Automated Implementation of the Digital Configuration Interface for Application Specific Integrated Circuits

Zhihong Lei

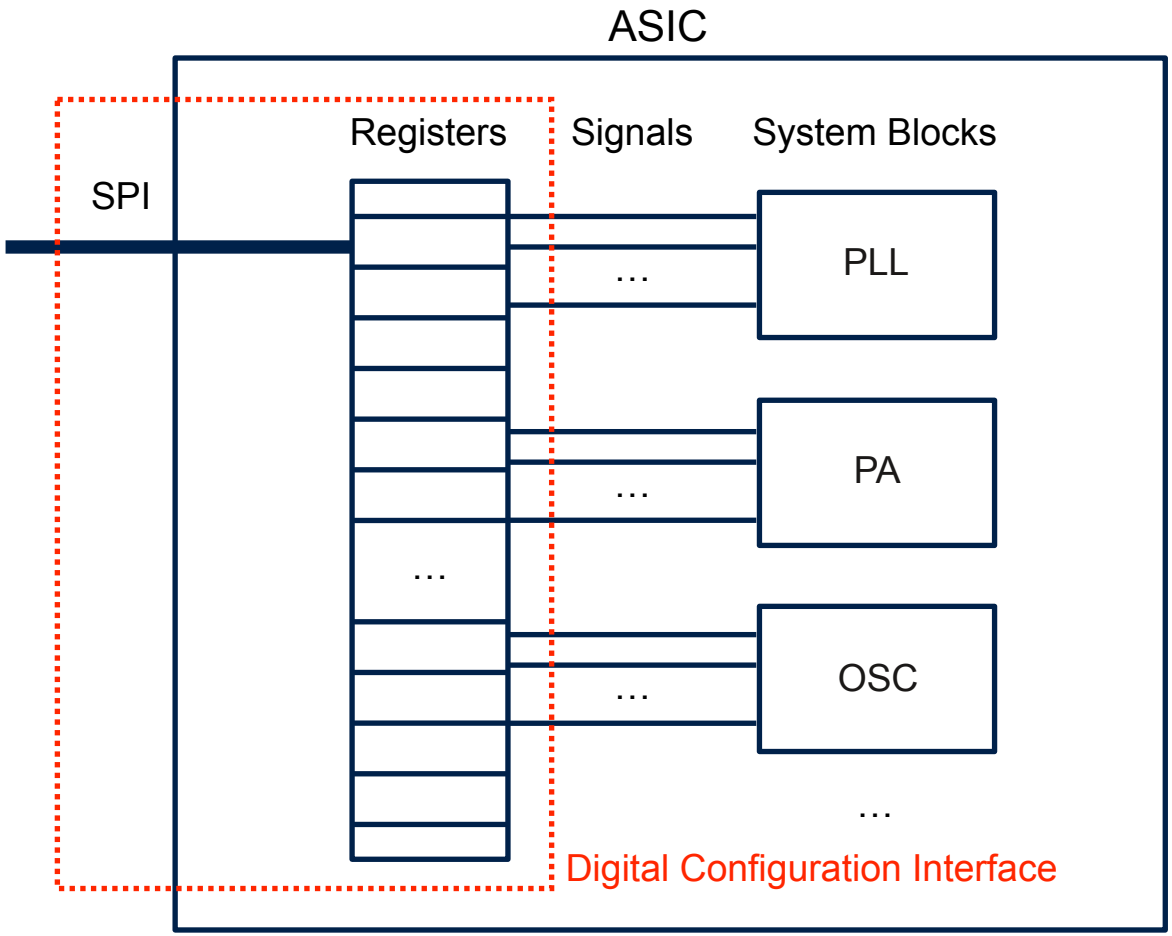
First Master Thesis Presentation

Supervisor: Johannes Bastl, M.Sc

Univ.-Prof. Dr.-Ing. Stefan Heinen
Integrated Analog Circuits and RF Systems Laboratory

