Hardware Engineering - Syllabus

Chip Design Verification

February 2020 Cohort

Summary (about 1 month each)

1. Fundamentals
2. Architecture and Design
3. Verification
4. Final project

Details

Fundamentals

The gist: Essential scripting languages and extension of basic hardware and software topics

This section of the program includes core components that are essential before approaching actual Design and Verification topics. Two major scripting languages are to be mastered for future mostly for automation and information extraction. More topics such as Linux, Git and more will also be practiced.

Architecture and Design

The gist: State of the art architectures, chip design and everything in between

In this section, you will get a glimpse into cutting-edge architectures, get to know a specification document, start practicing a HDL and implementing complex blocks, understand what is microarchitecture, learn about important protocols and models and more.

You will also encounter cutting-edge tools for simulation and debugging purposes, data extraction out of spec documents and more.

Once understanding what Design is and after practicing Verilog you will finally be able to understand the crucial need for high quality verification and why it is so important part in the chip design process.

During this phase will be hosting some very interesting workshops (some of them will take place in top companies' offices):

- Network protocols and layers mode – Given by Mellanox
- ARM Architecture – 3 days course in Marvell office
- Simulation and Debugging training given by Cadence
- More to be announced

Verification

The gist: What is Design Verification, different approaches and advanced tools

You will start with the fundamentals of Functional Verification and master its most common hardware verification languages. you will also explore and practice the UVM library brings much automation to the SystemVerilog language such as sequences and data automation features.

Another and super important verification technique, which is complementary and not alternative to Functional, is Formal Verification which you will also master.

Other verification matters such as timing, power and performance will be studied.

You will get to know and practice more important tools for planning, project management, logs and advanced debug purposes.

Assertions, block level, system level, direct vs. random and more important topics will be introduced and covered.

How a project in the industry looks like? What is the beginning and what is the end? How do we approach initial architecture details and start planning and implementing a verification

environment? We will deal with this.

What is a Regression and what are its advantages and problems? How can we automate this process?

Also during this phase will be hosting some very interesting workshops (some of them will take place in top companies' offices):

- A 5-days Formal Verification training in **Mellanox** office
- Edge Randomization, Randomization Gotchas, Re-seeding – given by **Samsung**
- PCIe protocol overview – given by **Apple**
- Intel Architecture – given by **Intel**
- Advance UVM topics – given by **Samsung**
- System level validation and integration validation – given by **Intel**
- Verification Planning – given by **Intel**.
- Emulation, Jasper, vManager, UPF trainings given by **Cadence**
- More TBA

Towards the end of the Verification chapter we will start working on the final project. The project will include all the different phases of a „real' project and will require arch, design and of course verification skills. You will be dealing with planning, coverage definitions and implementation and you will need to reach the target of 100% pass rate and coverage. All the companies mentioned above will send engineers to come and mentor you as you progress through the different phases of the project. You will need to use all the skills you have acquired so far and it will be a great opportunity to sharpen your knowledge, tools usage and coding skills before going out to the "real world".