

Difficulties in Designing in Advanced Technologies

GENNERET Bertrand

Apr., 2018

Agenda

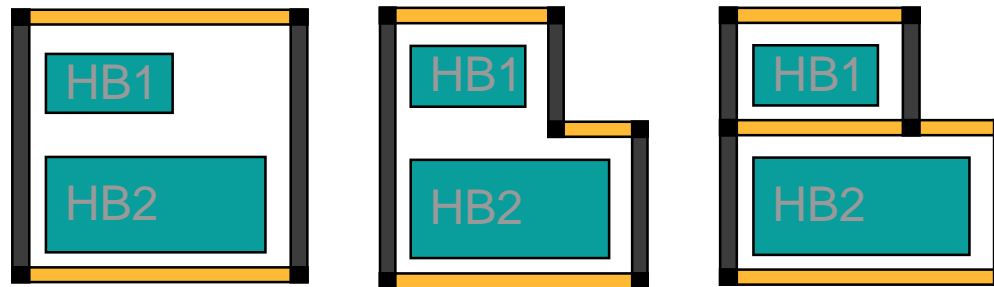
- **Block level analysis**
 - Floorplan / placement / clock tree for advanced nodes
 - New parameters to consider for routing
 - Timing and power analysis for advanced nodes
- **Top level analysis**
 - Handling partitions
 - Assemble design
 - System on chip (IP/Analog/Digital) > BigD / LittleA design methodology
- **New PDK and Tool requirements**
- **Problems seen with cern designs**
 - Radiation environment constraints
 - Flat timing/power analysis for a digital on top design with analog blocks
 - Timing convergence
- **Q&A**



Block level analysis

Floorplanning

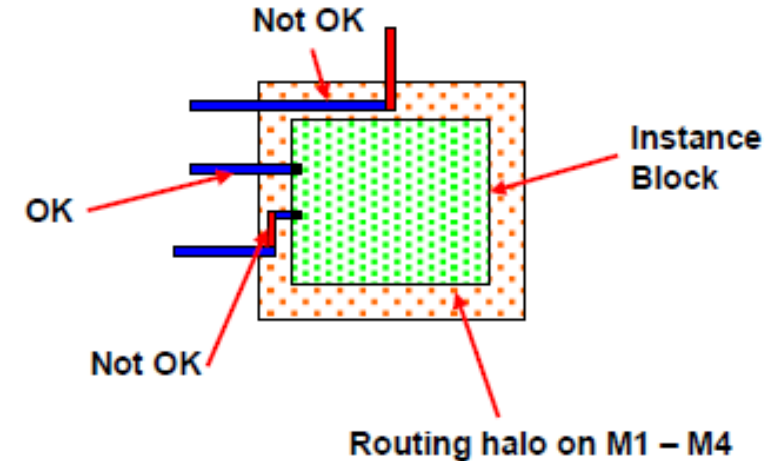
- Crosstalk impact is growing with advanced nodes, prevention becomes mandatory
 - To prevent signal integrity issues, you should define routing halo around blocks
- Block rings should be used to prevent irdrop issues. When generating rings, advanced options can be necessary to limit routing issues.



Case1: Square polygon

Case2:
Orthogonal polygon

Case 3:
Orthogonal polygon with shared edges

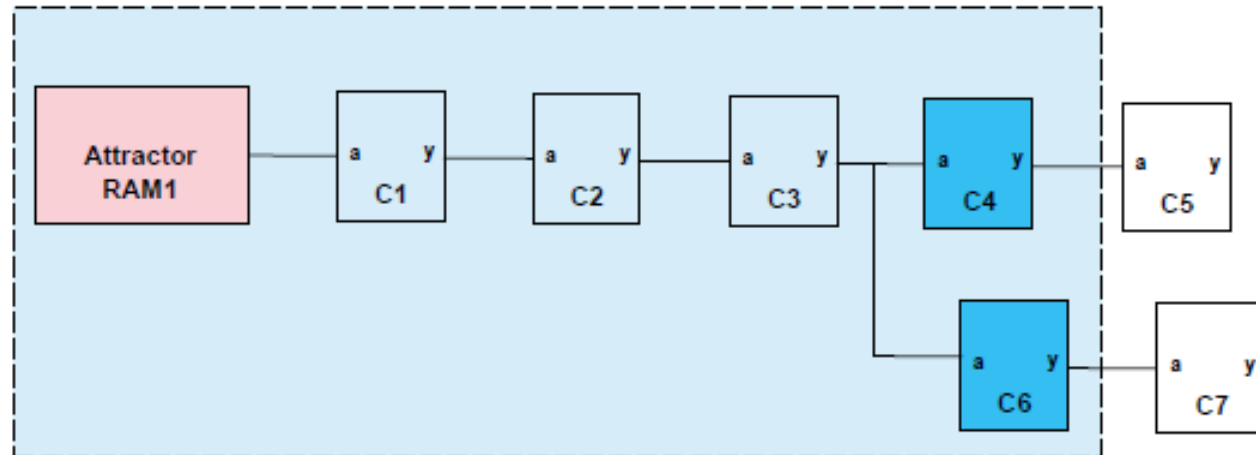


■ M1 routing
■ M2 routing

```
Commands  
addRoutingHalo  
deleteRoutingHalo
```

Placement

- It is recommended to prevent cells or pins from being placed under power stripes, thus avoiding pin access problems.
- For the stdcells with important number of pins it may be useful to keep empty space (filler cells) on each side of the stdcell.
- For particular time critical path you can force a max distance to an attractor cell

