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

Zhihong Lei

Final Master Thesis Presentation

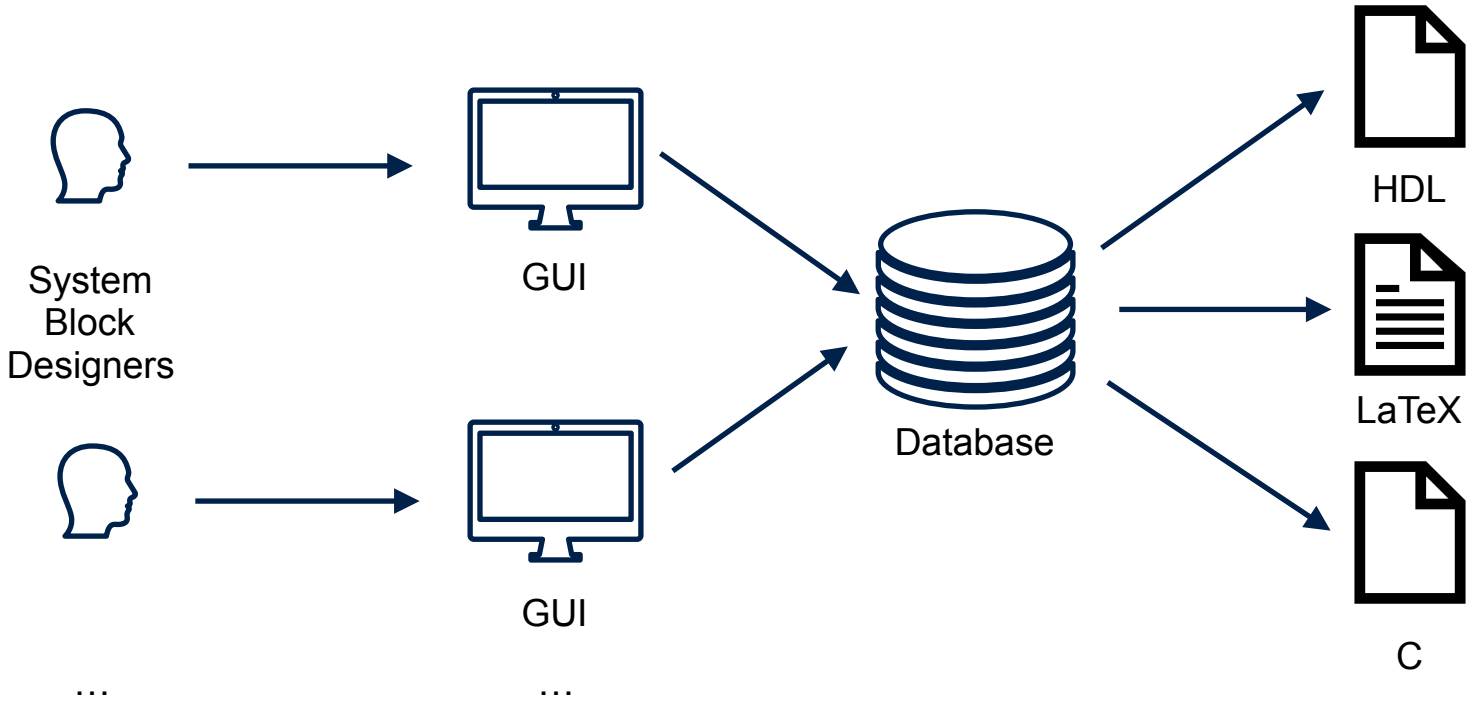
Supervisor: Johannes Bastl, M.Sc

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