Background

Configuration of ASIC System Blocks

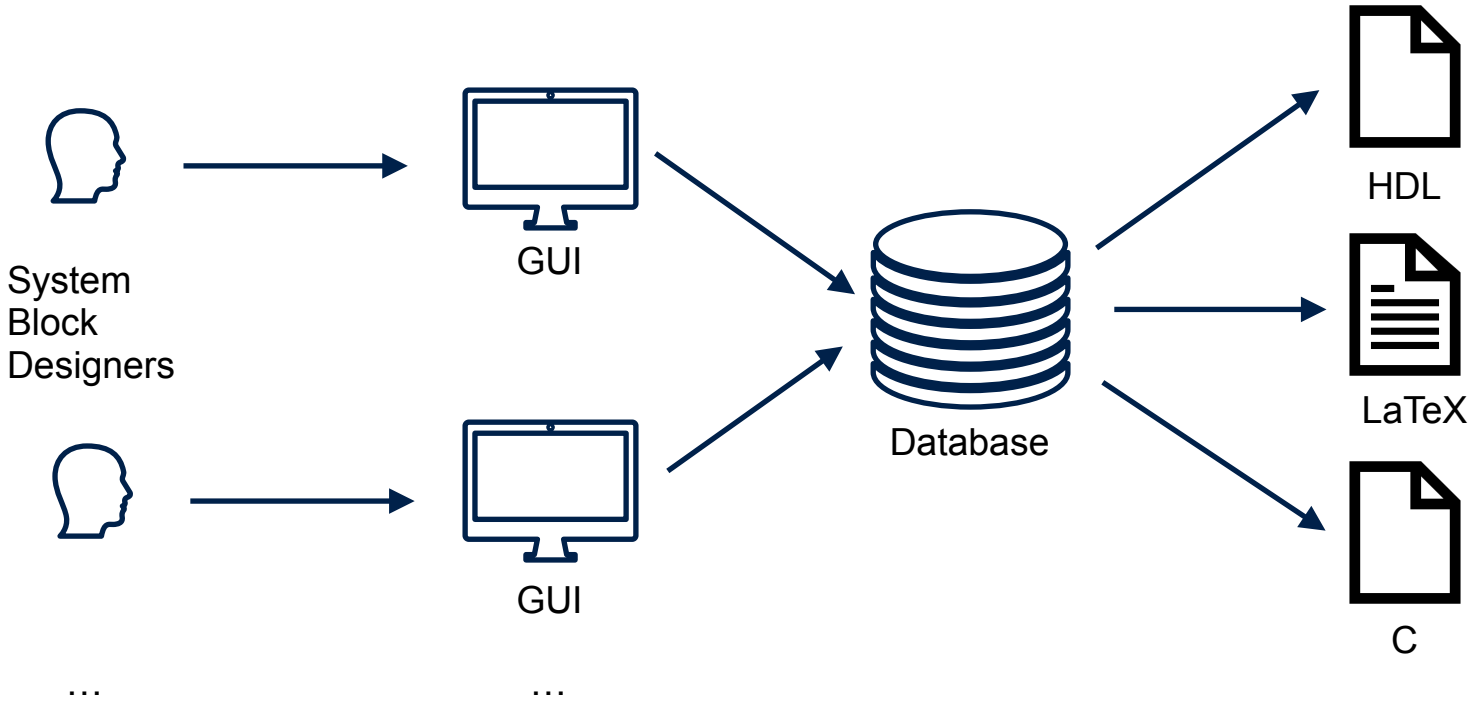


Problems

- Great amount of manual work
  - HDL source code
  - LaTeX documentation
  - Other programming languages (C, MATLAB etc)
- Communication and synchronization
- Complex ASIC: hundreds of registers

Proposed Solution

- A software that automatically generates HDL source code, LaTeX documentation and code in other formats (C, etc)