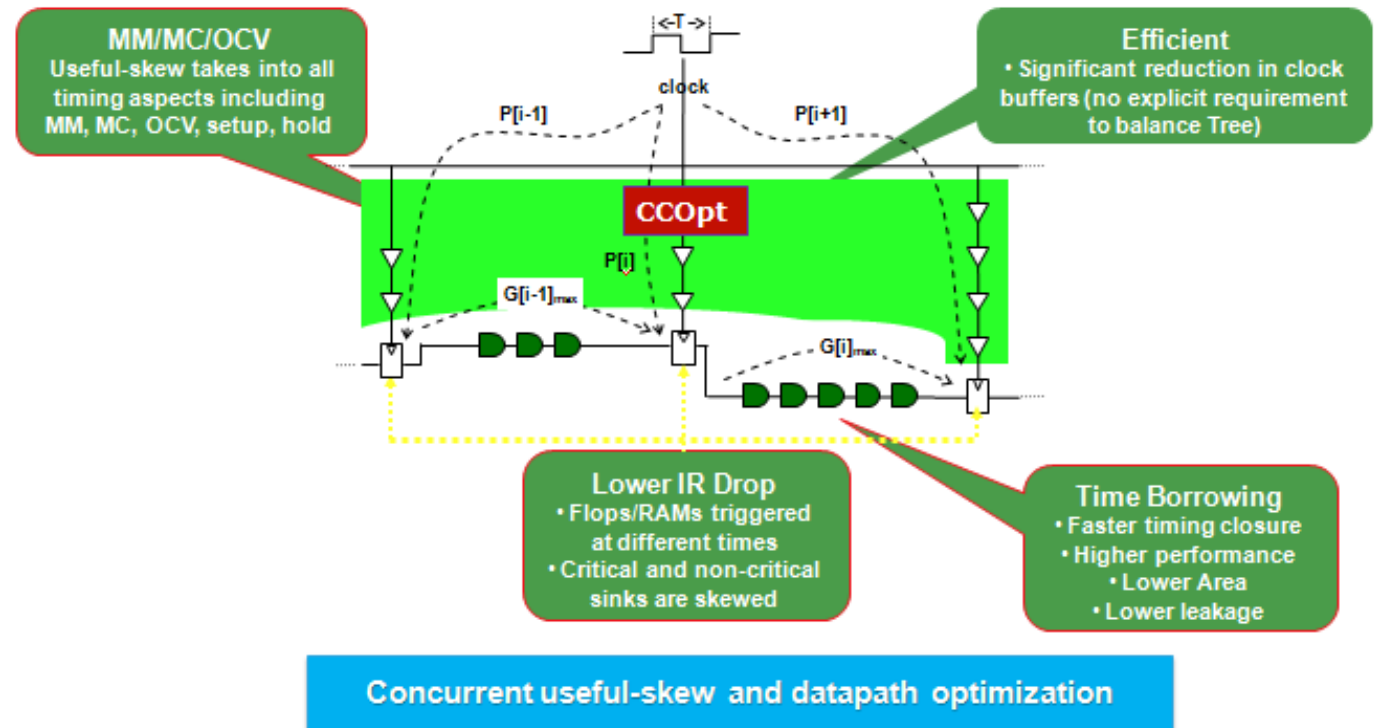


Clock tree

- It is recommended to use a 2x spacing 2x width non default rule for clock routing
- Clock shielding is recommended
- For better timing/power performance a new engine is used (ccopt)

This engine offers concurrent useful-skew and timing optimization.

Superior clock and datapath implementation

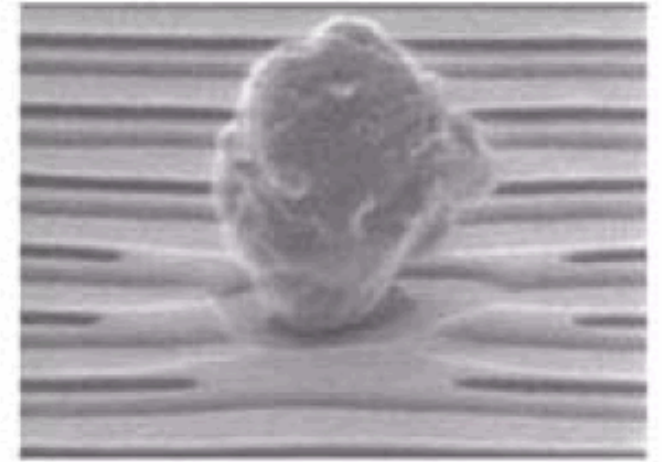


Routing: DRC / LVS complexity increase

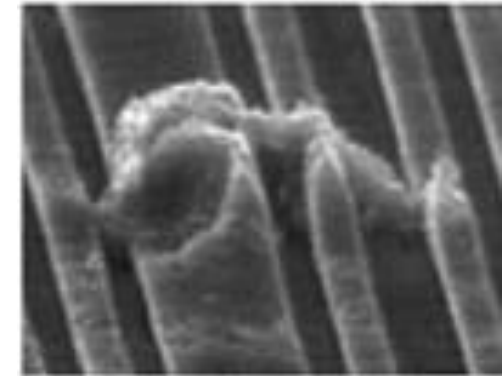
40nm and Above	32nm/28nm	20nm
Total > 20 rule types in tech lef	Total > 50 rule types in tech lef	Total > 120 rule types in tech lef
Simple two objects spacing	Context-based multi objects	Context-based multi objects on multiple masks, negative rules (deleting an object triggers violation)
		DPT
Placement needs to handle global congestion	Placement needs to handle global and local congestion, such as same net on abutted cells (MAR or minimum area rule vias)	Placement needs to handle color conflict, global, local congestion, such as same net on abutted cells, pin access
	Power routing needs to handle multi classes cuts	Power routing needs to handle complex rules, such as forbidden spacing, enclosure parallel
TR/GR need to model congestion accurately, but difficult to model some rules, such as EOL, MAR	TR/GR need to model congestion accurately, but difficult to model context- based multi objects rules, such as conditional MAR, opposite EOL, etc.	TR/GR need to model congestion accurately, but difficult to model context based multi objects rules, such as DPT, enclosure parallel, etc.
DR rips-up and re-routes to fix DRC violations	DR needs to surgically fix some of the context based multi objects violations	DR needs to resolve the pin accessibility “puzzle” due to complex rules and local pin density congestion

Routing:

- At 65nm and below, you need to take care of yield. Size of particles become non negligible compared to net size.
- To limit such problem there are options in router to:
 - Increase multicut via ratio
 - spread wires
 - widen wires



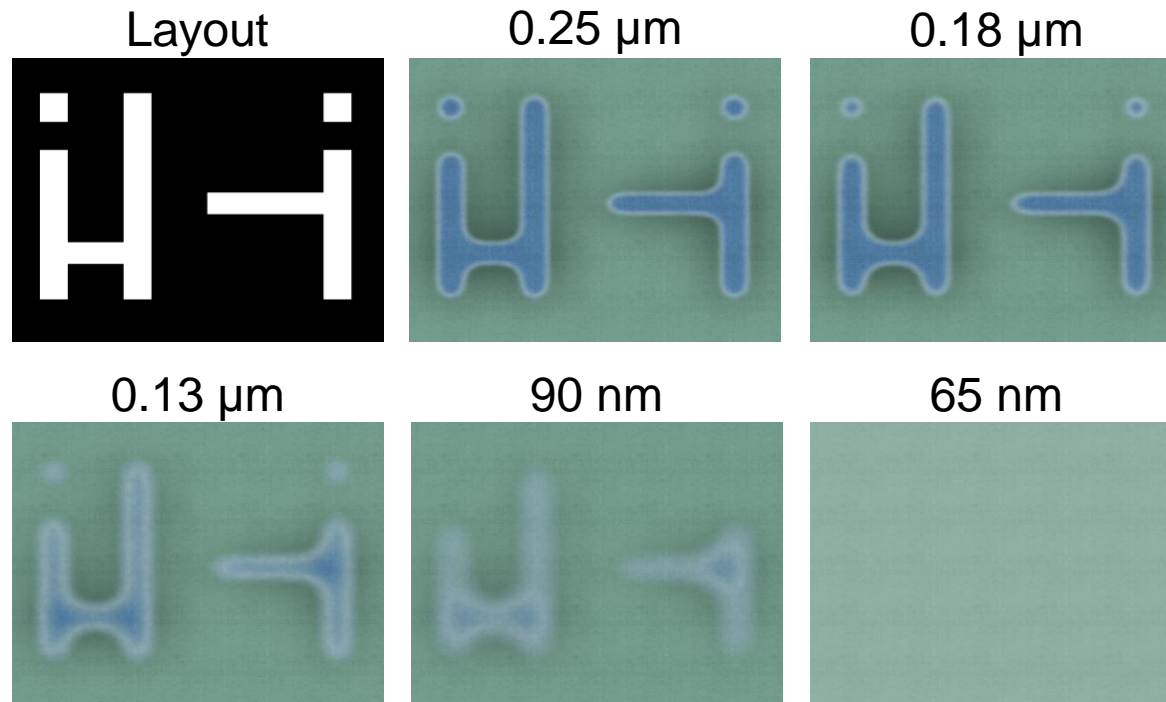
Some particles cause opens



Short Failure

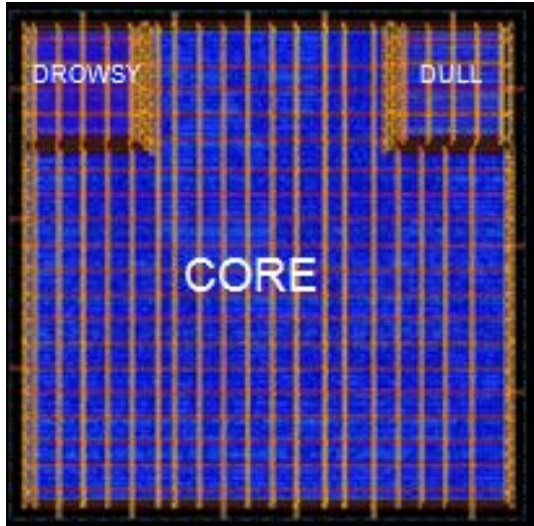
Routing: Lithography prevention / fixing

- For advanced nodes, due to lithography limitations (refraction,..) what you design is not exactly what you get on silicon. Router include prevention and fixing flow.