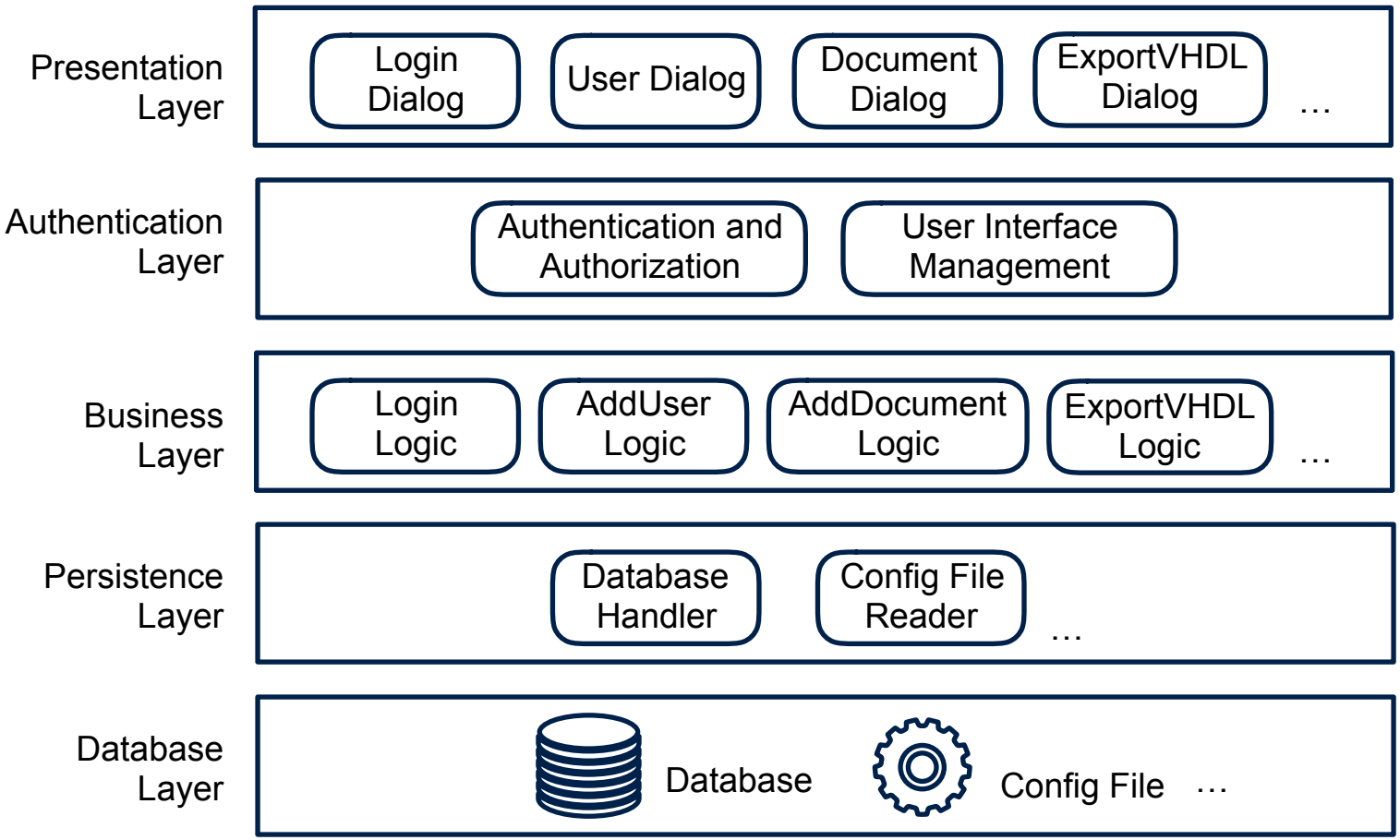
Recap

Proposal

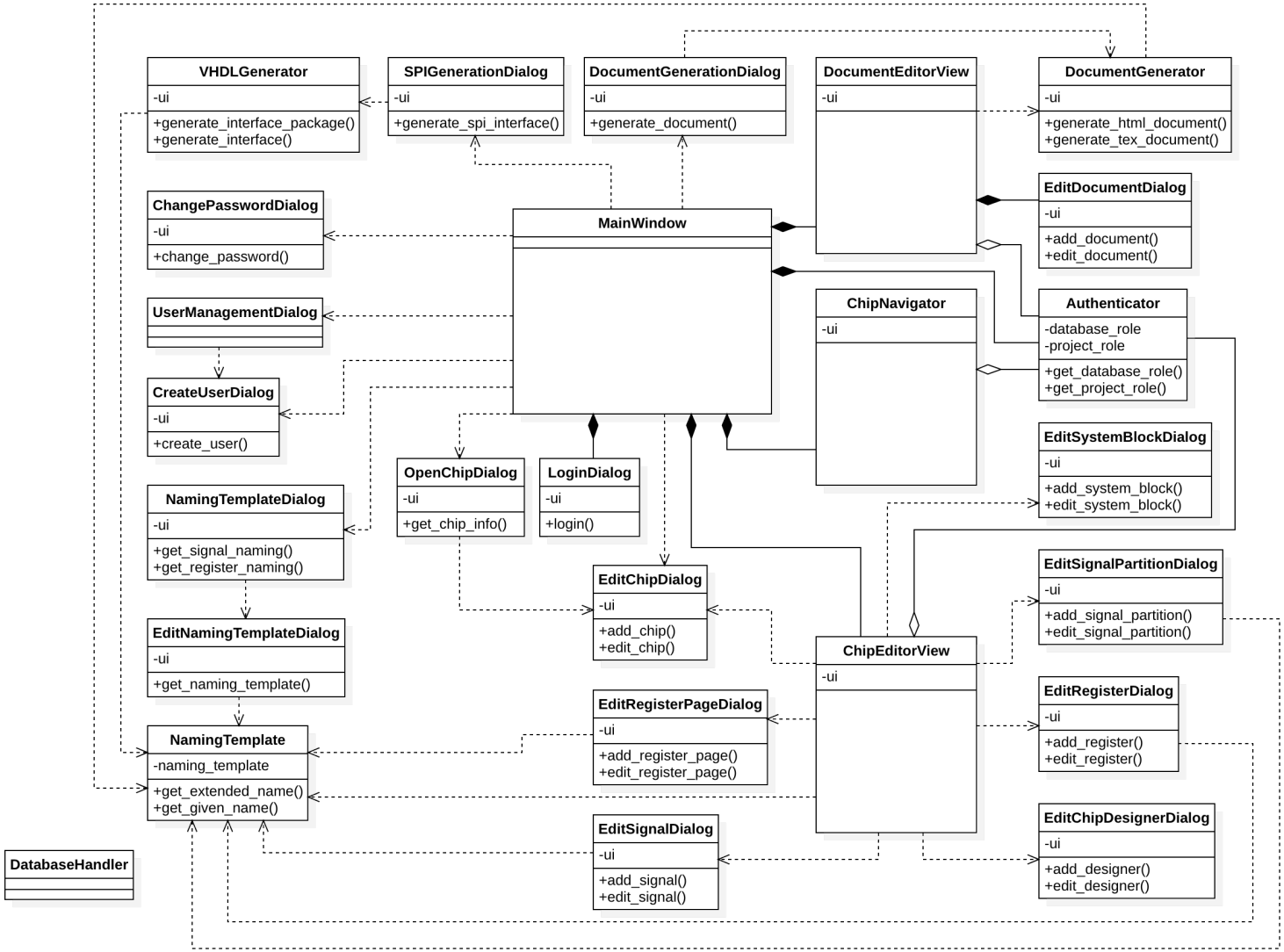
- A database centered software that aids in implementation of digital configuration interface for ASIC chips



