Projectplan

SystemC Computer Architecture Simulator

R.A.J. Wacanno

April 3, 2020

1 Context

Simulation is an important instrument in research. Having the ability to test models without requiring expensive resources and without being hindered by uncontrollable environmental variables makes this form of research more accessible and these methods easier to reproduce.

The former of these advantages is especially true in the case of system architecture design, where implementing and manufacturing working hardware is both time consuming and incredibly expensive. For this reason efforts are made to enable the construction and testing of system architectures using software based models.

In the case of this project an additional need exists for the creating of a framework which enables such software based system architecture simulation. This project aims to simulate the execution of instructions, while also showing the user clearly which data streams exist between components and what logic exists inside components. This way, designed architectures can easily be understood and debugs in case errors arise.

2 Literature

As it stands, this project won't be the first to tackle the issue of architecture simulation. Advances in the field of simulation for architecture design [Kohl et al., 2016] as well as research on more general purpose architecture simulators [Schoon, 2020] using SystemC already exist.

When it comes to VHDL synthesis based on SystemC specifications an article by Cote and Zilic [Cote and Zilic, 2002] and a more recent article by Chen [Chen, 2011] have been published diving into the details behind such mechanisms.

3 Research Question and Project Goal

Central to this project will be the following research question:

> How can a robust architecture synthesis and simulation be employing SystemC as its description language be developed?

This main research question will be answered—and expanded upon—by further answering the following sub-questions:

- How can SystemC based components be combined to construct more complex combinational logic?

- How can a SystemC based hardware architecture be simulated in a robust, extendable and accurate fashion?
- How can software-based hardware be clearly visualised such that its inner workings are explained during simulation?
- How can hardware described in SystemC be translated to VHDL and executed on FPGA hardware?

As illustrated in the previous section, some work has been done already in the field of system architecture simulation. Therefore this project aims to build upon this body of reference work by either extending certain functionality or by combining pieces of separate existing functionality into one convenient package.

In particular, the aim is to build a functionally robust framework for simulating system architectures described using SystemC code. Additionally, the package should also allow a user to construct new pieces of combinational logic based on existing SystemC modules or by constructing new ones. Another important part of the simulation will be the ability to view the architecture in a legible schematic way, as well as data streams moving through the RTL abstracted model.

Lastly, the simulation framework will be extended with features for generating VHDL code based of SystemC described architectures to enable users to run their models on FPGA hardware.

4 Method

4.1 Literature-based research

With such a broad goal description, more research on the required technologies and prior achievements in the same areas will need to be conducted. This research will also be used to construct a satisfactory body of publications, which will serve as the basis of this project

4.2 Implementation

Implementation of the framework will be split up into several parts. Some of these parts might be worked on concurrently as a result of their functional dependency.

SystemC Synthesis
The first part of the implementation will be constructing the basis for the SystemC architecture framework. This step involves implementing a framework for combining existing modules with connection descriptions into more complex structures. Additionally, based on configuration files, new basic modules should be able to be implemented. While headless in this stage, later on these pieces of functionality will hopefully be added to the simulators front-end.

Basic Simulation and Visualisation
To test the functional correctness of the generated SystemC components and as a basis for the eventual simulator, basic simulation logic will be implemented in this phase. Additionally, rudimentary visualisations enabling the monitoring of component in- and output will be implemented to accompany this basic simulation.

Implementation basic MIPS architecture
Once the basic framework functionality has been implemented, it will need te be tested using some (basic) system architecture. In order to do this, a (subset of the) MIPS architecture [Patterson et al., 2014] will be implemented in RTL using the SytemC module synthesis of the framework and run through the basic simulation functionality.

Connection with Assembly Simulator
As discussed in the literature section of this plan, parallel with this project, an assembly simulator will be constructed by a fellow student. The aim is for this simulator to be integrated into this project using an API which will be negotiated between me and the other student. This API ought to be a general communication interface between simulated processor and external instruction memory handler.

Finalisation Simulation Visualisation
Once all back-end functionality has been thoroughly implemented, attention will be directed to the front-end of the simulator. The simulated architecture should (during execution) show a schematic visualisation of the hardware components and should allow the user to see data streams inside and in between components as per the goals of the project.

Conversion of SystemC Hardware Description to VHDL
If time permits it, the framework will be extended with functionality which allows SystemC based logic to be converted to a functionally equivalent VHDL implementation. This VHDL description can in turn be executed on FPGA hardware.

