Solving the Physical Challenges for the Next Generation of Safety Critical & High Reliability Systems

Brian Li

PV Group Director – DSG, Cadence Design System Inc.

Agenda

- Market & drivers
- Challenges
- Forces of opportunity
- Electronic safety intent
- Examples
- Opportunities

Automotive Market and Key Trends

- Strong Revenue Growth Forecast
  - CAGR of 12.3% through 2025 (*Source: Omdia, 02/2022)*
  - Drive towards leading edge technology nodes
- Challenges of Agility and Supply
  - Software Defined Vehicle
  - ADAS and Autonomous Driving
  - Electrification (HEV/EV)
  - Shortage in semiconductor market, particularly automotive
- Industry Responding to Challenges
  - Automotive Makers, OEM and Tier-1s initiating SoC designs
  - Traditional semiconductor companies entering automotive market
  - Startup companies developing autonomous driving platforms and sensors

Software-Defined Vehicles

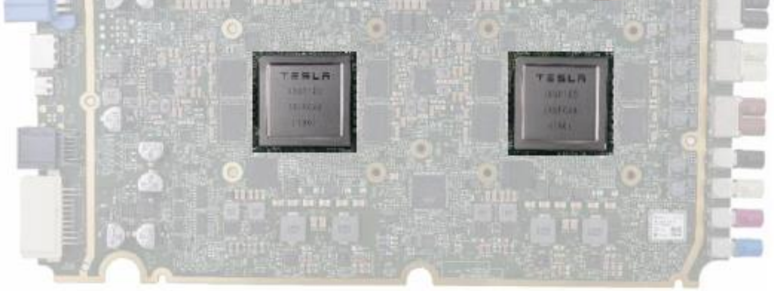
Electrification Hybrid & EVs

Discrete vs Integrated AD Platform

Audi zFAS

	Audi zFAS	Tesla HW 3.0
Processors	1x Nvidia, 2x Intel, 1x Infineon	2x Tesla SoC
Processor footprint	2985mm^2	2812mm^2
Processor die area	393mm^2	611mm^2
Technology node	28nm Tegra K1 28nm Cyclone 40nm EyeQ3 65nm Aurix	14nm Tesla

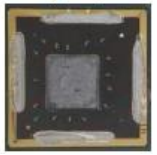
Tesla H\



NVIDIA Tegra K1 SoC

Intel Altera Cyclone

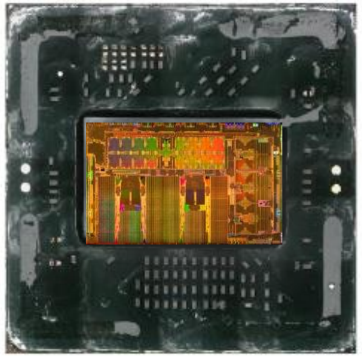
Intel MobilEye EyeQ3



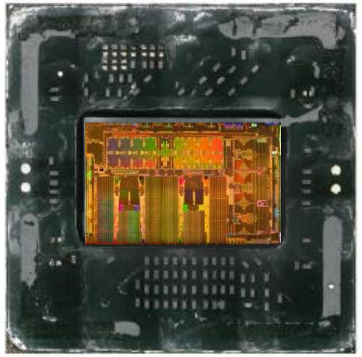
Infineon Aurix

150x Performance Improvement!

Tesla SoC



Tesla SoC



Performance <1 TOPS

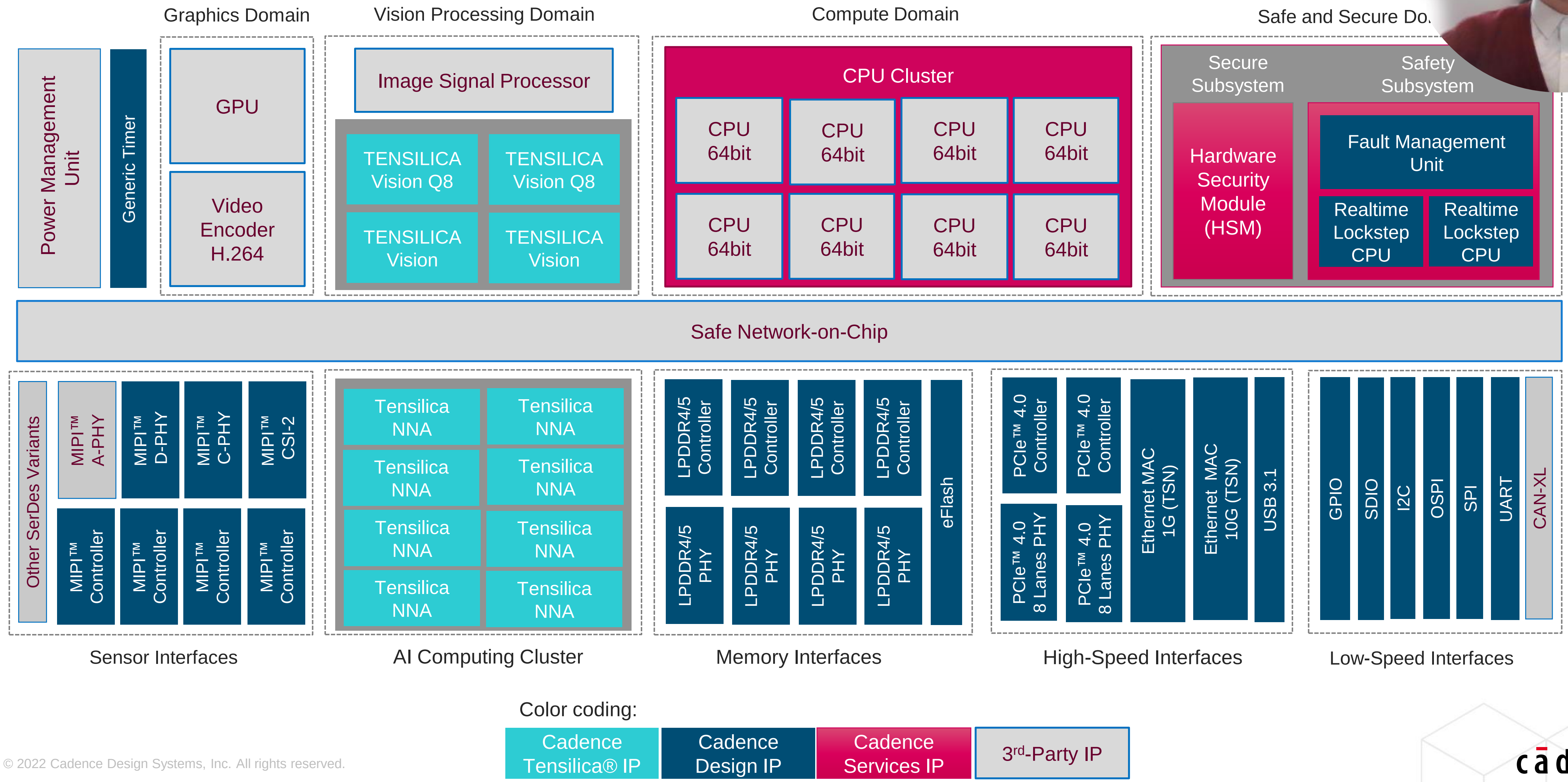
BOM cost: USD 275$

Performance: 144 TOPS

BOM cost: USD 400$

Source: SYSTEM Plus Consulting

High-Performance AI Compute SoC Architecture



cādence®

Physical Design Challenges

SAFETY
In event of a failure, prevent injury

RELIABILITY
Minimize failures during product life

QUALITY
Maximize working products from manufacturing

Technology Forces

Artificial Intelligence

Machine learning data and computation

LLM

Enabled by much greater computation power

Digital Twins

Virtual copy of physical objects

Early/Virtual prototyping

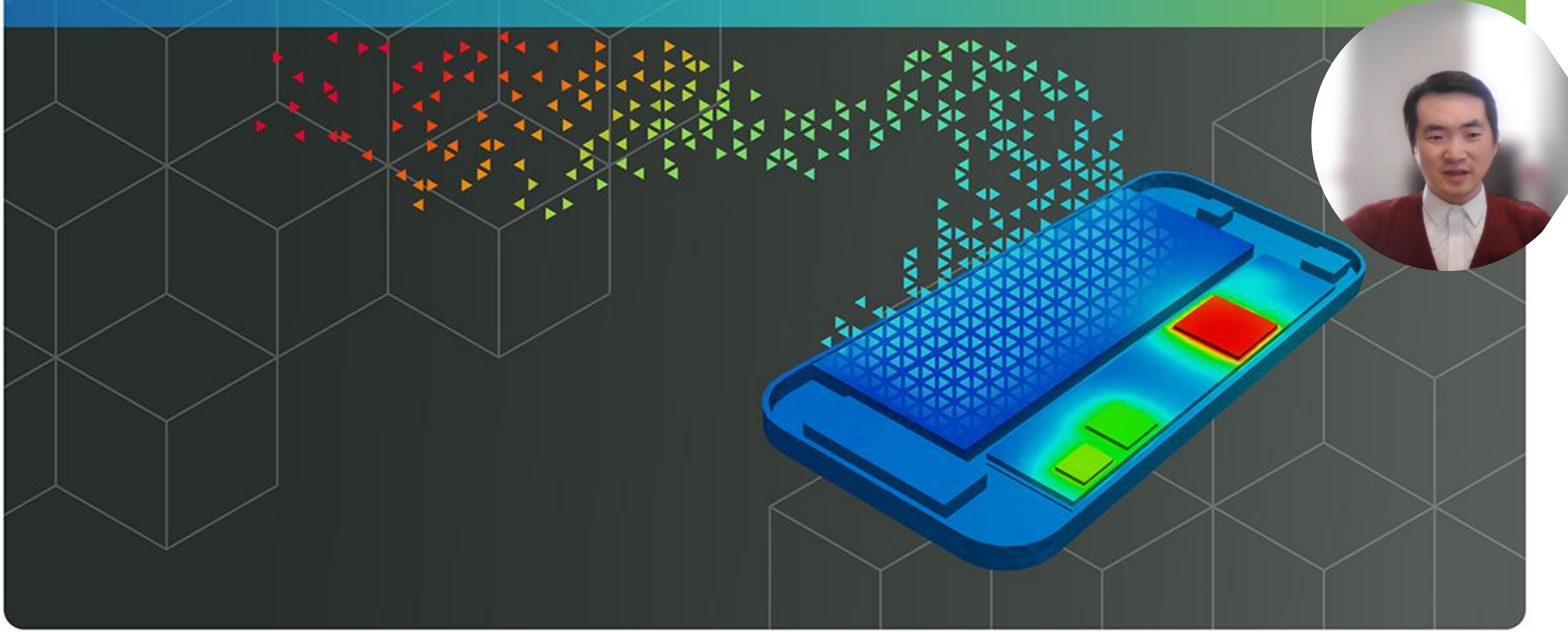
Agile development reducing ECOs

3DIC & Chiplet

Increased data bandwidth

System Design Flexibility

System Driven PPA



Automotive SoC Design Enablement with 3 Forces

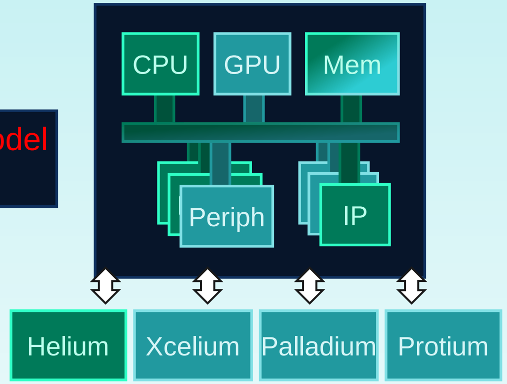
AI, Digital Twins, 3DIC

Digital Twins - Helium Virtual and Hybrid Studio

Platform Assembly

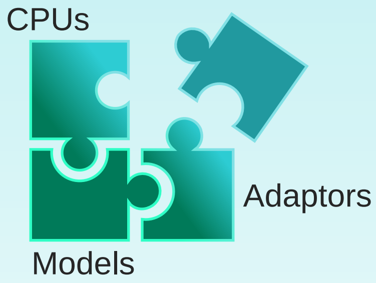
- GUI-based platform assembly
- Correct by construction
- Virtual model generator
- Helium™ virtual runtime engine

Hybrid Systems



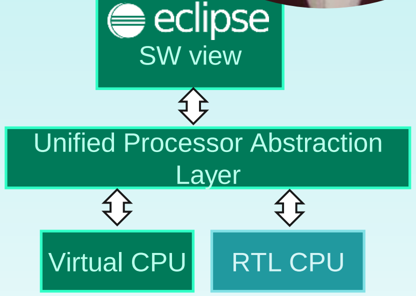
- Native integration with Palladium®, Protium™, and Xcelium™ technologies
- Gearshift from virtual CPU to RTL CPU dynamically
- Smart memory technology
- Virtual to RTL adaptors

Virtual Model Library



- Integration of full Arm Fast Models IP portfolio incl. V9
- SystemC™ TLM 2 Model Library TLM routers, I2C, UART, Ethernet, PCIe®, USB,…
- Engagement-ready virtual and hybrid reference designs