Layered Architecture Design

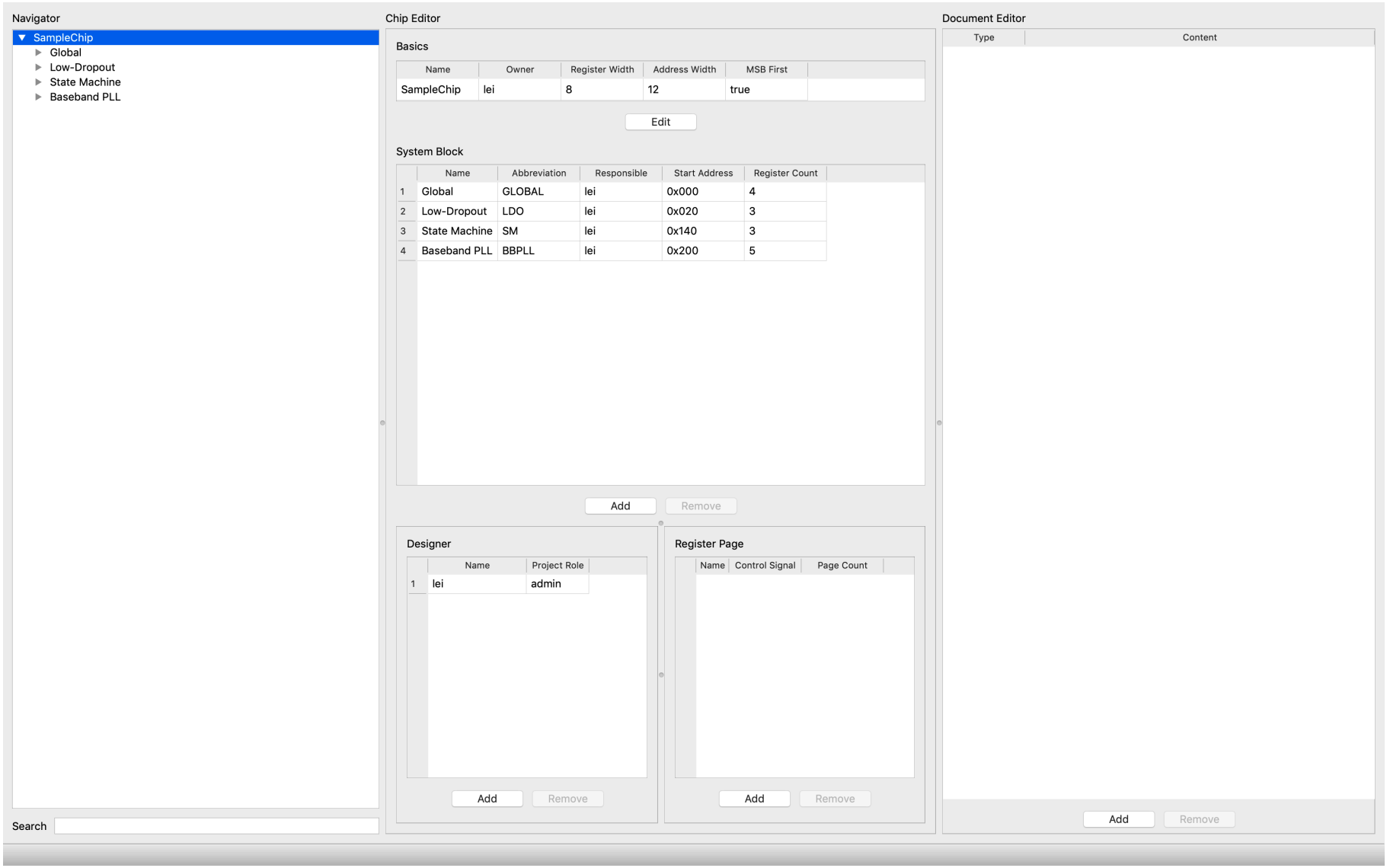


Object-Oriented Software Design



Module Design and Implementation

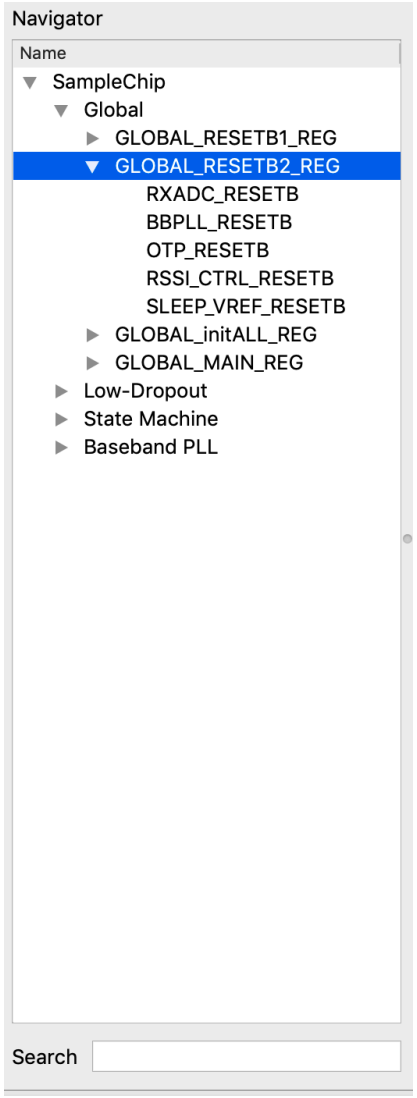
Main Window

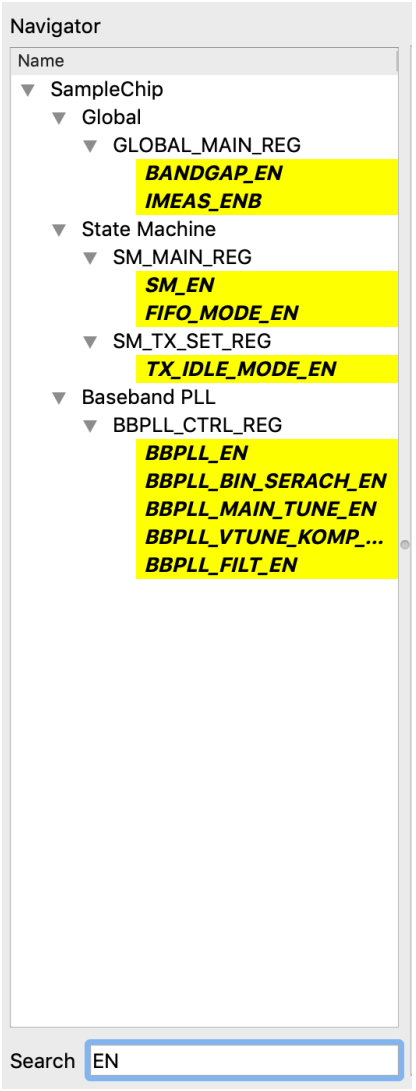

Navigator
SampleChip
Global
Low-Dropout
State Machine
Baseband PLL
Search
Chip Editor
Basics
Name
Owner
Register Width
Address Width
MSB First
SampleChip
lei
8
12
true
Edit
System Block
Name
Abbreviation
Responsible
Start Address
Register Count
1 Global GLOBAL lei 0x000 4
2 Low-Dropout LDO lei 0x020 3
3 State Machine SM lei 0x140 3
4 Baseband PLL BBPLL lei 0x200 5
Add
Remove
Designer
Name
Project Role
1 lei admin
Add
Remove
Register Page
Name
Control Signal
Page Count
Add
Remove
Document Editor
Type
Content
Add
Remove

