

Complete DFT RTL Verification Using Formal Techniques for Complex SoCs

Arvind Jain

Edmond George

Subir K. Roy

Sundarrajan Subramanian

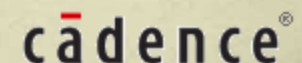
({a-jain, edmond.george, subir,
sundarrs}@ti.com)

Lopamudra Sen

(lopa@blr.interrasystems.com)

Lokesh P. Babu

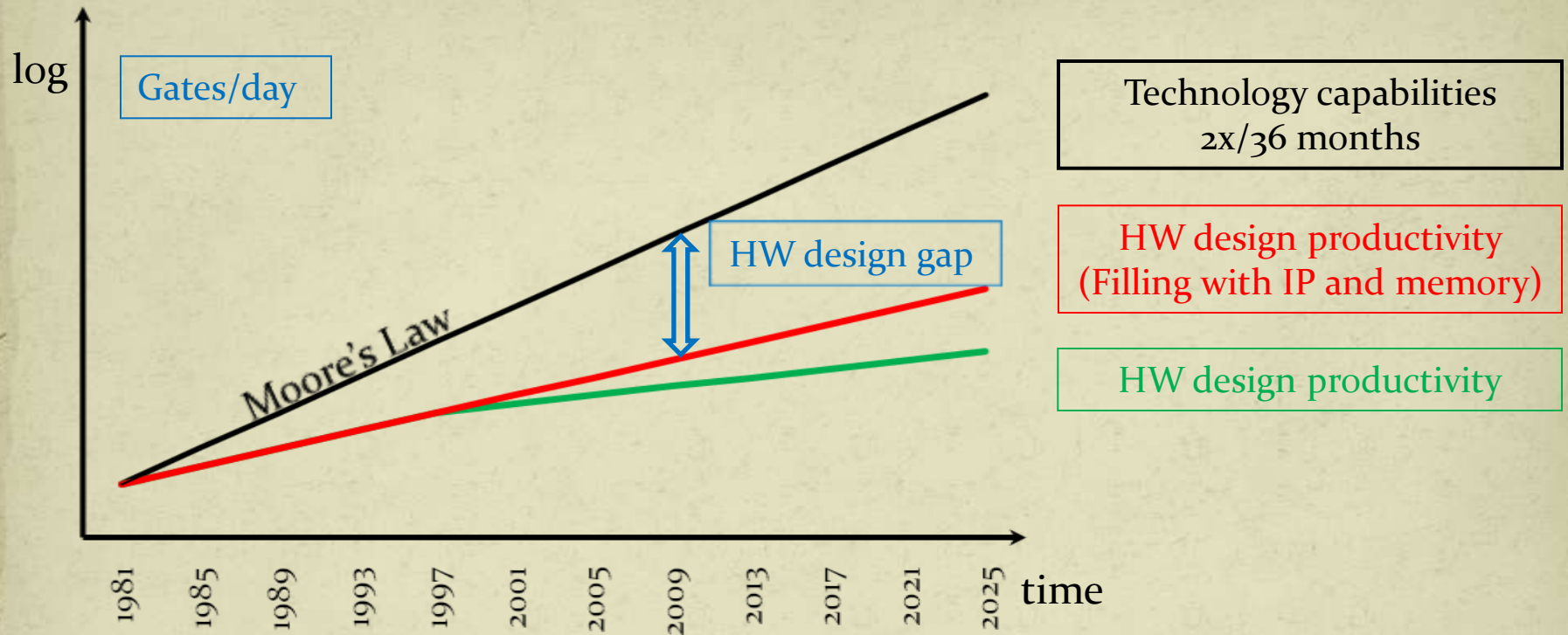
(lokeshp@cadence.com)



Outline

- Motivation
- Advantages of formal verification methods
- Scope of DFT RTL verification
- Case study1 : Test clocking verification
- Case study2 : DFT power management verification
- Limitations
- Conclusion/Future work

Productivity gap in IC design



“Cost (of design) is the greatest threat to continuation of the semiconductor roadmap”

