2023 DESIGN AND VERIFICATION™ DVCON CONFERENCE AND EXHIBITION EUROPE 10 YEAR ANNIVERSARY

Integration Verification of Safety Components in Automotive Chip Modules

Holger Busch

Infineon Technologies AG

Infineon

accellera SYSTEMS INITIATIVE

Agenda

1. Introduction
2. Formal-Property-Checking Approach
3. Structural Analyses
4. Automatic Integration Checks
5. Experience
6. Summary
7. Questions

Agenda

1. Introduction
2. Formal-Property-Checking Approach
3. Structural Analyses
4. Automatic Integration Checks
5. Experience
6. Summary
7. Questions

Introduction

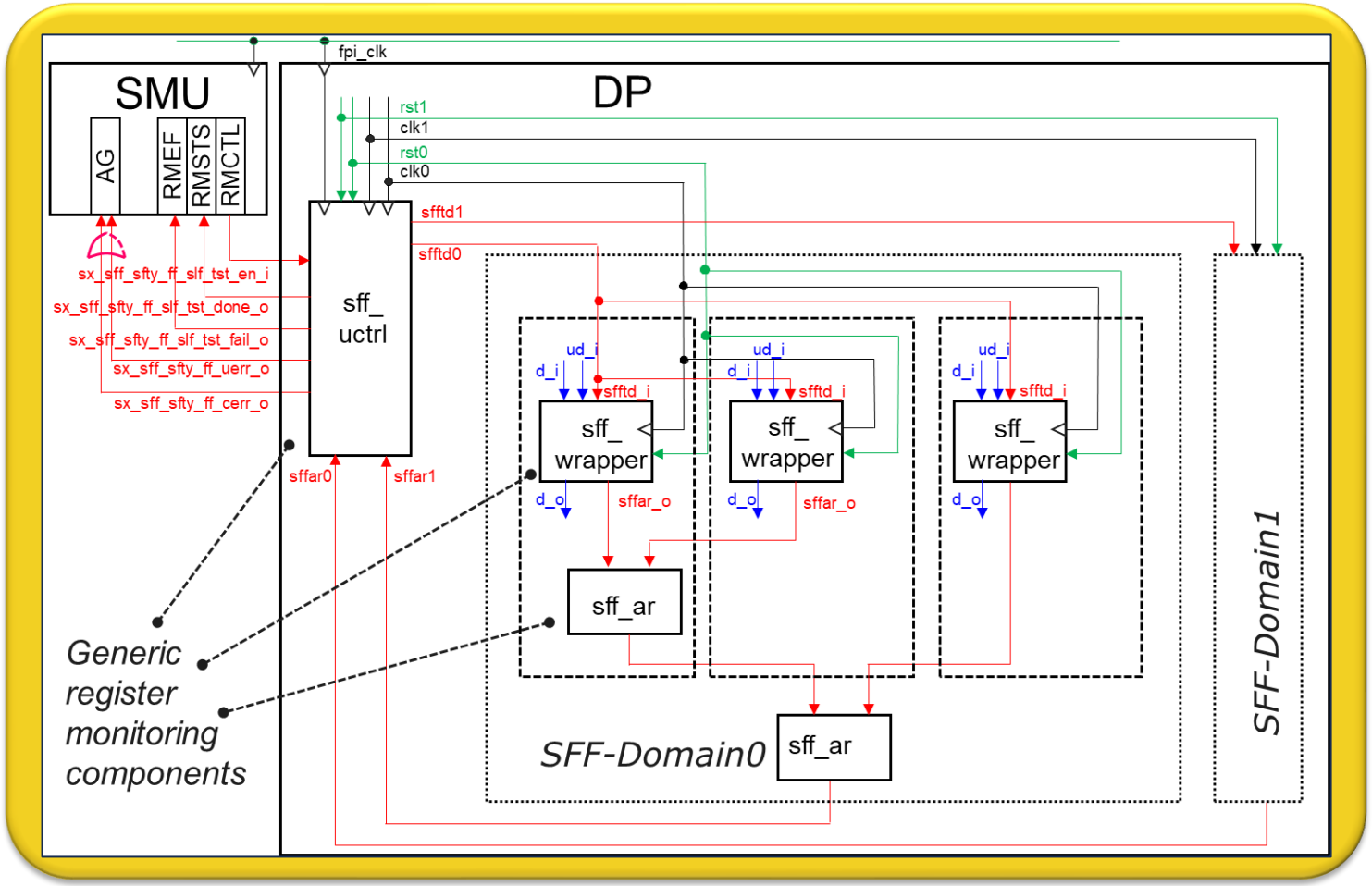
- Safety requirements: prevent failure modes by random faults
  - Error detection: raise error flag
  - Alarm reaction: drive system into safe state
- HW and/or SW safety mechanisms
  - Redundancy + comparison
  - Read-back + comparison
  - Configurable alarm reaction
- Safety verification objective
  - Evidence that safety mechanism yields diagnostic coverage
    Automotive **S**afety **I**ntegrity **L**evel D: detect 99% of all single point faults