Advantages

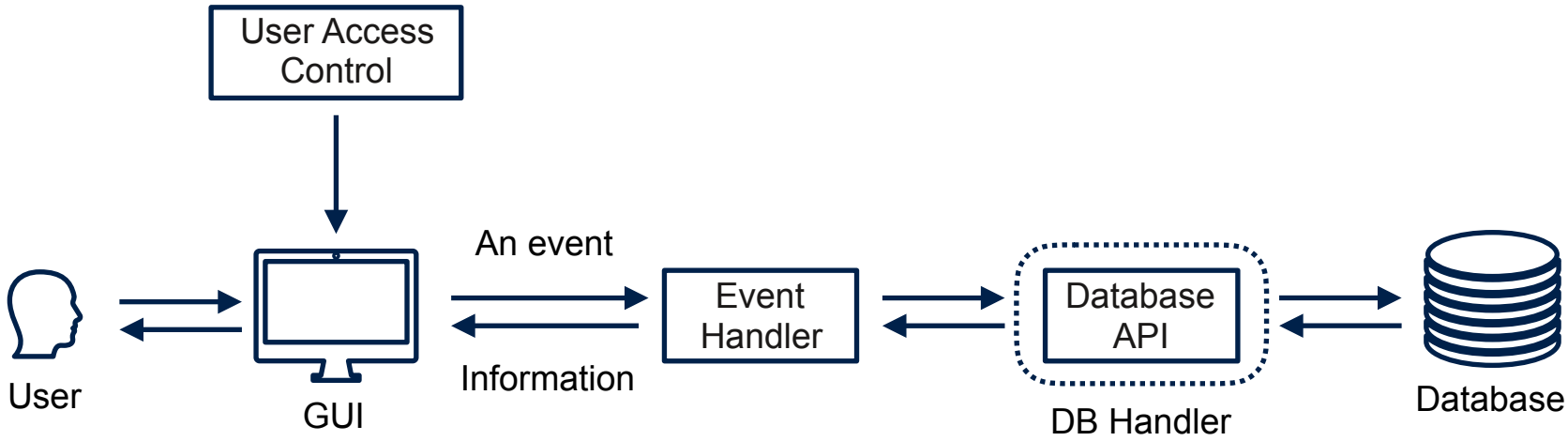
- Human labor will be greatly reduced
- Synchronization between different exports will be intrinsically ensured
- Errors can be eliminated (given that the register definitions themselves are correct)

