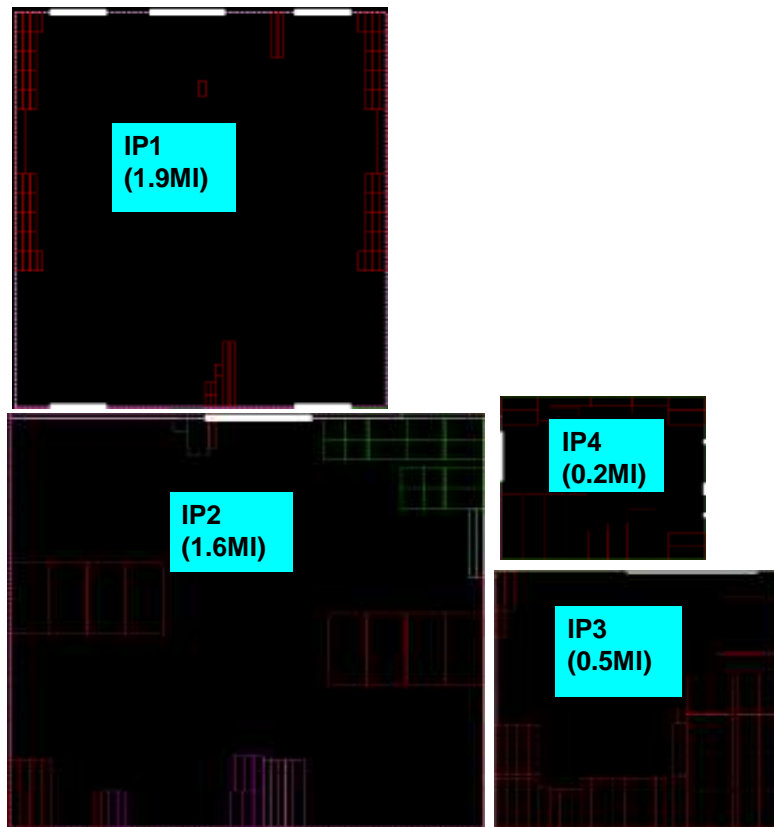


Super-IP: Hierarchical Design, Upside Down

Sreeram Chandrasekar, Hari Krishnamoorthy,
Nithin SK, Kumud Singh

Texas Instruments, Bangalore, India

SuperIP– Concept



Discrete IPs: Each IP requires separate physical implementation team/effort

(Placeable instances count in 45nm technology is mentioned for each, MI=Million Instances)