Chip Navigator

4-level tree widget
chip, block, register, signal

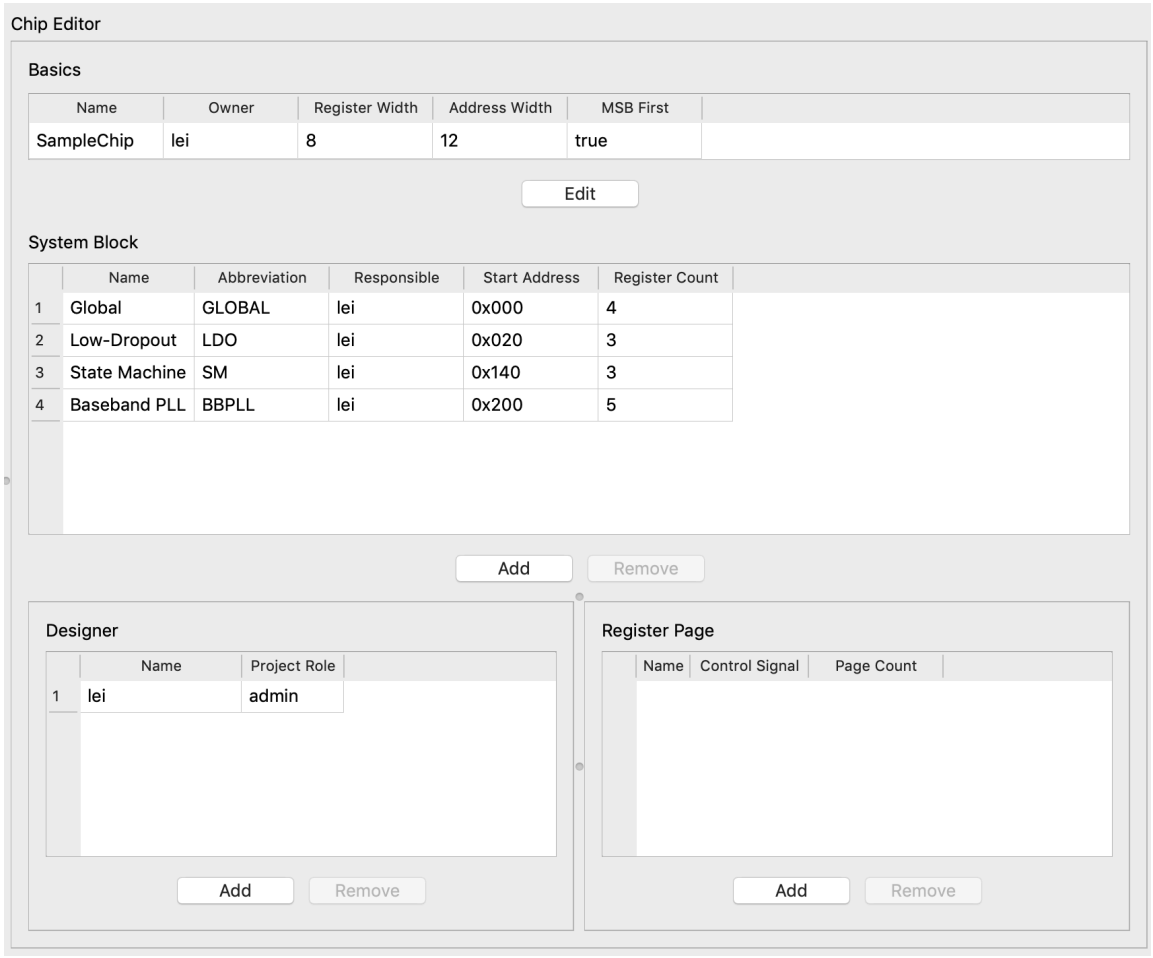
Search area





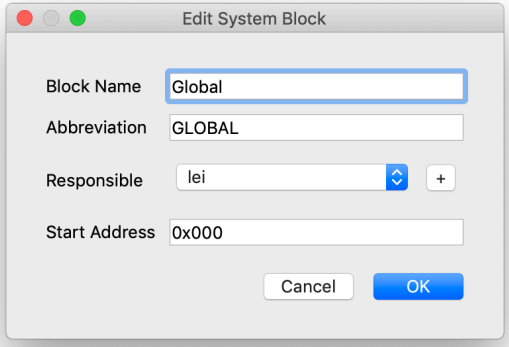
Chip Editor

- Chip-level view



Basic Information about the chip

List of system blocks



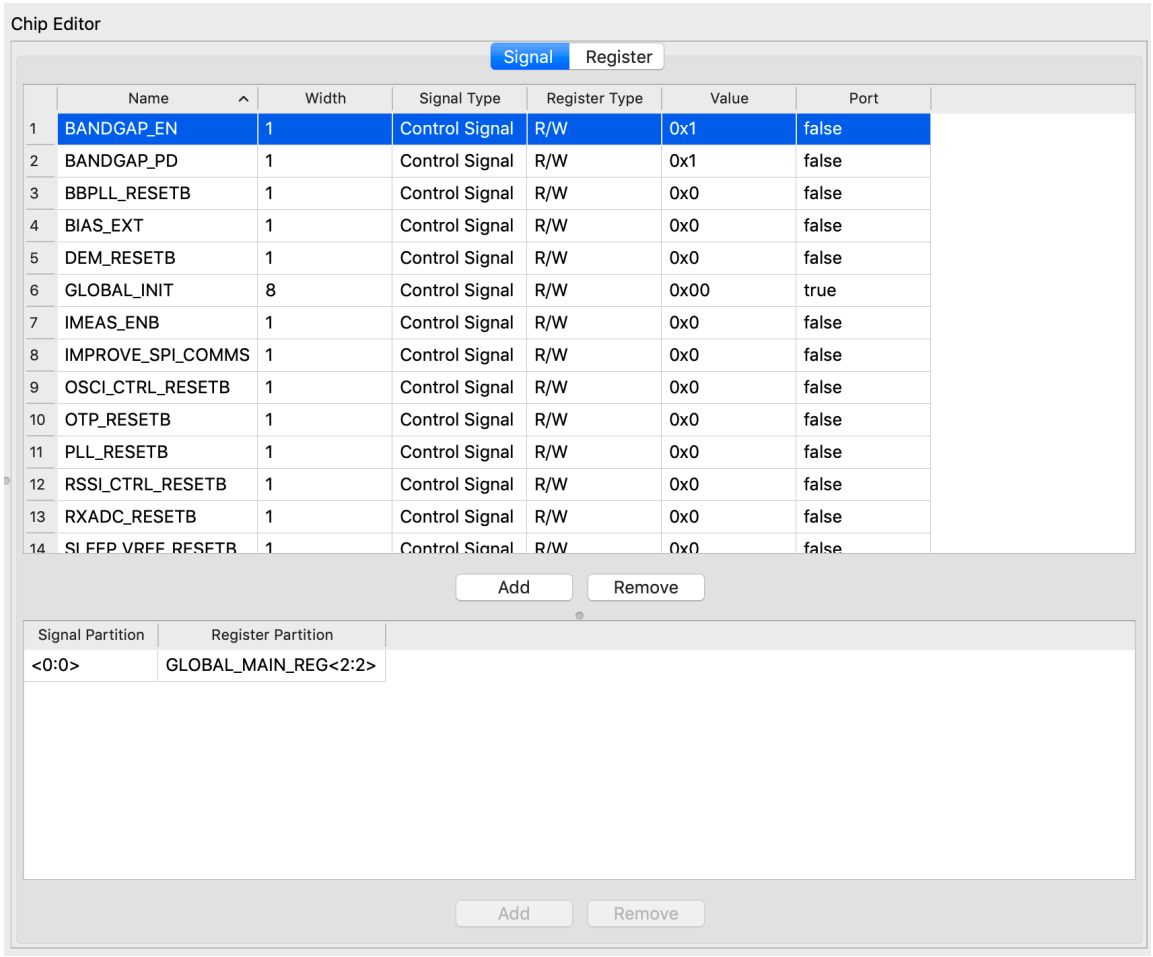
EditSystemBlockDialog

Designers

List of register pages

Chip Editor

- Block-level view - Signal

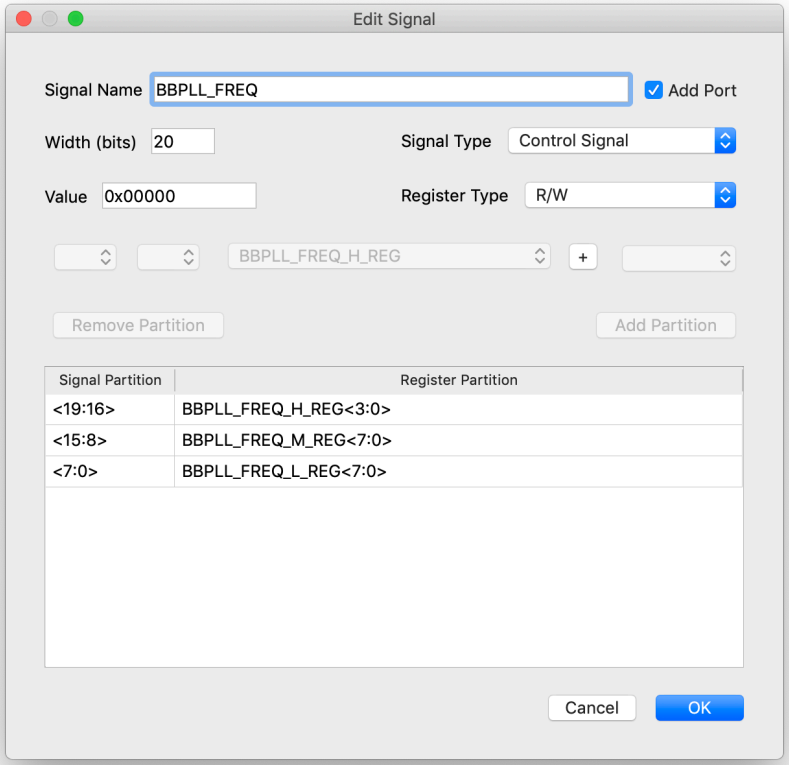


All signals in the current block

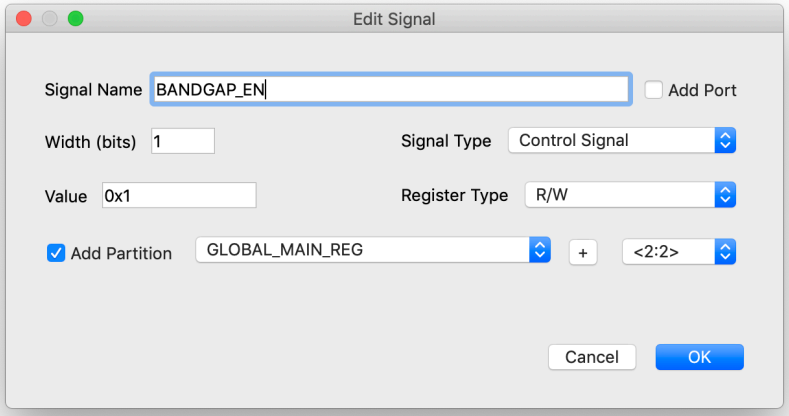
Signal-register mappings

Chip Editor

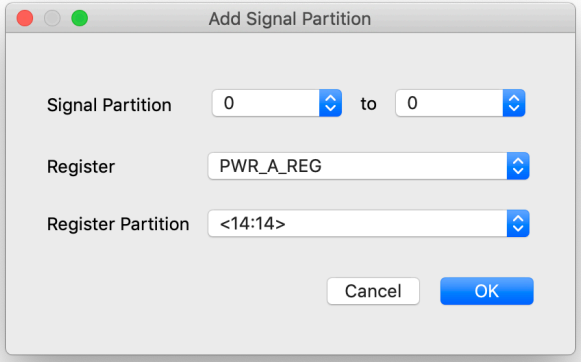
- Block-level view - Signal



EditSignalDialog - Multi-bit



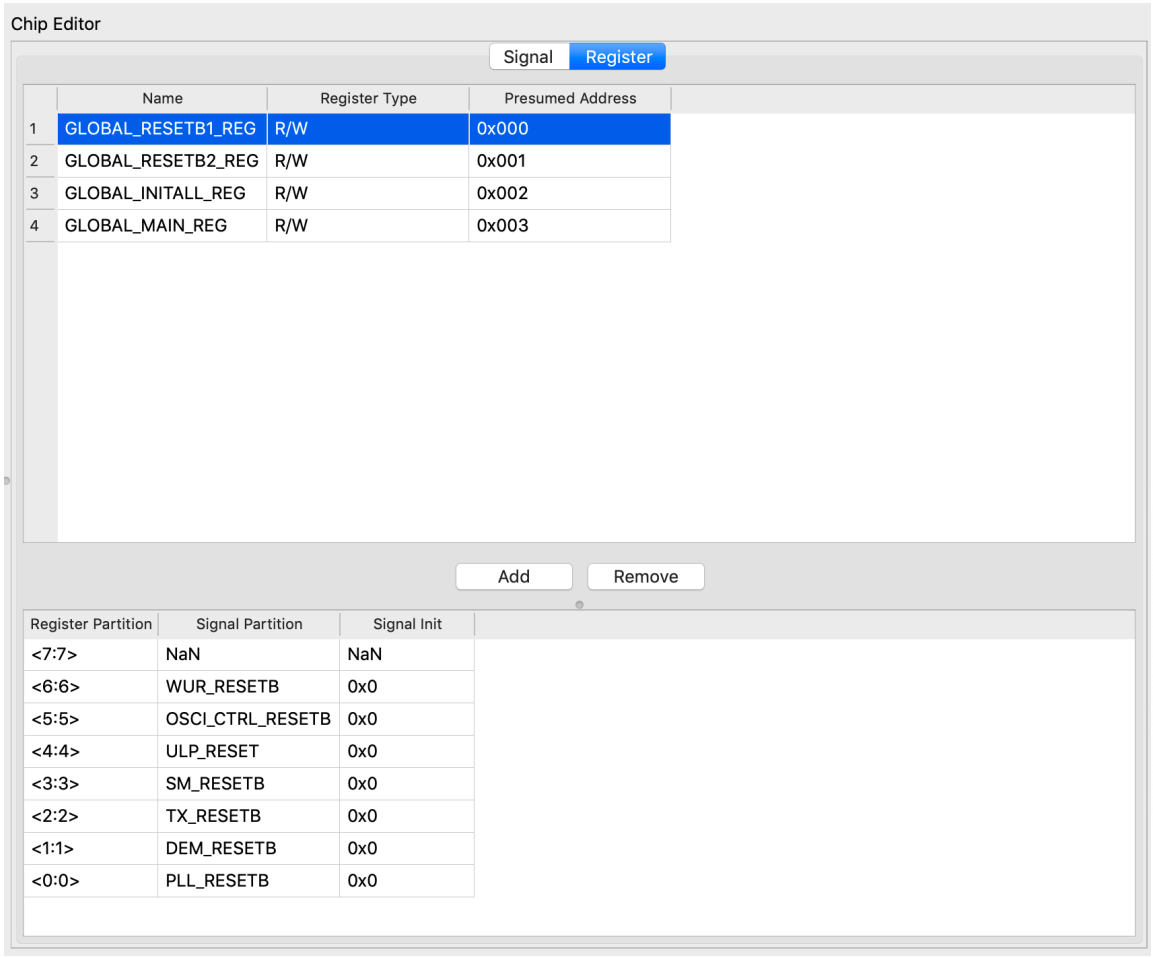
EditSignalDialog - Single-bit