Crucial Requirements

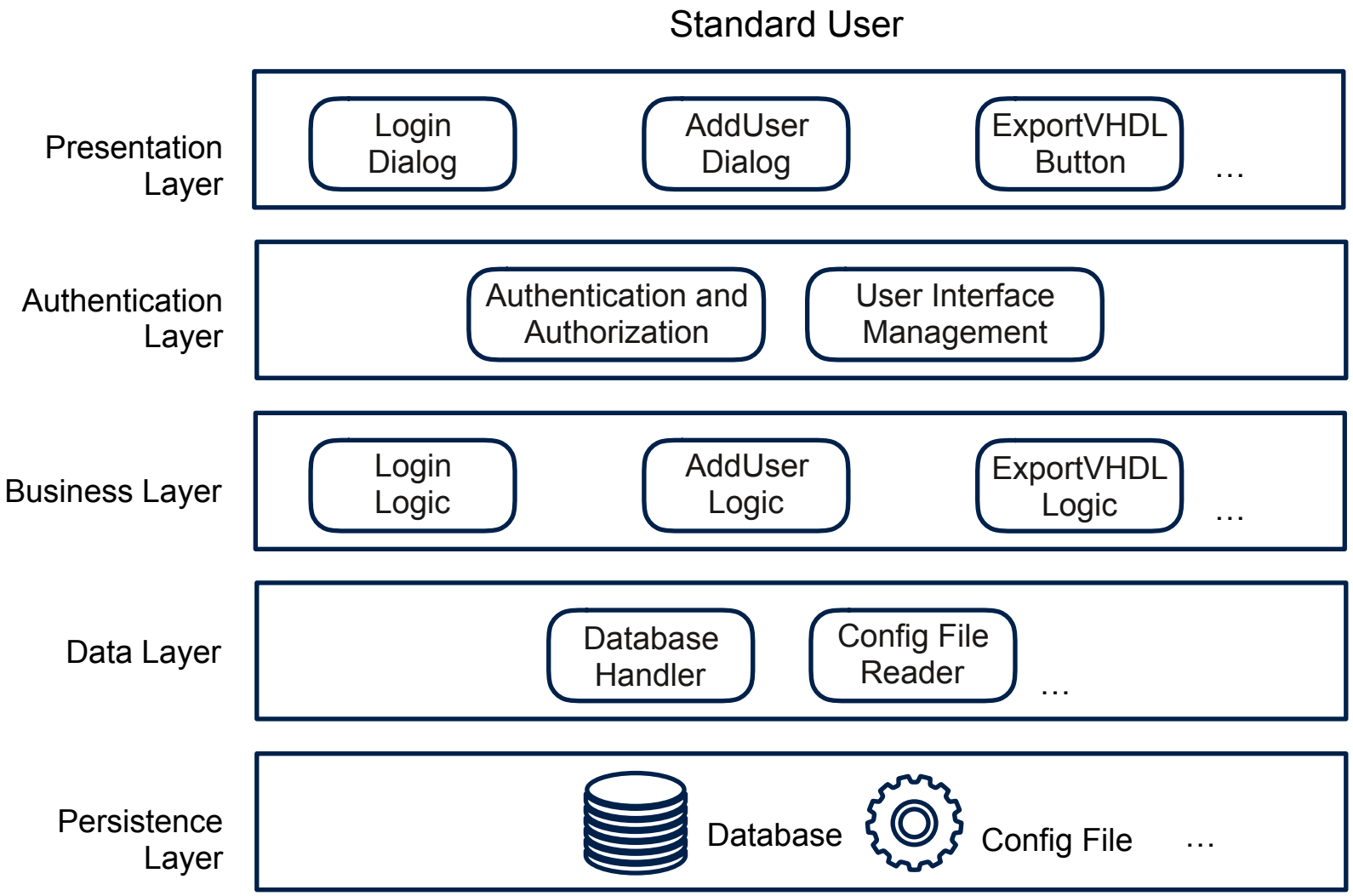
- Reliable storage and management of registers definitions and initial values
- Information security (user access control)
- Reliable export functions
- A friendly graphical user interface (GUI)

