Master Thesis Proposal

Didactic Illustration of Selected Clock Synchronization Algorithms in Exercises and Implementations

Christian Fiedler <christian.fiedler@student.tuwien.ac.at>
Advisor: Prof. Andreas Steininger
September 18, 2023

1 Motivation & Problem Statement

The ever-shrinking node sizes in semiconductor manufacturing allow ASIC designs to grow larger and larger. Implementors that choose a synchronous design face the challenges of a large clock network, among others: clock skew and reliability concerns. Clock synchronization algorithms such as the Welch-Lynch (1) or the Welch-Dolev (2) algorithm aim to solve these challenges while still keeping latency low. In [3], Lenzen et al. intend to compile clock synchronization algorithms and know-how for the design of advanced/large synchronous digital circuits. In a bottom-up approach, they introduce related concepts and algorithms together with formal proofs. A complementary exercise book together with solutions containing reference implementations of the presented algorithms may help readers intuitively comprehend and experiment with the algorithms. Implementations are going to be programmed in VHDL, an industry standard hardware description language, to illustrate the practical usability of the presented theory. Simulations of meaningful test cases need to be provided for the verification and analysis of the algorithms, along with solutions for the exercise book to make the material more easily accessible. Apart from the technical soundness of these solution templates, the didactic quality of the chosen exercises and questions as well as the presentation and discussion of the solutions are of vital importance for the value of the envisioned exercise book.

This thesis is set out to provide a first baseline for this exercise book.
2 Aim of the Work

The aim of the work is to provide a draft for both assignments and solutions of an exercise book for the topics discussed in [3]. Part of this is the implementation of selected clock synchronization algorithms in VHDL, including but not limited to the Welch-Dolev algorithm [2]. The implementation will include testbenches that highlight the function and also shortcomings of the algorithms (edge cases). For accessibility reasons, the implementation shall be compatible with edaplayground.com, a website that allows to run VHDL simulations in the web browser. The main aim is for these implementations to be of didactic value for engineers and scientists who want to familiarize themselves with clock synchronization and the challenges that come with synchronous large hardware designs.

The following research questions will be answered:

How can the theory about clock synchronization and the related algorithms presented in the book be best, from a didactic view, illustrated in practical exercises to a student reader?

Which practical aspects, like timing assumptions, limitations of hardware functionality, etc., must be considered when implementing complex clock synchronization algorithms (like the one by Welch-Dolev) in hardware?

How to select suitable corner cases to illustrate function as well as limitations of complex clock synchronization algorithms?

It is not the aim of this work to provide a complete or finished exercise book with full coverage of the whole material in the theory book. The target is rather to elaborate a solid basic setup and, at the example of selected chapters, a template for how this exercise book can be composed.
3 Methodological Approach

The following steps outline the methodological approach:

1. Literature Review
   The book by Lenzen et al. [3] provides the starting point for the literature review. Original papers and work based on them must be reviewed to check which of the algorithms have already been implemented and how useful such implementations are for the purpose.

2. Draft Exercise Sheets for Specific Algorithms & Program a Reference Implementation
   Exercise Sheets will be drafted which shall complement selected chapters of [3] and enable the reader to gain a practical understanding. A reference implementation will be programmed, tested and provided with useful comments. Based on feedback from a test student, the exercise sheets are improved.

3. Simulate Reference Implementations
   The reference implementation must be simulated for three reasons:
   - It is needed to verify the implementation.
   - The simulation is the main way to observe the behavior of the implementation.
   - Interesting edge cases can be simulated and pointed out to the reader.

4. Analysis & Conclusion
   Implementations are going to get verified and analysed. The behavior of the clock synchronization algorithms in edge cases is going to be compared to the theoretical expectations.
4 Structure of the Work

1. Introduction

2. General Approach and Methodology

3. State-of-the-art

4. Multiple Chapters with the following structure:
   (a) Book Chapter & Theoretical Background
   (b) Didactical Approach
   (c) Proposed Exercises
   (d) Solutions
   (e) Discussion

5. Conclusion & Future Work


5 State-of-the-Art

Research on clock synchronization protocols or algorithms was initially launched to establish a common sense of time in distributed systems with multiple processors [1, 5]. Fault-tolerant algorithms enable systems to stay synchronized even if some nodes do not abide by the protocol and show byzantine behavior [6]. Self-stabilizing algorithms, such as the algorithm of Welch and Dolev (Welch-Dolev) [2] can additionally establish synchrony when started from an arbitrary system state – as long as a certain fraction of nodes behaves correctly.

With ever more sophisticated clock-tree balancing efforts necessary in advanced synchronous hardware designs, different approaches are evaluated. Hardware implementations of clock synchronization protocols can be employed in computer chips to maintain clock synchrony between multiple clock regions or even generate the clock signal locally inside individual components of the chip, simplifying the task of distribution of clock signals and eliminating the single-point of failure of using a single clock [7].

The scientific literature on clock synchronization is broad and there is a large variety in proposed algorithms. Publications usually focus on proving correctness and properties of fault tolerance. Hardware description for implementations of such algorithms is usually not provided. Access to implementations could be very beneficial for this thesis, as it aims to illustrate such algorithms and allow their simulation.

Publications that implement clock synchronization algorithms for hardware clock generation include [8] where the algorithm of Welch and Lynch [1] has been implemented in a field programmable gate array (FPGA) and access to the VHDL implementation is available. Apart from that, the TTTech Time-Triggered Architecture (TTA) and FlexRay are industry-standard bus protocols based on that algorithm [9]. A different algorithm, the Gradient Clock Synchronization algorithm, has been implemented in [10] in a 15 nm ASIC.

We are not aware of publications that implement the Welch-Dolev algorithm, which will be implemented for this thesis in VHDL. We are also not aware of publications that focus on presenting HDL implementations of clock synchronization algorithms for didactic purposes.
6 Relevance to the Curriculum of Computer Engineering

The aim of this thesis to implement advanced hardware circuits, in particular clock synchronization algorithms, as well as to provide didactic experiment results aligns perfectly with the module **Digital Circuits and Systems** in the Computer Engineering Curriculum.

The most relevant courses that relate to the subject are listed in the following:

- 182.755 VU Advanced Digital Design
- 182.754 LU Advanced Digital Design
- VU Advanced Computer Architecture

References