EditSignalPartitionDialog

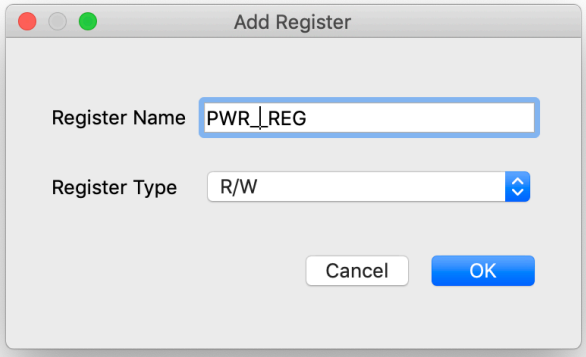
Chip Editor

- Block-level view - Register



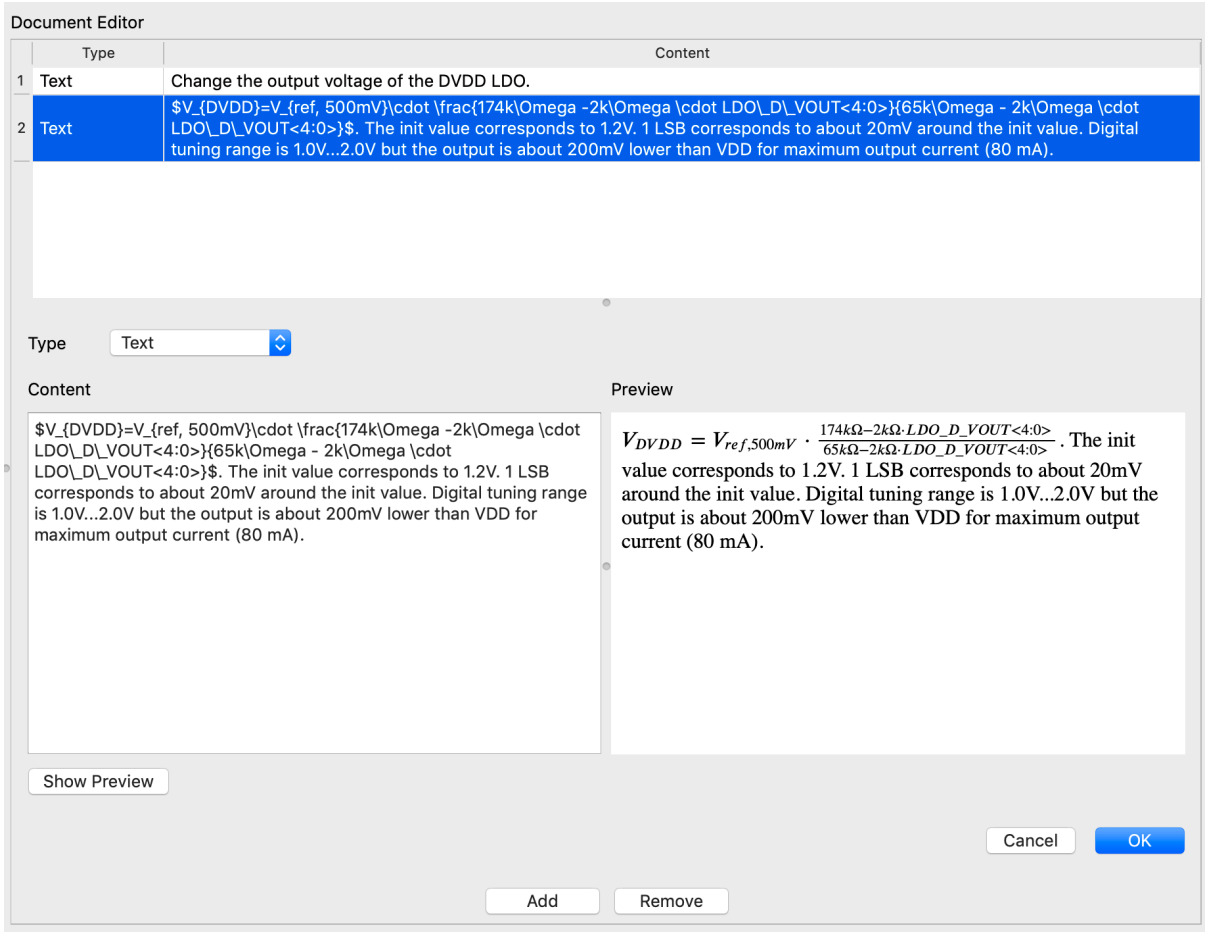
Register-Signal mappings

All registers in the current block



EditRegisterDialog

Document Editor



List of documents of currently selected block/register/signal

Document editing area

Preview: HTML web viewer supporting LaTeX

Document Editor

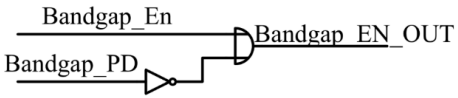
Type: Image

Caption: Bandgap power down mechanism

Path: /Users/zhihonglei/Documents/Thesis/sample_chip/bandgap.png Select

Width: 0.5

Preview



Bandgap power down mechanism

Image

Type: Table

Row: 8 Column: 2

Caption: SM_COMMAND

Table

SM_COMMAND<3:0>	Meaning