Unif



- Software debug experience identical from virtual to RTL
- Uniform support in all engines
- Mix software-level and signal-level debug commands in one TCL script
- Live RTL IP register view

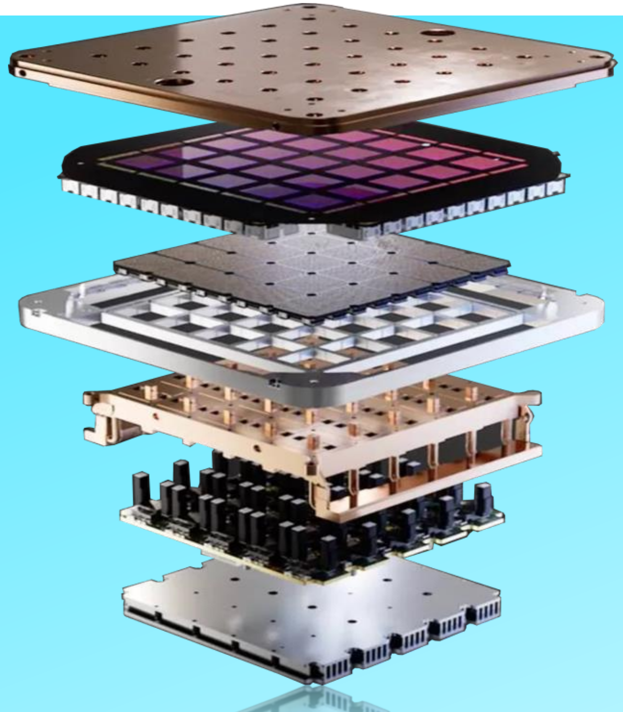
Tesla: GenAI and the Future of Semiconductor and System Design

DOJO AI Supercomputer

362 TFLOPs 7nm D1 AI Chip
designed with Cadence Digital Full flow

600K pins 3D-IC package
designed with Cadence Integrity™

9 TB/s I/O signal and power integrity
analysis with Cadence Sigrity™



Next Gen System

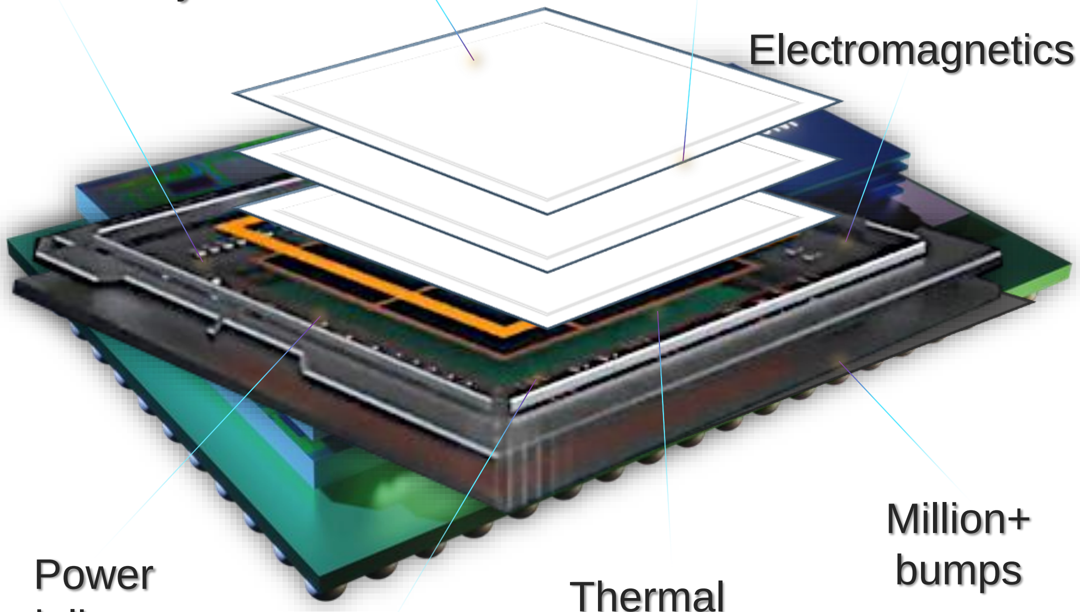
Full self-driving platform

AI Inference chip and solution

Next gen DOJO chip and solution

"We are very excited to extend our partnership to Tesla's next generation DOJO and FSD platforms."

Unified Design/Analysis Platform for 3D IC Design & Pack



Architecture iterations

Corner explosion

Connectivity

Electromagnetics

Million+ bumps

Power delivery

Thermal hotspots

Package alternatives

Integrity™ 3D-IC Platform

3D System Level Planning

Implementation

Digital	Analog	RF
Innovus™	Virtuoso®	Microwave Office®

Package / Board

Allegro® Package Designer / Allegro® PCB

IC, PKG, and Board Analysis & Signoff

DRC/LVS Timing Thermal Power SI/PI System LVS

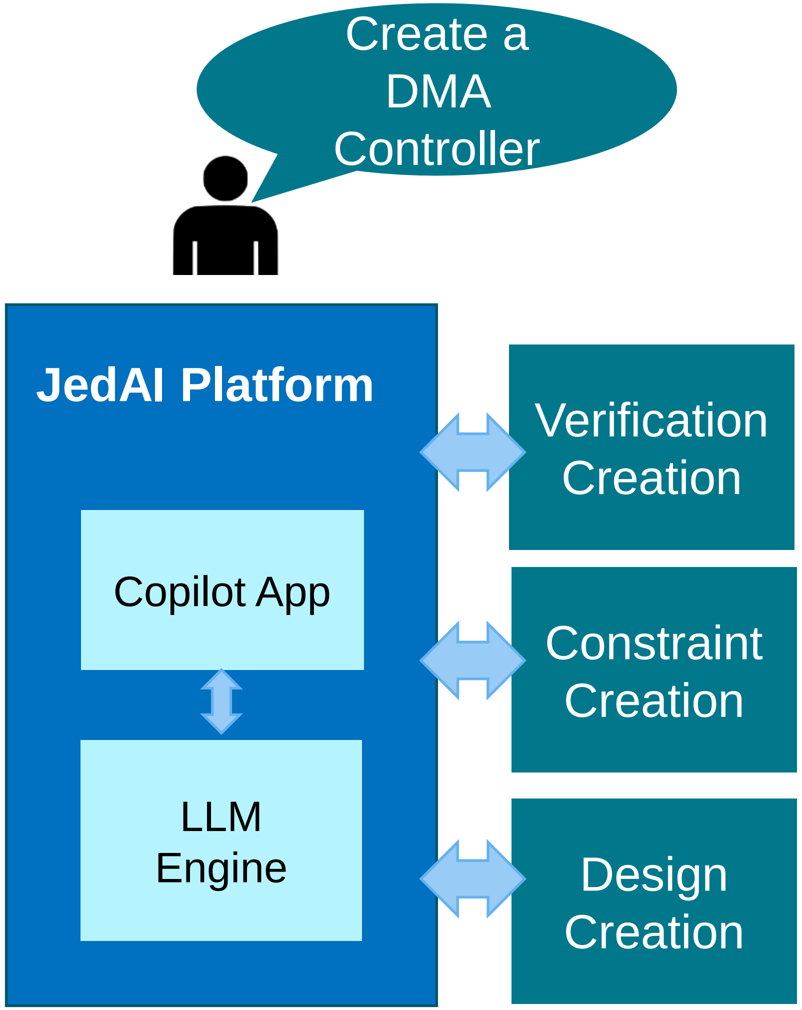
Clarity™ Sigrity™ Celsius™

Tempus™ Quantus™ Pegasus™

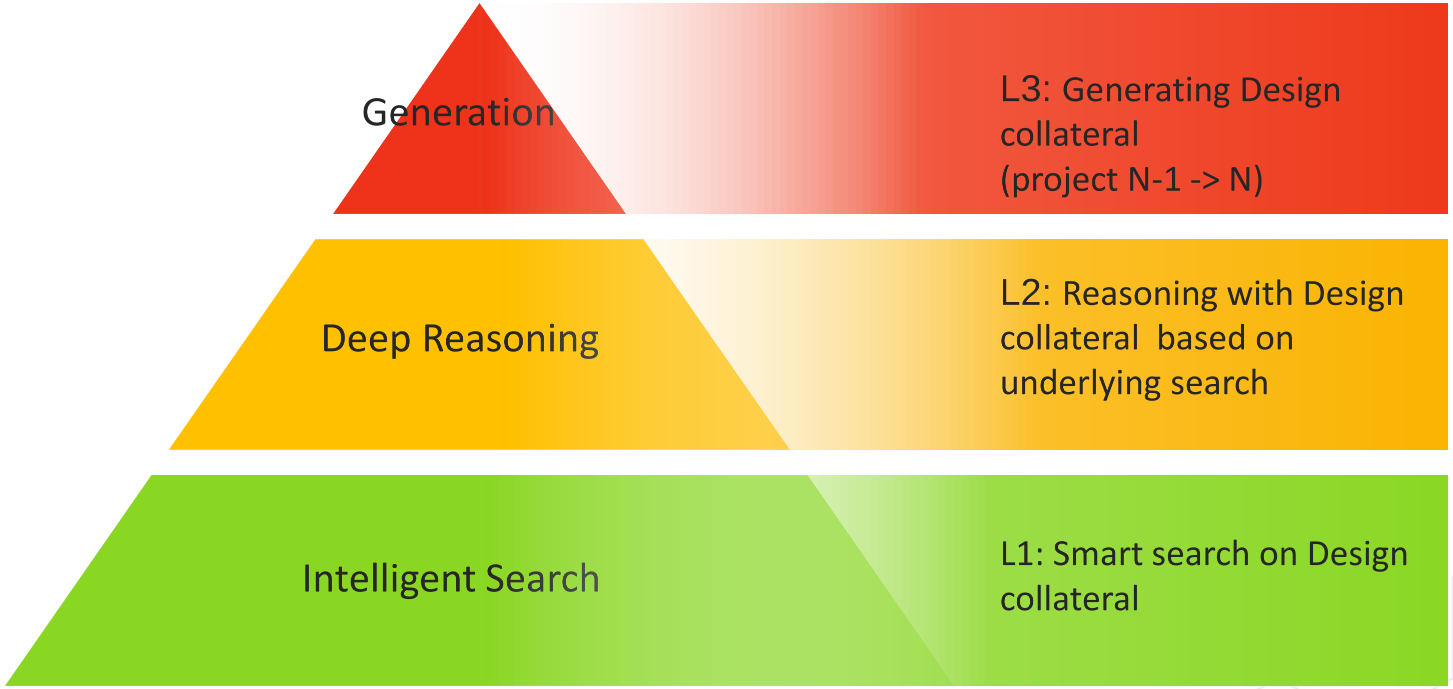
Reduces iterations between packaging and SoC.

Improves 3D-IC PPA , TAT and productivity.

AI - LLM Copilot

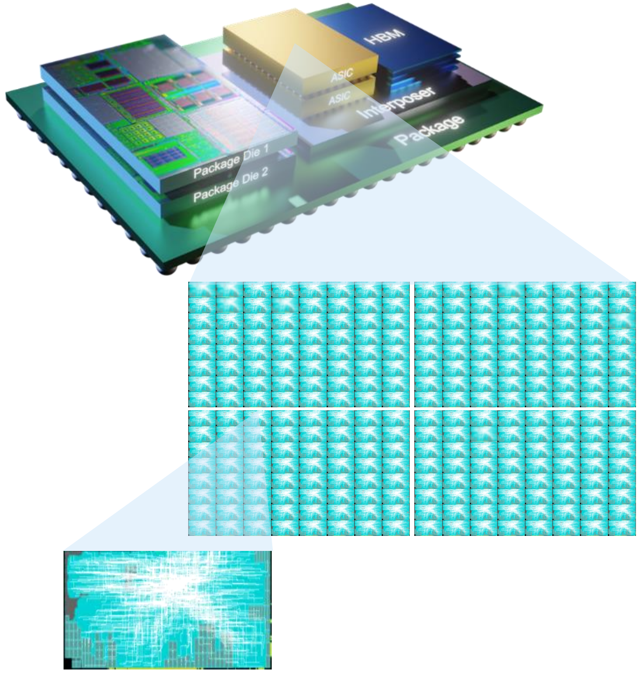


- JedAI Platform uses LLM as a Copilot to automatically generate design collateral
  - Reduces the manual engineering effort
  - Creates higher quality design and verification