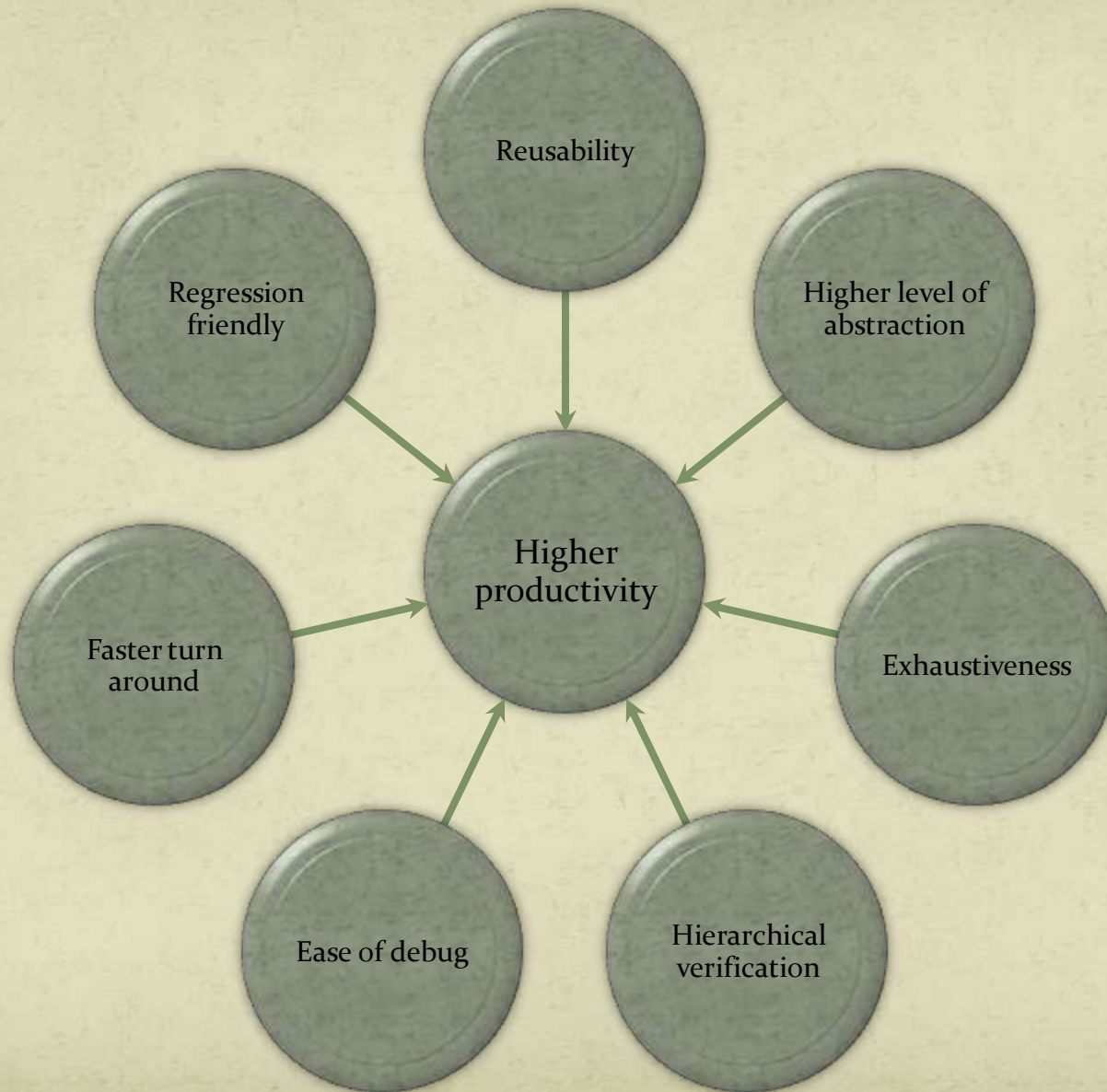
Manufacturing cycle times are measured in **weeks**, with **low uncertainty**

Design and verification cycle times are measured in **months or years**, with **high uncertainty**

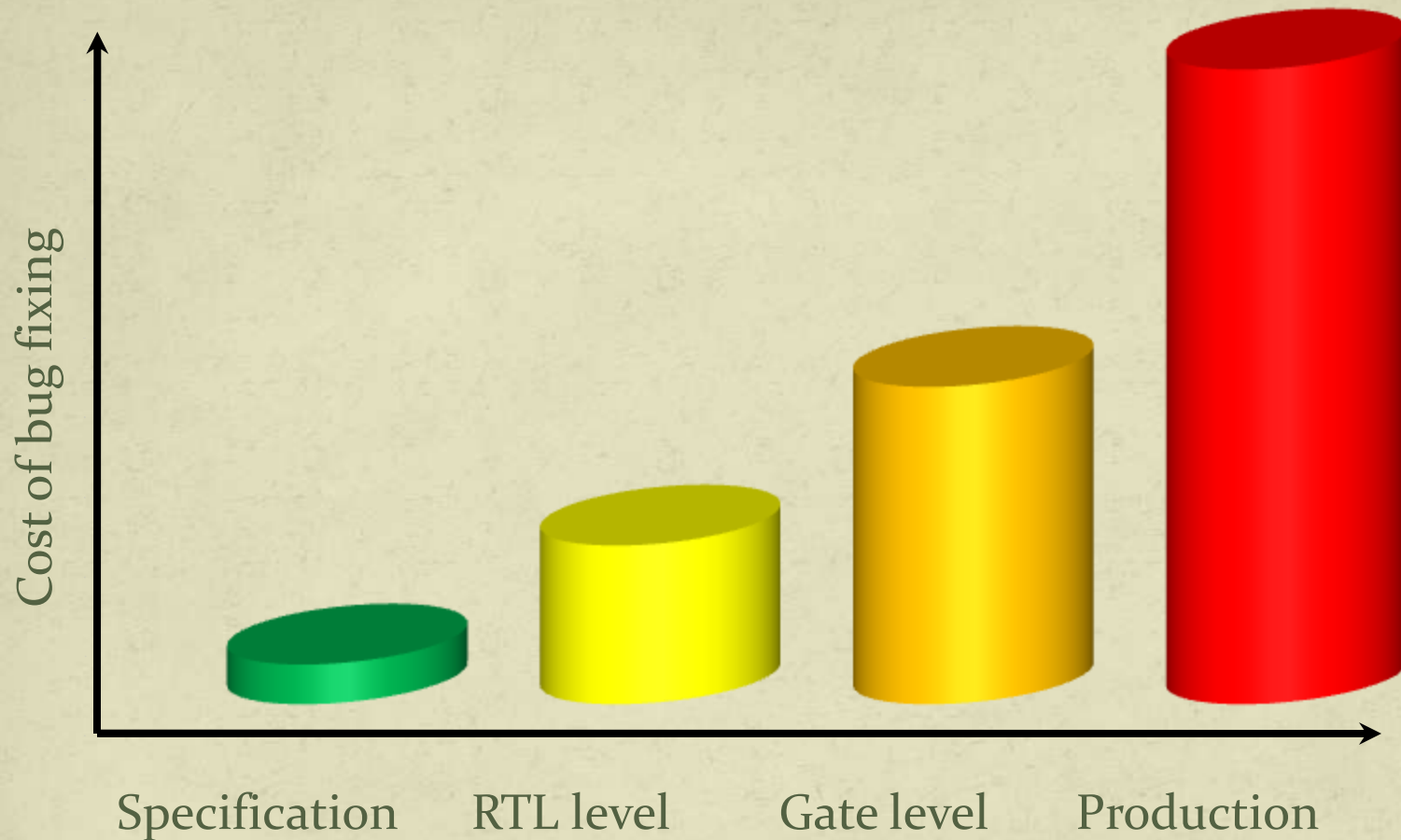
~50X increase in design productivity required to exploit the technology capabilities

