **Program editor**.

Simulation results are shown in the
Time sequence diagram

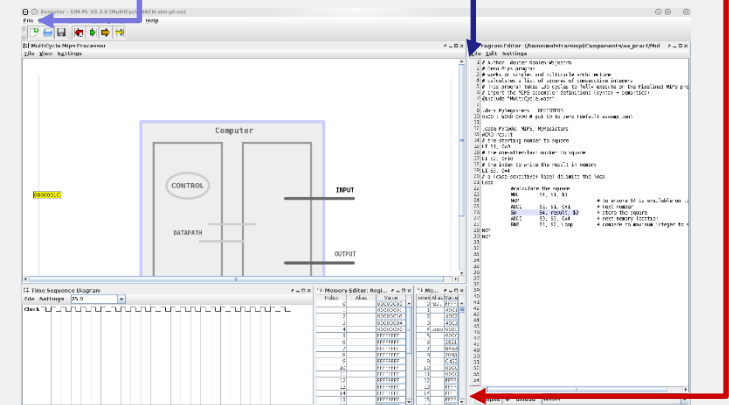
Memory editor

Register editor



SIM-PL – Executer Running Simulations

- Designs are loaded in the “Component view”.
- File -> open or new
- Programs are loaded in the “Program editor”.
- File -> open
- Compile
- Compiled code transferred to the design
- Press the Red arrow button
- Simulation starts by pressing the
- Green or Yellow or Orange button.



Downloads available

Manual

SIM-PL 2.3.2

Additional components

@Canvas

Lab 1

Performance of Computer Architectures

Lab 1 experiment - Performance

- **Goal**

To understand how performance is measured.

- **How**

By doing experiments in SIM-PL the logic simulator.

Researching by running three small programs on three architectures

- **Tools**

SIM-PL (Logic simulator)

- **Results**

The results of the experiments are used to answer the questions in the Canvas quiz.

Kick-start Lab 1

Description and Quiz

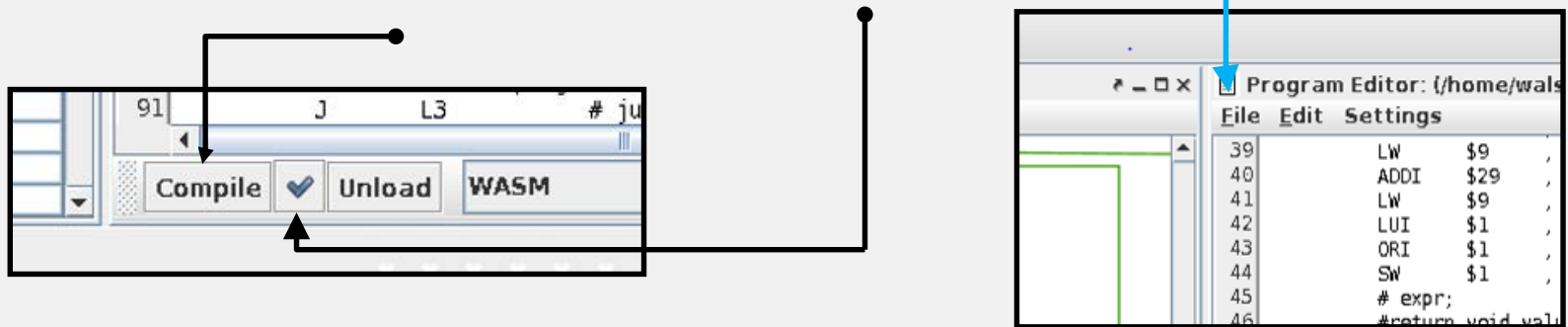


Lab 1 – Experiment 1

- Goal
 - Calculating the **CPI** of a computer architecture.
- How ?
 - Load an architecture in SIM-PL.
 - Execute a program.
 - Count the clock cycles and instructions until the program ends.
 - Determine the CPI ($\#clocks / \#instructions$).

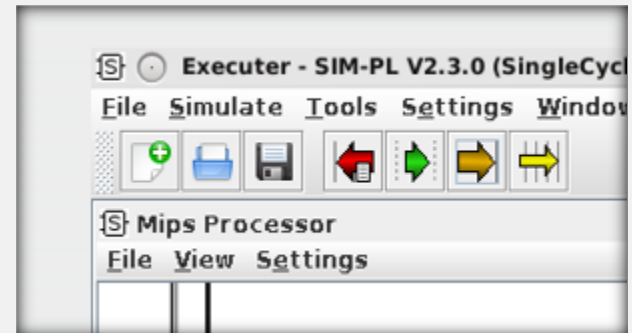
Lab 1 – Loading an architecture

- Start the SIM-PL executer.
 - `java -jar executer.jar` (Linux case sensitive)
- Load the **SingleCycle** architecture worksheet.
 - File->Open -> SingleCycle-architecture.sim-pl-ws
- Let's start with an addition of two numbers.
- Load the assembly program **addition.wasm** in the Program Editor.
 - File->Open -> addition.wasm
- Press the button “Compile” and wait for the checkmark



Lab 1 – Running the source code

- Load the compiled source code
 - Press the “Red arrow” button
 - First instruction is highlighted
- Execute the first instruction
 - Press the “Orange arrow” button
 - Clock cycle appears in Timing window
- Continue until all instructions are executed
 - No more highlighted instructions is end of program.
 - Don't forget to count !
- Write down the results in your lab-notes.
- Continue with the other programs and architectures.
- And complete this answer answer the Canvas quiz.



End of kick start session

Are there any questions ?

See you next time

Success !