Physical Implementation and Analysis of Functional Safety(FuSa) Features in Automotive Chips

Atul Bhattarai
Pavan Kudumula
Shashikiran Srinivasa
Badri Ramasubramanian

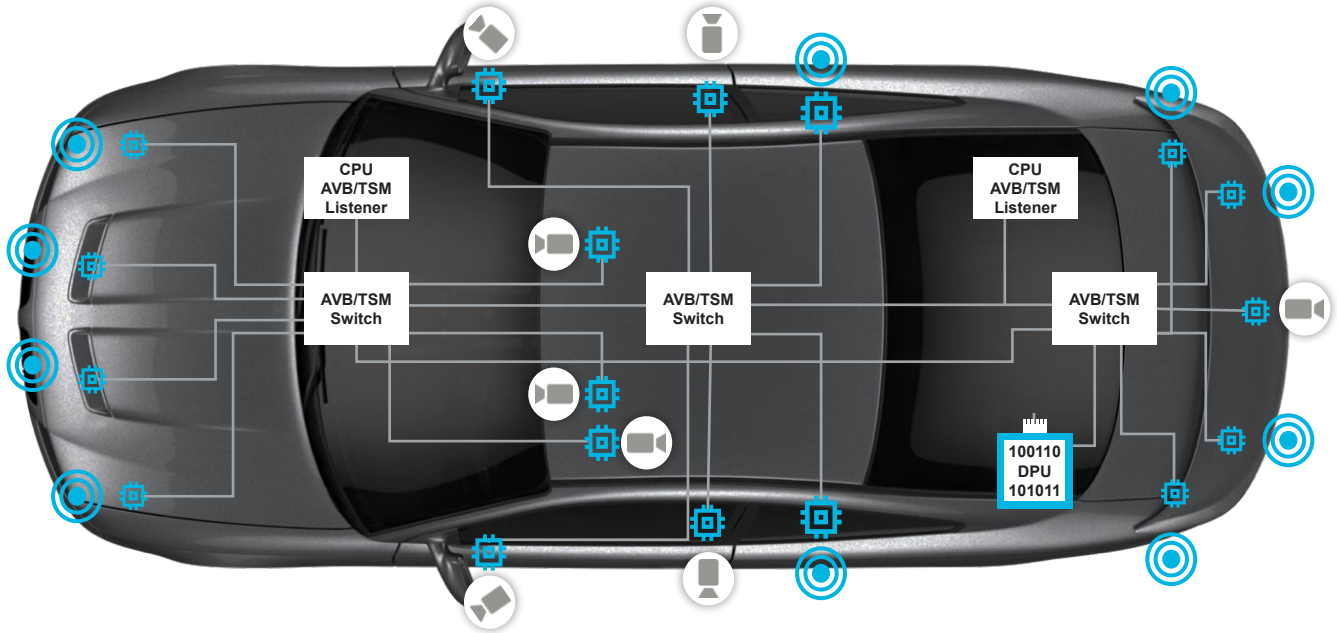
MARVELL™

Agenda

- The safety conundrum
- FuSa schemes
- PD techniques
- Methodology
- Results
- Summary
- Credits
- Q&A