Introduction (1)

- Safety verification approaches
  - Fault simulation:
    specialized tool
  - Functional simulation:
    forcing signal to inverted or constant value
  - Formal-Property-Checking:
    formal fault injection by signal cutting + fault assumption

Introduction (2)

- HW-safety mechanism for registers (SFF – Safety-Flip-Flop)
  - Highly configurable safety library components
- Integration in DP (**D**esign **P**art)
  - Replacement of previously unprotected registers by library components and wiring of test and alarm signals
  - Insertion of safety controller and connection to **S**afety-**M**anagement **U**nit

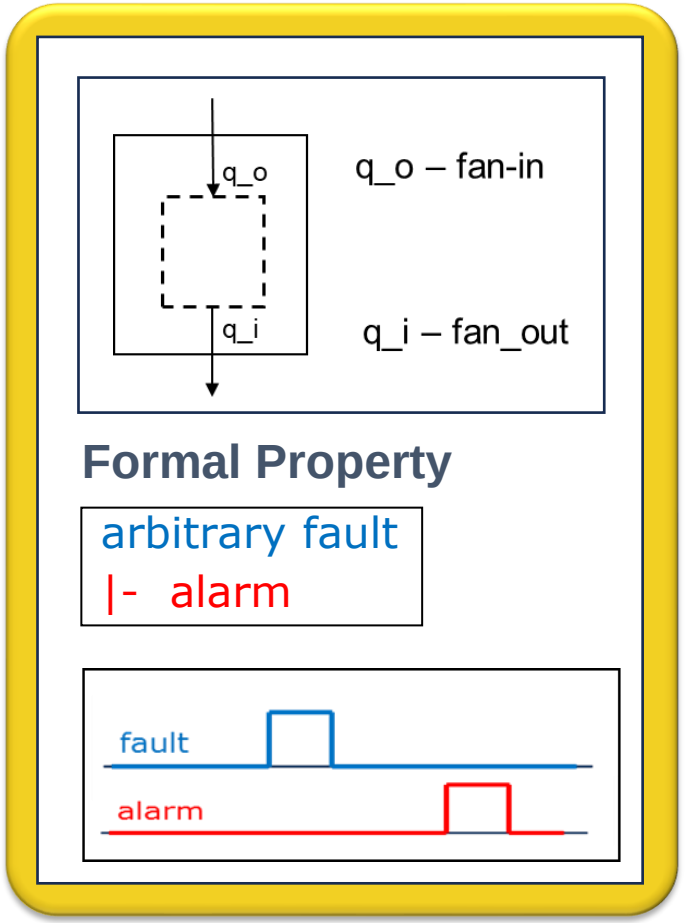


Agenda

1. Introduction
2. Formal-Property-Checking Approach
3. Structural Analyses
4. Automatic Integration Checks
5. Experience
6. Summary
7. Questions

Formal-Property-Checking Approach

- Instrument model: cut signals
- Provide pre-defined formal safety properties:
  - Error implies alarm
  - No error implies no alarm
  - Test activation causes alarm as expected
- Enhance functional properties:
  - Functional deviation implies alarm



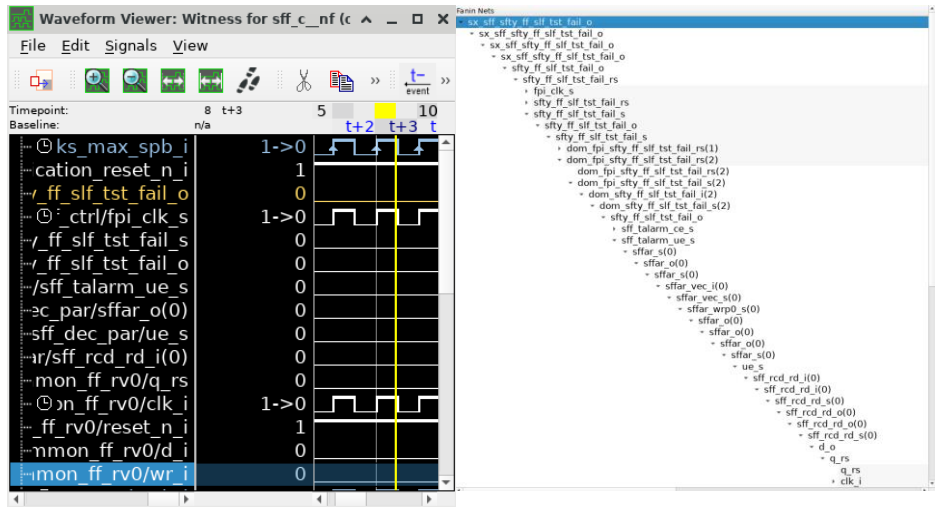
Formal-Property-Checking Approach (1)