000	CMD_NONE
002	CMD_SLEEP
003	CMD_DEEPSLEEP
004	CMD_TX
005	CMD_TXIDLE
006	CMD_RX
007	CMD_RXHOLD

Preview

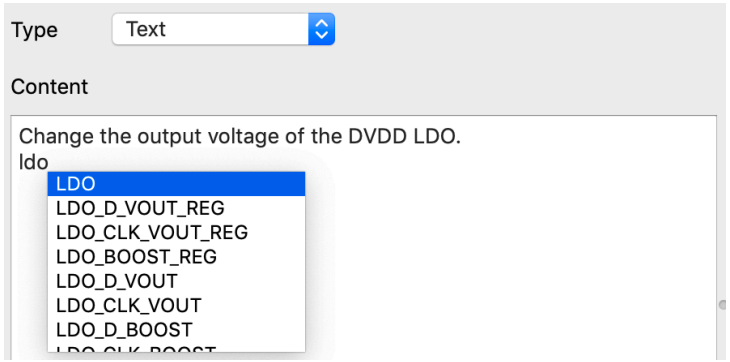
SM_COMMAND

SM_COMMAND<3:0>	Meaning
000	CMD_NONE
002	CMD_SLEEP
003	CMD_DEEPSLEEP
004	CMD_TX
005	CMD_TXIDLE
006	CMD_RX
007	CMD_RXHOLD

Table

Document Editor

- Auto-completion



Text content

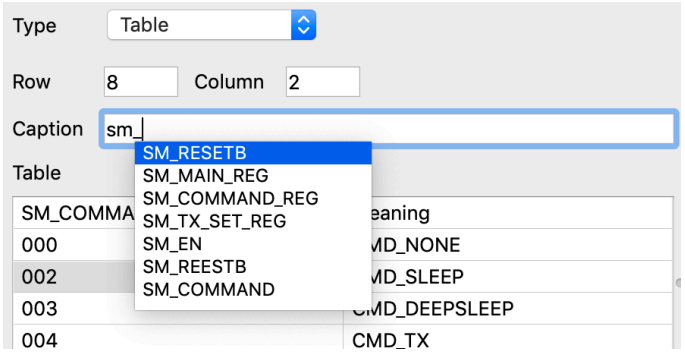


Table & image caption

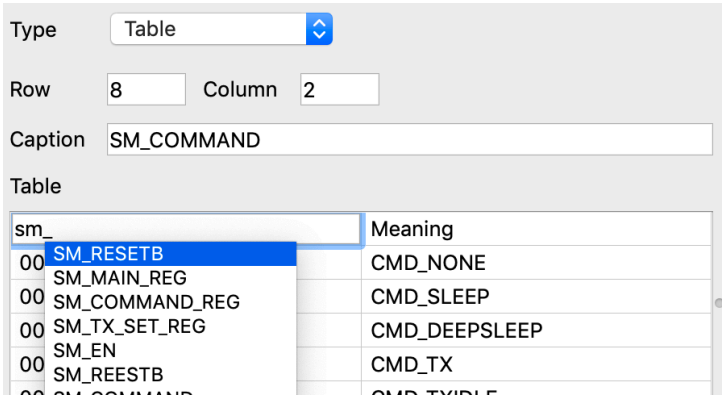
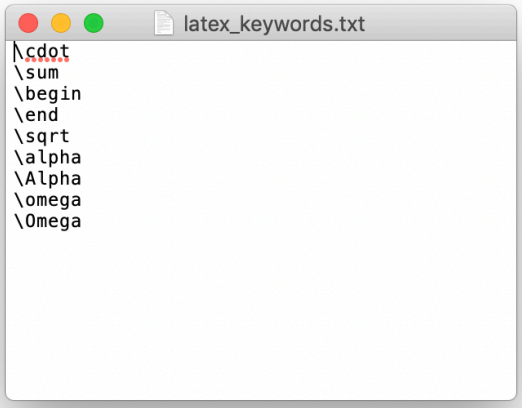


Table cell

Document Editor

- Auto-completion - candidate words
  - Builtin
    - Chip name
    - Block names and abbreviations
    - Register names
    - Signal names
  - User defined
    - ${root}/completion/*.txt