Source: The International Technology Roadmap for Semiconductors 2009

Productivity gains using formal methods

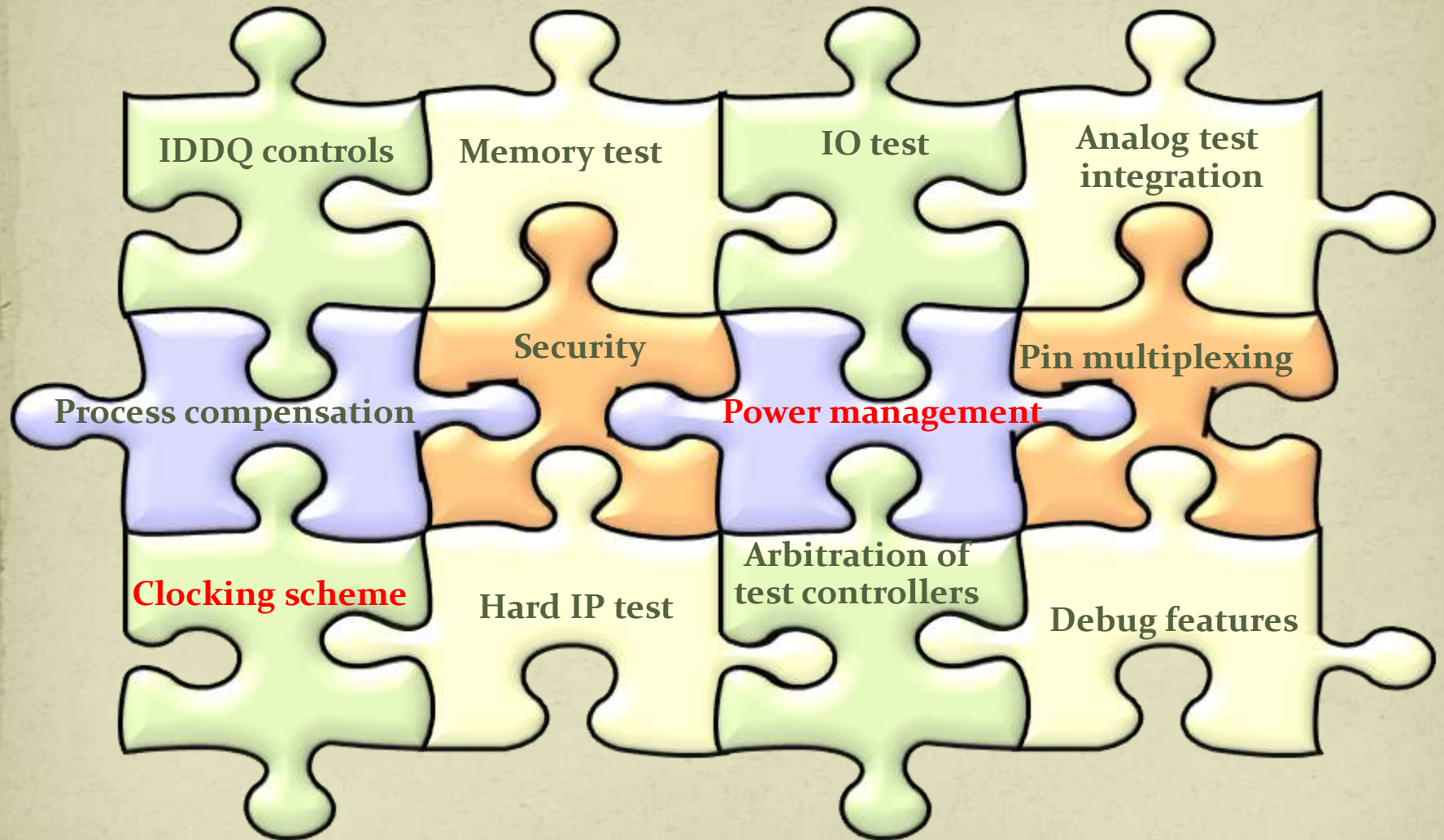


Importance of DFT verification at RTL stage



Exhaustive RTL level verification leads to tremendous savings!

