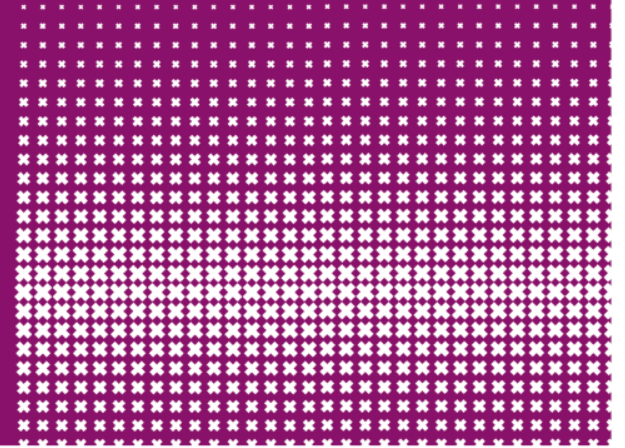




ing. Edwin Steffens

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
- Reports filenames must be conform  
`<Labname>-<YourCompleteName>.pdf`
- Source code filenames must be conform  
`<Labname><exercisenum>-<YourCompleteName>.pdf`
- Handed in via email before deafline: 21-09-2016 12:00 UTC.  
`e.h.steffens@uva.nl` and `cc t.r.walstra@uva.nl`

**No ZIP Files ! Not conforming? It will be rejected !**

More to read at the course page

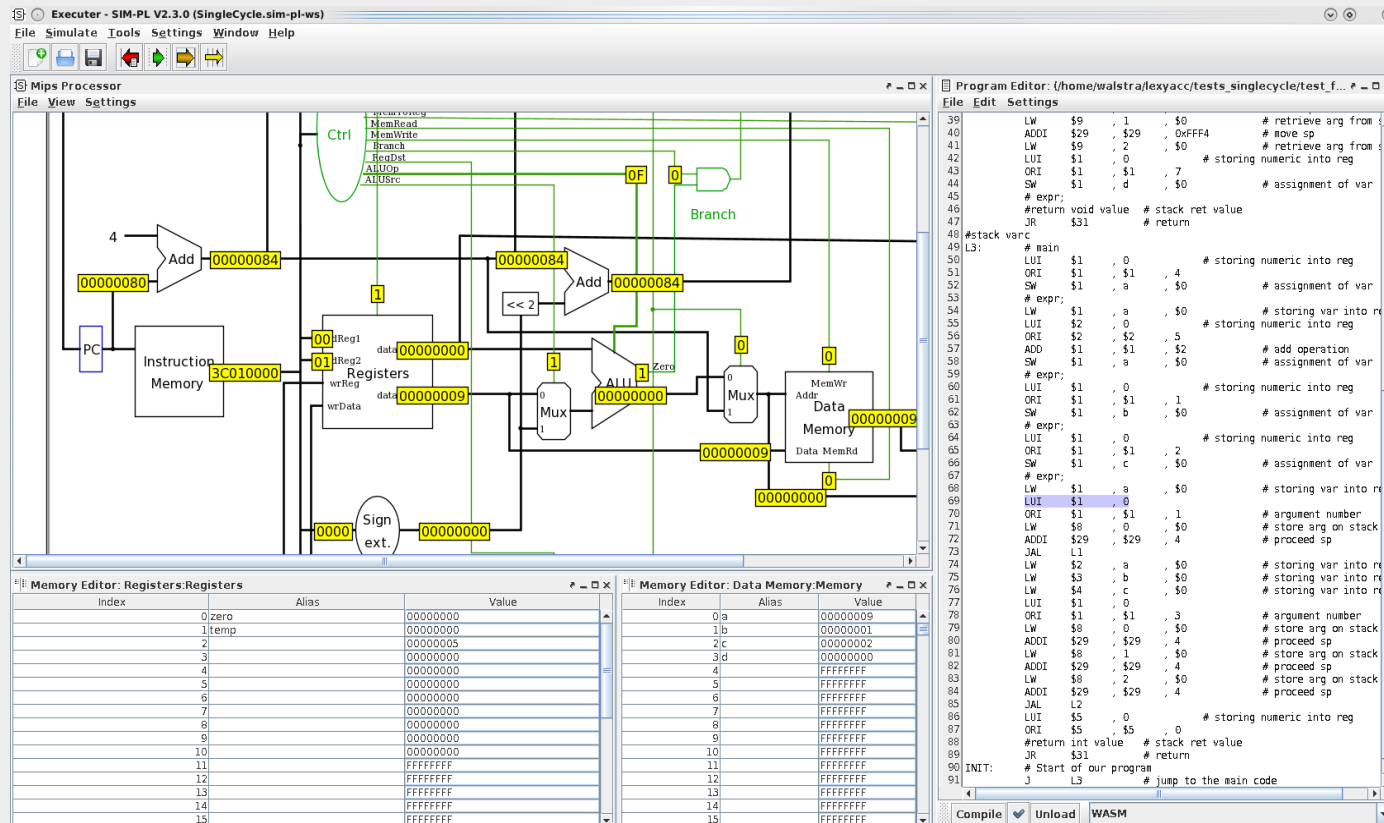
[staff.fnwi.uva.nl/e.h.steffens/?page\\_id=18](http://staff.fnwi.uva.nl/e.h.steffens/?page_id=18)