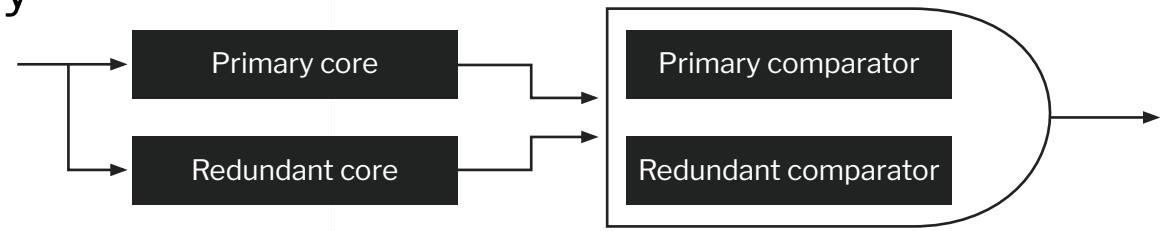
The safety conundrum

- Functional Safety (FuSa)
  - Practice of ensuring systems operate safely
  - Paramount importance in Automotive chips
- ISO26262 Standards
  - Automotive Safety Integrity Level (ASIL)
    - A->D in increasing order of hazard severity
    - Implausibility of failure arguments in reviews
- Physical Design FuSa Goals
  - Physical Independence
    - **Achieve true independence b/w elements**
  - Robustness
    - Consider every PD methodology improvement
    - Remove unnecessary risk to FuSa



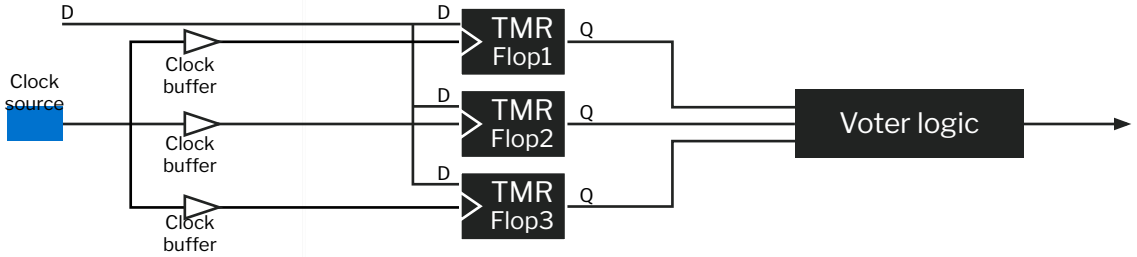
FuSa Schemes

Redundanc
y



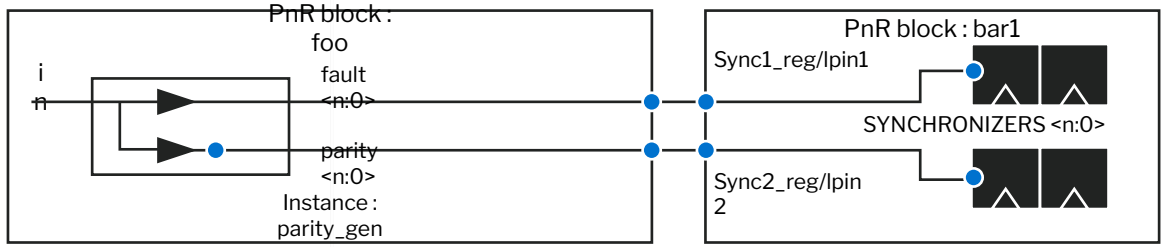
Dual Core Lock Step (DCLS)

IP is instantiated twice in design, each copy processing same set of inputs. Outputs are continuously compared to ensure both the copies are functioning identically



Triple Modular Redundancy (TMR)

Redundant registers with voter logic that provides both error detection and correction



Safety critical signal/Parity Lines

Source IP sends data with relevant inverse parity
Target IP checks valid data is received

