Power Intent Automation for Ultra Low Power Mixed-Signal SoC

Sai Sudarsan Chitrapu, Ruchi Shankar, Ankitha M, Penchal Kumar Gajula, Tejas Salunkhe, Lakshmanan Balasubramanian, Gaurav Kumar Varshney

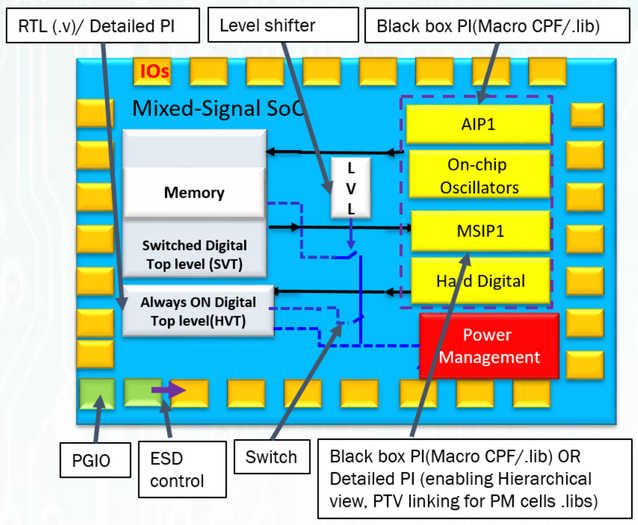
Texas Instruments (India) Pvt. Ltd.

1. Motivation & Problem Statement (1/2)

Context

- Newly evolving embedded processing SoCs developed for applications like automotive and IoT are heavily power managed with complex functionality
- Power management (PM) information like power domain (PD) definition, level shifting, isolation, state retention and power switches is captured in a file called power intent (PI)
  - PI can be specified in Common Power Format (CPF) or Unified Power Format (UPF)
  - PI from here on represents PI written in CPF
- Tens of digital IPs, analog and mixed signal IPs, SoC top-level and verification test bench require PI
- High integration complexity due to multiple voltage, power, reset* and clock* domains→ several hundred power modes

**Not in scope of this paper*



2. Motivation & Problem Statement (2/2)

Prior Art

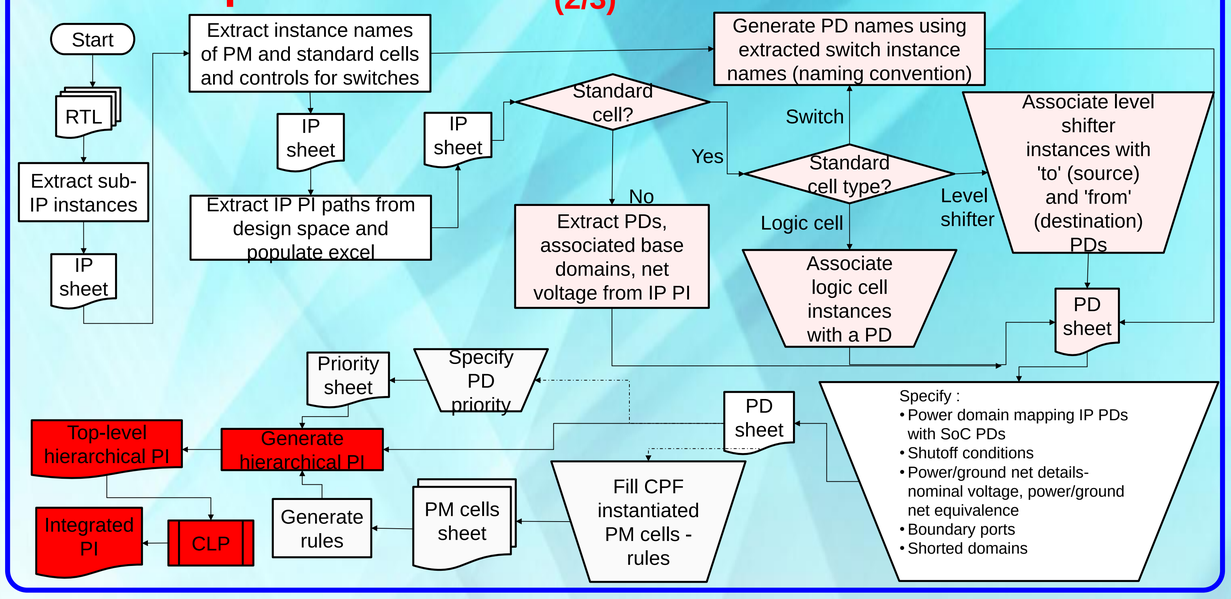
- SoC PI uses black box PI for MSIP[1], adequate for driving backend implementation
- Detailed / Hierarchical PI view is required for enabling early DV[1], Flat STA, faster debug of PI violation(s)
- Absence of PGIOs and PGIO driven ESD control connections in SoC PI, eventually defined downstream (Physical Design), results in late findings of incorrect domain mappings causing unprotected IOs under ESD events[2,3]
- SoC development involves four formal compile milestones and tens of integration iterations that require continuous PI modification
  - Takes about 2 to 3 man months
  - Significant manual effort to build SoC PI and multiple iterations as IPs and SoC RTL mature
  - Error prone due to manual intervention and hence impact design execution cycle for any late findings

SoC PI Complexity		
SL.NO	CPF Construct	Count
1	create_power_domain	100
2	update_power_domain	100
3	-shutoff_condition (for all create_power_domain)	70
4	create_power_nets	50
5	create_nominal_condition	50
6	set_instance	20
7	create_power_modes	20
8	-domain_conditons (for all power_mode)	2000
9	create_level_shifter_rule	100
10	update_level_shifter_rule	100
11	create_global_connection	350
12	Manual coded lines for SoC PI	3000