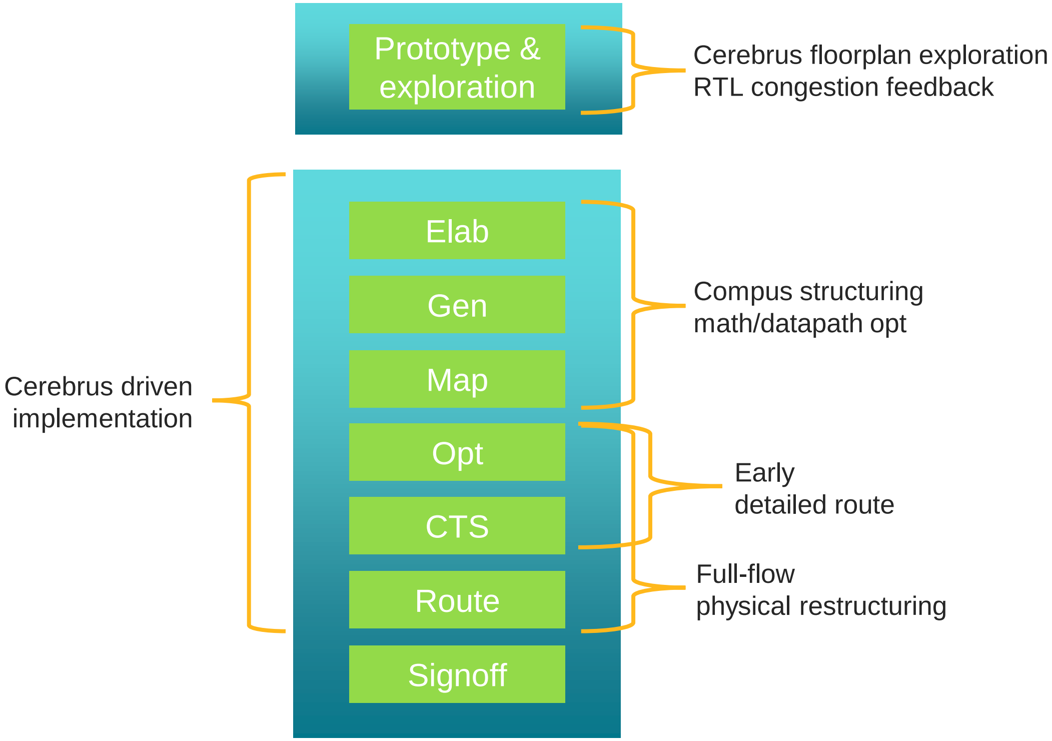

Generation
L3: Generating Design collateral (project N-1 -> N)
Deep Reasoning
L2: Reasoning with Design collateral based on underlying search
Intelligent Search
L1: Smart search on Design collateral

cādence®

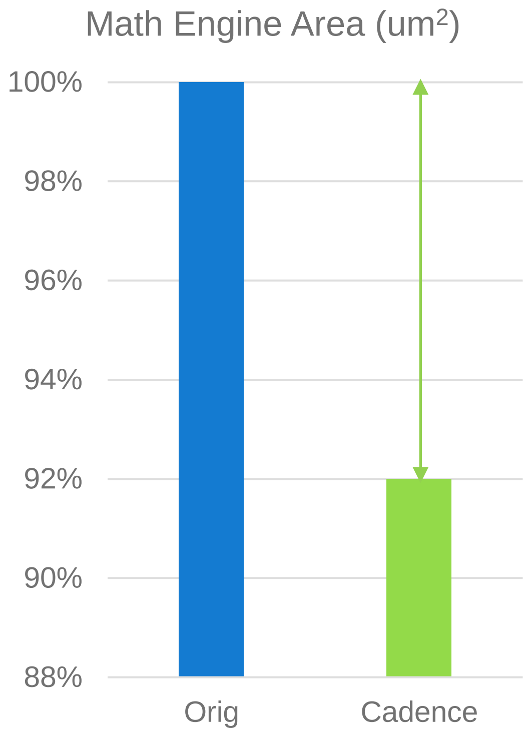
AI - Critical Compute Element: Smallest Routable



Size & routability of critical compute element matters



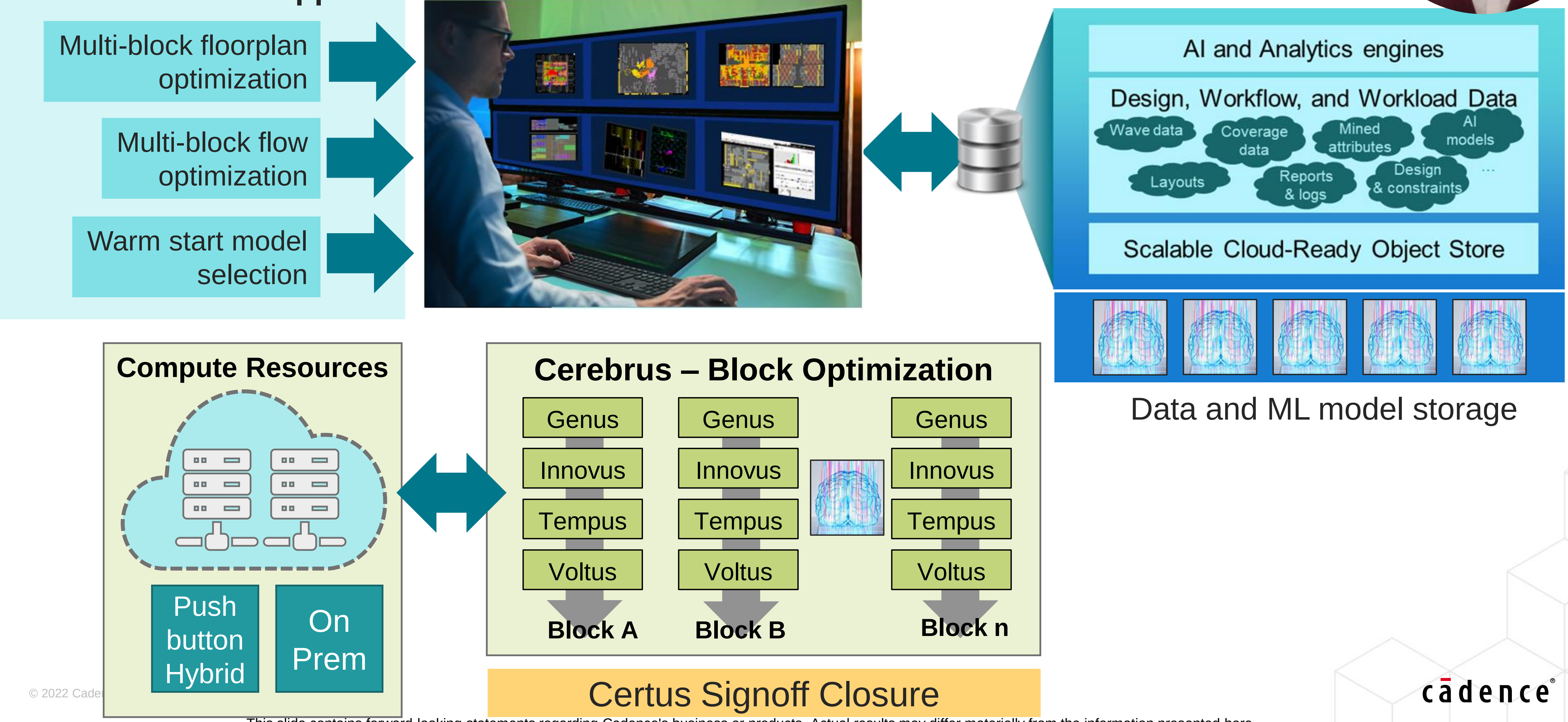
AI-driven early exploration and full-flow optimization for smallest, routable blocks



Up-to 10% smaller math engines

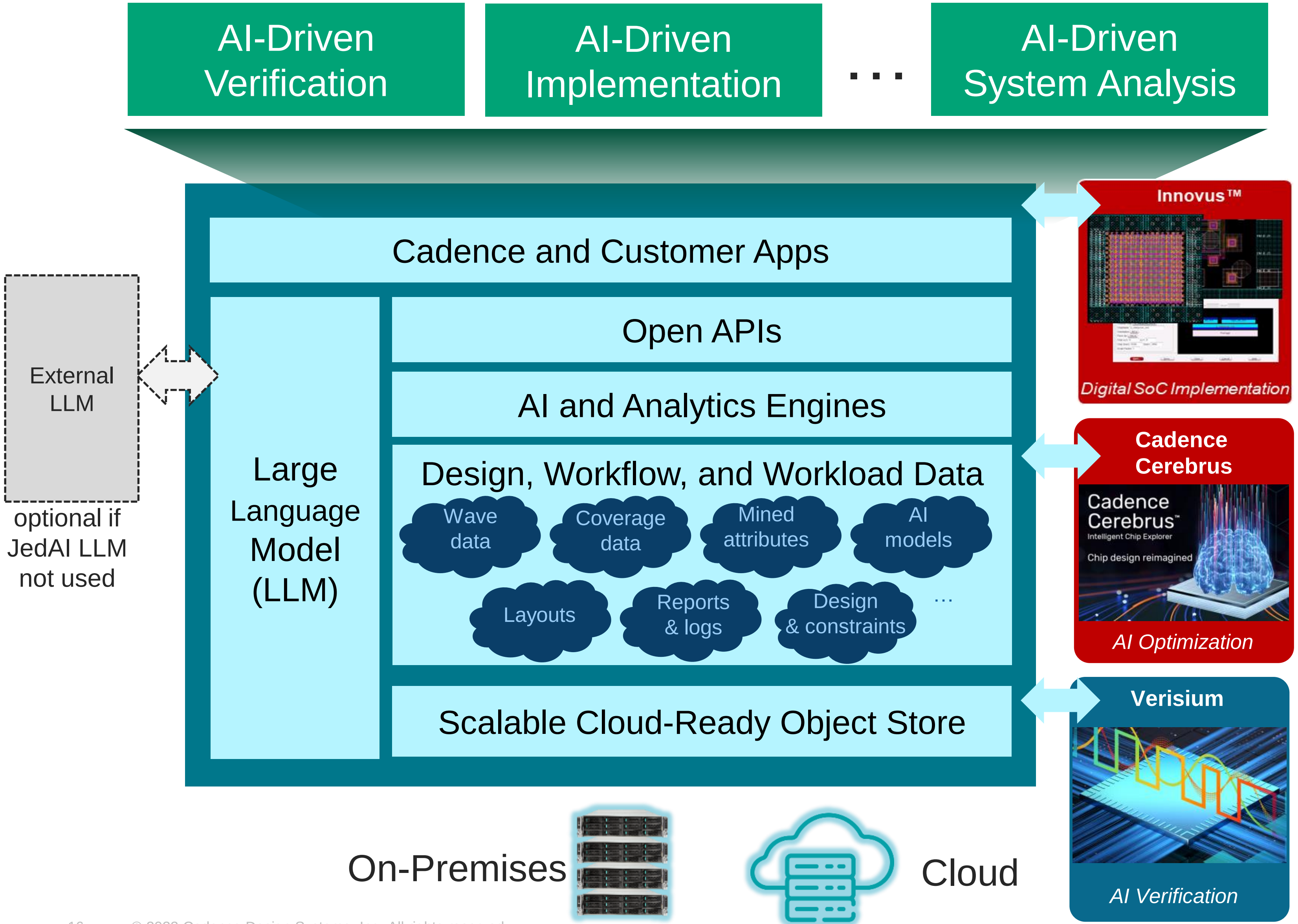
AI - 50M cell Design Automation

Multi Design – Multi Run - Design Closure



This slide contains forward-looking statements regarding Cadence's business or products. Actual results may differ materially from the information presented here.

AI - Joint Enterprise Data and AI (JedAI) Platform

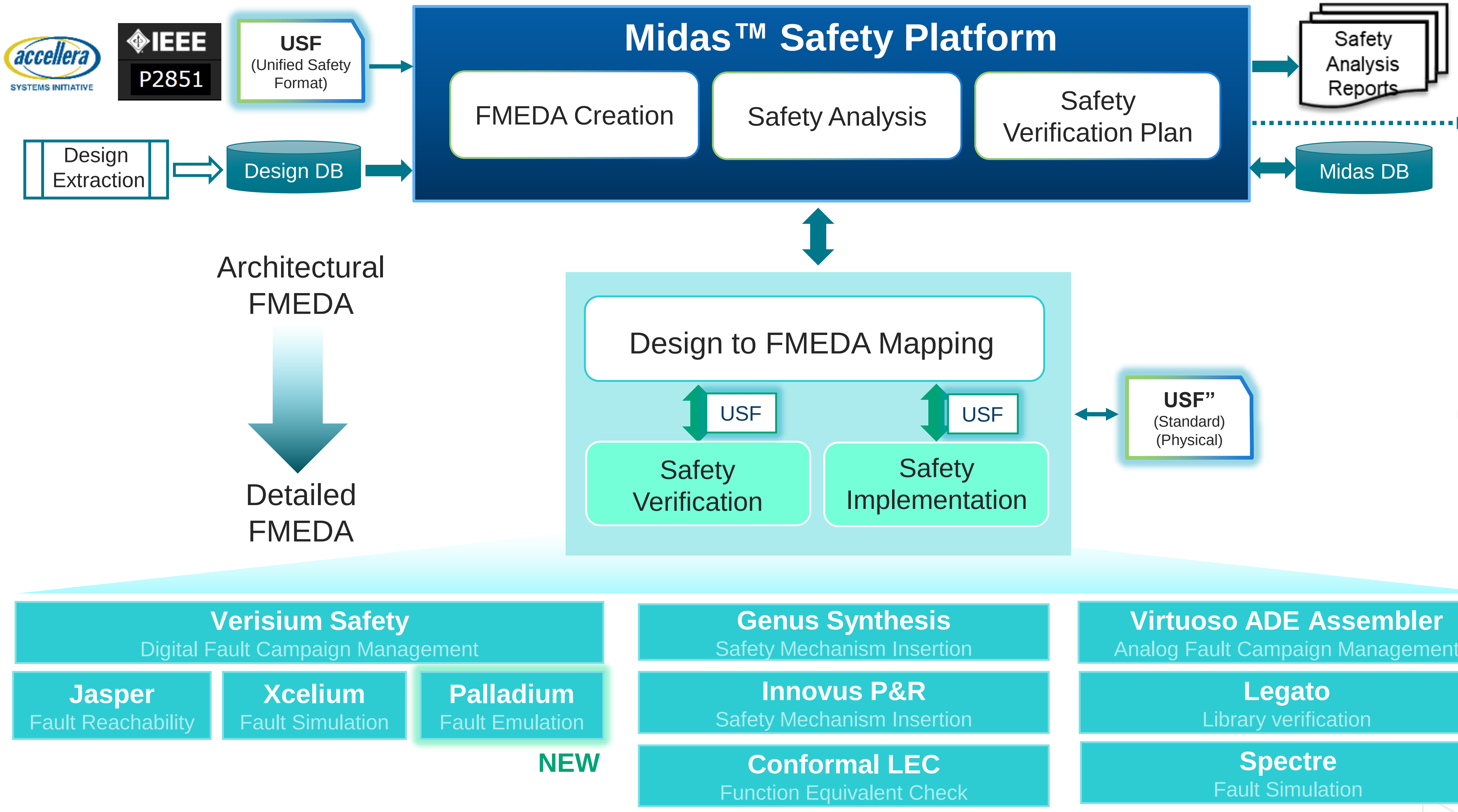


- Enterprise scale AI and LLM driven data platform
- Connectors to many Cadence tools for complete chip design analysis
- Open API enabling Cadence and Customer Apps
- Integrated LLM for secure Copilot Apps

