



# Master's Thesis

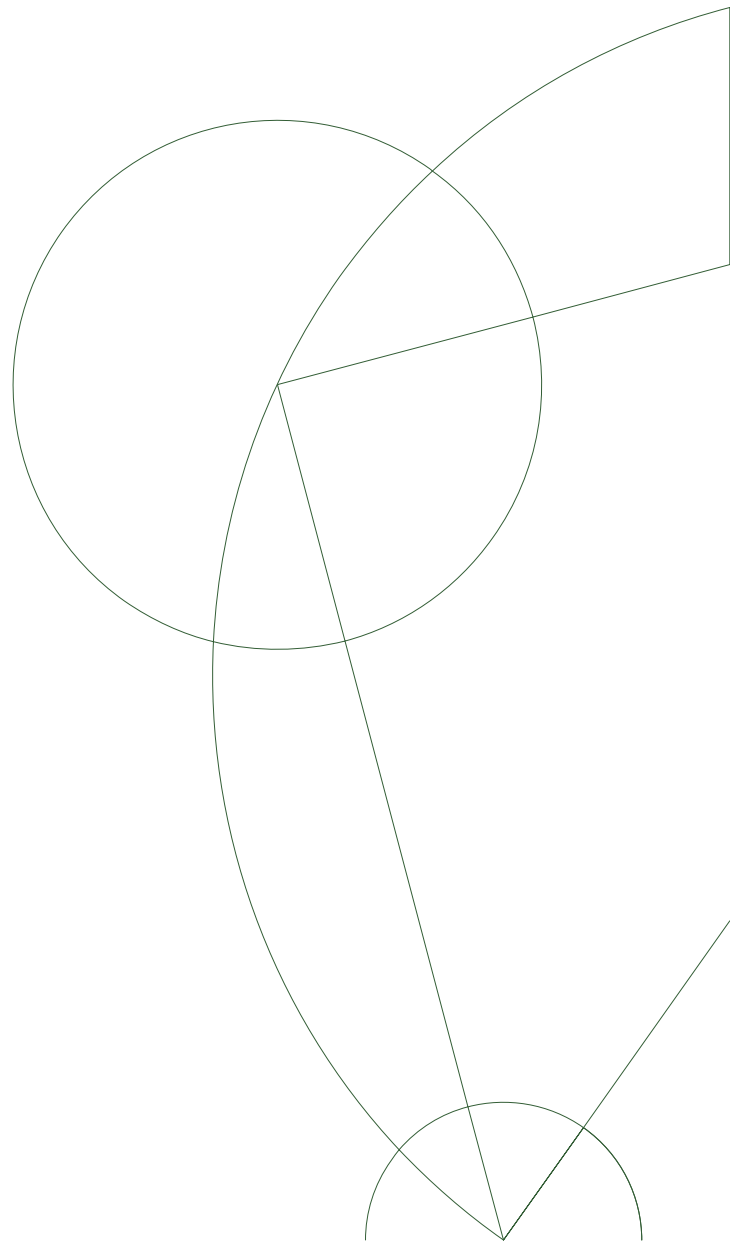
Alberte Thegler - [alberte@thegler.dk](mailto:alberte@thegler.dk)

## Towards formal verification of FDR4

Department of Computer Science

Professor Brian Vinter

August 2018



**Abstract**

Bla bla bla bla

## Contents

<b>1</b>	<b>Introduction</b>	<b>1</b>
<b>2</b>	<b>Related work</b>	<b>1</b>
2.1	Software Verification . . . . .	1
2.2	Hardware Verification . . . . .	1
<b>3</b>	<b>Theory</b>	<b>2</b>
<b>4</b>	<b>Implementation</b>	<b>2</b>
<b>5</b>	<b>Testing</b>	<b>2</b>
<b>6</b>	<b>Conclusion</b>	<b>2</b>
<b>7</b>	<b>Future Work</b>	<b>2</b>

## Todo list

Add reference to SPIN . . . . .	1
add reference to promela . . . . .	1
add reference to Linear Temporal Logic . . . . .	1

# 1 Introduction

When we create programs, we wish to verify that it is also correct. There are several ways to do this, one commonly used is **testing** which require that the programmer creates several different scenarios and its expected output, or that the programmer programs a test-generator to create the scenarios and expected output. This, however, is not adequate for (word for important systems). Therefore it is of high interest to create a verification of the system or program.

Talk about how verification was first created and how it became to be used for concurrent systems. Then write about how it works and then write about the different systems and formal languages that is used for it.

In this thesis we look at model checking, that is, verifying that a specific property will always hold for a piece of code.

## 2 Related work

**Related work should be about describing the history of formal verification and formal languages in both software and hardware.** Formal verification is the process of checking whether a program satisfies specific properties.

### 2.1 Software Verification

Several different software verification tools exists today, and they all have different advantages.

#### SPIN

SPIN is a verification system that uses process interactions to prove correctness for a system. The system is described in the formal language PROMELA (PROcess MEta LAnguage) and the correctness properties are specified in Linear Temporal Logic or LTL. Spin was developed at Bell Labs, starting in 1980. Since 1991 it has been freely available and today it is used by thousands of people worldwide. The job is to find other types of models and compare them, as well as the types of languages. so different tools and different languages..

Occam-pi is a programming language that came from occam and was merged with pi-calculus.

Add reference to SPIN

add reference to promela

add reference to Linear Temporal Logic

### 2.2 Hardware Verification

The job here is to find some examples of stuff that is relevant to SME and then see how I can compare them.

### **3 Theory**

### **4 Implementation**

### **5 Testing**

### **6 Conclusion**

### **7 Future Work**