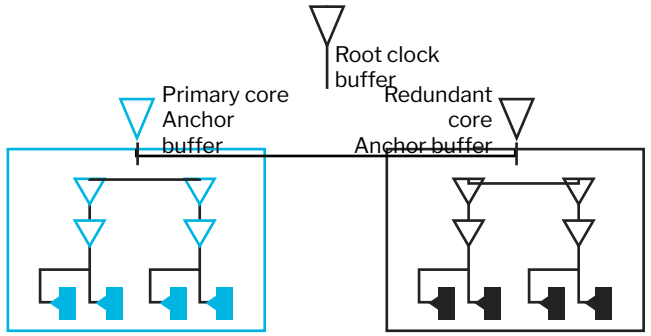
PD techniques

Achieving “true independence” between redundant elements for safety schemes

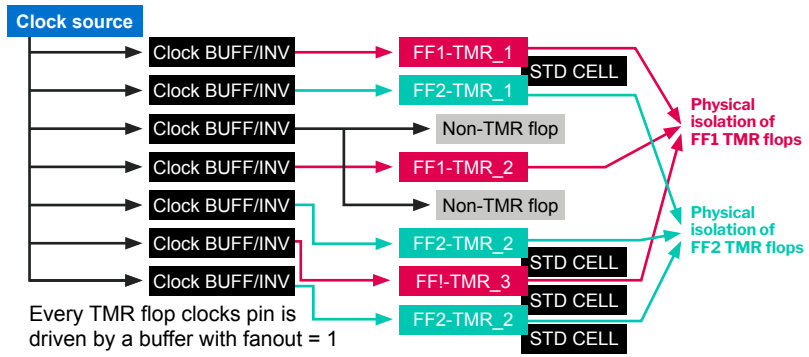
Physical separation
Independence from process variation, PI etc