Scope of DFT RTL verification in current SoCs

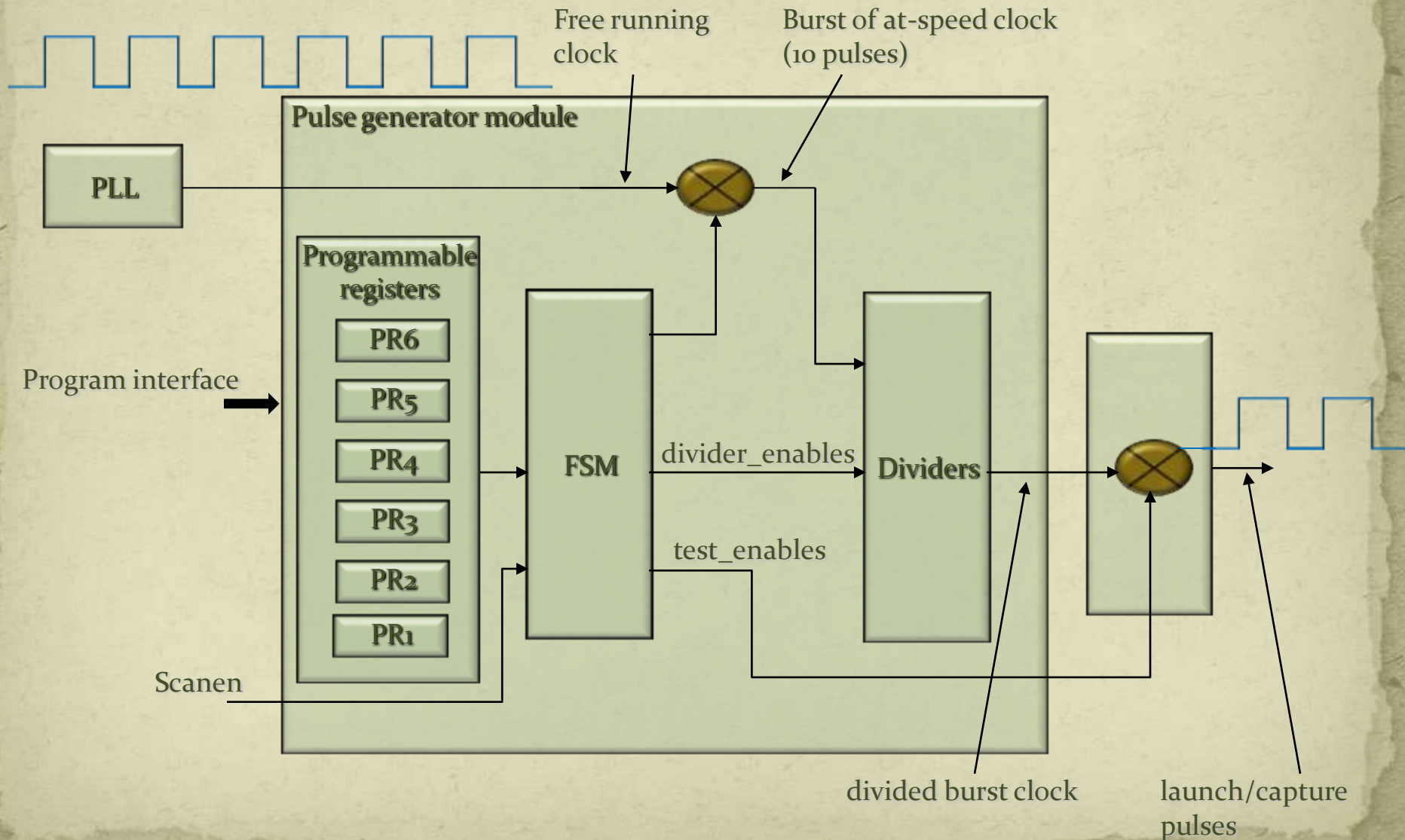


Following slides cover test clocking and power management verification using formal methods as case studies

