Planning and Timeline for my Master Thesis on Hardware Acceleration of Homomorphic Encryption

Jonas Bertels

October 2020

1 Introduction

This document lays out the planning for my master thesis on hardware acceleration of Homomorphic Encryption.

> "Fully homomorphic encryption (FHE) is a class of encryption algorithms that support any computation on encrypted messages without revealing anything about these messages in unencrypted form except for their maximal size. Using FHE, a party that owns private data can securely outsource computations on this data to another party. Due to this functionality, FHE finds many applications both in practice (e.g. cloud computing) and in the design of new cryptographic algorithms." [1]

However, current algorithms are not efficient enough to be truly applicable which is why there is significant interest in hardware acceleration of these FHE schemes.

2 Planning

The thesis proposal suggested the following distribution of work:

- 20% on literature
- 20% on theoretical work
- 60% on the hardware implementation

In terms of my Individual Study Program (ISP), 9 study points were allocated to the first semester and 15 to the second semester. The FPGA on which the hardware implementation will be created will only be available in the second semester, and obviously most of the literature and theoretical work should be done before the hardware implementation. More specifically, these are the main tasks that must be finished:

2.1 Milestones

1. **Determination of the scheme (by 26-10-2020)** Before being able to implement a scheme, the scheme (or part of a scheme) that is to be implemented must be chosen. Therefore several papers on the available schemes must be read, both theoretical papers and papers outlining how these schemes can be implemented into hardware.

2. **Deeper investigation into the chosen scheme (by 09-11-2020)** After the scheme or part of the scheme to be accelerated is chosen, the different parts of the scheme must be broken down into modules that can be implemented piece by piece. This can only be done properly after the scheme is chosen, but for the purpose of this planning, we have assumed we would implement the FV scheme.

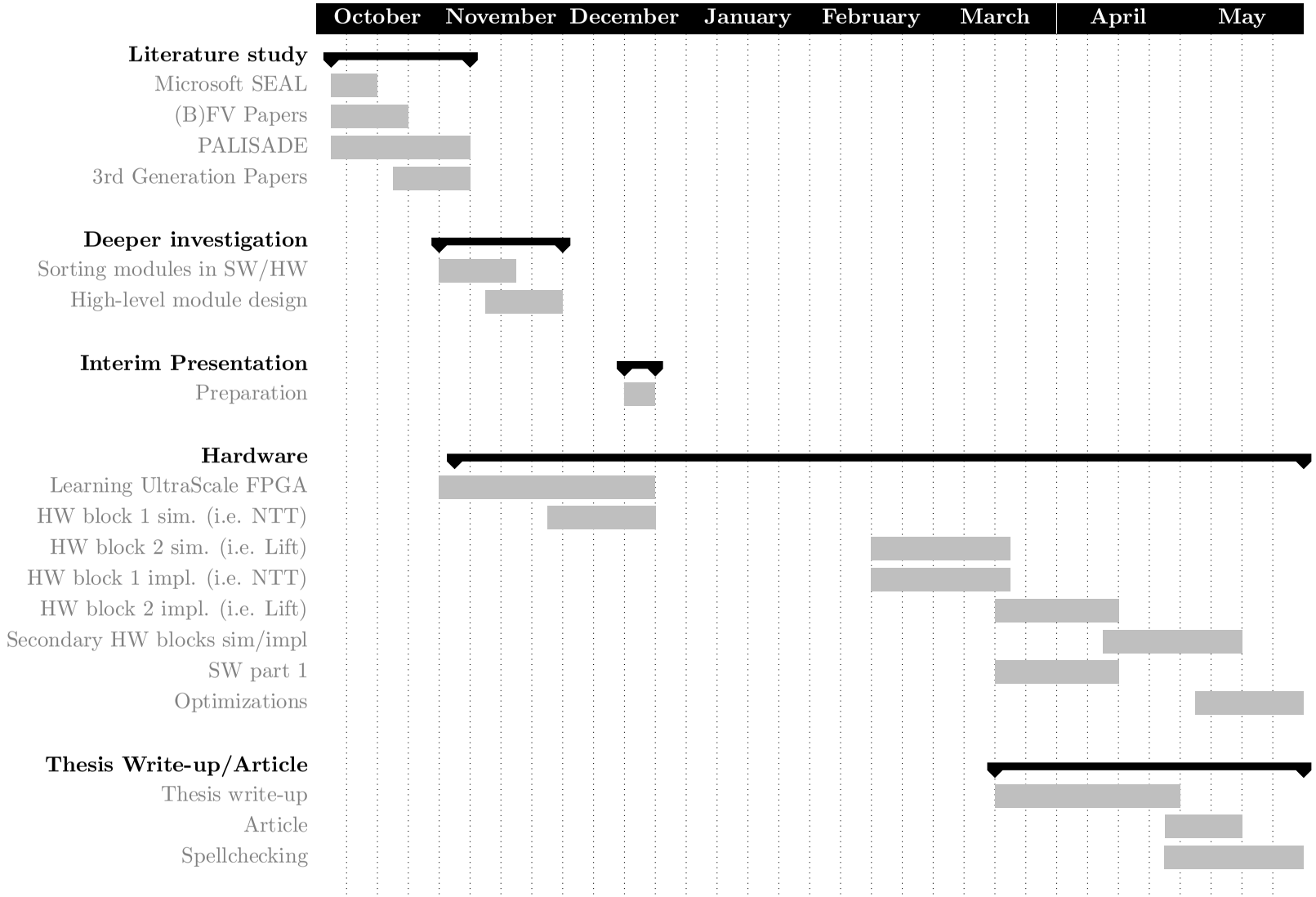
3. **Hardware (by 03-05-2020)** This can be broken down into several parts: learning to work with the UltraScale+ architecture, building the hardware modules, simulating to verify correctness and writing the required SW part. Once the FPGA is available, the design can then be implemented.

4. **Thesis write-up (written by 01-05-2020, spell-checked by 31-05-2020)** As the thesis will mirror my work done, this could be broken down in a literature survey, theoretical work and hardware section, but should mainly feature the latter.

5. **Article (written by 16-05-2020, spell-checked by 31-05-2020)** Writing the article after the thesis is finished should be a relatively straightforward task. While the article is being written, time can also be spent on the last milestone, namely further optimizing the HW and SW implementation.

6. **Further optimizations (by 31-05-2020)**

3 Gantt Chart


October
November
December
January
February
March
April
May
Literature study
Microsoft SEAL
(B)FV Papers
PALISADE
3rd Generation Papers
Deeper investigation
Sorting modules in SW/HW
High-level module design
Interim Presentation
Preparation
Hardware
Learning UltraScale FPGA
HW block 1 sim. (i.e. NTT)
HW block 2 sim. (i.e. Lift)
HW block 1 impl. (i.e. NTT)
HW block 2 impl. (i.e. Lift)
Secondary HW blocks sim/impl
SW part 1
Optimizations
Thesis Write-up/Article
Thesis write-up
Article
Spellchecking

References

[1] Ilia Iliashenko. *Optimisations of fully homomorphic encryption*. PhD thesis, KU Leuven, May 2019.