Example word list

Accounts and Passwords

- Database account: shared to all software users

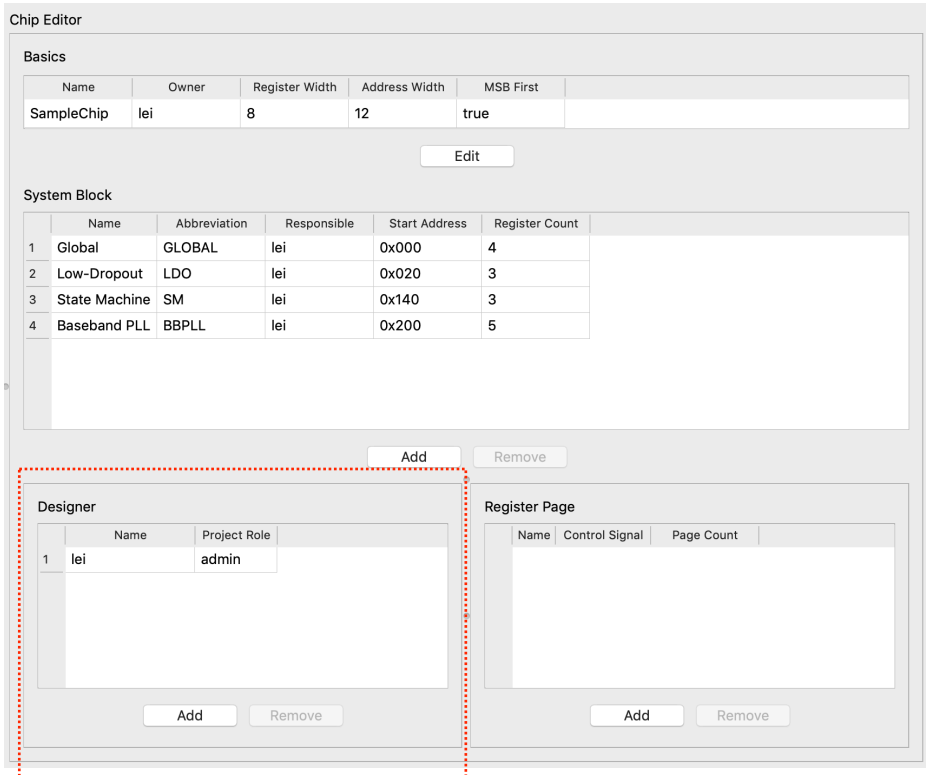
```
# ${root}/global_settings.ini
[database]
database=ias
encrypted_password="TvCYyAGe6hcIyj4ftIg31w=="
host=localhost
key=vFCI399bBoVpu7o89rGbQ/TYyJjOG8ft
port=3306
username=root
```

- Software account

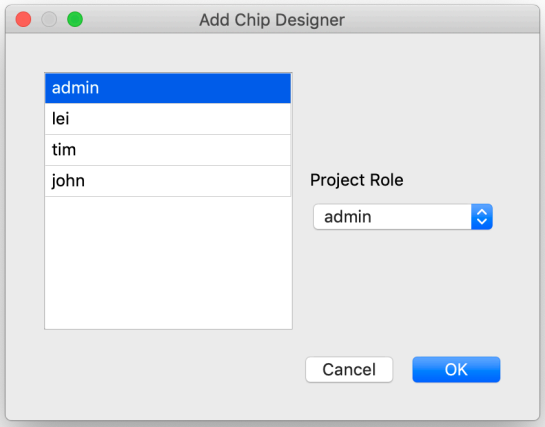
LoginDialog

Accounts and Passwords

- Chip designer account



List of Chip Designers



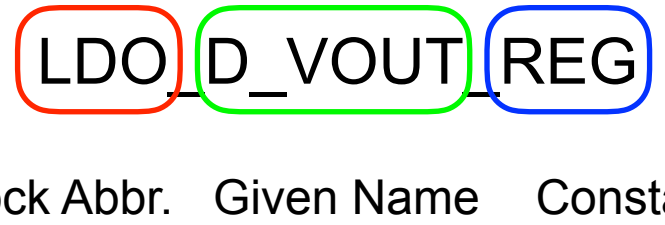
EditChipDesignerDialog

Permissions and Authentication

- Software/database permissions: software users
  - Add or remove a user account
  - Add or remove a chip project
- Project permissions: chip designers
  - Add system blocks
  - Remove system blocks the user is responsible for
  - Read all system blocks
  - Add or remove chip designers
  - Edit system blocks the user is responsible for (implicitly true for all chip designers)
- Authenticator class

Signal and Register Naming

- Signals and registers may be named after a certain pattern

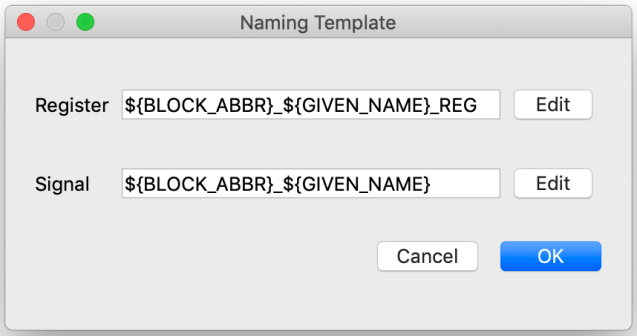


- Question: what if block abbreviations change?
- Solution
  - Only store the given names in the database
  - Define naming templates for signals and registers

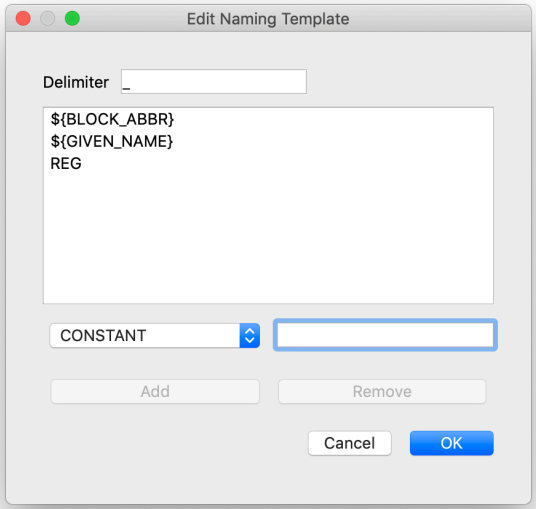
    Example: **${BLOCK_ABBR}_${GIVEN_NAME}_REG**
  - Recover the extended name

Signal and Register Naming

- NamingTemplate class
  - naming_template: string
  - get_extended_name(given_name: string) -> string
  - get_given_name(extended_name: string) -> string
- Naming templates are easily adjustable