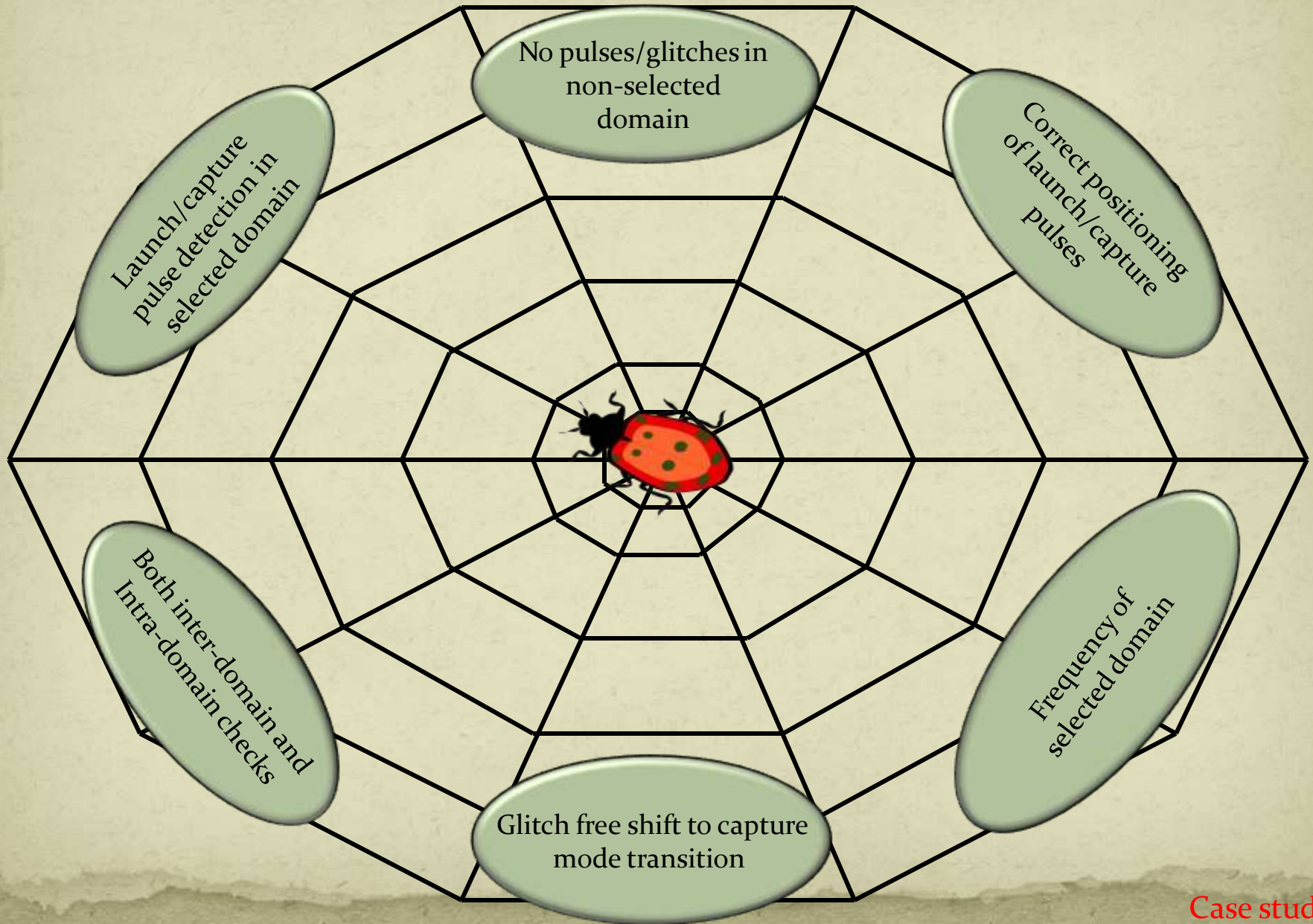
Case study-1: At-speed test clocking verification

- At-speed tests have become necessary due to growing design frequencies and technology shrinking.
- Most of the SoCs use on-chip PLL and complex pulse generator circuitry to generate at-speed pulses required for at-speed tests.
- Pulse generator circuits are complex FSMs which can be programmed for delays and spacing of launch-capture pulses.
- Exhaustive verification of at-speed test clocking at RTL stage is very important as bugs will be caught really late in the design cycle while simulating at-speed scan patterns on netlist.
- This case-study discusses FV based verification of at-speed test clocking for large sized SoCs with multiple clock-domains.

