ETH
Eidgenössische Technische Hochschule Zürich
Swiss Federal Institute of Technology Zurich

Institut für Integrierte Systeme
Integrated Systems Laboratory

Department of Information Technology and
Electrical Engineering

Spring Semester 2025

Ensuring Fault Tolerance in the CROC SoC: A Hardware-Centric Approach to Watchdog Timer Implementation

VLSI 2 Course Project

Miguel Correa and Elio Wanner
cream@ethz.ch, ewanner@ethz.ch

May 15, 2025

Professor: Frank Kagan Gürkaynak, kgf@iis.ee.ethz.ch

Abstract

This report summarizes the work done in the VLSI 2 course project. The project consisted of extending the Croc SoC with a Watchdog Timer.

Contents

Chapter 1

Introduction

1.1 Overview of the Croc SoC and its open-source nature

- Picture of Croc SoC
- Quick explanation of its main parts and mostly what we are using ==> $\text{user}_{d}omain$
- Mention that it is open-source and the importance of this, but also why it has to be robust for it to be competitive with closed source in real life scenarios

1.2 Why a watchdog timer (WDT) was needed

- Real life scenario as an example of anoying stuck in a loop chip (if possible, true historic reference)
- Explain what is a watchdog timer and how it would prevent this
- From this go on and conclude it was needed on the Croc SoC

1.3 Goals and expected improvements

- Emphasize on the main goals of the watchdog, with bullet points if possible to emphasize even more. THe main goals being: improve robustness while keeping area and throughput as low as possible. (should we add more goals?)

Chapter 2

Related Works

2.1 Overview of existing watchdog timer implementation

2.2 Justification for our approach

Chapter 3

Methodology

3.1 Design choices for WDT (e.g., countdown vs. count-up, reset duration)

3.2 Testing strategy (testbenches, simulation tools)

3.3 Debugging and iteration process

Chapter 4

Hardware Architecture

This chapter describes the hardware architecture of the Watchdog Timer that we implemented in the Croc SoC. In the different sections we talk about different implementations with different features and complexities.

4.1 Simple Watchdog Timer

4.1.1 Hardware implementation

The first implementation of the Watchdog Timer is the simplest one we could imagine. As shown in Figure 4.1, the WDT is composed of very few components. The main component is the counter, which is incremented every clock cycle and implemented as a flip-flop. The counter is connected to a comparator that checks if the counter has reached a certain value. If the counter reaches the value, the comparator will trigger a reset signal that will reset the system. Otherwise, we keep incrementing the counter. As soon as we get a kick signal, the counter is reset to zero.

4.1.2 Tests and Results

4.1.3 Conclusion

4.2 Watchdog Timer with 2 stages

This is for later.

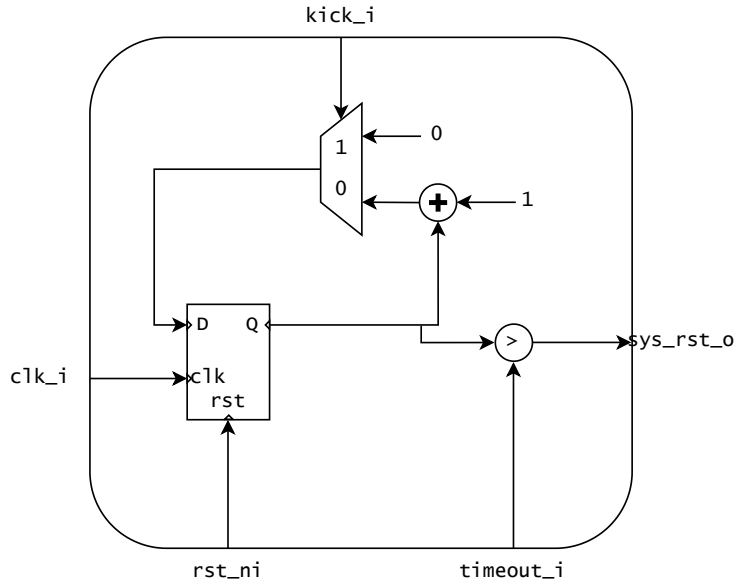

Figure 4.1: Simple Watchdog Timer Architecture

Chapter 5

Evaluation

5.1 Simulation results for v1 and v2

5.2 Performance comparison (goal achieved?)

5.3 Trade-offs between versions

Chapter 6

Discussion

6.1 Strengths and limitations of each approach

6.2 Potential improvements (e.g., three-stage WDT)

6.3 Lessons learned from the implementation

Chapter 7

Conclusion

7.1 Summary of key findings

7.2 Future work and next steps