# Introduction SIM-PL

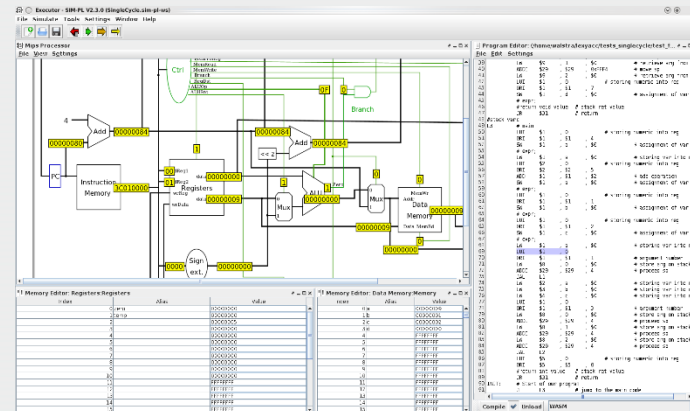
## Logic simulator

# SIM-PL – Our tool to build and simulate computer architectures



# SIM-PL – Our tool to build and simulate computer architectures

- “Editor” section for building (complex) components.
- “Executer” section for simulating
- Written in Java



# SIM-PL - Executor

The screenshot displays the SIM-PL Executor interface, which is a multi-window application for simulating a MIPS processor. The main window, titled "MultiCycle MIPS Processor", shows a block diagram of the processor with a "Computer" block containing "CONTROL" and "DATAPATH" units. The "INPUT" and "OUTPUT" buses are visible. A yellow box highlights the address "0000001C".

Below the main window, there are three smaller windows:

- Time Sequence Diagram:** Shows a clock signal and a settings dropdown set to "25.0".
- Memory Editor: Regi...:** A table showing the state of registers. The first column is "Index" (0-15), the second is "Alias", and the third is "Value".
- Me...:** A table showing the state of memory. The first column is "Index" (0-15), the second is "Alias", and the third is "Value".

On the right side, there is a "Program Editor" window showing the assembly code for the simulation. The code includes comments and instructions for a MIPS program that calculates the square of a number and stores it in memory.

```

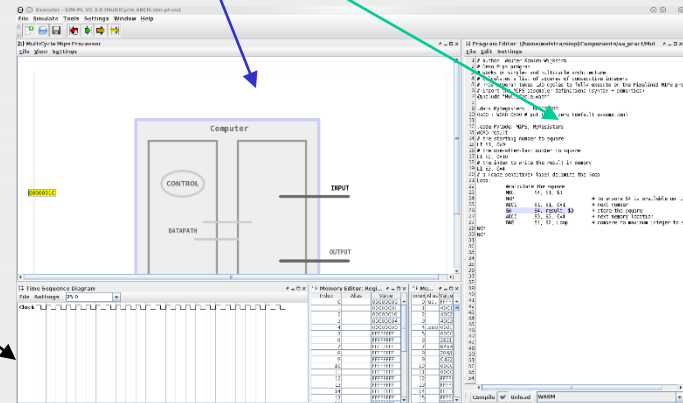
1 # Author: Wouter Koelen-Wijkstra
2 # Demo MIPS program
3 # works on single- and multicyle architecture
4 # calculates a list of squares of consecutive integers
5 # This program takes 130 cycles to fully execute on the Pipelined MIPS pro
6 # import the MIPS assembler definitions (syntax + semantics)
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 ti
26     SW $4, result, $3 # next number
27     ADDI $3, $3, 0x4 # store the square
28     BNE $1, $2, Loop # next memory location
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
55

```

At the bottom right, there are buttons for "Compile", "Unload", and "WASM".

# SIM-PL - Executer

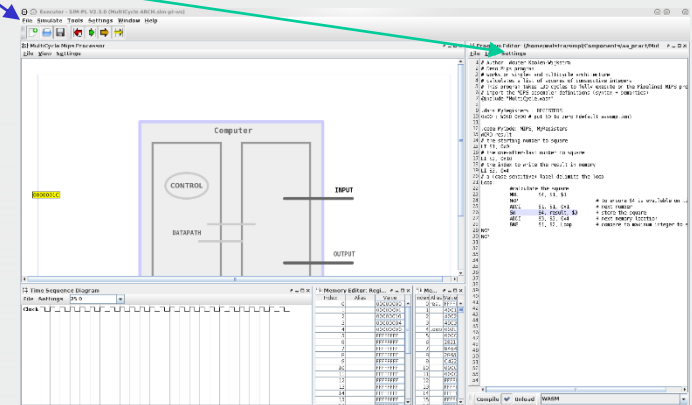
- Executer is started at the command prompt
- Java -jar Executer.jar
- Designs are loaded in the “Component view”.
- Programs are loaded in the “Program editor”.
- Simulation results are shown in
  - Time sequence diagram
  - Memory editor
  - Register editor





# 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 2.3.0

Additional components

Course webpage [e.h.steffens](http://e.h.steffens)



# 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 assignments are handed in for grading.



# Kickstart Lab 1

Description

Course webpage

[staff.fnwi.uva.nl/e.h.steffens](http://staff.fnwi.uva.nl/e.h.steffens)

# Lab 1 – Experiment 1

- Goal is to determine 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

- Download and install SIM-PL
- Start the 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
- 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 highlighted instructions is end
  - Don’t forget to count !
- Write down the result in your report.
- Continue with the other programs and architectures





# End of kick start session

Are there any questions ?



# See you next time

# Success !