Typical at-speed clock generation architecture



Test clocking verification: Scope and complexity



Testcase size complexity of test clock verification

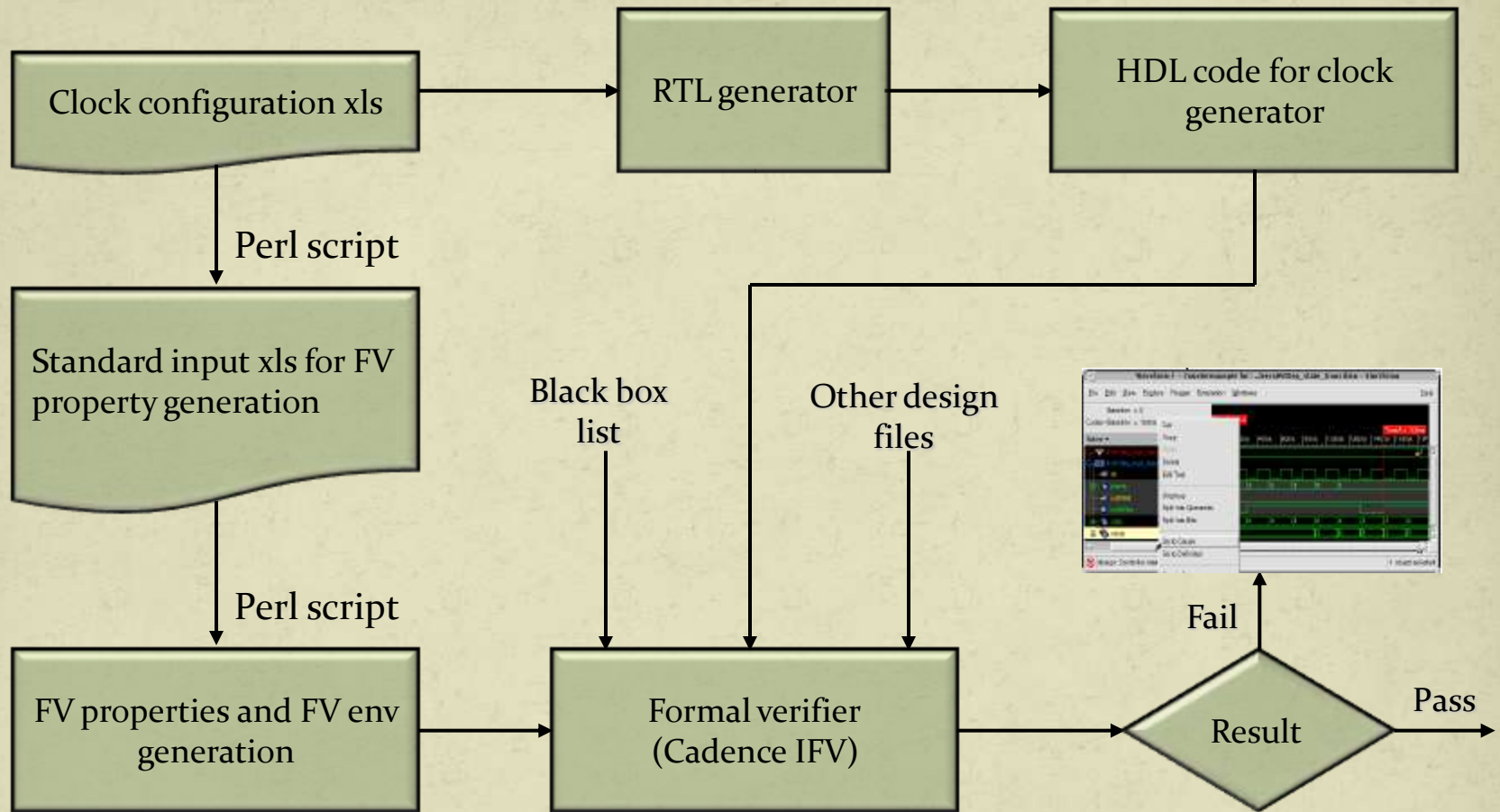
- Verification of these testcases becomes cumbersome with the increasing number of clock domains as shown below.
- If design has X clock domains and on an average Y clock generation points per clock domain, assume that N testcases will verify all the above properties for a clock generation point, then
 - Intra-domain testcases = $N * X * Y$
 - Inter-domain testcases per clock domain = $N * (X - 1) * Y$
 - Total no of inter-domain testcases = $X * [N * (X - 1) * Y]$
 - Total no of testcases = $N * X * Y * (1 + X - 1) = N X^2 Y$ hence total no of testcases $\propto X^2$