Timing analysis:

- For advanced nodes there are multiple corners / modes to analyze



Mode	Core	Drowsy	Dull
Baseline	1.08V 125MHz	1.08V 125 MHz	1.08V 125 MHz
Slow	1.08V 125MHz	1.08V 125MHz	0.9V 66MHz
Standby	0.0V	1.08V 125MHz	0.0V

Multiple constraints (.sdc)

Example: baseline.sdc, ios.sdc, dull.sdc, drowsy.sdc

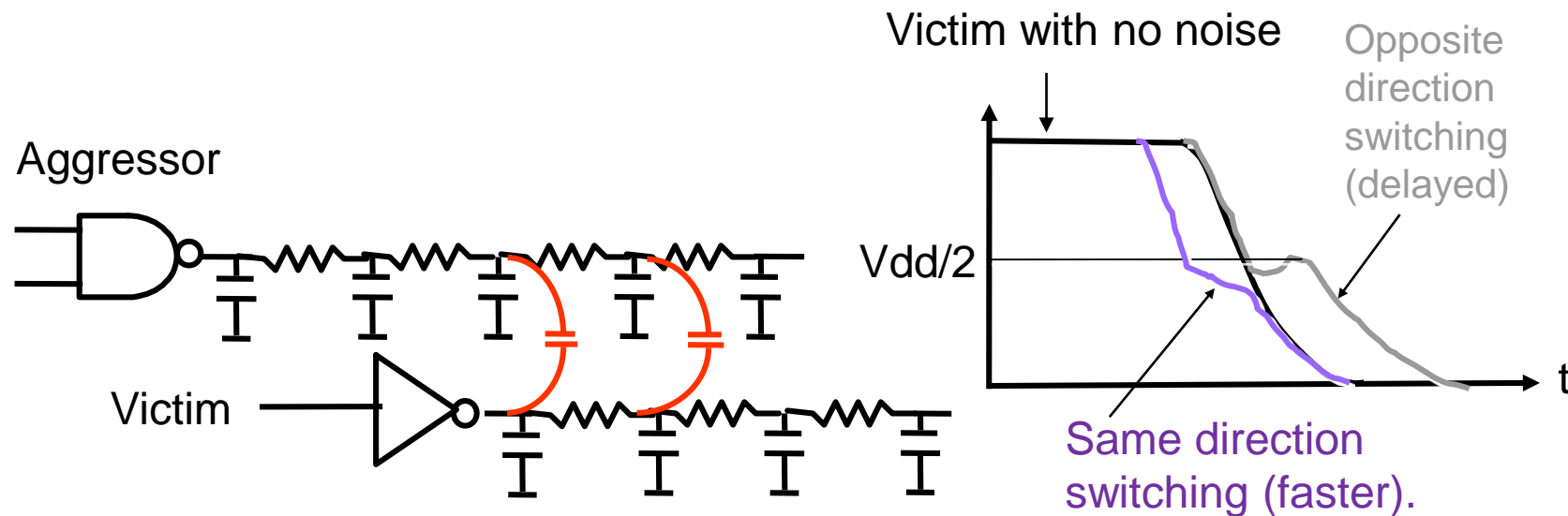
Libraries

stdcell_1.08sl.lib,
stdcell_0.9sl.lib,
stdcell_1.08fs.lib,
stdcell_0.9fs.lib

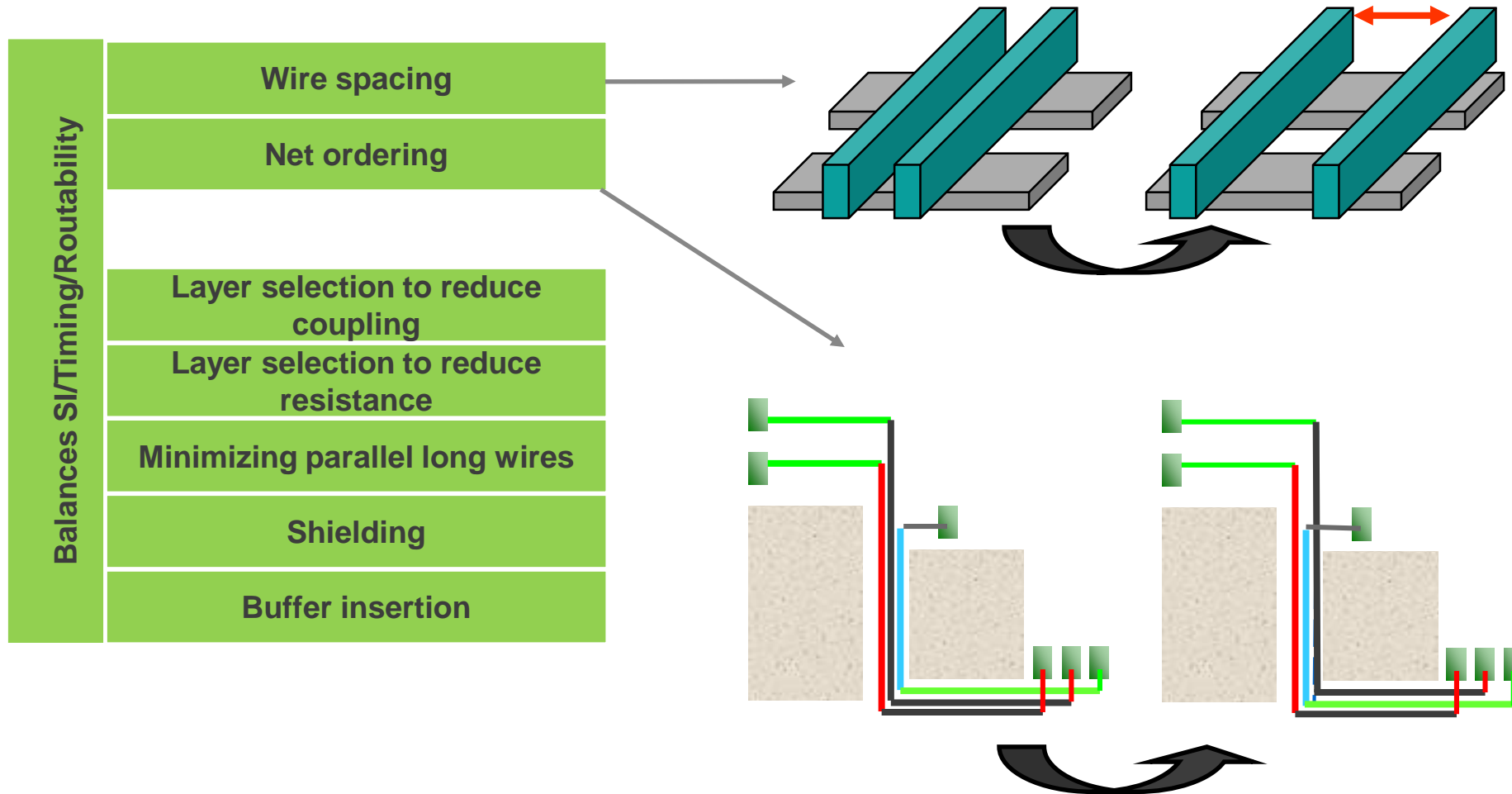
- Multiple modes need to be analyzed/optimized for multiple corners
- Setup analysis for (WC, 1,125C) corner