- Benefit:
  - Exhaustive coverage of all potential faults
- Bottleneck:
  - Module complexity
  - Debugging of proof failures
    - Design familiarity
    - Tool expertise
- Experience:
  - Library components correct
  - Bugs by wrong integration: structural root cause

Debugging:

Tracing through deep instance-hierarchy

```
prove:
  [+]during [t, te]:
    sx_sff_sfty_ff_slf_tst_fail_o = '0';
```



Formal-Property-Checking Approach (2)

- New Approach:
  1. Exhaustive formal verification of library components in parameterizable test-architecture using actual configurations
  2. Structural integration checks of library components in modules
     - ➢No formal proofs required

Agenda

1. Introduction
2. Formal-Property-Checking Approach
3. Structural Analyses
4. Automatic Integration Checks
5. Experience
6. Summary
7. Questions

Structural Analyses

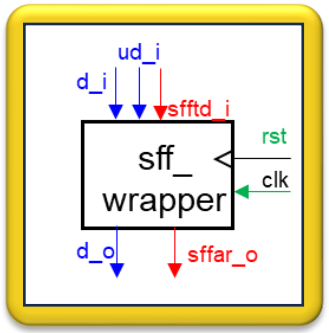
- Clock and reset domains

A	B	C	D	E	F	G
no	**clk**	**par_clkL**	**sel**	**en**	**clk0_diff**	**uctrl_clk_en**
0	clocks__0/clk_gate_2/clk_o	1		clk_gate_en_i(0)	0	-1
1	clocks__0/clk_gate_1/clk_o	2		ks_spb_dft_ctrl_i	2	-1
2	sx_clocks_clks_max_spb_i	-1			2	-1

A	B	C	D	E
no	**rst**	**par_rstL**	**sel**	**en**
0	sx_reset_application_reset_n_i	-1		
1	sx_reset_system_reset_n_i	-1		

Structural Analyses (1)

- Safety registers
  - Protection method
  - Data width
  - Testability
  - Alarm connectivity

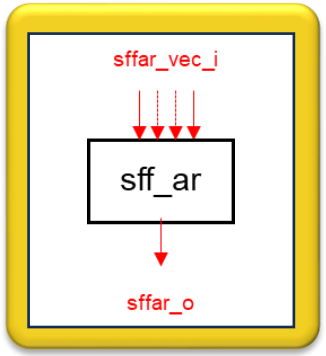


A	B	C	D	F	H
no	signal	wrp_no	type	width	sfr_bf
5	_oven5_s	6	DED	1	OVCENABLE.OVEN5
6	ble_god_s	6	DED	1	OVCENABLE.GOD
7	0_edav_s	7	DED	1	PRDCFG0.EDAV

A	B	C	D	E	F	G	H	I	J	U	V	X
no	wrp	par_comp	clk_no	rst_no	ctrl_fo	ctrl_fi	ar_fo	pmeth_g	sff_ste	dw	rcw	pdw
6	_ovcenable	int_ubs_sfr	0	0	3	3	5	2	true	7	4	7
7	let_prdcfg0	int_ubs_sfr	0	0	3	3	5	2	true	32	6	32

Structural Analyses (2)