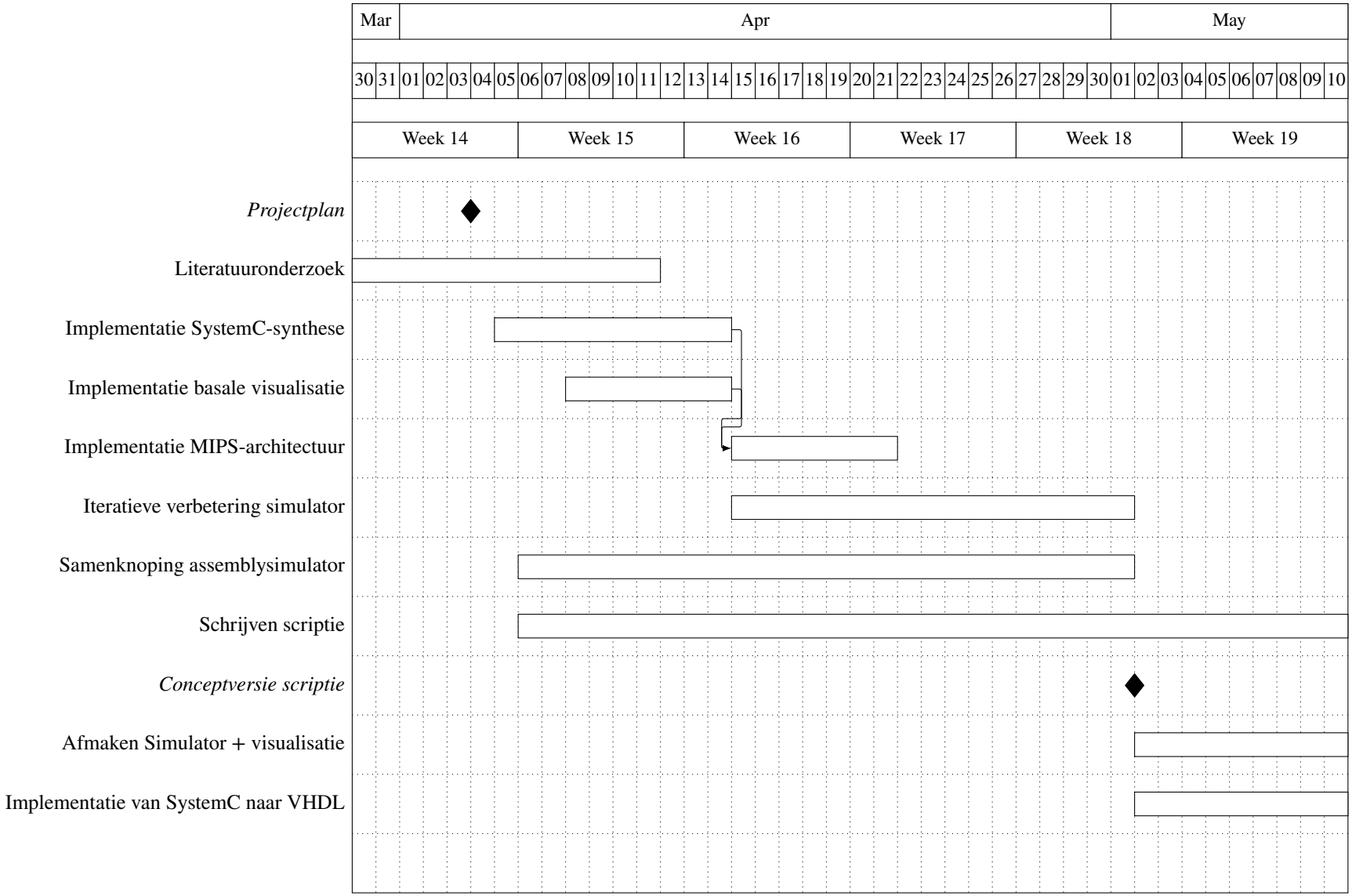
4.3 Testing with Assembly Simulator

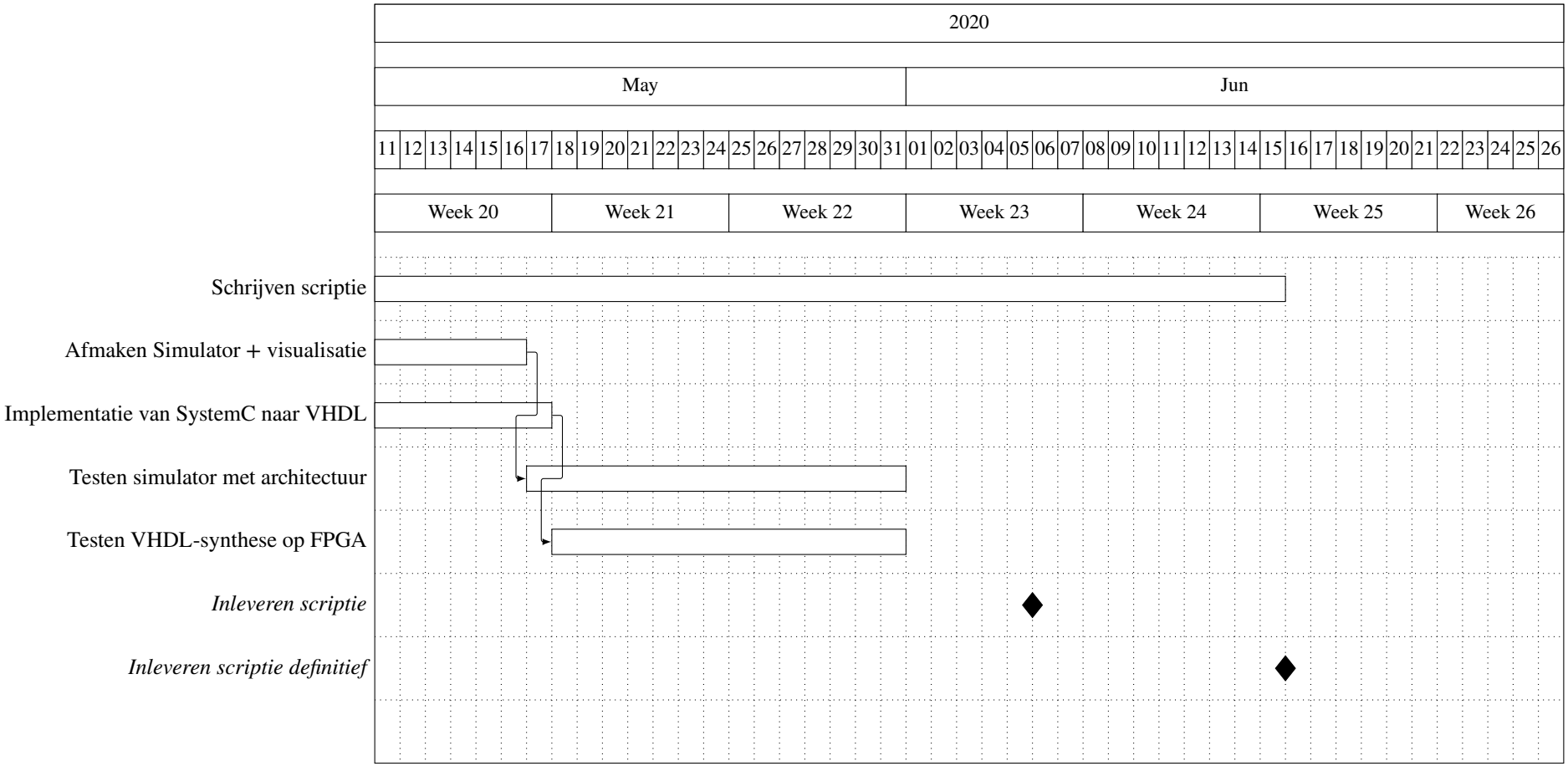
During the construction of the simulator, after the first 3 phases of implementation have been more or less finished, its functionality will be tested using a MIPS architecture implementation in SystemC. During the absence of the integration of the aforementioned assembly simulator, a dummy byte-code generator will be used in order to execute and test the implemented instructions.

4.4 Testing on FPGA hardware

If (basic) conversion from SystemC to VHDL is able to be implemented, its functionality has to be tested. This extra phase will involve comparing converted VHDL descriptions to descriptions directly programmed in VHDL of the same logic components and comparing their results when being executed on FPGA hardware.

5 Planning


Mar
Apr
May
30 31 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 01 02 03 04 05 06 07 08 09 10
Week 14
Week 15
Week 16
Week 17
Week 18
Week 19
Projectplan
Literatuuronderzoek
Implementatie SystemC-synthese
Implementatie basale visualisatie
Implementatie MIPS-architectuur
Iteratieve verbetering simulator
Samenknoping assemblysimulator
Schrijven scriptie