Electronic Safety Intent

USF (Unified Safety Format) Driven Design & Implementation

Cadence Functional Safety Solution



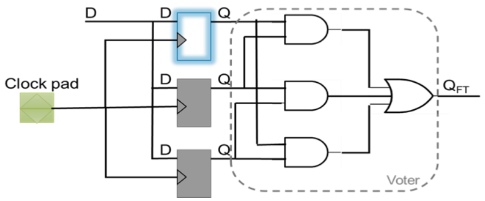
Safety Features Implementation (in a nutshell)

SAFETY

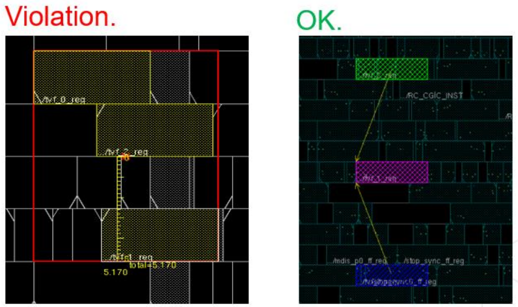
TMR: triple modular redundancy

tmr cloning.

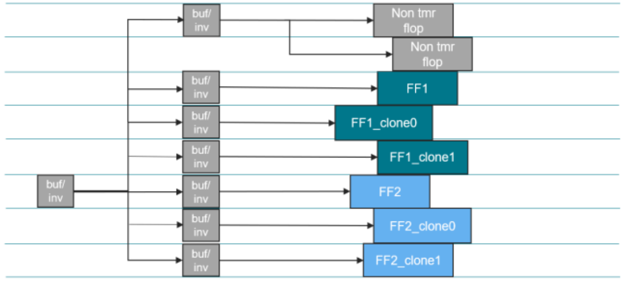
Triple Modular Redundancy (TMR)



Placement.
(x/y & radial separation).

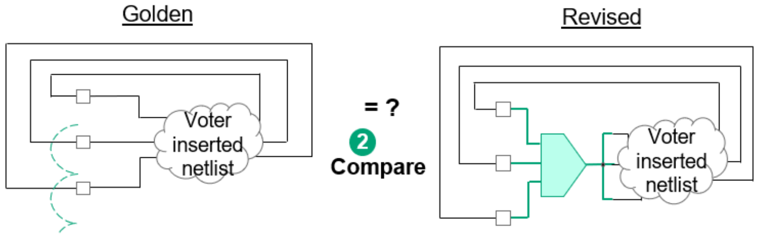


Clock isolation.



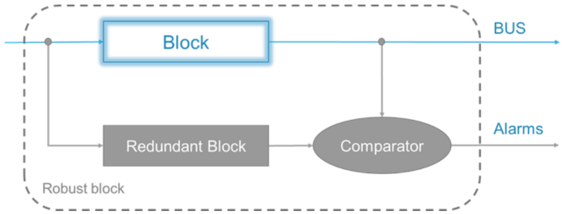
Every tmr flop clock-pin is driven by a buffer with fanout = 1

Conformal.
Equivalency Checking.
tmr logic validation.

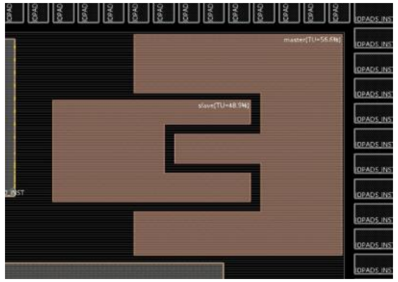


DCLS: Dual Core Lock Step for modular redundancy

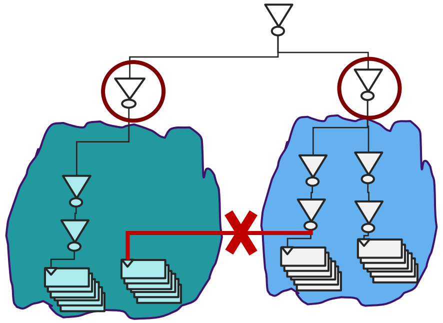
Dual Core Lock Step (DCLS)



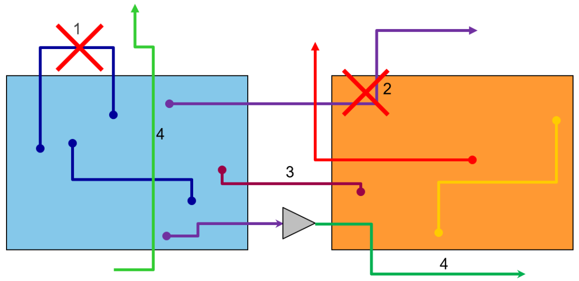
Placement.
Physical separation.



Clock isolation.



Routing.
Routing isolation.



SAFETY

Example - USF-Phys File

```
create_failure_mode FM_tmr -insts {FF1 FF2…..}
create_safety_mechanism SM_tmr -type tmr -class hw
apply_safety_mechanism SM_tmr -to FM_tmr
```

- Define a list of safety-critical flops
- Apply tmr safety-mechanism to those flops
- “-generated” option is not used as Genus™ will create the safety-mechanism, ie. clone the flops and add voting logic

```
create_failure_mode FM_mission -group {Master}
create_failure_mode FM_passive -group {Slave}
create_safety_mechanism SM_dcls -type dcls -class hw
apply_safety_mechanism SM_dcls -to FM_mission -generated
apply_safety_mechanism SM_dcls -to FM_passive -generated
```

- Define safety-critical groups
- Apply dcls safety-mechanism to those groups
- “-generated” option is used as the groups should already be defined in the RTL and floorplan

```
set_safety_mechanism_rules
  -tmr_spacing <>
  -tmr_isolate_clock

set_safety_mechanism_rules
  -dcls_spacing <>
  -dcls_isolate_clock <>
```

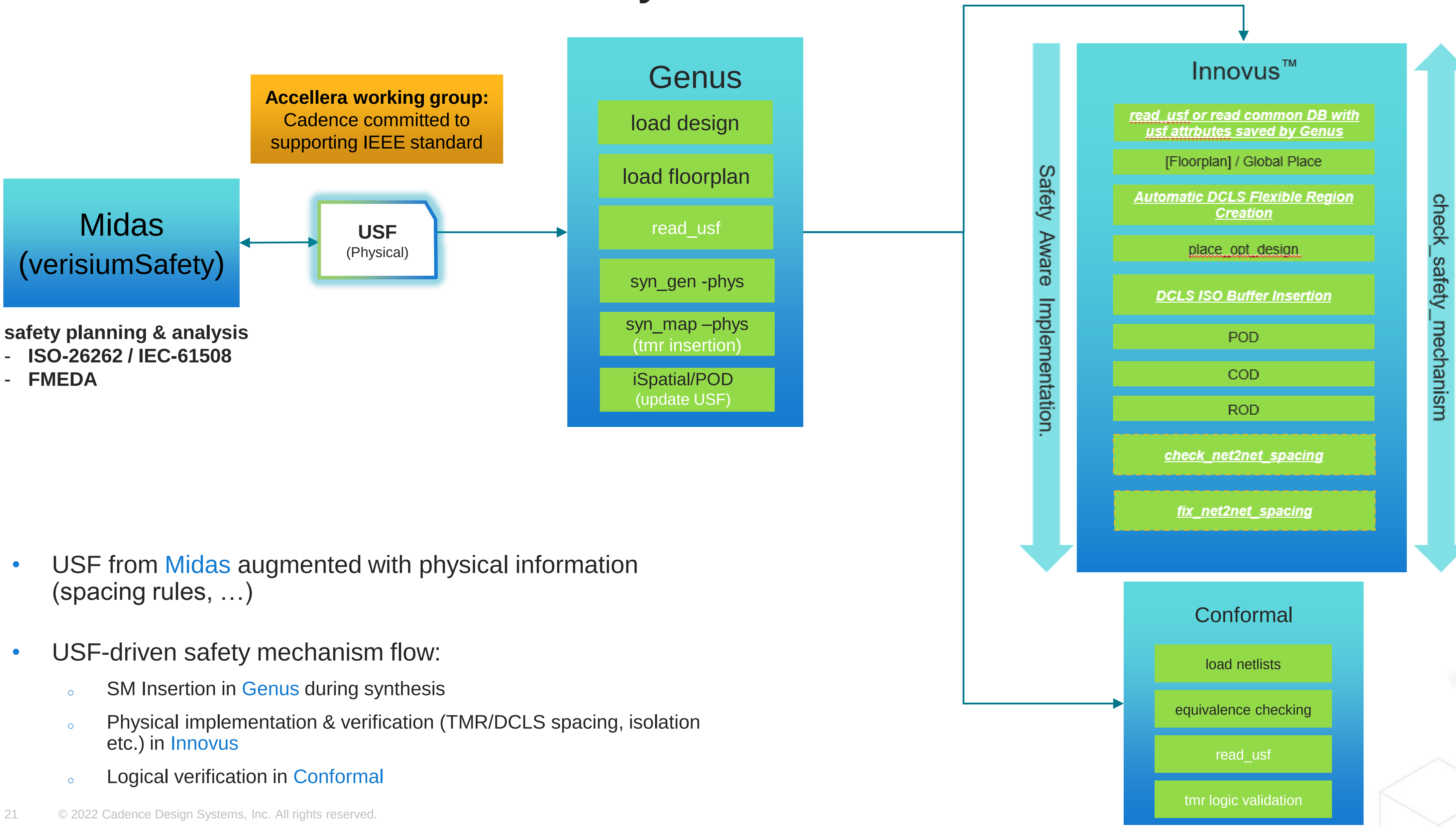
- Define implementation rules for safety mechanisms

```
read_usf <file>
write_usf <file>
reset_usf
read_safety_tmr <> : import tmr from third-party tool or rtl.
```

cādence®

SAFETY

Cadence Automotive Safety / USF-Driven Flow



Accellera working group:
Cadence committed to supporting IEEE standard
Midas (verisiumSafety)
USF (Physical)
Genus
load design
load floorplan
read_usf
syn_gen -phys
syn_map –phys (tmr insertion)
iSpatial/POD (update USF)
Safety Aware Implementation.
Innovus™
read_usf or read common DB with usf attrbutes saved by Genus
[Floorplan] / Global Place
Automatic DCLS Flexible Region Creation
place_opt_design
DCLS ISO Buffer Insertion
POD
COD
ROD
check_net2net_spacing
fix_net2net_spacing
check_safety_mechanism
Conformal
load netlists
equivalence checking
read_usf
tmr logic validation

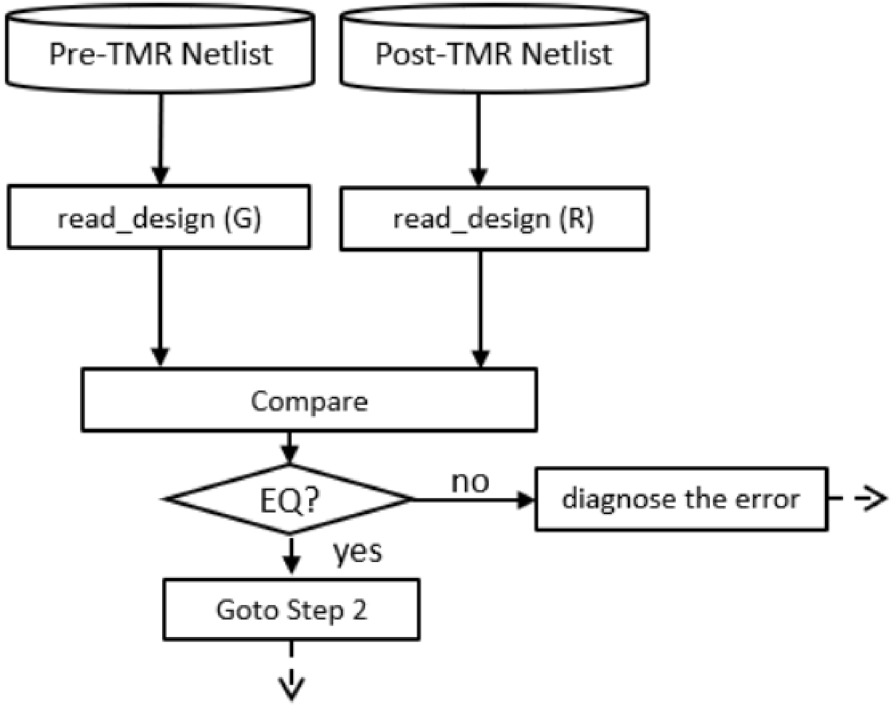
safety planning & analysis
- **ISO-26262 / IEC-61508**
- **FMEDA**