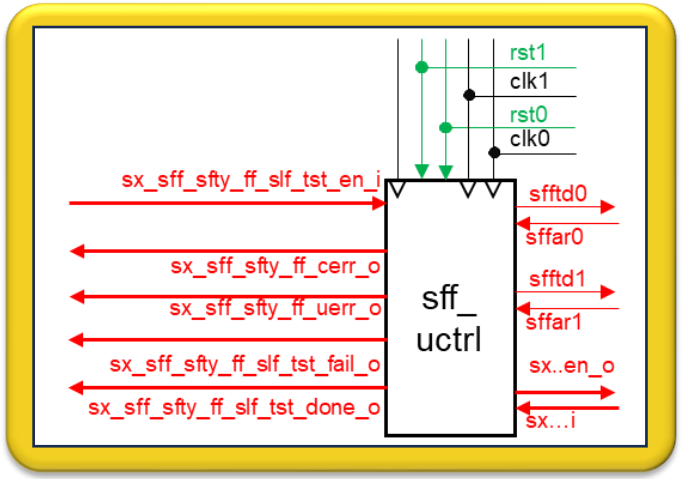
- Alarm reductors
  - Protection method
  - Alarm connectivity



A	B	C	D	E	F
no	**ar**	**par_comp**	**ctrl_fo**	**ar_fo**	**pmeth**
4	r_sfr/sffar1	int_ubs_sfr	2	-1	2
5	r_sfr/sffar2	int_ubs_sfr	3	-1	2

Structural Analyses (3)

- Controller
  - Test support
  - Domains
  - Clock relations
  - Test connectivity
  - Alarm connectivity
  - Interface to SMU

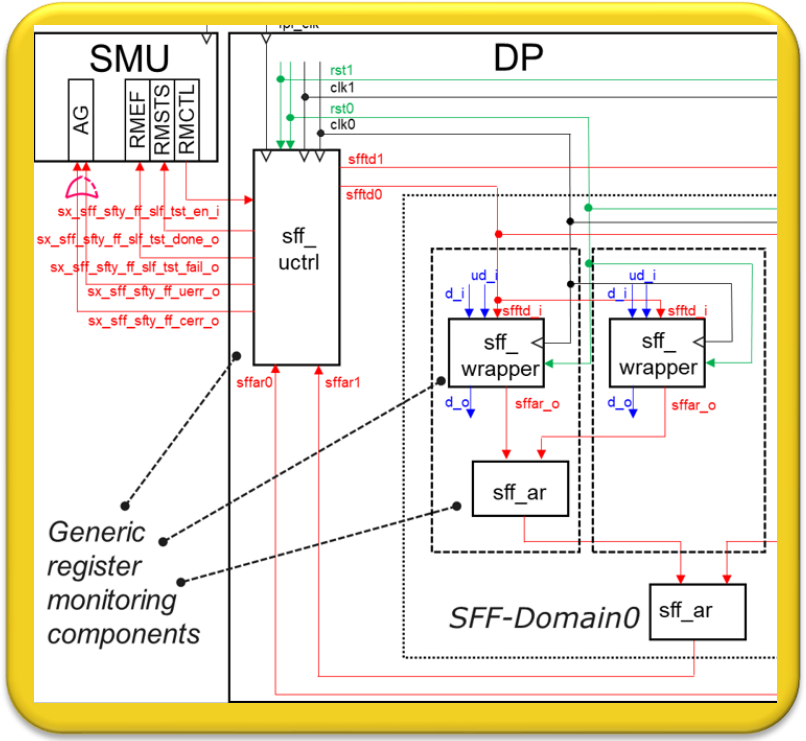


A	B	C	D	E	F	G	H	M
no	**uctrl**	**par_comp**	**clk_no**	**rst_no**	**pmeth_g**	**ste_g**	**top_g**	**clks_diff_g**
0	st_sff_uctrl	pwr_ucw	0	0	2	true	true	8'h0

A	B	C	D	E	F	G	I	J	K
no	**ctrl**	**par_comp**	**clk_no**	**rst_no**	**uctrl_no**	**dom_no**	**clk_diff**	**f_pmeth_g**	**ste_g**
1	dom_ctrl	sff_uctrl	0	0	0	1	0	2	true
2	dom_ctrl	sff_uctrl	0	1	0	2	0	2	true

Structural Analyses(4)

- Connectivity of library components
  - Clock domains
  - Reset domains
  - Test control
  - Alarms
- Data connectivity
  - Localization of registers specified to be protected