Super-IP: Combined implementation of multiple IPs “together” by a single team

4 IPs done “together” by half the team size in same schedule

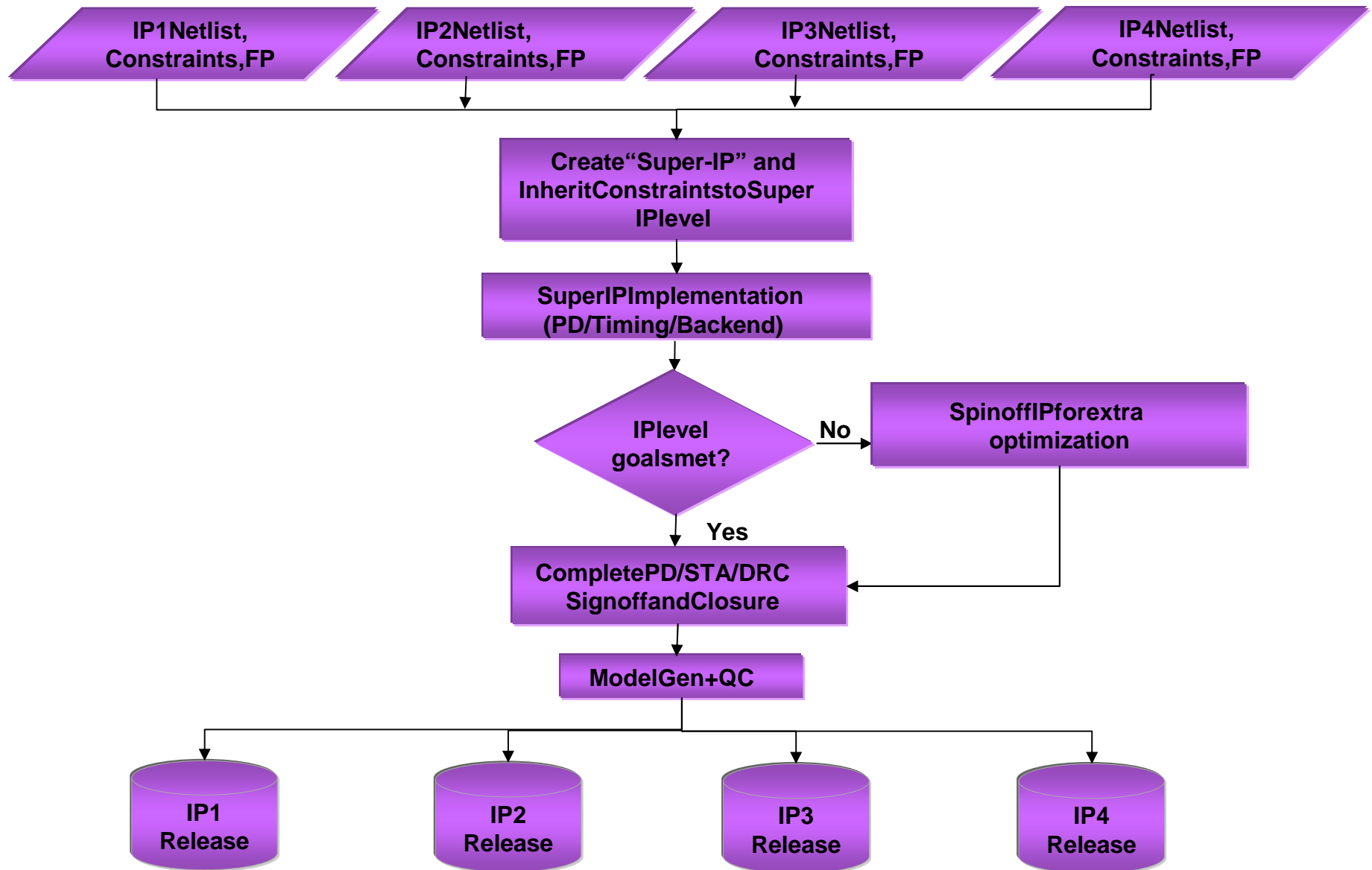
Motivation

- Complex IP Subsystem development schedules are critical to SOC Time-to-Market
 - Often, IP timelines (also bugs, architecture changes) are asynchronous to SOC
 - Several IPs /versions to implement, catering to multiple SOCs
- Constraints
 - Typically complex IPs are implemented as “Hard” IPs
 - Multiple IP developments concurrent with SoC programs
 - IP team cannot expand/ramp up to cater to peak IP development effort
 - Peak compute/resource availability is also a concern
- So, Need a High Productivity Solution
 - That can scale up to implement multiple IPs without requiring larger teams
 - Utilize computer resources optimally
 - Bring in consistency in physical design specs/practices/ quality

Solution: “SuperIP”

- A Method to combine multiple designs/IPs into a single implementation
 - The designs can be completely unrelated
 - Different frequencies/clocking are ok
 - Mainly a PD/STA/Design Closure vehicle
 - Does not involve DFT, Gate Simulation etc.
- Super-IP in a Nut-shell
 - Combine IPs A, B, C to create a virtual design (Super-IP)
 - Implement as one design (in the same schedule for individual IP)
 - Release A, B, C as individual IPs as before
- Drastically reduce resources to implement a number of IPs
 - By combining the common things to be done together
 - By utilizing tool capacity/parallelization ability
 - By enabling one resource to work on multiple IP together
- Can achieve 50% resource reduction – Quantified later

SuperIPFlow



What's New vs. Existing Solutions?

Common Hierarchical Flow	SuperIP
<ul style="list-style-type: none"> • Context-driven top-down optimization, where IP shapes and budgets are discovered • Reuse of IP across SoCs not possible • Used to partition a large design into smaller pieces, to manage the size/capacity 	<ul style="list-style-type: none"> • New Method to <u>Combine multiple IP implementations into one effort</u> <ul style="list-style-type: none"> – The designs may be unrelated to each other <ul style="list-style-type: none"> • In terms of frequency, clocking, Voltage condition etc. • <u>Reuse of standard interface timing budgets for IPs</u> <ul style="list-style-type: none"> – Not in-context budgeting, or optimization across IP boundary – Avoids iterations due to re-budgeting at various stages (CTS, STA closure etc.) • Provides method to <u>port multiple IP constraints to Super-IP level – for optimization and signoff</u> • <u>Combines IPs for implementation, but allows release/maintenance as individual IPs</u> <ul style="list-style-type: none"> – Critical for IP Reuse, ECOs • Not Tied to Flat or Hierarchical flow; adaptable to either as required by different design stages

Benefits/Advantages

FlowStep	Resource need for Super-IPVs4 Discrete IPs
Synthesis/EC	50%
Floorplanning, Power routing, DRC, IR Drop analysis	25%
Placement, Clock Tree Synthesis, Multi- Mode/Corner Optimization	50%
Detailed Route+Timing	<50%
PM Checks/ Optimization	50%
SCAN/GLS/TDLSims	100%
STA+Final Timing Closure	50%
DRC/Rely Analysis+ Closure	<50%
Release	<50%
Total Resources	<50%

- Theme: Utilize increasing tool capacity/parallelism to get common tasks done “together” across IPs
- Drastic reduction in the resource required for IP physical implementation
- Capacity Driven Efficiency: Combined IP Implementation
 - Floor-planning, power routing, DRC clean-up of power grid, etc. can be done together
 - Combined Placement stage optimization (timing, congestion) that is tool runtime dominated
 - Example:
 - Discrete IP Runtime = 24hrs
 - For Super-IP with 3 IPs, it is 31hrs
 - 3 IPs done almost in the time of one through Super-IP
 - STA/Timing Closure across Modes/Corners – particularly extraction, hold fixing, crosstalk
 - DRC/Elec and Rely signoff Checks + Fixing
 - Also results in lesser CPU needs – Extraction, STA, DRC
- Other significant benefits in design execution
 - Uniform deliverables QC across IPs
 - Easy to apply best practices/optimization knobs
 - Common & Quality Design reviews

Care-Abouts

- WhentoSuper-IP?
 - “Known” or mature IPs
 - Revisions– Technology migration, minor changes, bug fixes, etc.
 - Not for Brand new IP architecture with unknown constraints
- ECOs
 - ECO in one or more IPs in SuperIP
 - Gate ECO done as in the traditional ECO flow
 - Apply the ECO at SuperIP level in the appropriate flow stages
 - Major ECO/Re-do of a Sub-IP in SuperIP
 - Blackbox the IP from SuperIP
 - Respin the IP from the appropriate stage of the flow
 - There-spun IP can intercept SuperIP at later stage
 - If ECO after Release
 - Impacted IP alone can be re-spun

Results

- Placement done with the physical and timing constraints of the 4 IPs
- Timing:
 - 10k Violations/450ps WNS
 - Similar to Individual IP
- Routing/Congestion
 - Each IP routed within its boundaries
 - Similar status as discrete IPs
- 50% resource reduction for IP implementation