Test case volume illustration

Design	N	Y	X	Total testcases
SoC1 (45nm)	2	3	49	14112
SoC2 (45nm)	2	3	65	25350

FV based automation flow was used to handle huge volume and complexity of testcases.

FV based flow for clock generator verification

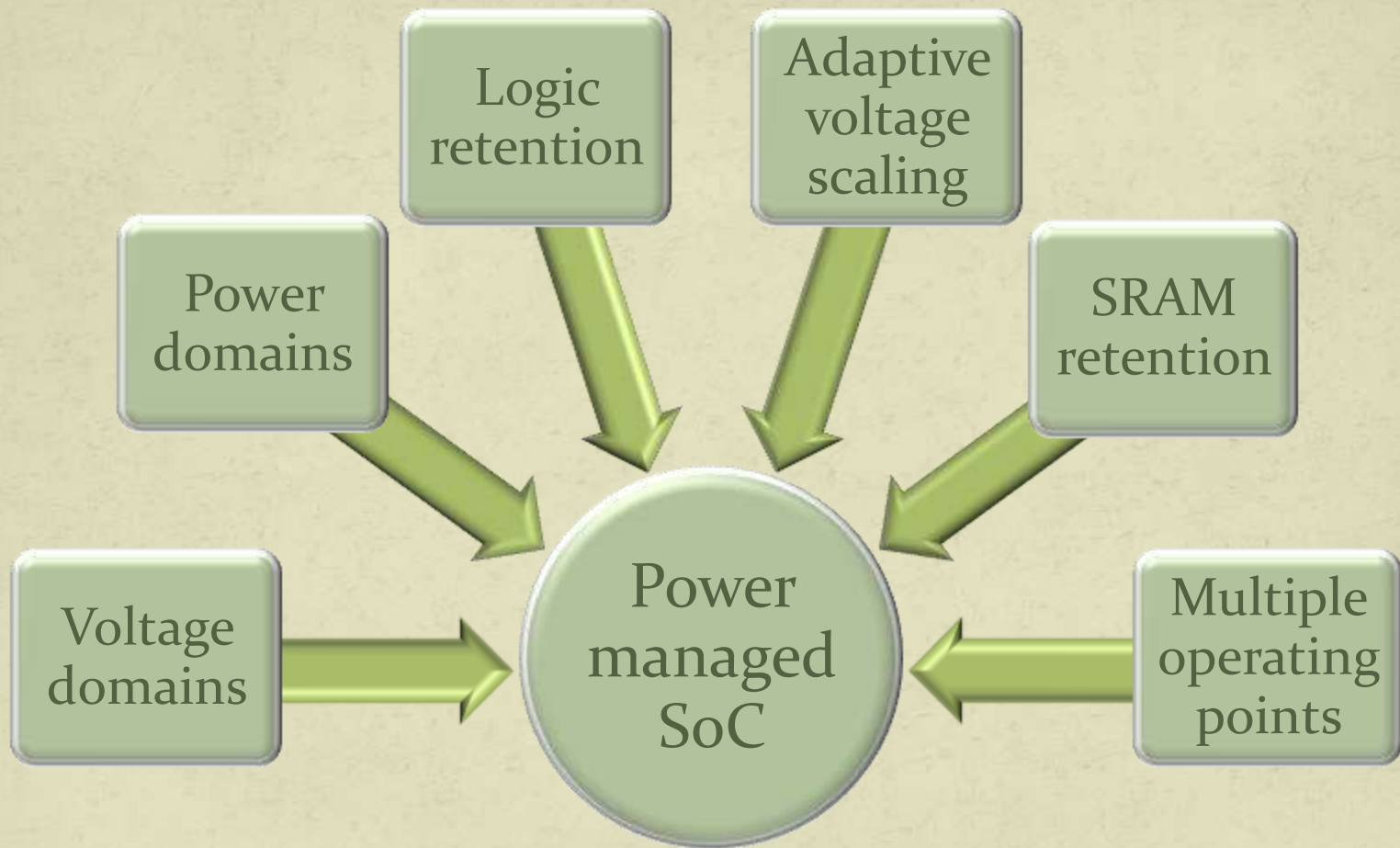


Test clock verification: Results

Design	No. of clock domains	No of testcases	Effort for complete verification	Effort savings with FV based flow compared to simulation based flow
SoC1 (45nm)	49	14112	1 man-month	1 man-month
SoC2 (45nm)	65	25350	.5 man-months	2 man-months

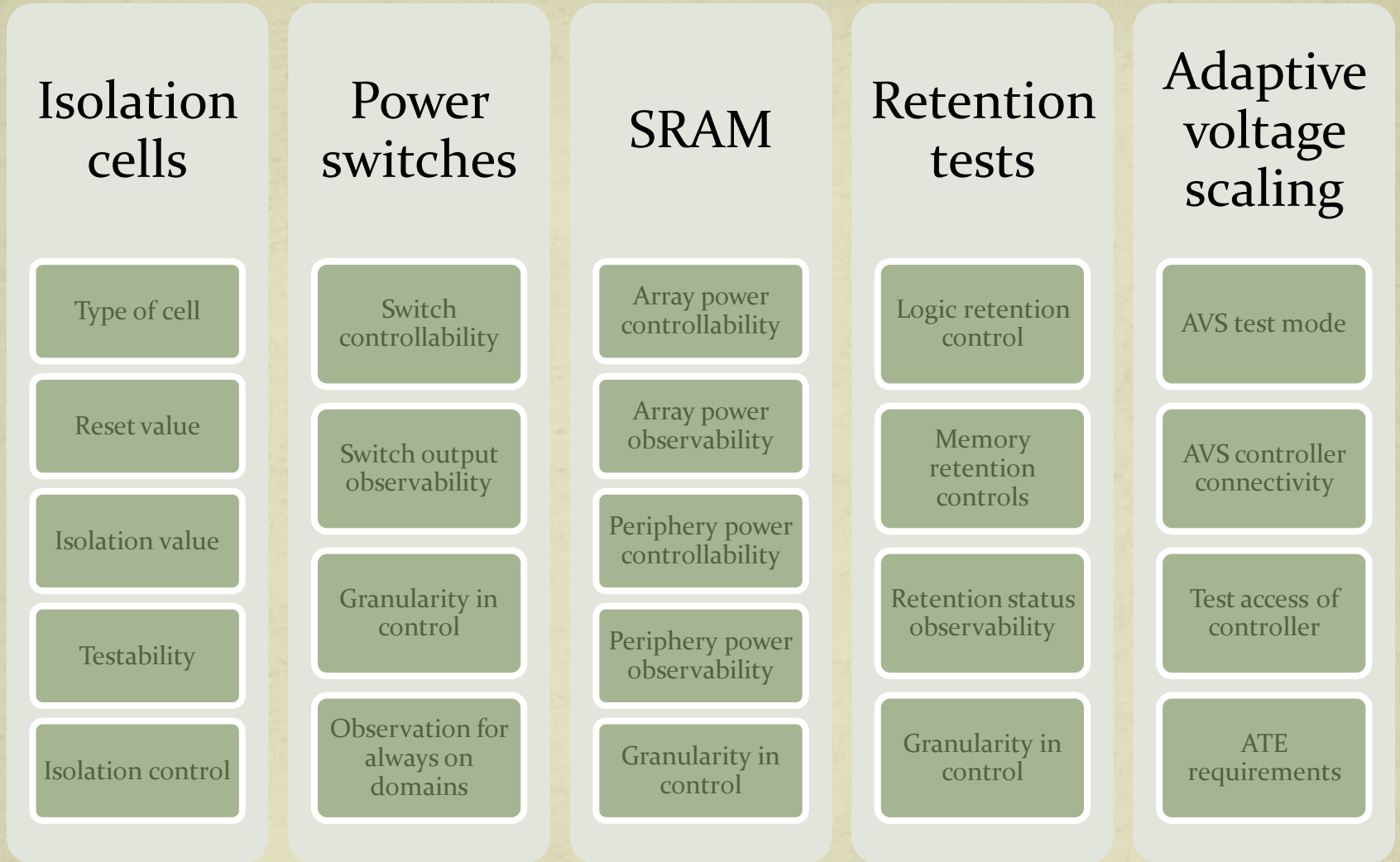
- Flow for FV based test clocking verification was developed for SoC1.
- SoC1 data includes flow development effort also.
- Huge effort saving on SoC2 due to reuse of flow/automation from SoC1.
- Same flow will be reused for multiple SoCs and IPs in future.