➢Base technology

- Efficient functions for transitive fan-in determination

Agenda

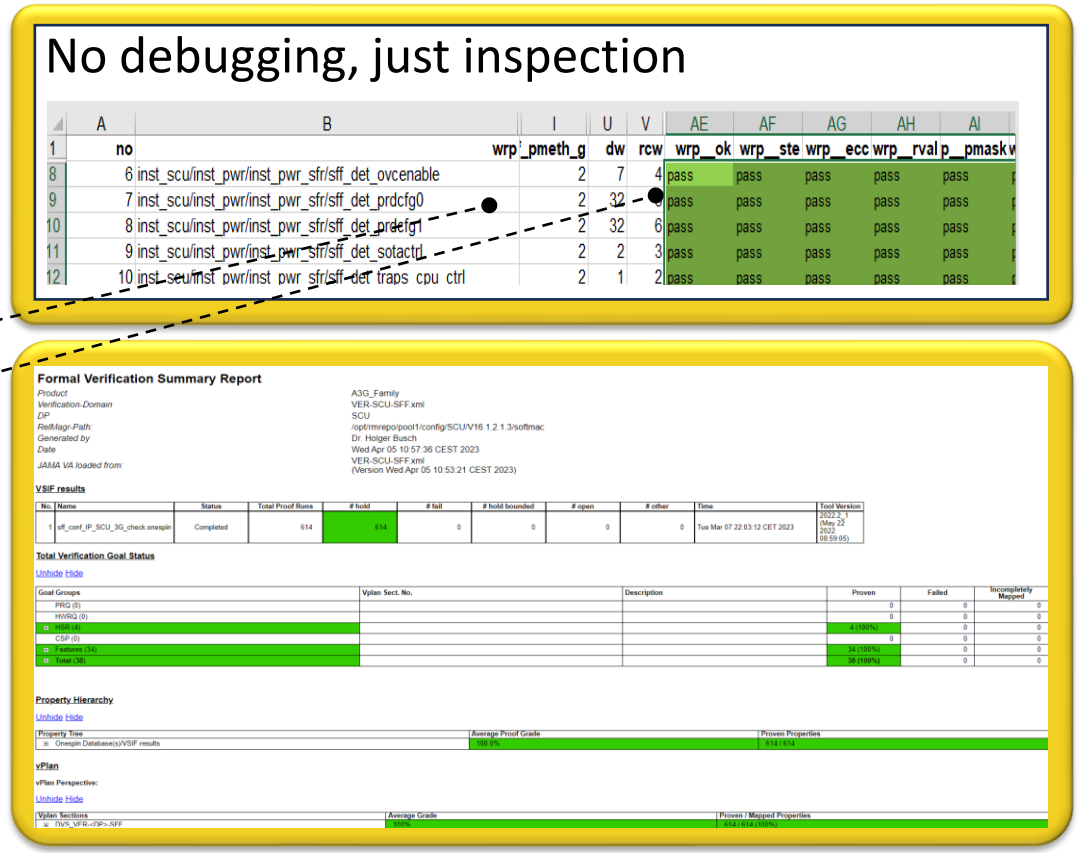
1. Introduction
2. Formal-Property-Checking Approach
3. Structural Analyses
4. Automatic Integration Checks
5. Experience
6. Summary
7. Questions

Automatic Integration Checks

- “Just” evaluation of extracted structural data
- Compatibility of configuration parameters of connected components
  - Protection
  - Self-testability
  - Domain-controllers
  - Synchronization
- Connectivity
  - Clock and reset inputs
  - Test enabling
  - Alarms in domains and global

Automatic Integration Checks (1)

- Automatic integration verification flow
  - Collection of RTL libraries
  - Design compilation
  - Location of library components
  - Extraction SFF-data
- Result tables
  - Extraction data
  - Check results
- Report Generation



Agenda

1. Introduction
2. Formal-Property-Checking Approach
3. Structural Analyses
4. Automatic Integration Checks
5. Experience
6. Summary
7. Questions

Experience

- More than 100 module instances of MC product automatically verified
- Automatic compilation works for almost all modules
  - Manual set-up adjustment in few very specific cases
- Automatic integration checks in few seconds to minutes
- Integration bugs caused by manual wiring or configuration
- In case of findings:
  - No debugging required:
    Result tables with direct references to extraction data
  - Easy correction
  - No additional findings by formal-property checks

Agenda

1. Introduction
2. Formal-Property-Checking Approach
3. Structural Analyses
4. Automatic Integration Checks
5. Experience
6. Summary
7. Questions

Summary

- New integration verification flow:
  1. Exhaustive formal library verification-for reduced test-architectures
  2. Fast automatic structural integration verification for modules
  3. Representative limited checks by sub-system or SoC-simulation
- Benefits
  - Efficiency
  - Ease-of-use
    - No expert knowledge for set-up, execution, root-cause analysis
  - Comprehensiveness
  - Uniformity
  - Substantial efforts saved

Agenda

1. Introduction
2. Formal-Property-Checking Approach
3. Structural Analyses
4. Automatic Integration Checks
5. Experience
6. Summary
7. Questions

Thank You!

Questions ?