Timing analysis:

- Noise libraries (.cdb) must be taken into account for
 - Glitch violations
 - Signal Integrity issues



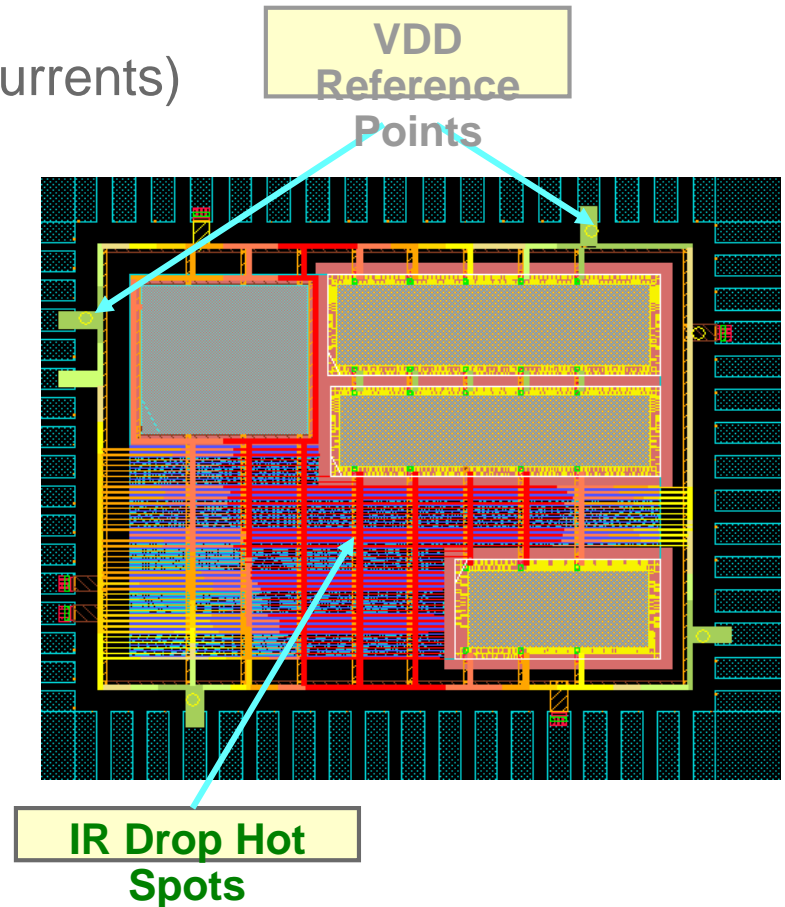
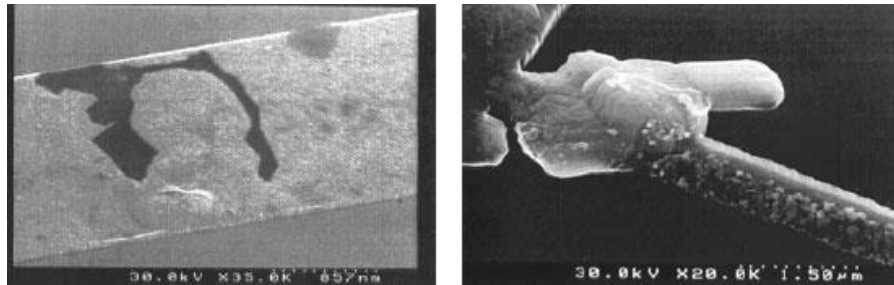
Timing analysis: crosstalk prevention / fixing



Power analysis:

- At 65nm and below, several checks must be done:
 - Power calculation based on different simulation scenario (func, test,..)
 - Irdrop analysis (static and dynamic)
 - Capacitance analysis (decap cells used to decrease peak currents)
 - Electromigration analysis
 - Power switch analysis if some blocks of the design may be turned off

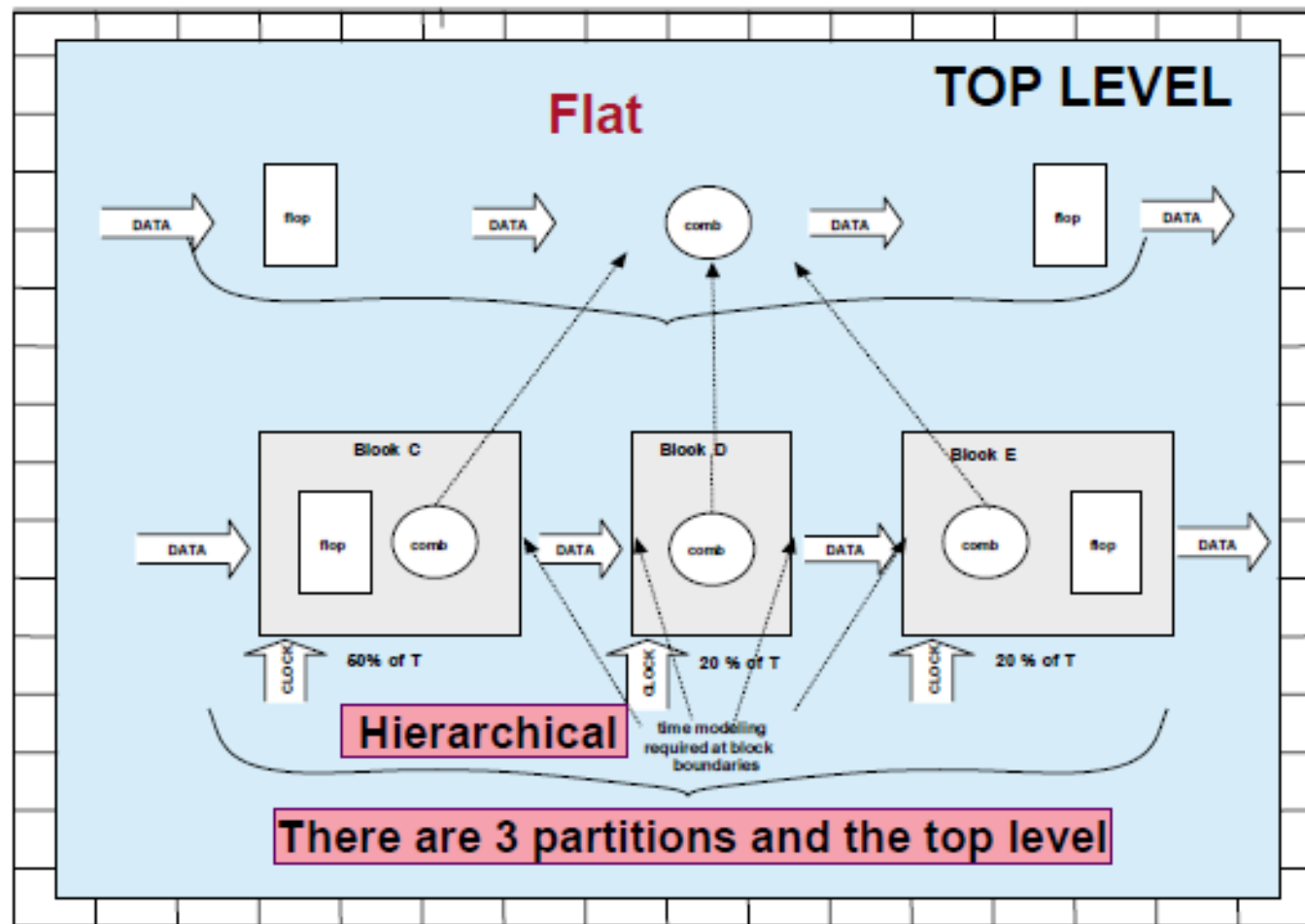
EM Failures as seen through a Scanning Electron Microscope (SEM)