- USF from Midas augmented with physical information (spacing rules, …)
- USF-driven safety mechanism flow:
  - SM Insertion in Genus during synthesis
  - Physical implementation & verification (TMR/DCLS spacing, isolation etc.) in Innovus
  - Logical verification in Conformal

SAFETY

Conformal USF Verification for TMR

- Step 1: Pre-TMR Netlist versus Post-TMR netlist Verification



Pre-TMR Netlist

Post-TMR Netlist

read_design (G)

read_design (R)

Compare

EQ?

no

diagnose the error

yes

Goto Step 2

GENUS/safety_compare_pre_vs_post_tmr_netlist.do

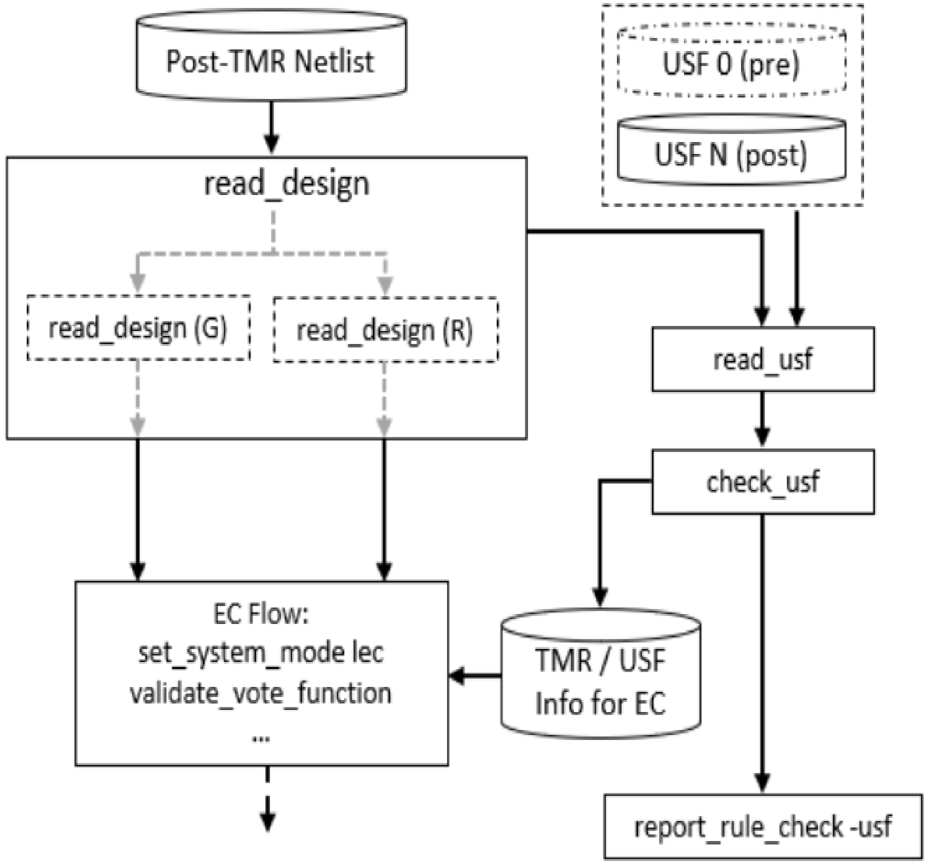
```
tclmode
set_dofile_abort off
set_naming_rule -field_delimiter "." ""

read_library -liberty \
   [list ./DB/libs/fast_vdd1v2_basicCells.lib\
   ./DB/libs/fast_vdd1v2_basicCells_hvt.lib\
   ./DB/libs//fast_vdd1v2_basicCells_lvt.lib\
   ./DB/libs/CDK_S64x10.lib]\
   -both ;

read_design ./pre_tmr_clone.v  -root cpu_10bit -rootonly -
golden ;
read_design ./post_tmr_clone.v -root cpu_10bit -rootonly -
revised ;

# perform sequential merge on tvf flops
set_flatten_model -all_seq_merge
set_x_conversion E -both
set_system_mode lec
add_compared_points -all
compare
```

cadence

SAFETY

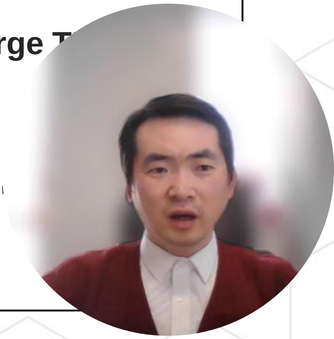
Conformal USF Verification for TMR

- Step 2: Post-TMR Netlist Verification



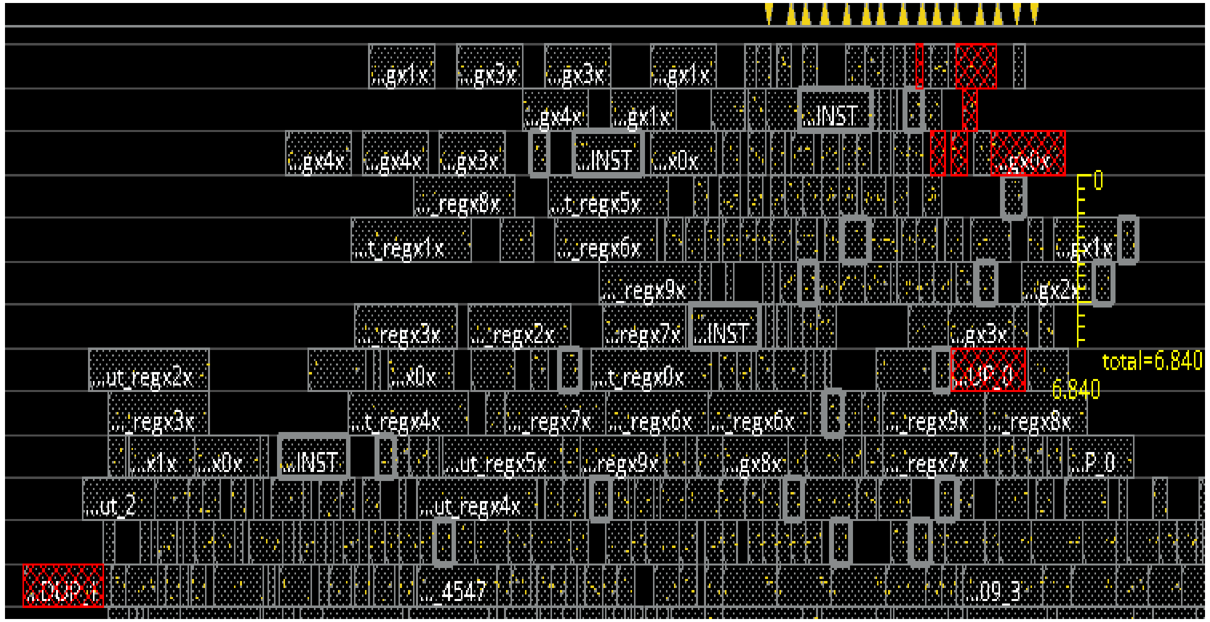
GENUS/safety_tmr_validate_vote_function.do (manual)

```
tclmode
set_dofile_abort off
set_naming_rule -field_delimiter "." ""

# set_usf_option -hiernameprefix rootmodule ;         # pathnames in usf file start with
root module
                                                       # uncomment as needed
set_usf_option -flow tmrvalidation ;                   # the flow is for TMR validation

read_library -liberty \
   [list ./DB/libs/fast_vdd1v2_basicCells.lib\
   ./DB/libs/fast_vdd1v2_basicCells_hvt.lib\
   ./DB/libs/fast_vdd1v2_basicCells_lvt.lib\
   ./DB/libs/CDK_S64x10.lib]

read_design ./post_tmr_clone.v -root cpu_10bit -rootonly ; # post TMR insertion netlist
read_usf post_tmr_clone.usf_no_dcls ;                 # post TMR usf file
set_analyze_option -noseq_merge ;              # do not perform sequential merge T
set_x_conversion E -both ;
set_system_mode lec ;

check_usf ;       # perform usf TMR front-end check; generate TMR info file
validate_vote_function -file lec_tmr.spec ;      # validate TMR functionality
report_compare_data -class noneq ;
```