Clock isolation
Common clock tree reduction

DCLS – Physical and clock isolation



TMR – Physical & clock isolation

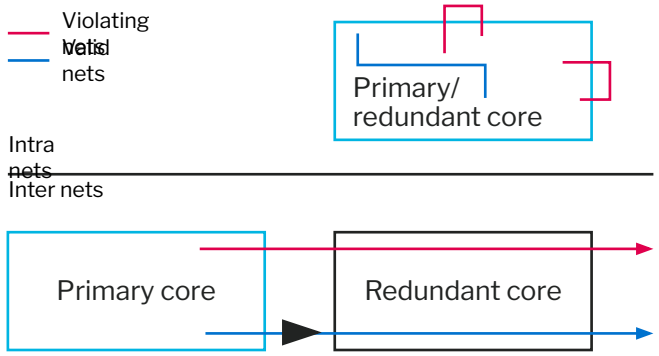


Physical separation
Avoid soft-error Mbit fails

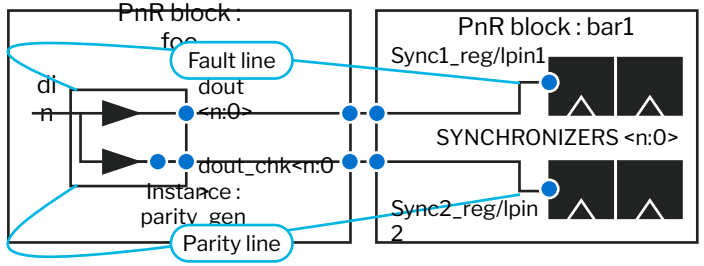
Clock isolation
Leaf level protection

Route isolation
Zero crosstalk interference

DCLS – Route isolation



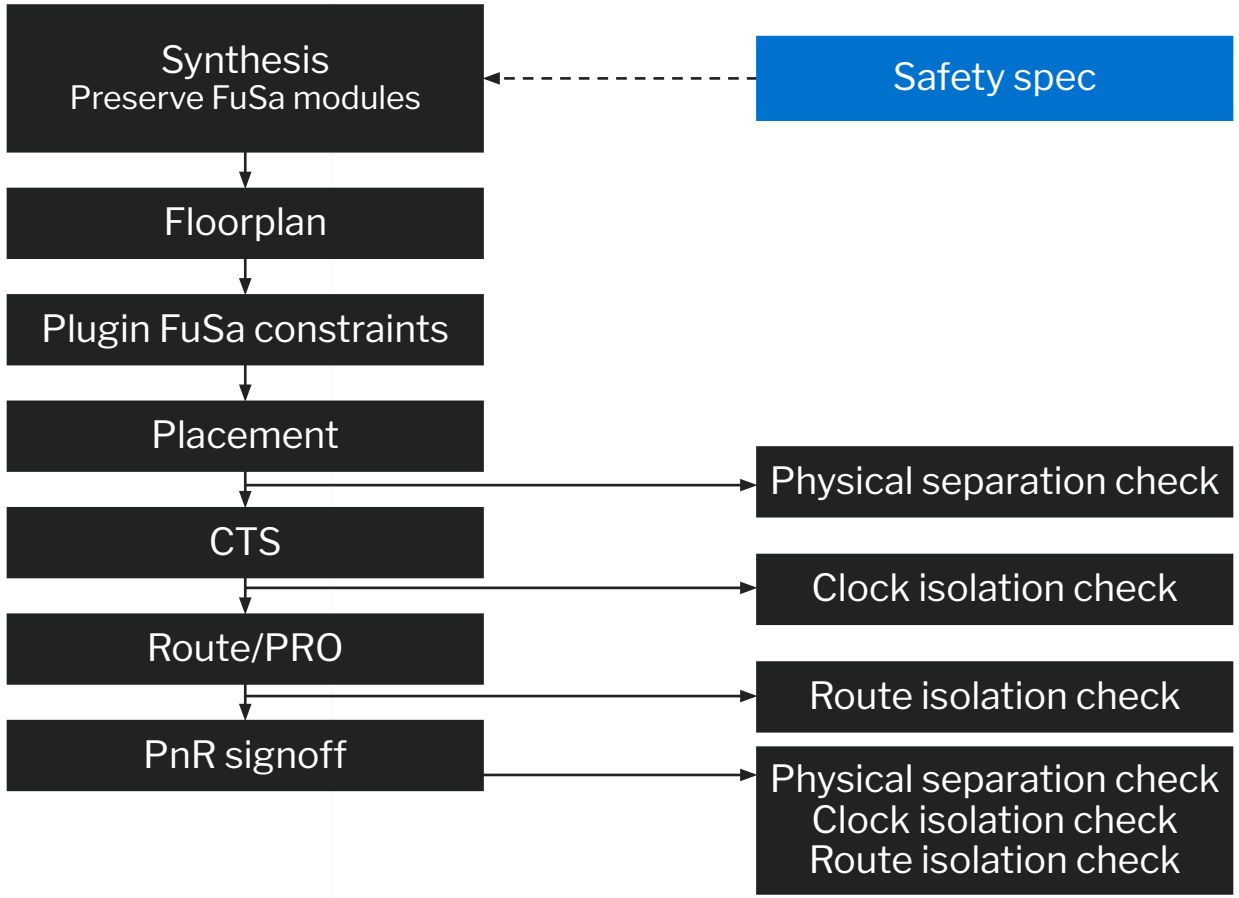
Fault/parity route isolation & delay matching



Safety critical signals
Route isolation
Zero crosstalk interference
Delay Matched Lines
- Same cycle valid data