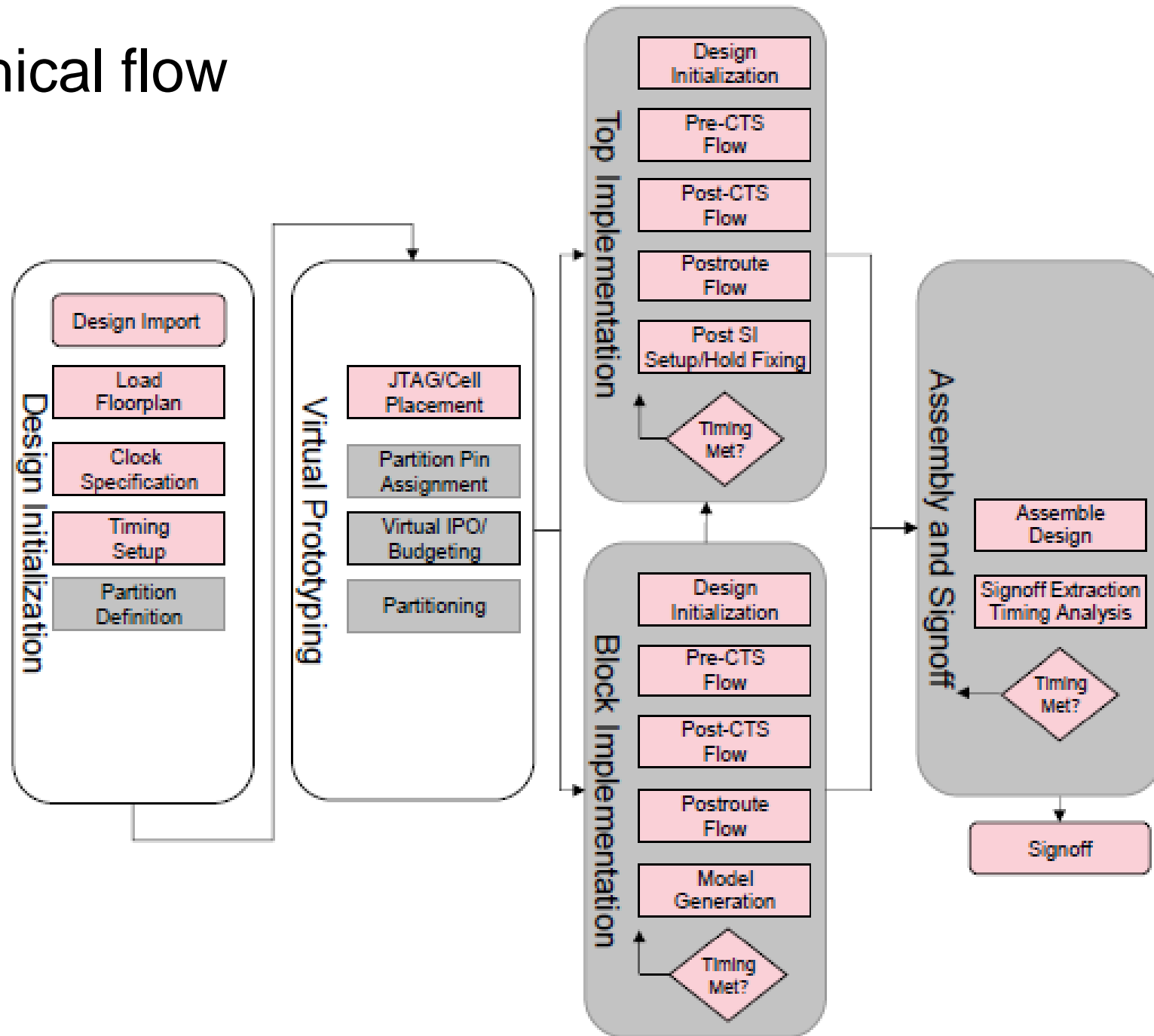


Top level analysis

An example of hierarchical design

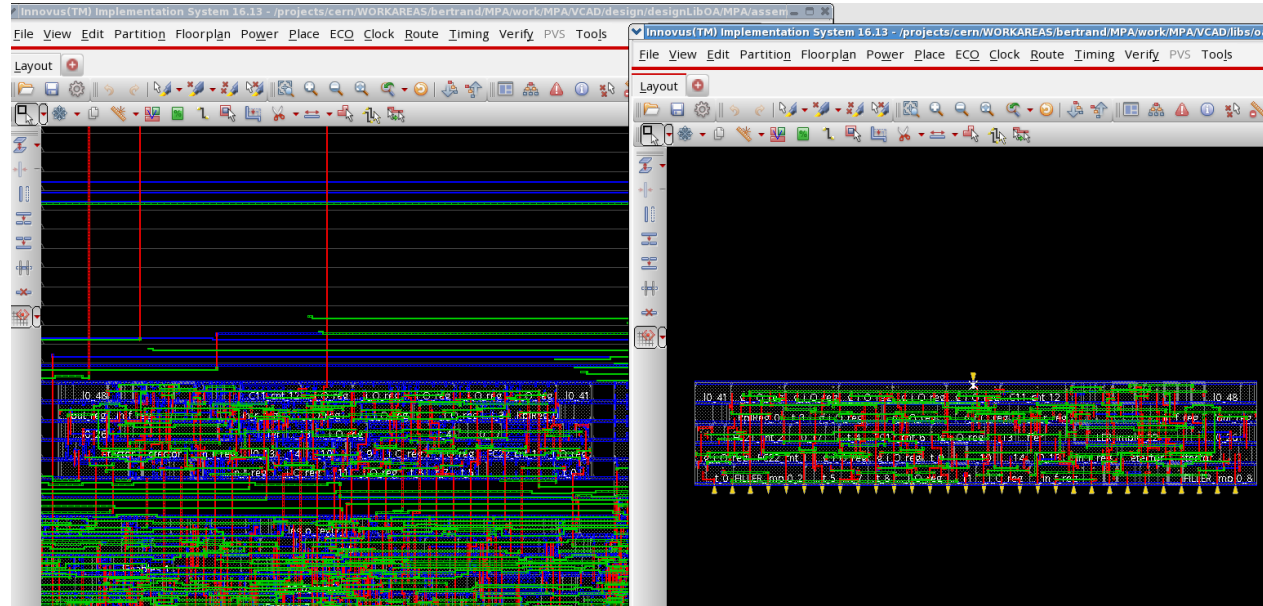


Hierarchical flow



Assemble Design

- It is when you merge the routed block into the top level



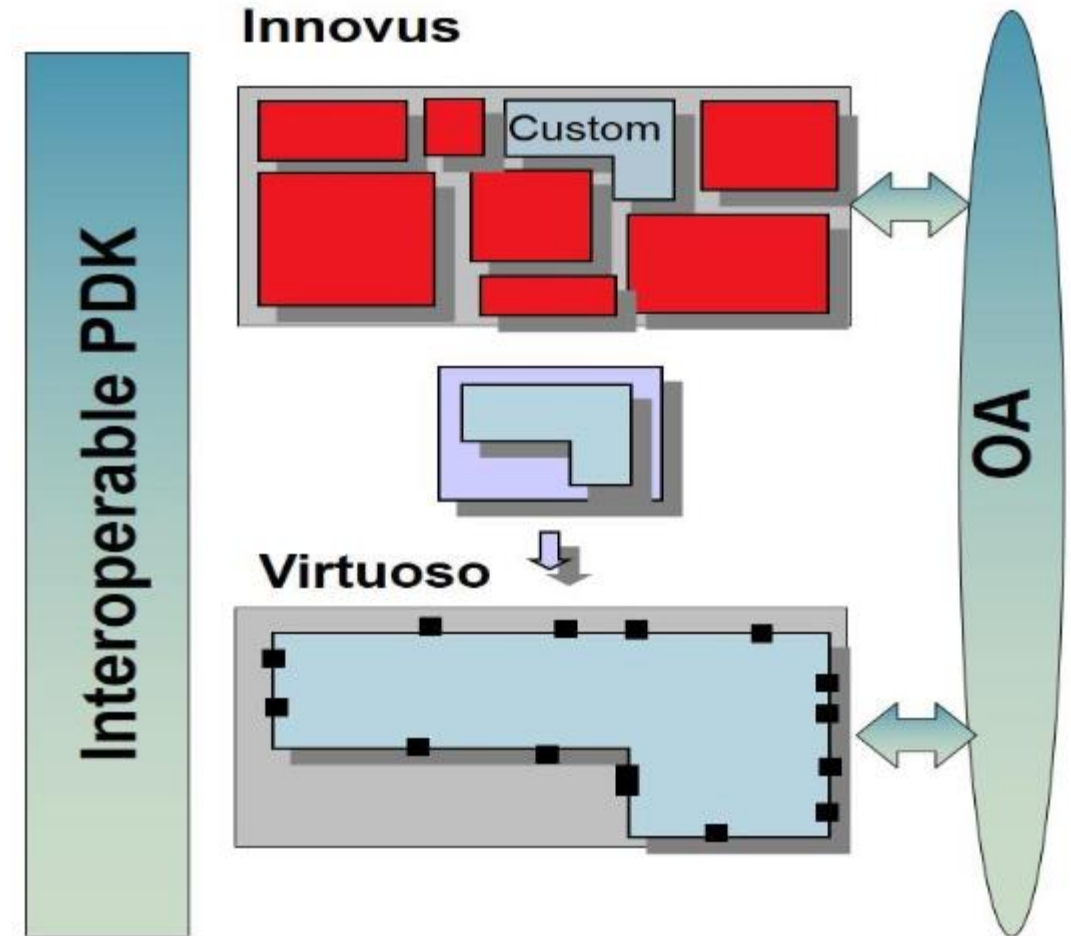
- Crash may happen if multiple module definitions of same cell are defined in top as well as in block being incremental assembled.
- The incremental capability of assembleDesign does not support blocks with mixed data types (If physical information for some blocks are saved in Innovus database while others are saved in OA database)

What to consider

- They are 3 factors to consider to go for hierarchical implementation: size, complexity, and maturity.
 - Size of the design is too big to implement it as a flat design. Placement/routing runtimes prevent easy iterations on the flow.
 - Complexity from a timing perspective (nb of corners, nb of modes)
 - Design reuse is well supported in hierarchical design
- Hierarchical flow is an advantage when the work is distributed across a wide geographic area.
- What is optimal?
 - Maximize gate count, instance count, minimize area in each block
 - Minimize pin count on each block
 - Minimize number of top level nets
 - Minimize timing critical paths at the chip level

System on chip (IP/Analog/Digital) > BigD / LittleA design methodology

- Required Data:
 - Interoperable PDK
 - Open Access database
 - Innovus/Virtuoso tools
- Below are the different steps:
 - Design import and early timing analysis in Innovus
 - Implemented the AMS block in Virtuoso
 - Create FTM (Full Timing Model) for the AMS block
 - Handling ECOs and redoing FTM analysis