Software Design

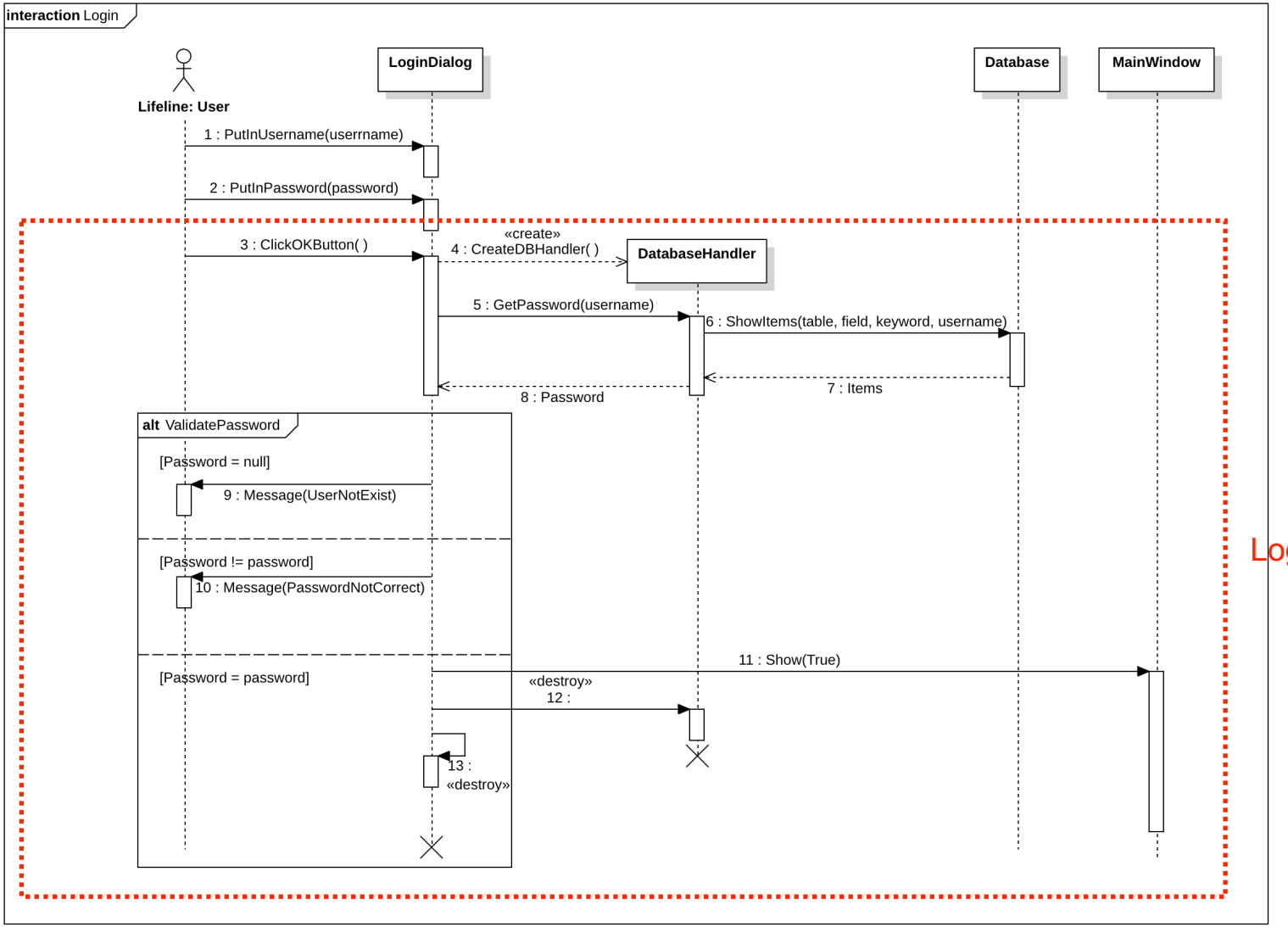
Event-Driven Application Design



Architecture and Module Design



Workflow of the Layered Architecture



Advantages of the Architecture

- Each layer is clearly defined
- Layers are ideally only dependent of neighboring layers
- Modules are well encapsulated and the boundaries are clear

Selection of Infrastructure

Selection of the GUI Framework

Toolkit Name	Cross-platform	Programming Language	Specialized IDE	Licence
Qt	✓	Multiple	✓	LGPL
MFC	✗	C++	✓	Proprietary
GTK	✓	Multiple	✗	LGPL
Swing	✓	Java	✓	-
Tk	✓	Multiple	✗	BSD
wxWidgets	✓	Multiple	✗	WxWindows license

- Qt is a preferred choice

Why Qt?

- Qt Designer - specialized IDE for Qt, with which we can design GUI in a WYSIWYG manner
- Cross-platform
- Free software with a very loose license
- Easy to learn and to use
  - Well organized documentation and a strong developer community
  - Concise, elegant and user friendly APIs

Requirements for Data Storage

- Database shall be accessible to all users (on different local machines)
- Database shall be concurrently read or modified by multiple users
- Data shall be reliably stored
- Data access should be efficient
- Data should be well organized

A well-designed relational database can fulfill these requirements

Selection of the Database Management System

DBMS Name	Cross-platform	License	Client/Server	Comment
MySQL	✓	GPL	✓	Most widely used open-source database system
Microsoft Access	✗	Proprietary	✗	Expensive
SQLite	✓	Public domain	✗	Limited functionalities
Microsoft SQL Server	✗	Proprietary	✓	Expensive
Oracle	✓	Proprietary	✓	Expensive
MariaDB	✓	GPL	✓	Compatible to MySQL

- MySQL or MariaDB are the best choice

Work Schedule

Work Process

- So far
  - Architecture design
  - Selection of GUI frameworks
  - Selection of database systems
  - GUI prototyping (ongoing)
  - Database design (ongoing)
- To do
  - Implementation of software modules
  - Testing and trial run
  - Documentation of the software
  - Thesis writing

Time Table

Timeline	Work
31.03	Requirement engineering; Database structure design finalization
30.04	Presentation and authentication layers finalization; Part of modules on the business layer
31.05	Implementation of LaTex, VHDL and other formats export
30.06	Intensive test and verification; implementation of additional functionalities
31.07	Trial run; documentation
31.08	Thesis writing
30.09	Finalization

Thank you