Methodology

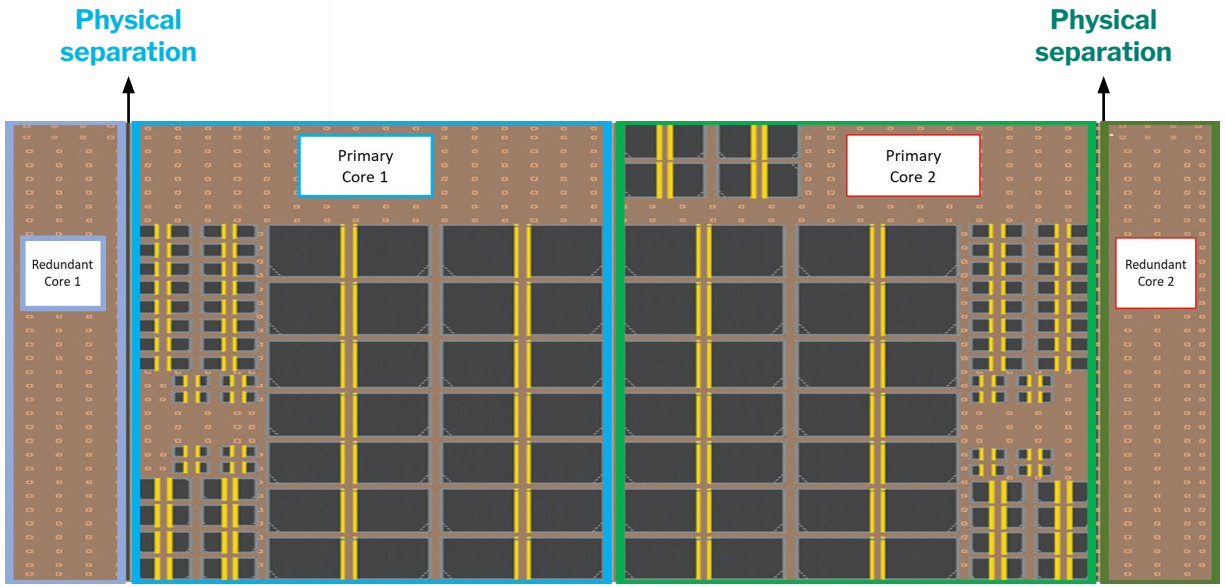
PD FuSa flow



- Preserve hierarchical modules
  - Required for independence and enables calculation of fault metrics
- Execute scripts to perform the implementation
  - Exception from our standard methodology
- After each step, run specific checkers
  - Review violations and apply corrective actions
  - Apply waivers as necessary
- Run all checkers as part of Sign-off
  - Ensure no escapes at signoff