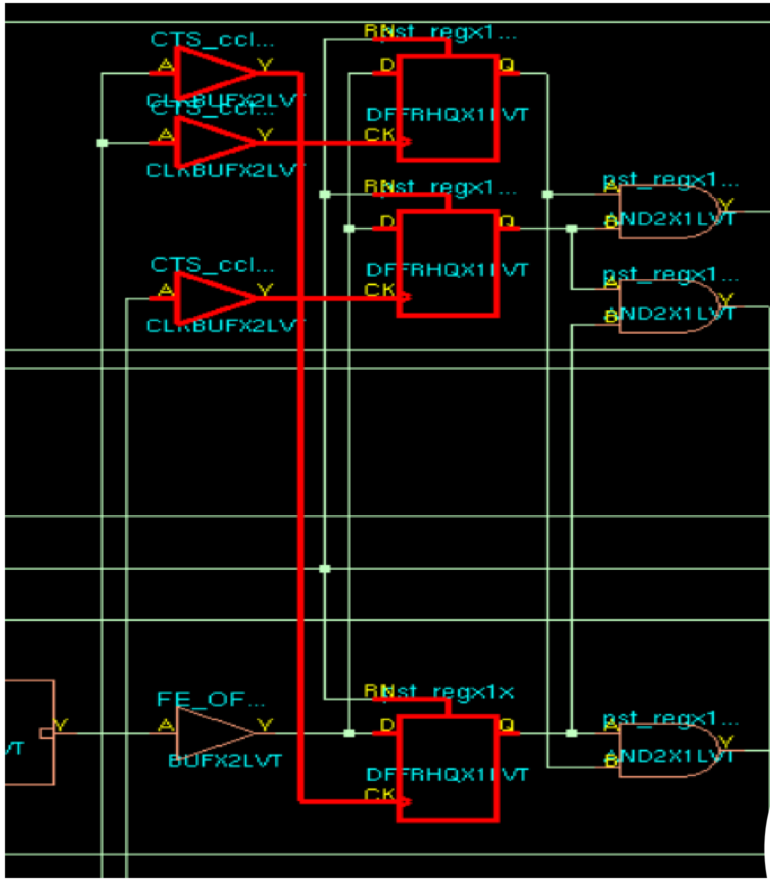


Example - TMR implementation

safety_tmr_spacing_y: 6.0



safety_tmr_isolate_clock: unique_driver



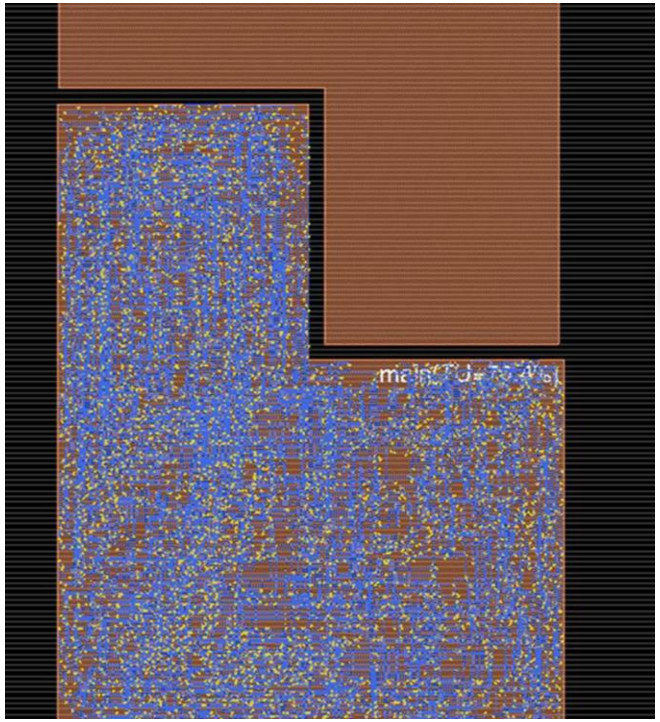
cadence®

Flexible DCLS (Dual-Core-Lock-Step) Floorplan/Placement

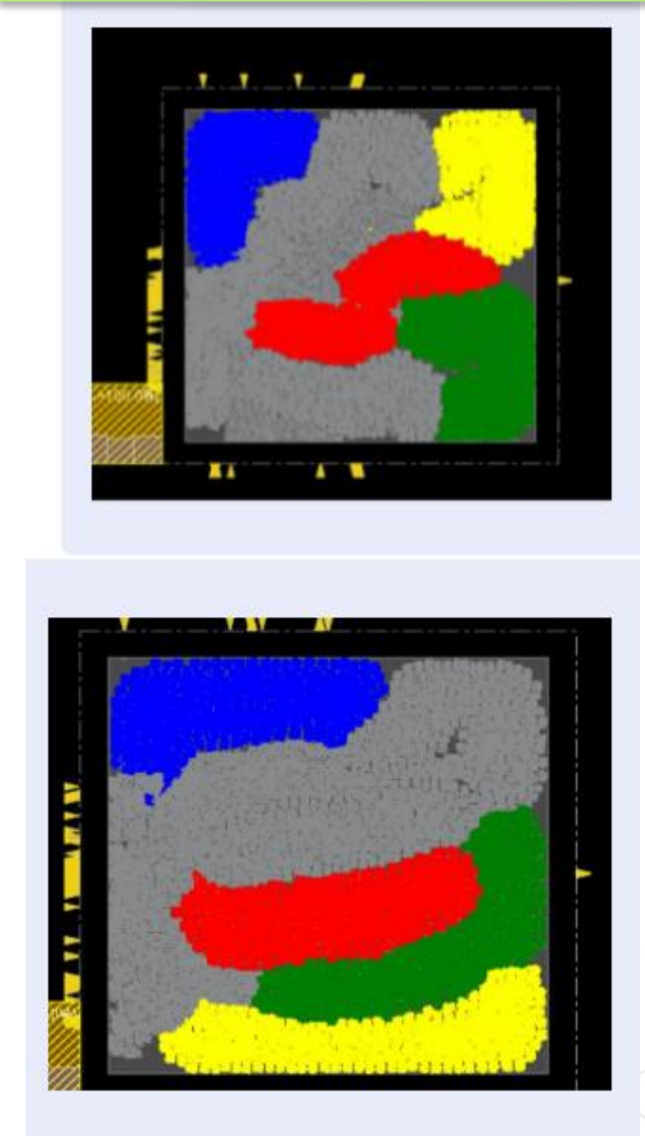
Innovus Implementation System

SAFETY

Old Fence Based Solution



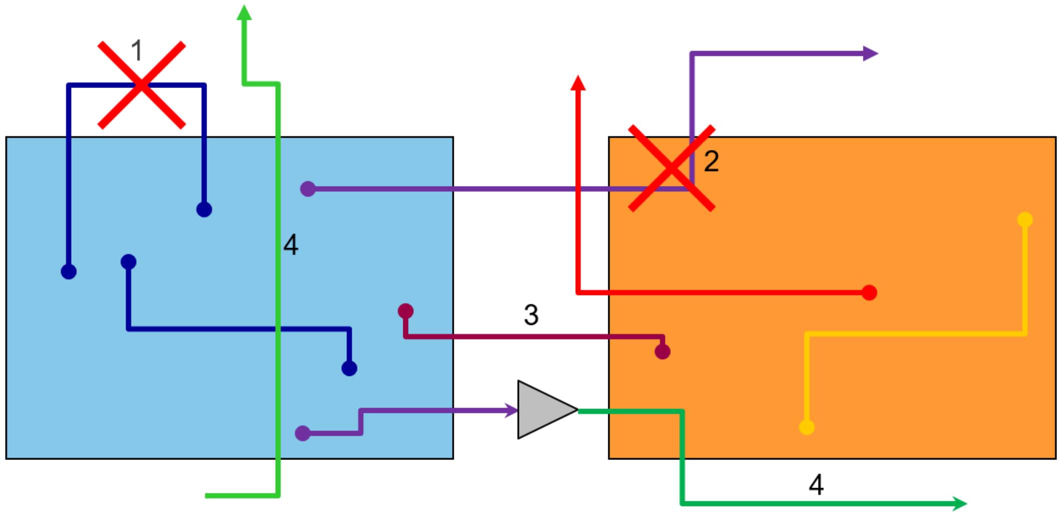
New DCLS Flexible Region Based Solution



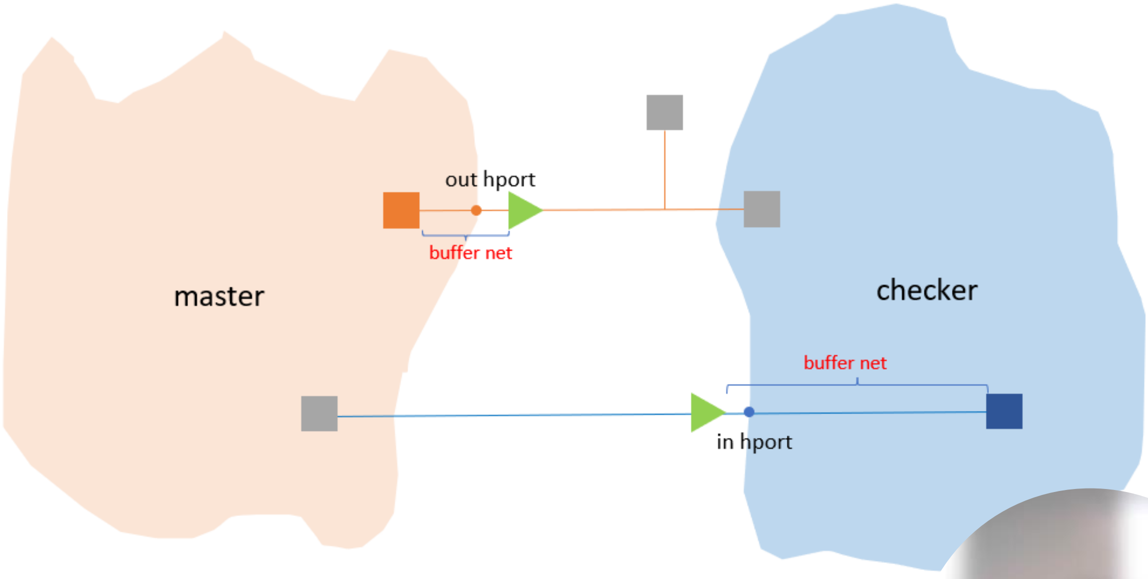
Avoid DCLS FuSa Interface Net Violation From Pre-Route

add_dcls_iso

- FuSa routing restrictions



- All interface net FuSa violation will be avoided

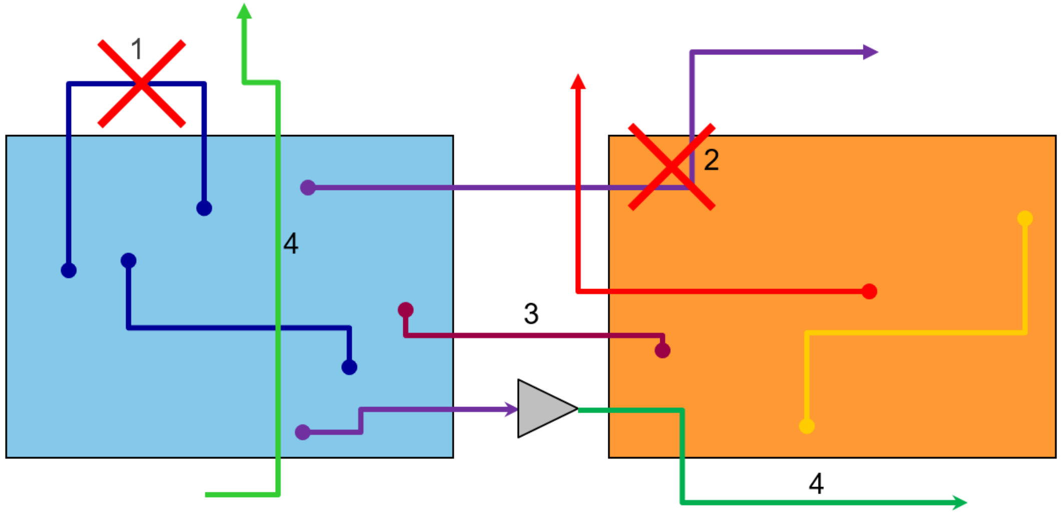


SAFETY

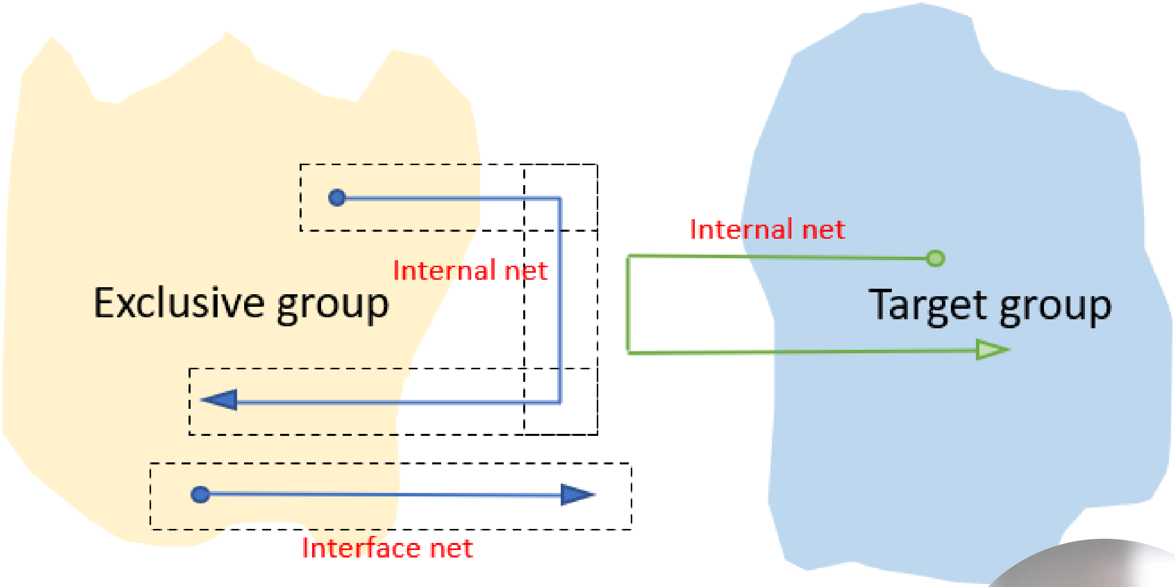
Avoid Over Pessimism in FuSa Routing

check_net2net_spacing / fix_net2net_spacing

- FuSa routing restrictions (over-pessimistic region boundary based checks)



- Identify the *real* FuSa violations without relying on region boundary

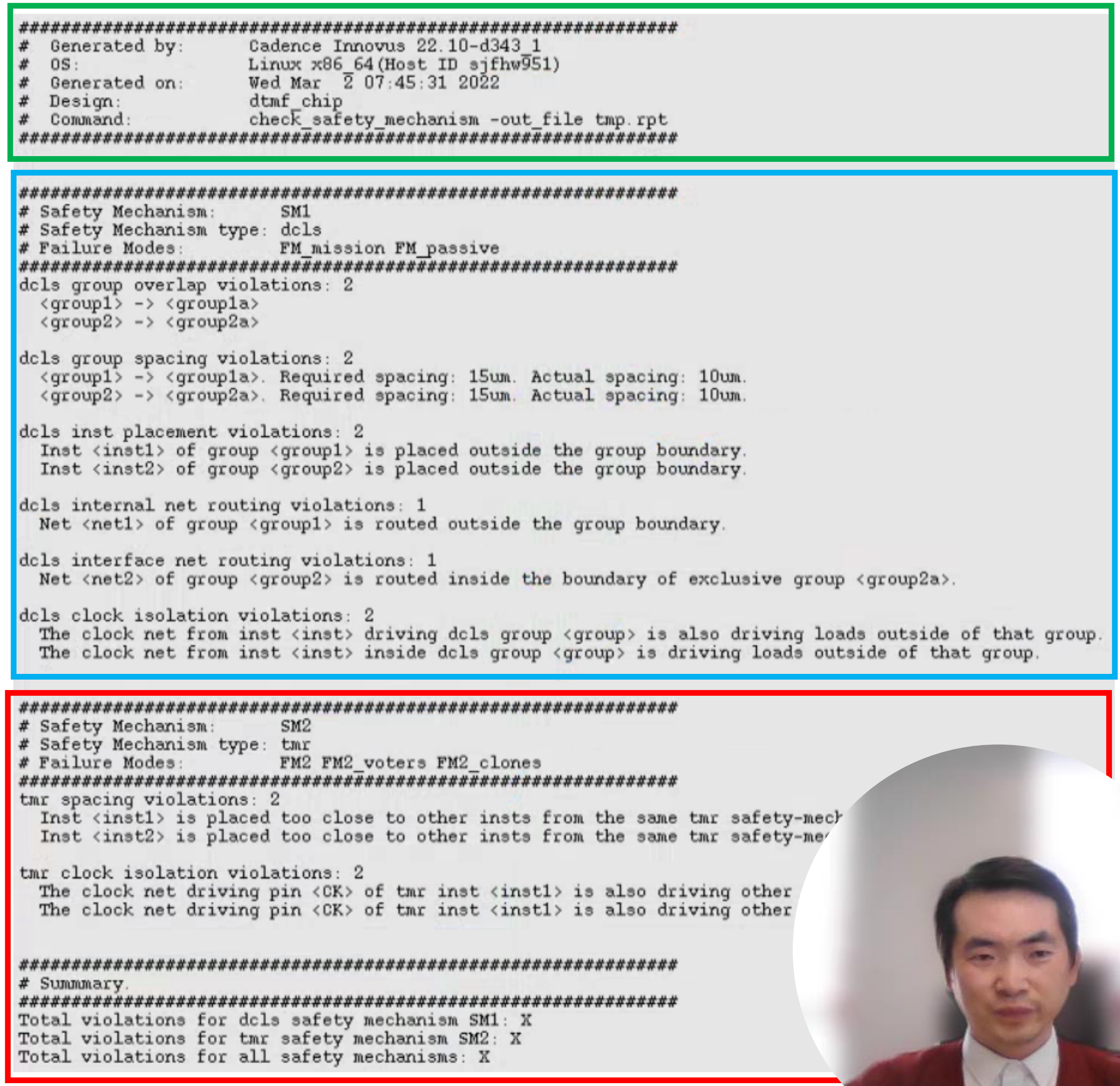


cadence

SAFETY

Unified Safety Reporting

- Single command to check all safety mechanisms
  - check_safety_mechanism
- Single report for all safety mechanisms
  - DCLS: placement, routing, clock isolation
  - TMR: placement, clock isolation