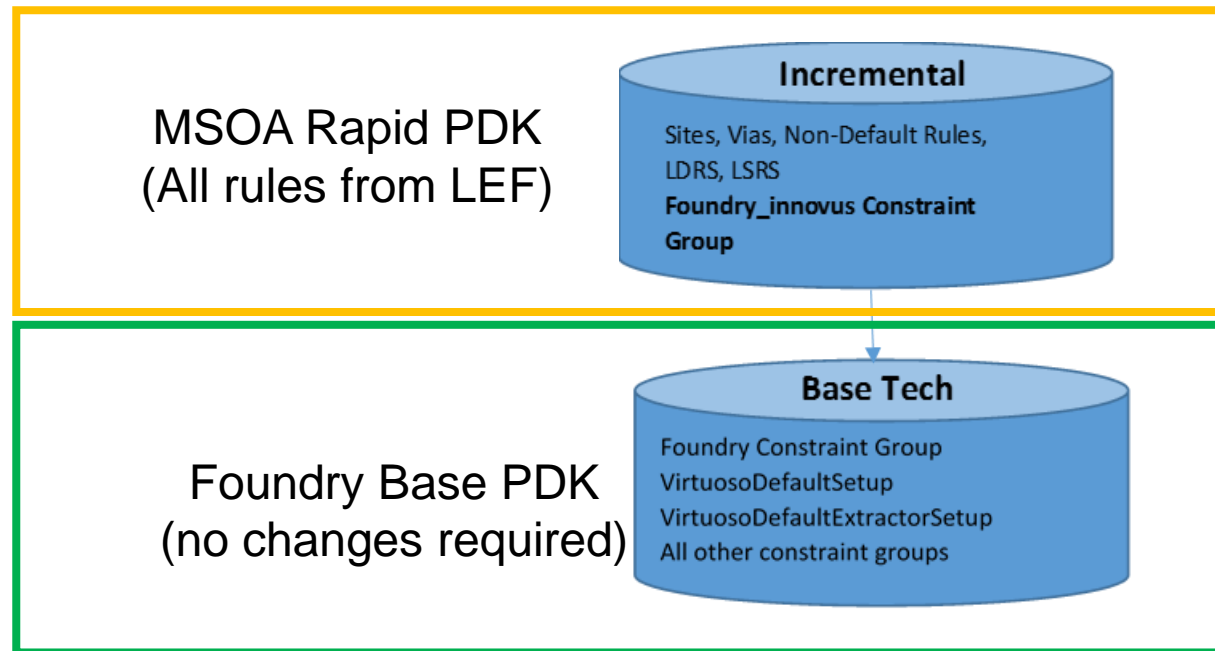




New PDK and tool requirements

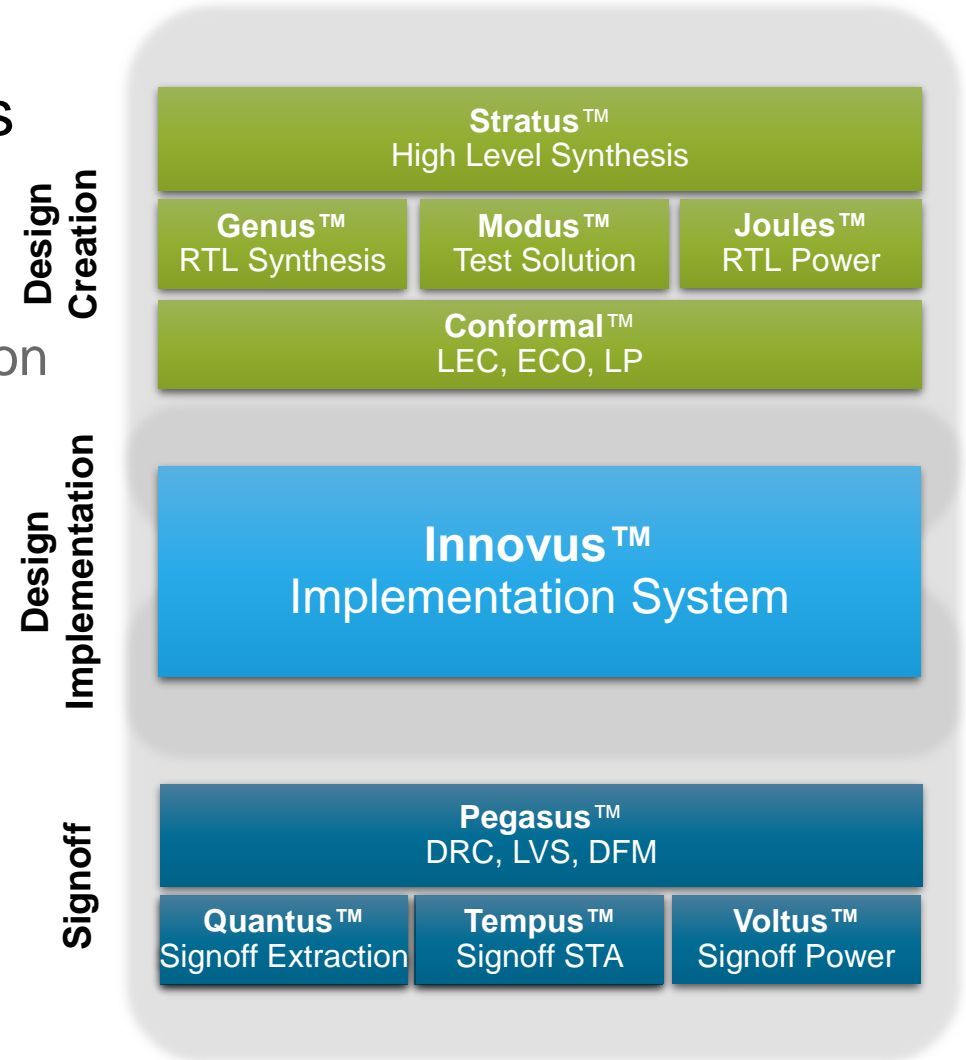
Rapid MSOA PDK

- **Use when foundry doesn't provide MSOA PDK**
- Presence of a new foundry group in the incremental tech file.
- **Innovus (18.1)** will use **foundry_innovus** constraint only
- Can be made by any user even when base PDK is read only



Recommended cadence tool versions

- Move from RTL Compiler to Genus for synthesis
 - up to 10X improvement in RTL design productivity
- Move from EDI to Innovus for P&R
 - Ccopt/gigaopt engines for clock and timing optimization
- Move from QRC to Quantus for extraction
 - Better debugging interface
- Move from ETS to Tempus for signoff timing
 - path-based analysis to improve the overall accuracy
- Move from EPS to Voltus for signoff power