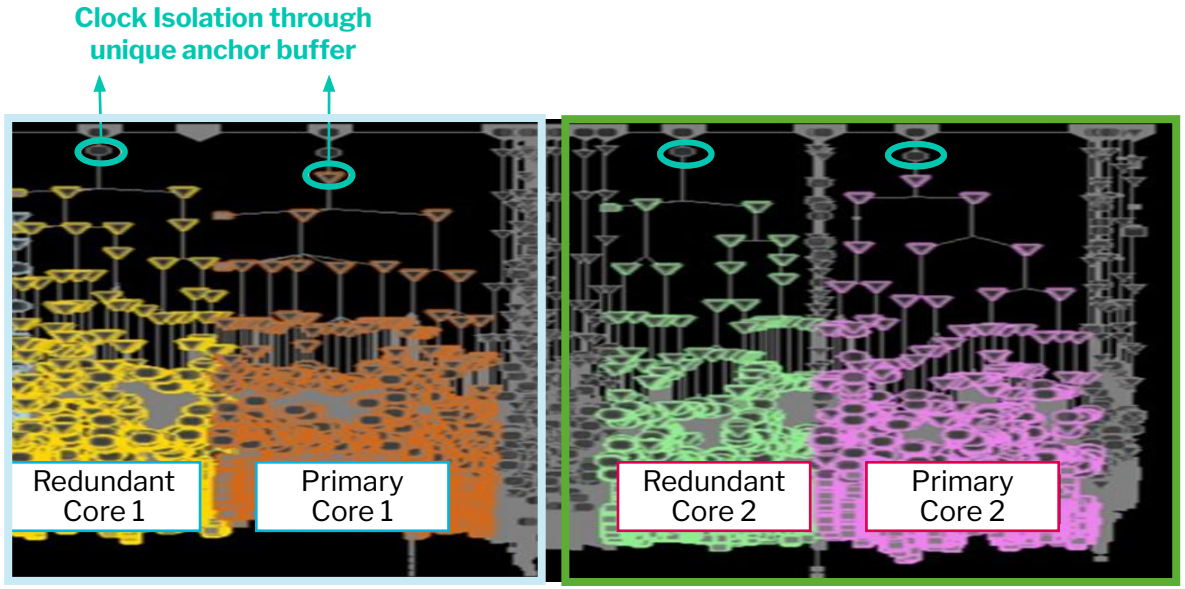
Results | DCLS

Visual inspections



Physical separation of DCLS modules

*No undue impact to QoR metrics experienced

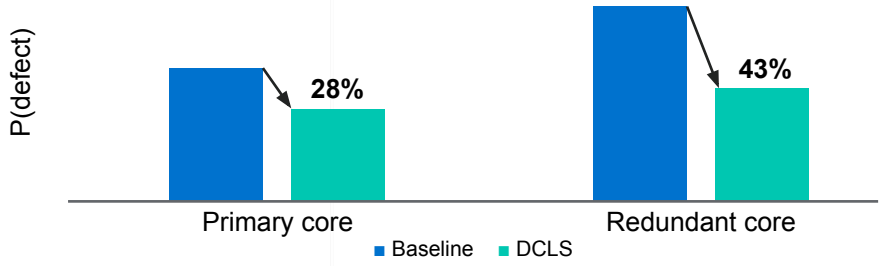


Clock tree debugger logical view

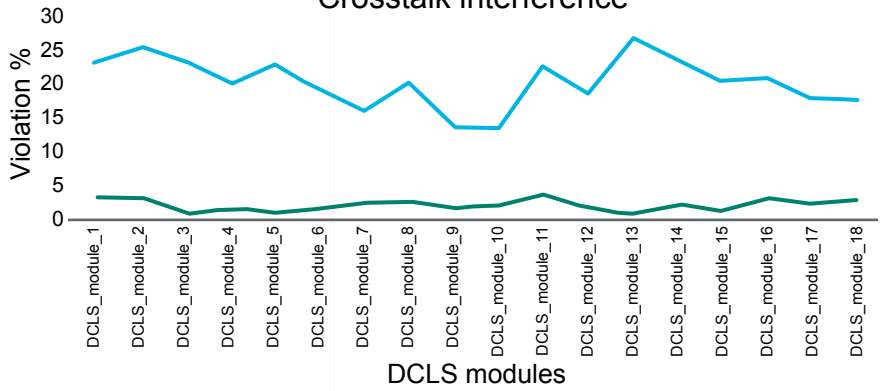
Results | DCLS

Plots

Common clock tree defect density



Crosstalk interference



Checkers

DCLS – Physical separation checks

Status	Spacing (µm)	Primary_group	Redundant_group
PASS	XX	foo_0_primary_group	foo_0_redundant_group
PASS	YY	foo_1_primary_group	foo_1_redundant_group
PASS	ZZ	foo_2_primary_group	foo_2_redundant_group
PASS	XY	foo_3_primary_group	foo_3_redundant_group
PASS	YZ	foo_4_primary_group	foo_4_redundant_group

DCLS – Clock isolation checks

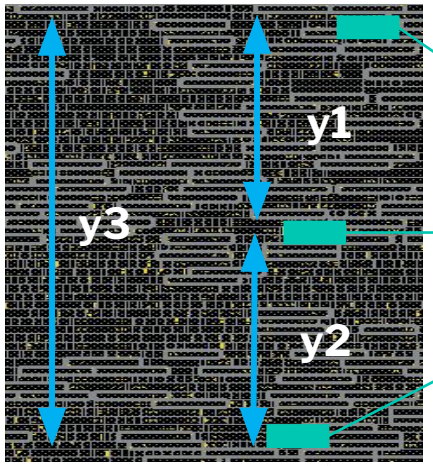
Status	Module	Module seq count	Clock_buffer	Level	Clock buffer sink count	Core module sinks	Redundant module sinks	Non-DCLS sinks
PASS	foo_0_primary_module	3183	foo_0_primary_module/anchor_clock_buffer	12	3184	3183	0	1
PASS	foo_0_redundant_module	3185	foo_0_redundant_module/anchor_clock_buffer	12	3185	3185	0	0
PASS	foo_1_primary_module	10	foo_1_primary_module/anchor_clock_buffer	18	10	10	0	0
PASS	foo_1_redundant_module	10	foo_1_redundant_module/anchor_clock_buffer	18	10	10	0	0
PASS	foo_2_primary_module	25613	foo_2_primary_module/anchor_clock_buffer	4	26862	25613	0	1249
PASS	foo_2_redundant_module	25058	foo_2_redundant_module/anchor_clock_buffer	6	25058	25058	0	0