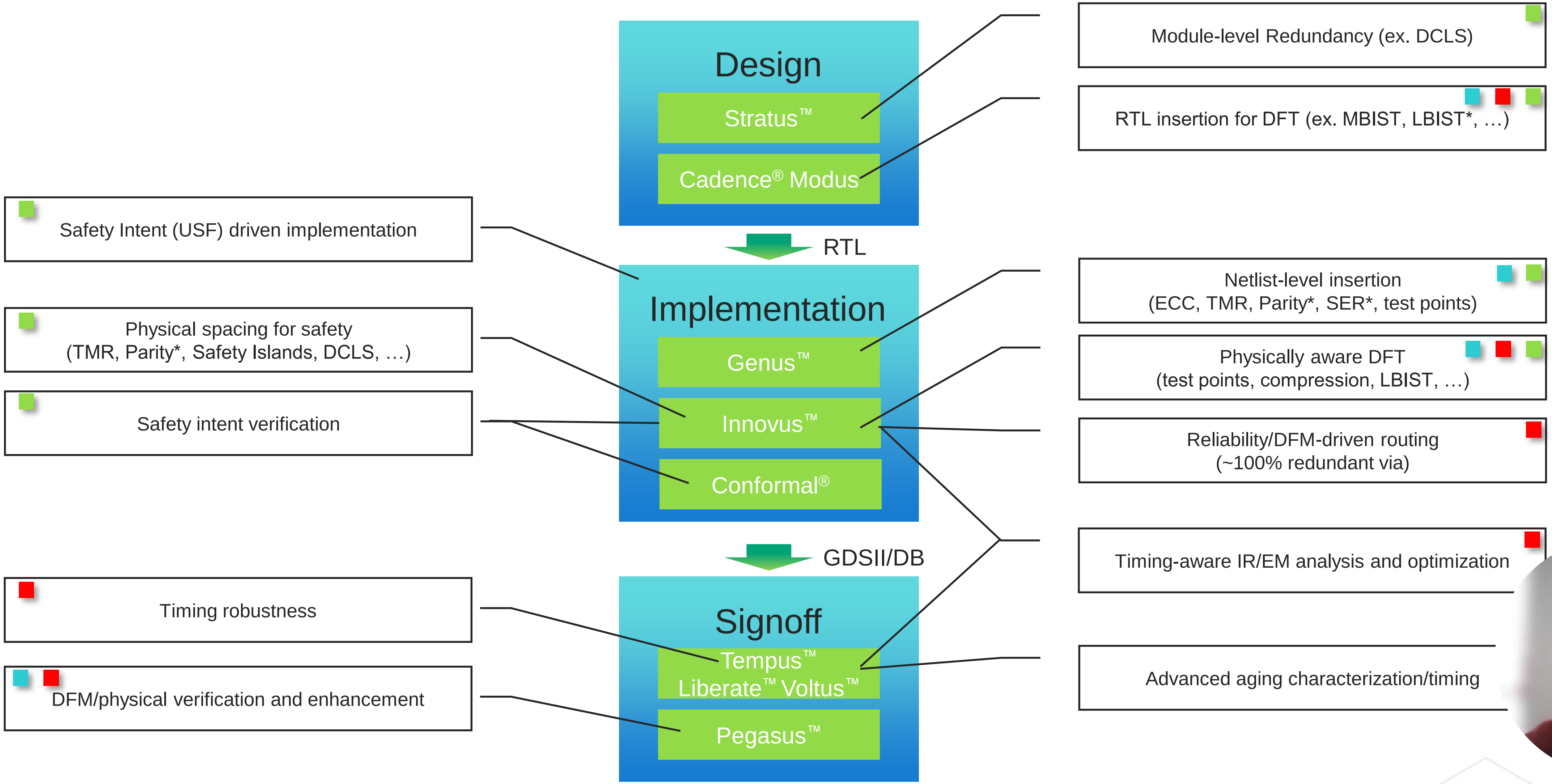
```
###############################################################
#   Generated by:          Cadence Innovus 22.10-d343_1
#   OS:                    Linux x86_64(Host ID sjfhw951)
#   Generated on:          Wed Mar  2 07:45:31 2022
#   Design:                dtmf_chip
#   Command:               check_safety_mechanism -out_file tmp.rpt
###############################################################
```

```
###############################################################
# Safety Mechanism:        SM1
# Safety Mechanism type:   dcls
# Failure Modes:           FM_mission FM_passive
###############################################################
dcls group overlap violations: 2
  <group1> -> <group1a>
  <group2> -> <group2a>

dcls group spacing violations: 2
  <group1> -> <group1a>. Required spacing: 15um. Actual spacing: 10um.
  <group2> -> <group2a>. Required spacing: 15um. Actual spacing: 10um.

dcls inst placement violations: 2
  Inst <inst1> of group <group1> is placed outside the group boundary.
  Inst <inst2> of group <group2> is placed outside the group boundary.

dcls internal net routing violations: 1
  Net <net1> of group <group1> is routed outside the group boundary.

dcls interface net routing violations: 1
  Net <net2> of group <group2> is routed inside the boundary of exclusive group <group2a>.

dcls clock isolation violations: 2
  The clock net from inst <inst> driving dcls group <group> is also driving loads outside of that group.
  The clock net from inst <inst> inside dcls group <group> is driving loads outside of that group.
```

```
###############################################################
# Safety Mechanism:        SM2
# Safety Mechanism type:   tmr
# Failure Modes:           FM2 FM2_voters FM2_clones
###############################################################
tmr spacing violations: 2
  Inst <inst1> is placed too close to other insts from the same tmr safety-mech
  Inst <inst2> is placed too close to other insts from the same tmr safety-me

tmr clock isolation violations: 2
  The clock net driving pin <CK> of tmr inst <inst1> is also driving other
  The clock net driving pin <CK> of tmr inst <inst1> is also driving other


###############################################################
# Summary.
###############################################################
Total violations for dcls safety mechanism SM1: X
Total violations for tmr safety mechanism SM2: X
Total violations for all safety mechanisms: X
```

Automotive Electronics Implementation Summary

🟦 QUALITY 🟥 RELIABILITY 🟩 SAFETY



Design
- Stratus™
- Cadence® Modus

RTL

Implementation
- Genus™
- Innovus™
- Conformal®

GDSII/DB

Signoff
- Tempus™ Liberate™ Voltus™
- Pegasus™

🟩 Safety Intent (USF) driven implementation

🟩 Physical spacing for safety (TMR, Parity*, Safety Islands, DCLS, …)

🟩 Safety intent verification

🟥 Timing robustness

🟦 🟥 DFM/physical verification and enhancement

Module-level Redundancy (ex. DCLS) 🟩

RTL insertion for DFT (ex. MBIST, LBIST*, …) 🟦 🟥 🟩

Netlist-level insertion (ECC, TMR, Parity*, SER*, test points) 🟦 🟩

Physically aware DFT (test points, compression, LBIST, …) 🟦 🟥 🟩

Reliability/DFM-driven routing (~100% redundant via) 🟥

Timing-aware IR/EM analysis and optimization 🟥

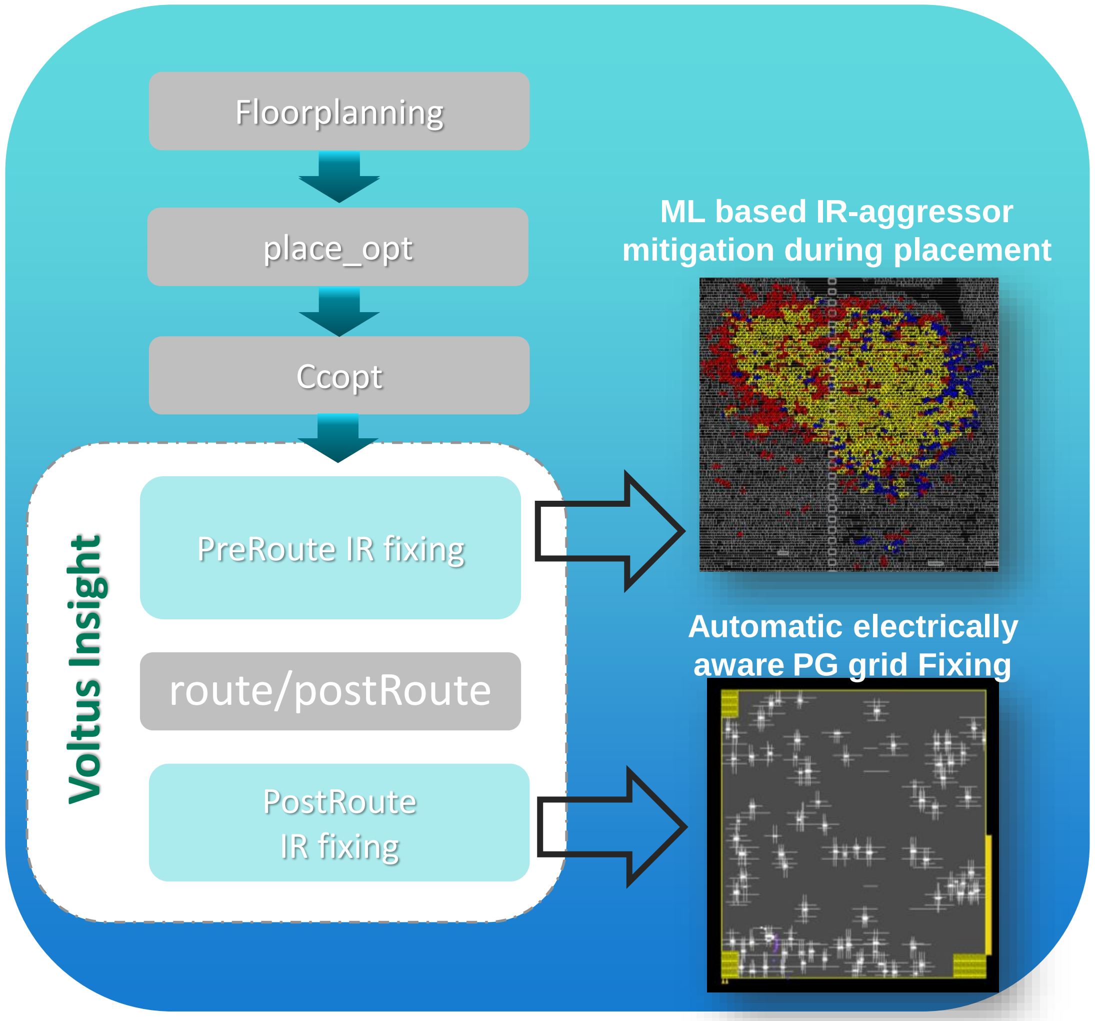
Advanced aging characterization/timing

cādence®

RELIABILITY

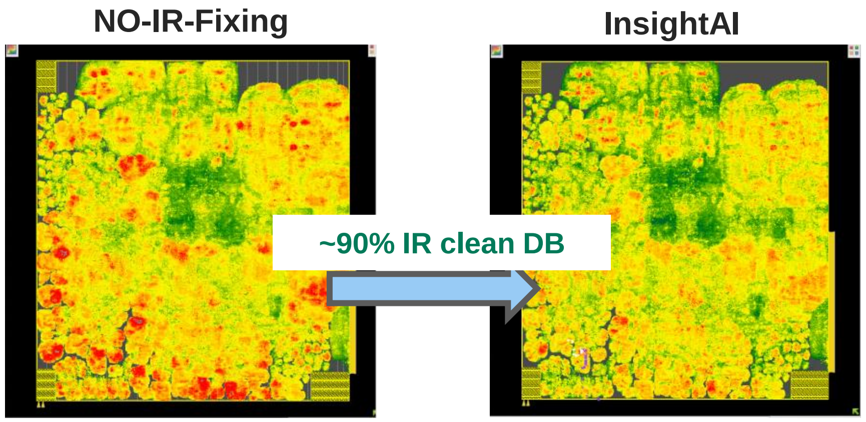
Reliability - Voltus InsightAI

InsightAI within Innovus Digital Implementation Flow



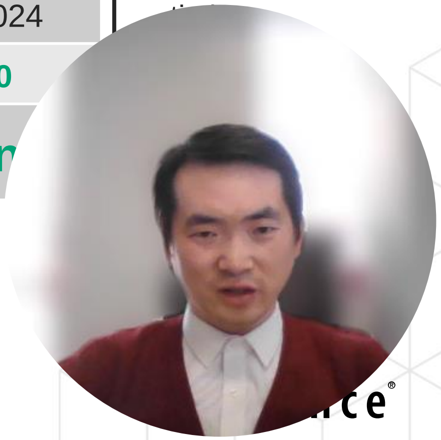
ML based IR-aggressor mitigation during placement