 - Advanced analysis, complex em rules handling





Problems seen on cern designs

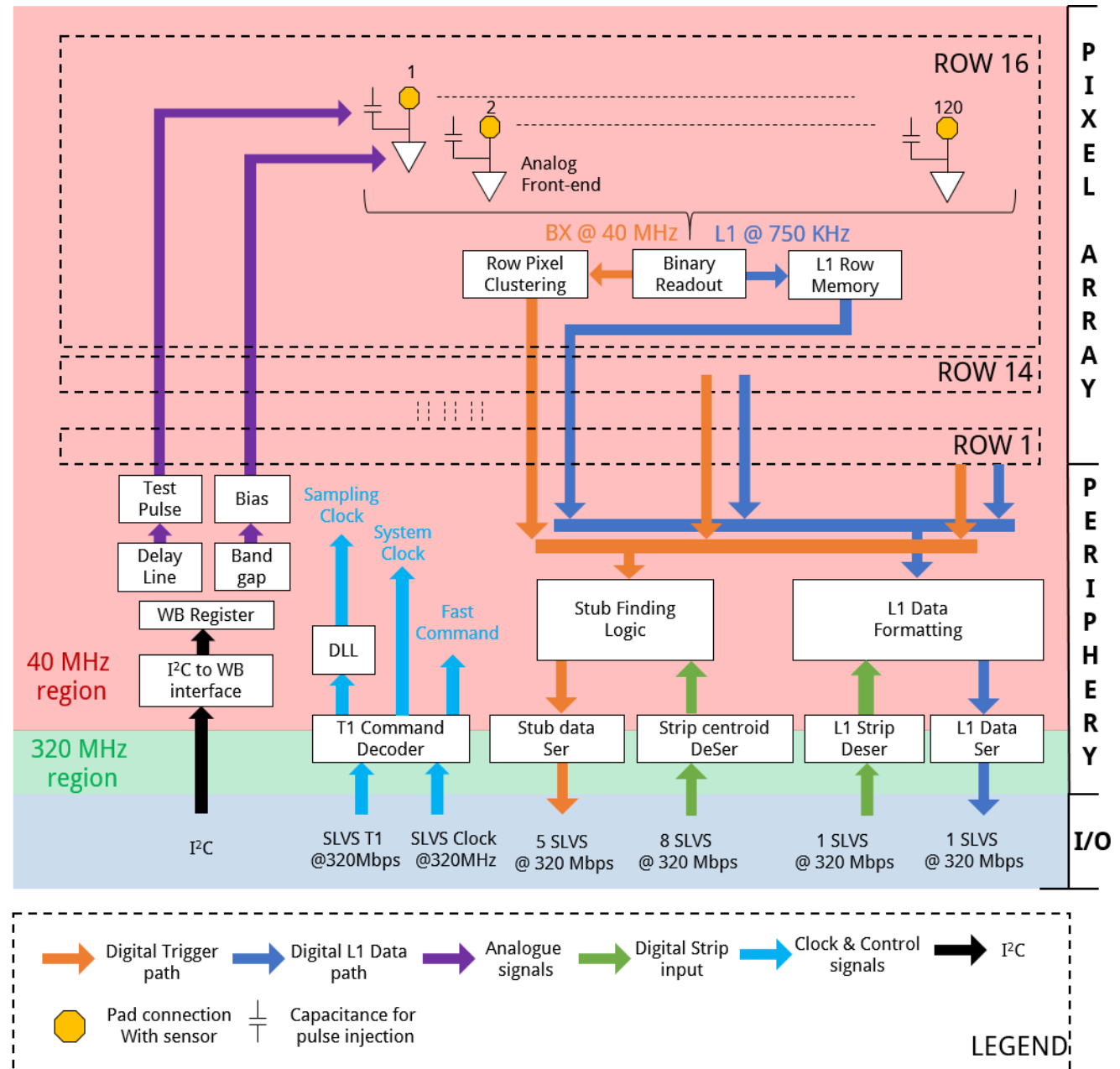
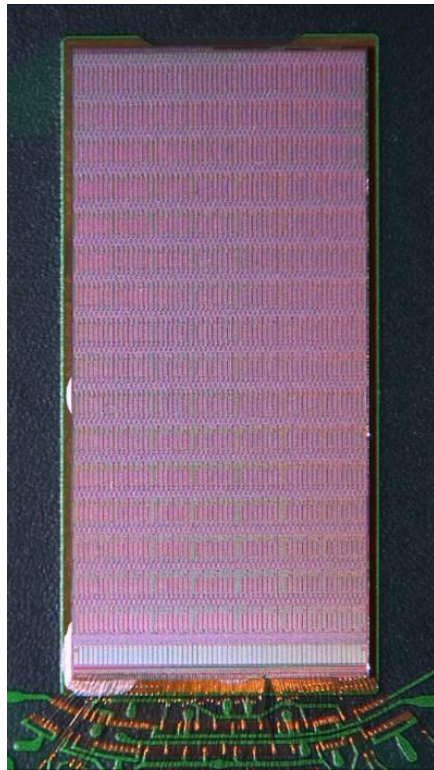
Techniques in P&R flow for radiation environment

- Use of inverters rather than buffers on critical nets (clocks, resets, scanenable)
 - Buffers are more sensitive to radiations than inverters, because of short internal net between inv stages.
- Min capacitance constraint on all nets
 - Nets with a very low capacitance will propagate a small glitch too easily
- Use of HiRel (high reliability) cellTypes for critical flops
 - The list of critical flops is identified by the designer, based on functional sim with fault injection
 - The HiRel flops are radiation-hardened, functionally equivalent but larger and slower
- Use of Triple Voting Flops
 - OK for a few flops, but too costly to use widely.