DCLS – Route isolation checks

DCLS group	Total_nets	Intra_nets	Inter_nets	Dangling nets	Intra violate nets	Inter violate nets	Both violate nets	Total violate nets	Violate %
foo_0_primary_group	24858	24635	221	0	0	2	0	2	0.008
foo_1_primary_group	24490	24306	183	0	1	0	0	1	0.004
foo_2_primary_group	464	302	145	0	0	17	0	17	3.664
foo_3_primary_group	467	285	163	0	0	19	0	19	4.069
foo_4_primary_group	24520	24314	205	0	1	0	0	1	0.004
foo_0_primary_group	24933	24699	232	0	1	1	0	2	0.008

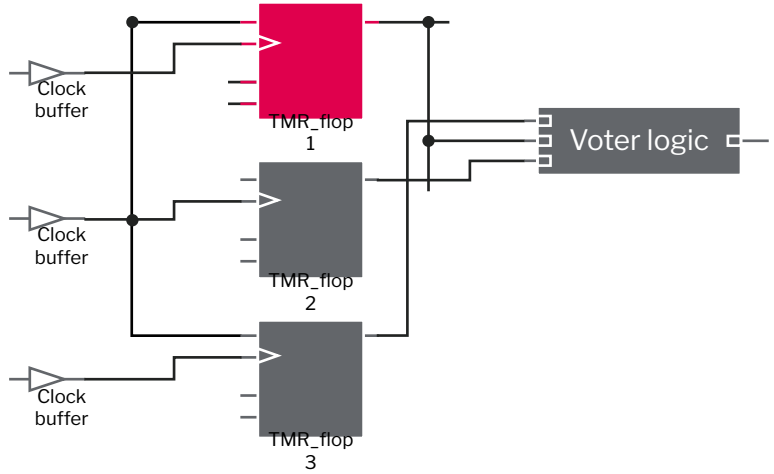
Results | TMR

Visual inspection



Physical separation of TMR flops

TMR flop clock pins are driven by a buffer with fanout = 1



Checkers

TMR – Physical separation checks

Status	Y_distance	TMR_flop	TMR_flop
PASS	y1	tmr_abc_0_reg_0_	tmr_abc_1_reg_0_
PASS	y2	tmr_abc_1_reg_0_	tmr_abc_2_reg_0_
PASS	y3	tmr_abc_2_reg_0_	tmr_abc_0_reg_0_
PASS	y4	tmr_abc_0_reg_10_	tmr_abc_1_reg_10_
PASS	y5	tmr_abc_1_reg_10_	tmr_abc_2_reg_10_
PASS	y6	tmr_abc_2_reg_10_	tmr_abc_0_reg_10_

TMR – Clock isolation checks

Status	TMR_flop	TMR_clock_buffer
YES	tmr_abc_0_reg_0_	tmr_clock_buffer_abc_0_reg_0_
YES	tmr_abc_1_reg_0_	tmr_clock_buffer_abc_1_reg_0_
YES	tmr_abc_2_reg_0_	tmr_clock_buffer_abc_2_reg_0_
YES	tmr_abc_0_reg_10_	tmr_clock_buffer_abc_0_reg_10_
YES	tmr_abc_1_reg_10_	tmr_clock_buffer_abc_1_reg_10_
YES	tmr_abc_2_reg_10_	tmr_clock_buffer_abc_2_reg_10_