Automatic electrically aware PG grid Fixing



Tech Node		N3E	
Block Type		CPU	
		No IR Fixing	InsightAI
Utilization		58.297	58.059
Timing (TNS)		-90.931	-93.210
Power (mW)		2133.547	2140.024
IR-Drop errors	Total	13226	1330
	InsightAI Advantage	~90% IR clean	

Similar
Area , power ,

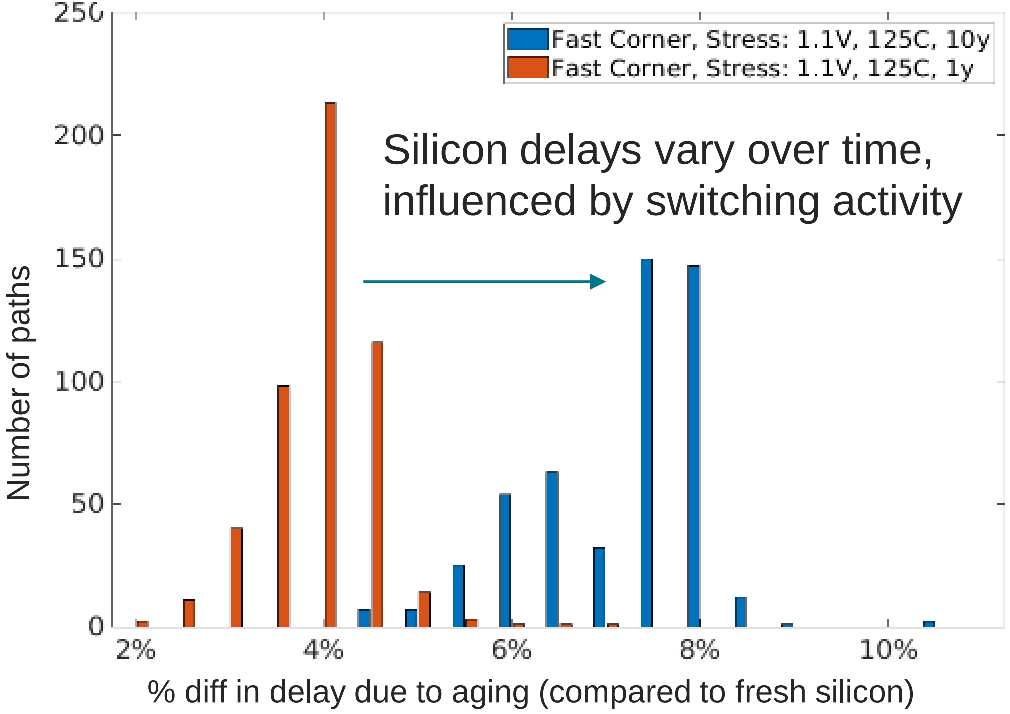


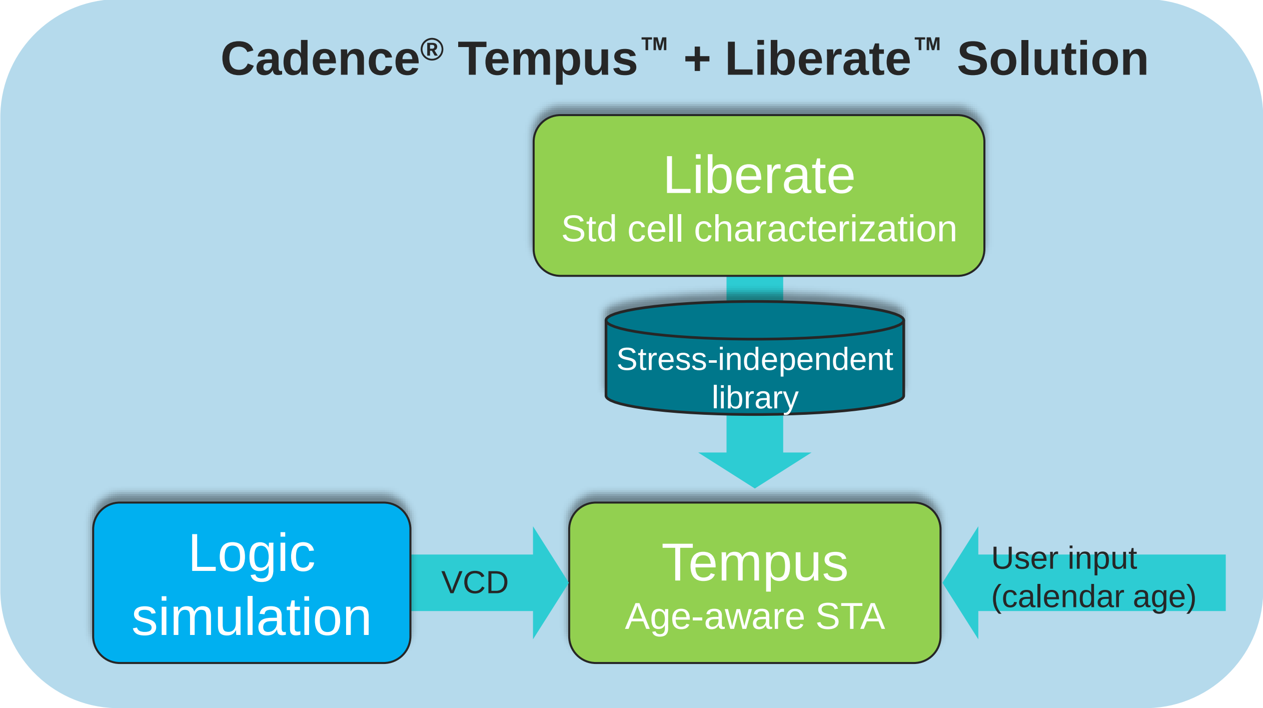
8-10X improvement in IR-Closure cycle

RELIABILITY

Reliability - Tempus Aging-Aware Robustness Solution

Aging impact on delay, advanced node





Cadence® Tempus™ + Liberate™ Solution

Liberate
Std cell characterization
Stress-independent library
Logic simulation
VCD
Tempus
Age-aware STA
User input (calendar age)

- Existing aging solutions are expensive and inaccurate
  - Derate-based flow: too inaccurate, overly pessimistic
  - Fixed-age libs: not accurate enough, costly to characterize
- Cadence aging solution addresses both co
  - Single stress-independent library characte
  - Switching activity-driven non-uniform agir

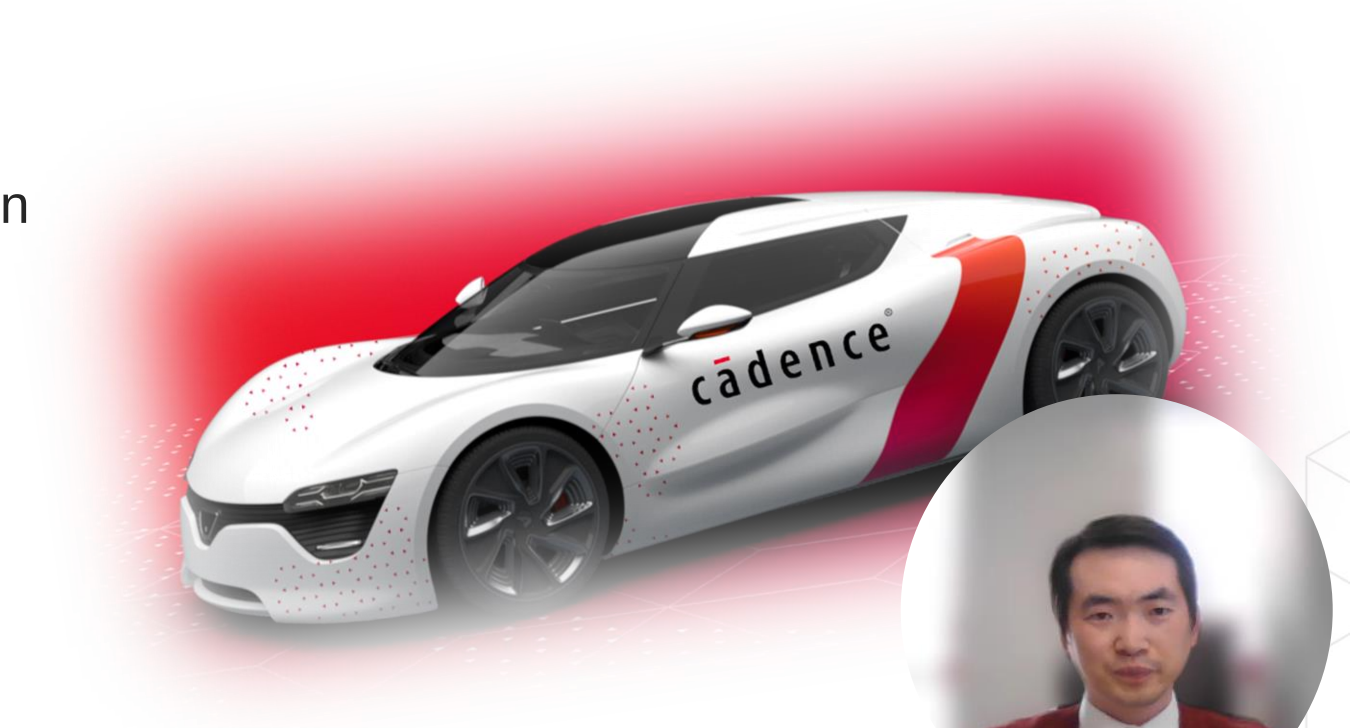
Accurate Analysis → Avoid Over-Design → Improve PPA

Summary

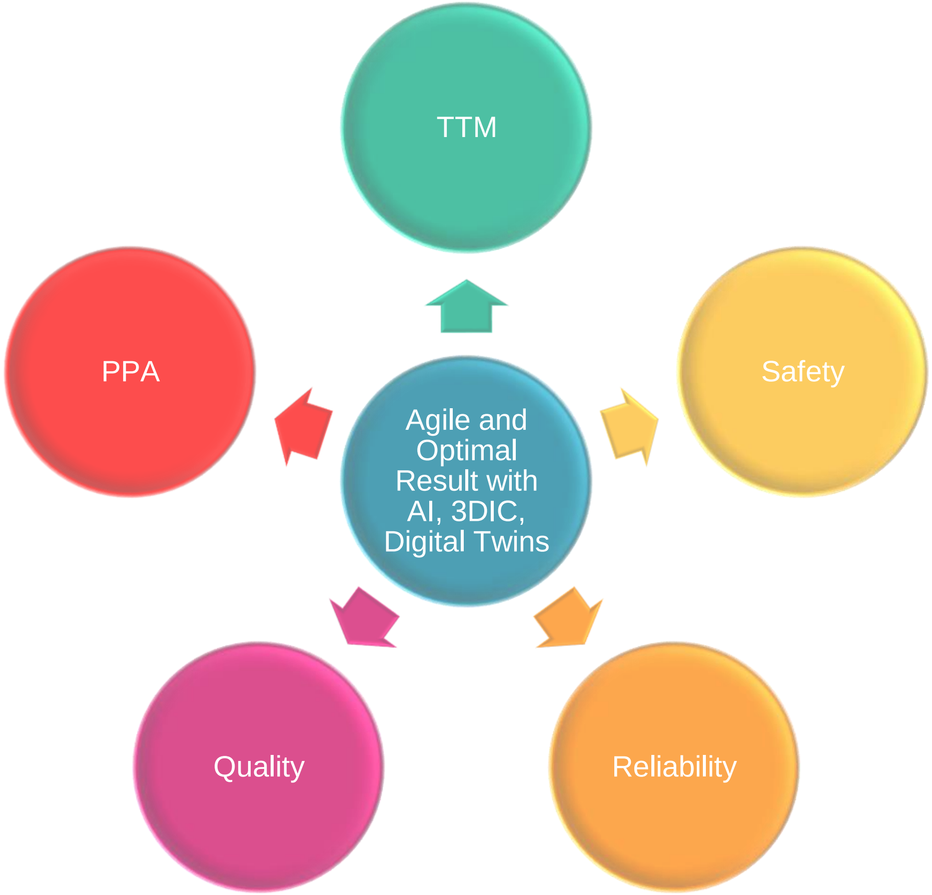
- Automotive Electronics Physical Design is demanding and challenging

- Various new technologies have emerged

- Blessed with powerful EDA products, we can achieve automotive electronics design of excellence!



Opportunities?



cādence®