Case study-2: Power management DFT verification



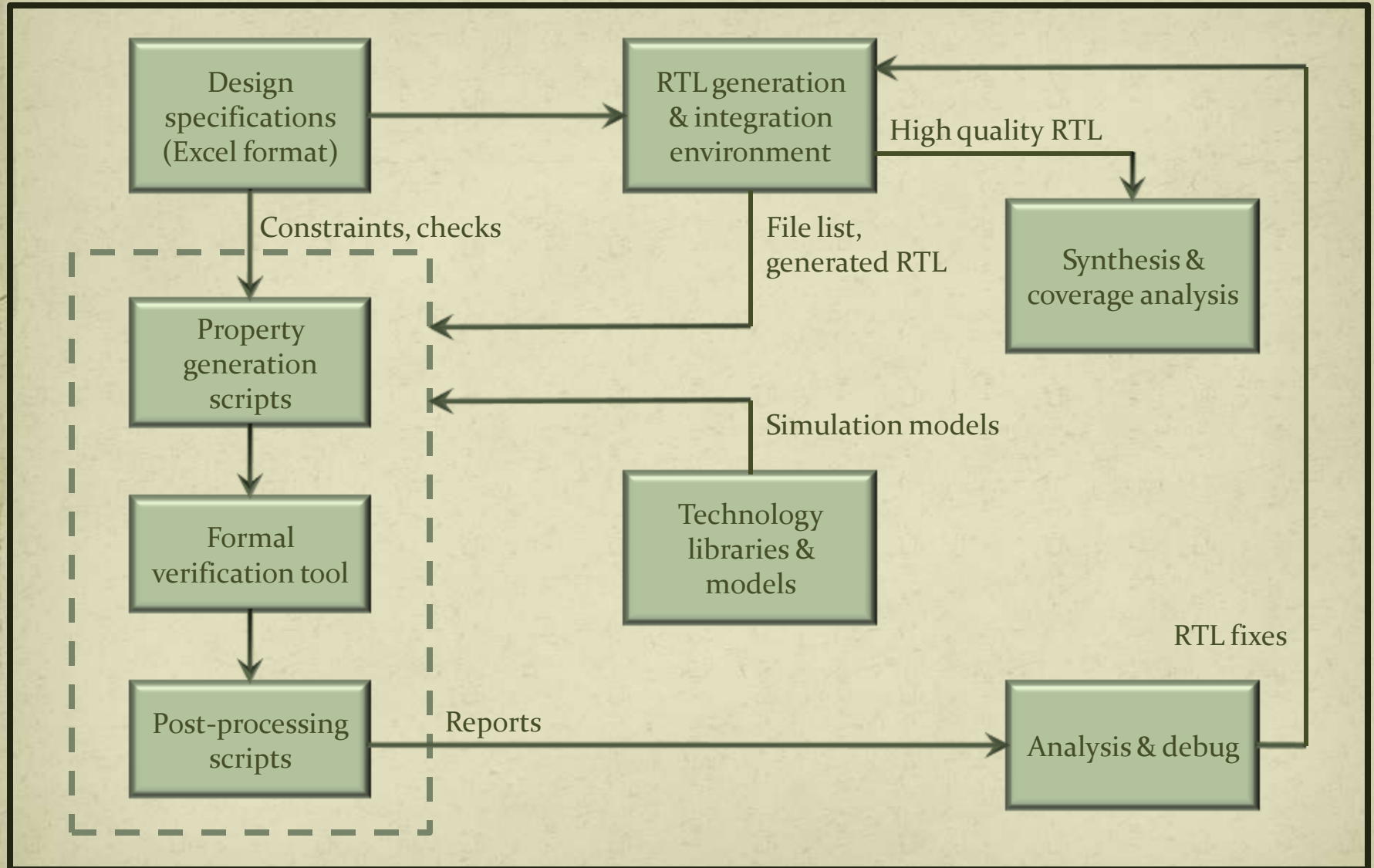
Components of power management in SoC context

Scope of DFT power management verification



All these were verified using formal verification methods

DFT power management verification flow diagram



DFT power management verification: Summary

Ease of use

- Inputs provided in XLS format
- Easily reviewable by designers

Handling of complex designs

- Created and used on a design with 13 power domains and 2 voltage domains with numerous signals crossing between them

Reusability

- Flow was reused in a similar SoC with 17 power domains and 3 voltage domains with little additional effort

Quality impact

- Exhaustive verification of power control signals resulted in higher ATPG coverage due to better testability of power management structures

Faster ATPG bringup

- Faster pattern bringup by avoiding unnecessary failures and debug due to power management issues

Schedule impact

- ~2 man-months effort saved across two SoCs
- Schedule predictability improved by avoiding ECOs and iterations in final dash

Limitations

- Environment setup time
 - Initial FV environment setup time can be high as there is no direct way to identify required black-boxes for particular set of properties.
 - Incomplete list of black-boxes may impact run time of the property.
- Handling of checks which require huge design space
 - FV checks may explode for the checks which requires huge active design space. For example, RTL level scan checks may not be handled effectively by FV.
- PSL (Property Specification Language) dependency
 - Lack of PSL knowledge may limit exhaustive usage.

Conclusion/Future work

Conclusion

- DFT verification complexity is increasing with increasing design sizes.
- Two case-studies of DFT verification on 45nm Complex SoCs were discussed.
- Several benefits of formal techniques over traditional simulation based approaches were discussed in terms of exhaustiveness, automation, reuse, efforts savings and results were also presented.

Future Work

- Property based automatic black-boxing to reduce environment setup time.
- Extend concept to cover RTL based scan checks.