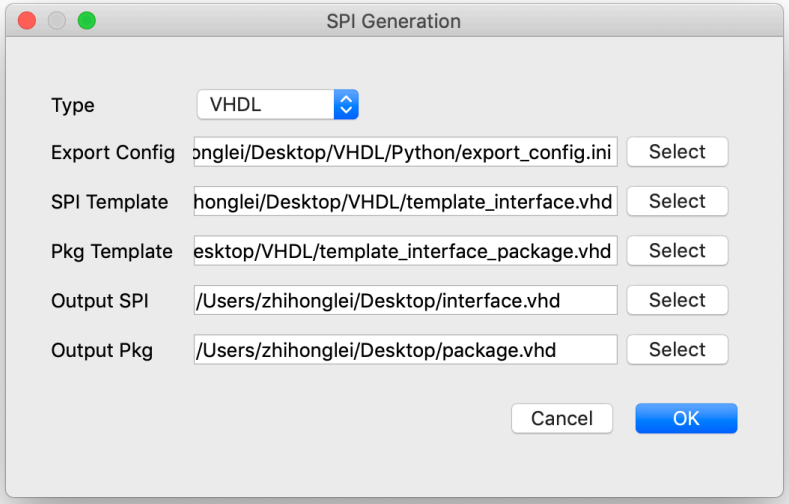


NamingTemplateDialog

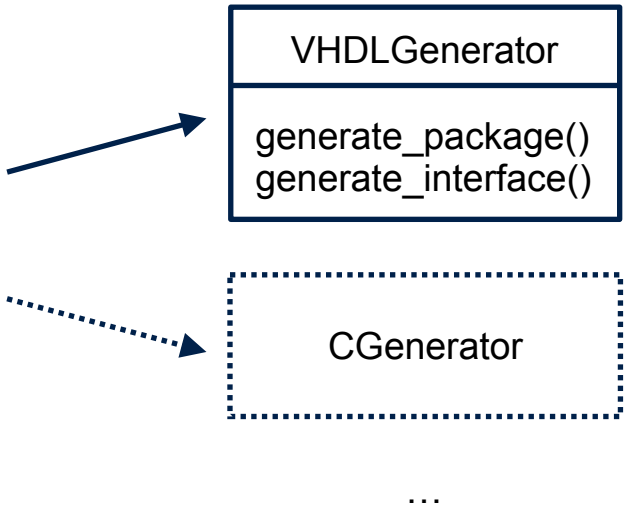


EditNamingTemplateDialog

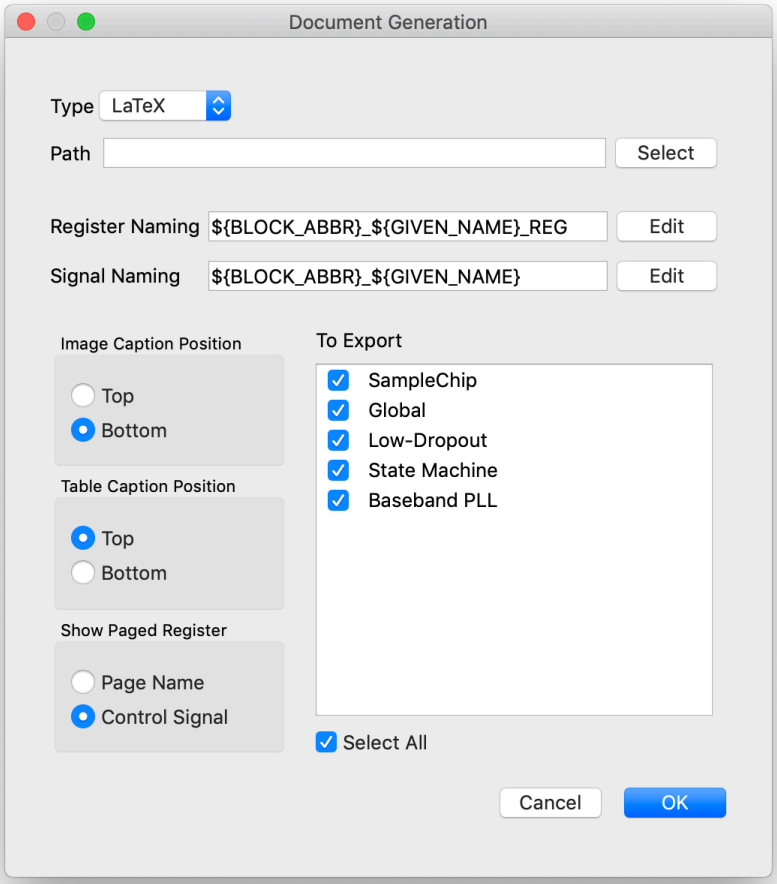
SPI Interface Generation and Export



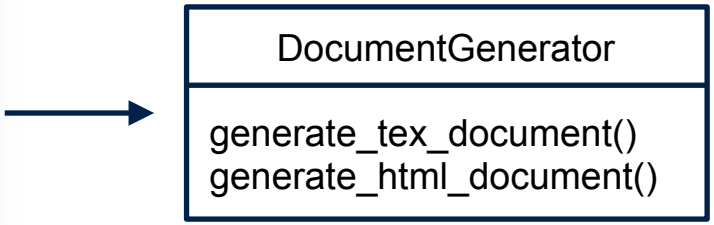
SPIGenerationDialog



Document Generation and Export



DocumentGenerationDialog



Documentation Preview

Navigator

- SampleChip
  - Global
  - Low-Dropout
  - State Machine
  - Baseband PLL

Search

Document Preview

SampleChip

1. Global
   - GLOBAL_RESETB1_REG
   - GLOBAL_RESETB2_REG
   - GLOBAL_INITALL_REG
   - GLOBAL_MAIN_REG
2. Low-Dropout
   - LDO_D_VOUT_REG
   - LDO_CLK_VOUT_REG
   - LDO_BOOST_REG
3. State Machine
   - SM_MAIN_REG
   - SM_COMMAND_REG
   - SM_TX_SET_REG
4. Baseband PLL
   - BBPLL_CTRL_REG
   - BBPLL_FREQ_H_REG
   - BBPLL_FREQ_M_REG
   - BBPLL_FREQ_L_REG
   - BBPLL_POSTDIV_OSCICTRL_REG

Global

GLOBAL_RESETB1_REG - 0x000

7	6	5	4	3	2	1	0
...	WUR_RESETB	OSCI_CTRL_RESETB	ULP_RESET	SM_RESETB	TX_RESETB	DEM_RESETB	PLL_RESETB
.	0	0	0	0	0	0	0

- WUR_RESETB is written to WAKEUP<3>
- OSCI_CTRL_RESETB is written to OSCI_CTRL<0>
- ULP_RESET is written to ULP_MAIN<1> and inverted to to ULP_MAIN<4>
- SM_RESETB is written to SM_MAIN<1>
- TX_RESETB is written to TX_MAIN<5>
- DEM_RESETB is written to DEM_MAIN<7>
- PLL_RESETB is written to PLL_MAIN<3>

GLOBAL_RESETB2_REG - 0x001

7	6	5	4	3	2	1	0
...			RXADC_RESETB	BBPLL_RESETB	OTP_RESETB	RSSI_CTRL_RESETB	SLEEP_VREF_RESETB
.	.	.	0	0	0	0	0

- RXADC_RESETB is written to RXADC_SET_REG<7>
- BBPLL_RESETB is written to BBPLL_CTRL_REG<7>
- OTP_RESETB is written to OTP_CONFIG_REG<0>
- RSSI_CTRL_RESETB is written to RSSI_CTRL_REG<3>
- SLEEP_VREF_RESETB is written to SLEEP_VREF_REG<2>

Integration, Testing and Deployment

Integration and Testing

- Module testing: right after implementation
- Integrated testing
  - Beta version delivered to users
  - The software was used on real-world projects
  - Errors and bugs were found and removed
  - Some functionalities were improved

Deployment

- The software depends on external softwares in form of shared libraries
  - Qt modules
  - MySQL C++ Connector
- Problem: users have to manually install all dependencies and link them to the executable
- Deployment
  - Package the executable and all dependencies
  - Relink the executable to these dependencies
  - Deliver the software package to users
- We deployed the software using linuxdeployqt tool

Thank you