MPA project

- Pixel Strip module

Technology: TSMC 65



XL compliance issues (for power analysis with Voltus)

Block designed in Virtuoso Should be XL-Compliance

if not follow these steps:

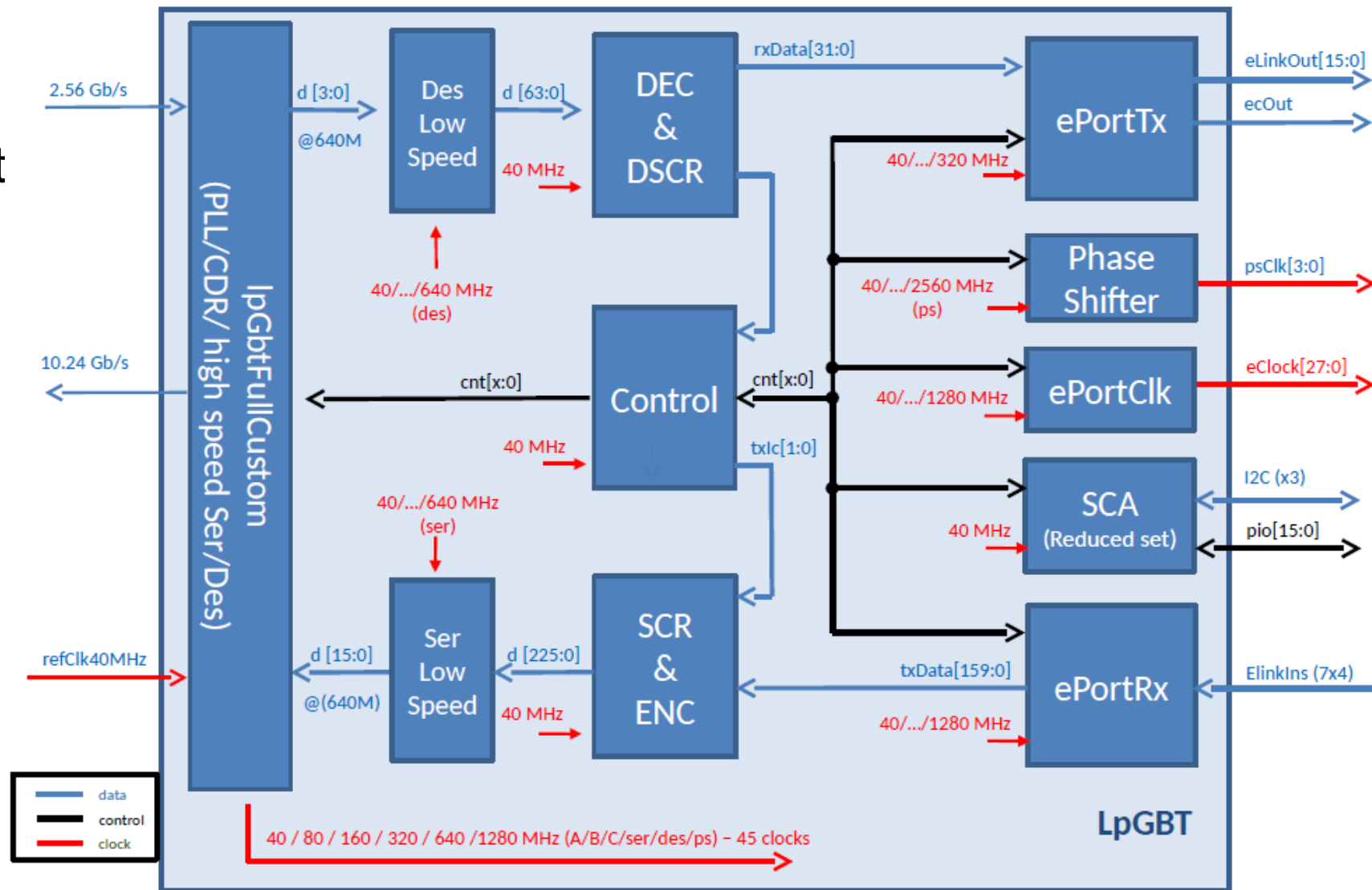
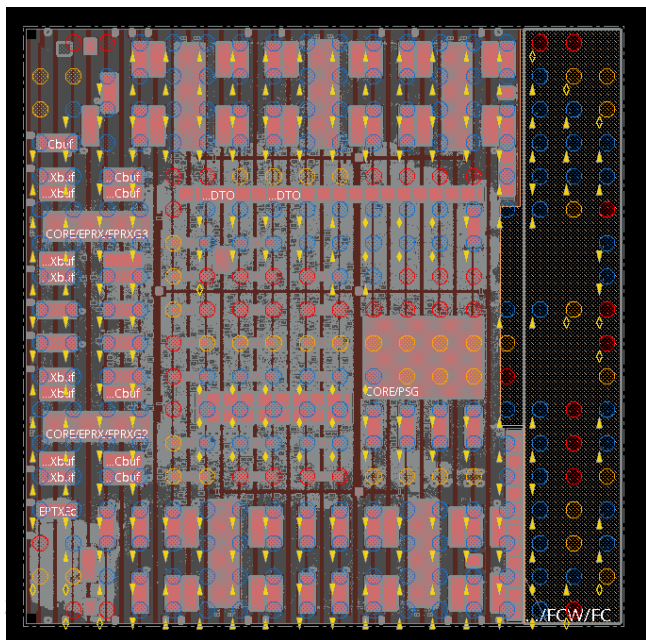
- All physical Pin MUST match schematic
- Step1 : Open TOP block and launch VLS-XL
- Step2 : Extract Connectivity
- Step3 : Flatten first level of hierarchy (level 1)
 - Select all instances
 - Preserve Pin geometries
 - **Do not preserve Pins**
 - Keep Via and Pcells
- Extract the connectivity again (Step2)
- Do that until you get the complete polygons that describe power/ground nets

IpGBT project

Design: low power GigaBit Transceiver

Technology:

TSMC 65 (1p6m3x1z1u)



© confidential.

Timing convergence issue (preCTS – postCTS)

- Extraction engines need user to define rc factors to better correlate pre CTS postRoute timing results
- User have to take care about innovus / tempus setup to ensure good correlation between tools. (delay calculator, signal integrity, extraction setup have to be align and to be setup for signoff quality results)
- Badly characterized libraries may produce a different delay calculation between clock tree builder and timing checker leading to non optimized clock tree delay.

Routability issues

- Due to highest placement density pin access may become a challenge that require some advanced options to be setup for routing.
- High number of drc rules increase the routing complexity, more iterations are required to reach drc clean layout
- Runtimes increase and require more cpu/memory usage
- Debug require more knowledge and more time investment in parsing rule manual



Q&A

cā dence[®]