Conceptversie scriptie
Afmaken Simulator + visualisatie
Implementatie van SystemC naar VHDL


2020
May
Jun
11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26 27 28 29 30 31 01 02 03 04 05 06 07 08 09 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25 26
Week 20
Week 21
Week 22
Week 23
Week 24
Week 25
Week 26
Schrijven scriptie
Afmaken Simulator + visualisatie
Implementatie van SystemC naar VHDL
Testen simulator met architectuur
Testen VHDL-synthese op FPGA
Inleveren scriptie
Inleveren scriptie definitief

References

[Chen, 2011] Chen, R. (2011). Synthesizable systemc to vhdl compiler design.

[Cote and Zilic, 2002] Cote, C. and Zilic, Z. (2002). Automated systemc to vhdl translation in hardware/software codesign. volume 2, pages 717 – 720 vol.2.

[Kohl et al., 2016] Kohl, J., Reichenbach, M., Demel, C., and Fey, D. (2016). A SystemC based framework for cycle accurate processor simulation and parameter analysis. *IFAC-PapersOnLine*, 49(25):92–97.

[Patterson et al., 2014] Patterson, D. A., Hennessy, J. L., and Alexander, P. (2014). *Computer Organization and Design: the hardware/software interface*. Morgan Kaufmann.

[Schoon, 2020] Schoon, S. (2020). Hierarchical logic simulator using systemc.