Results | Safety critical routes

Checkers

Safety critical net pair delay checks

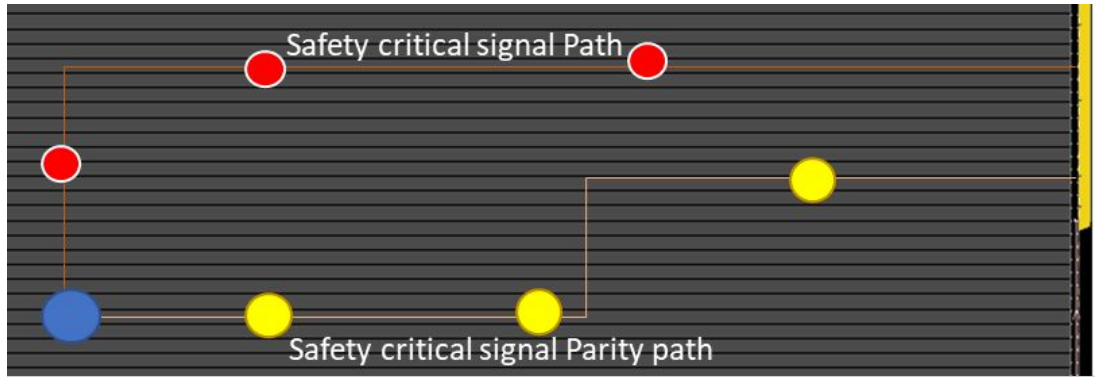
Path type	Start point	End point	Path_delay	Difference
Fault	foo1_in	fault_1	0.098	0.009
Parity		parity_1	0.107	
Fault	foo2_in	fault_2	0.071	0.022
Parity		parity_2	0.049	
Fault	foo3_in	fault_3	0.074	0.001
Parity		parity_3	0.075	
Fault	foo4_in	fault_4	0.216	0.004
Parity		parity_4	0.220	

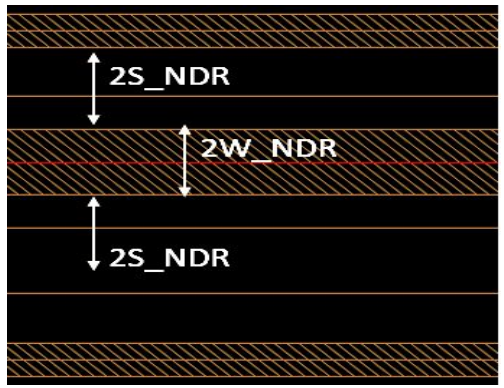
Safety critical net attribute check

Path type	Net_name	Bottom_metal_layer	Top_metal_layer	Route_rule
Fault path	fault_net1	Mx	My	2w2s_ndr
	fault_net2	Mx	My	2w2s_ndr
	fault_net3	Mx	My	2w2s_ndr
	fault_net4	Mx	My	2w2s_ndr
Parity path	parity_net1	Mx	My	2w2s_ndr
	parity_net2	Mx	My	2w2s_ndr
	parity_net3	Mx	My	2w2s_ndr
	parity_net4	Mx	My	2w2s_ndr
	parity_net5	Mx	My	2w2s_ndr

Visual inspection

Safety critical signal paths with nets and instance





Summary

Guidelines

- ISO26262
  - Guidelines for semiconductors
- IP
  - Safety/DFA manuals
- Tool
  - ISO26262 certified
  - FuSa manuals

Design methodology

- Independence
  - Achieve true independence using custom scripts, tool features
  - Checkers to validate
- Robustness
  - Tighter sign-off: Timing, EMIR
  - Latch up, DFM considerations
  - Immunity to process variation

Quality management

- Design release mechanisms
  - Tags, traceability enabled
- Data reproducibility
- Periodic internal reviews
- Documentation
  - PD analysis for FuSa documents must be well-managed/retained
  - Implausibility of failure arguments

Credits

Many thanks to:

Anup Kumar (Cadence Design Systems)
and to our colleagues Rahul Mandal, Arun Hegde and Brian Rogers

Q&A

Thank you