3. Proposed Solution (1/3)

- **Automation:** The time consuming and manually intensive process of PI development warrants an automation to generate correct by construct PI with RTL as key input
- **Flavors of PI**: Supports multiple flavors – detailed and macro PI views generated automatically with same interface
- **Hierarchical generation of PI**: PI generated is hierarchical in nature – generates intermediate PI which are used to generate final SoC PI
- **Simple and efficient user interface**: Easy and intuitive interface for specifying PI with minimum user inputs
- **Adaptation:** PI automatically adapts to RTL incremental updates without manual intervention, reducing manual effort and time consumption

4. 4. Proposed Solution (2/3) – Automation Flow



5. Proposed Solution (3/3) – Tabular Framework

IP Sheet:

rtl_wrapper	IP group hierarchy	IP instance name	IP module	PI (CPF) file path	PI type
rtl_hier1	ana_hier	ip_1	IP_1	ip_1_macro_cpf	macro
rtl_hier3	svt_hier	ip_2	IP_2	ip_2_design_cpf	design
rtl_hier3	svt_hier	ip_3	IP_3	ip_3_design_cpf	design
rtl_hier3	svt_hier	ip_4	IP_4	ip_4_macro_cpf	macro
rtl_hier3	svt_hier	ip_5	IP_5	ip_5_macro_cpf	macro
rtl_hier3	svt_hier	ip_6	IP_6	ip_6_macro_cpf	macro

PD Sheet:

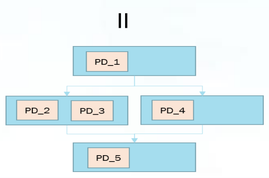
rtl wrapper	IP group hierarchy	IP instance name	ip power domain	SoC level power domain mapping	PM/standard cell	default PD flag	base domains	shutoff conditions	shorted_domain	power net name	internal flag	power net voltage	nominal condition	equivalent power nets	ground net	equivalent ground nets	boundary ports
rtl_hier1	ana_hier	ip_1	ip_pd_1	PD_1		1				VDD_1		v1	n1		gn1	gn2	bp1,bp2
rtl_hier1	ana_hier	ip_1	ip_pd_2	PD_2			PD_1			VDD_2		v2	n2	VDD_1	gn1		bp3
rtl_hier2	svt_hier	ip_2	ip_pd_3	PD_3		1				VDD_3		v3	n3		gn2		
rtl_hier3	svt_hier	ip_3	ip_pd_4	PD_4						VDD_4		v4	n4		gn1		
rtl_hier3	svt_hier	ip_4		PD_5	switch		PD_1	con1, con2	1	VDD_5	1	v5	n5		gn1		
rtl_hier3	svt_hier	ip_5		PD_5	switch		PD_2	con1, con2	1	VDD_5	1	v6	n6		gn1		
rtl_hier3	svt_hier	ip_6		PD_2	level_shifter		PD_1										

PM rules Excel:

PM rule name	rule_type	power_domain	secondary domain	include instances	exclude instances	target	type	conditions	cells
iso_rule1	Isolation	{from domain, to domain}		[pins]		from	high		
iso_rule2	Isolation	{from domain, to domain}		[pins]		to	low		
iso_rule3	Isolation	{from domain, to domain}		[pins]		drop_list	drop_list		
iso_rule4	Isolation	{from domain, to domain}		[pins]		drop_list	hold		
ret_rule1	Retention	power_domain		[instances]		flop	save_edge		
ret_rule2	Retention	power_domain		[instances]		latch	restore_edge		
ret_rule3	Retention	power_domain		[instances]		both	restore_edge-save_edge		
ret_rule4	Retention	power_domain		[instances]		drop_list	drop_list		

Priority Excel:

Priority		
1	PD_1	
2	PD_2,PD_3	PD_4
3	PD_5	



6. Results and Conclusions

Sl.No	Aspect	Conventional flow	Automated flow
1	Time required for **initial** SoC PI bring up	2-3 weeks	2-3 days, (86% reduction in manual effort and cycle time)
2	Iterations	PI adaptation per SoC RTL incremental updates is time consuming	PI automatically adapts to SoC RTL incremental updates
3	SoC PI views	Manual generation of different SoC PI flavors (with Black box PI for MSIP)	Automatic generation of: • SoC PI with black box PI for MSIP • SoC PI with detailed PI of MSIP for early DV, flat STA, faster debug
4	Reliability	Error prone due to manual modifications	More reliable since tool extracts necessary information from RTL and IP PI
5	Prerequisite skills	• Knowledge of power architecture of SoC and IPs • Knowledge of PI language (CPF/UPF)	• User only needs to input data in predefined fields in excel • Agnostic to PI language & syntax
6	Portability	Manually porting from CPF to UPF is very tedious and error prone	Plan to update the script for UPF format by Q2 2023
7	Schedule	• Significant effort is required to generate PI and manually align PI with design changes • Repetitive iterations and late findings can affect time to market	• Generates a reliable and exhaustive PI with minimum user intervention, which easily adapts to changing design intent • Improves time to market

- ***Source:*** *[1] Aswani Kumar Golla, et al, "Hierarchical Power Intent Driven Efficient Power Integration Methodology for Ultra Low Power Mixed-Signal SoC,", DAC 2019*
- *[2] Lakshmanan Balasubramanian, et al, "Advances to CPF Based Low Power Mixed Signal Integration and Verification," Cadence Live India 2021*
- *[3] Vijay Kumar Sankaran,et al, "Modern Recipes for Brewing the Inevitable Methodology for Today's ICs: Low-Power Mixed-Signal Design Verification", DAC 2019*

Acknowledgements