Engineering Trade-offs in the Implementation of a High Performance ARM® Cortex™-A15 Dual Core Processor

Bernard Ortiz de Montellano
Product Manager
Processor Division
ARM

Joe Walston
Staff Applications Consultant
Synopsys

March 26th, 2013

Synopsys Users Group

Today's Session

ARM-Synopsys Project Introduction — Bernard Ortiz de Montellano

Engineering Trade-offs in the Implementation of a High Performance Cortex-A15 Dual Core Processor — Joe Walston

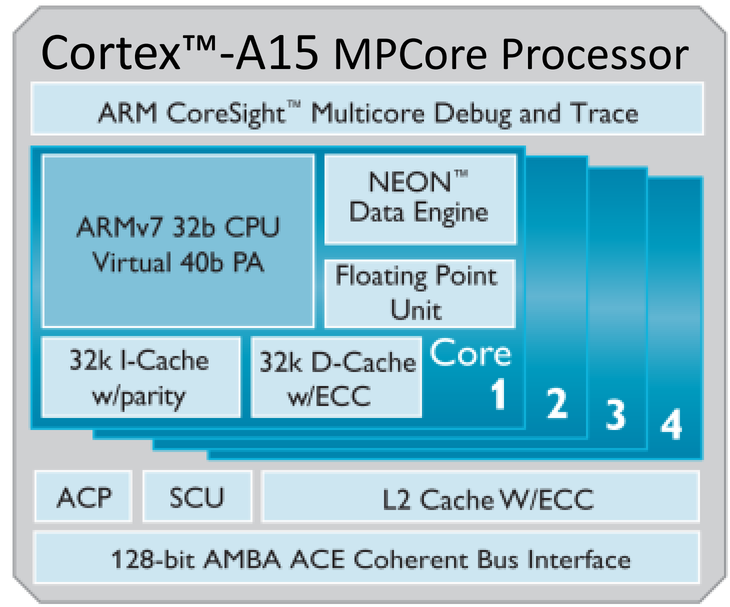
ARM-Synopsys Project Introduction

The ARM® Cortex™-A15 MPCore™ Processor

Implementation Optimization for the “big” core in a big.LITTLE SoC

The ARM Cortex™-A15 MPCore Processor

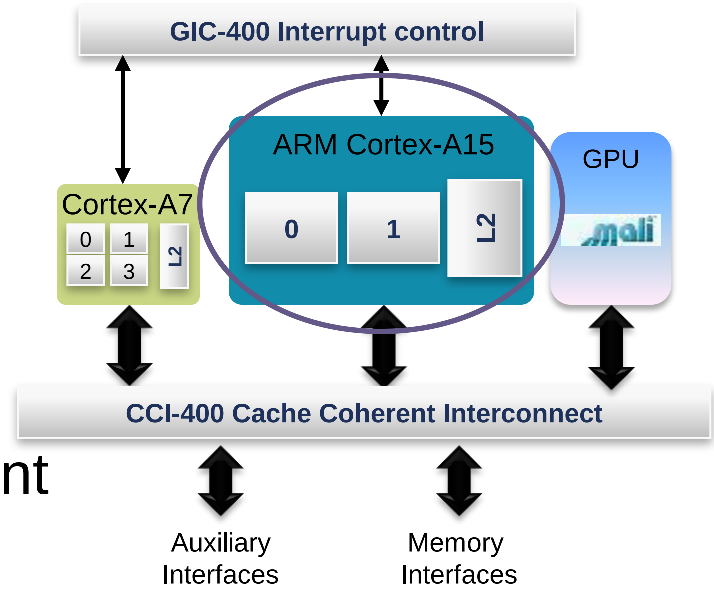
- Highest performance ARMv7 (32b) application processor
  - Multi-issue, out-of-order pipeline
- Best for superphones, tablets, laptops, servers, infrastructure
- Processor cluster includes
  - 1-4 processor cores with NEON and FPU
  - ACP, SCU, L2 and bus interface
- Architectural enhancements
  - Hardware enhanced OS virtualization
  - 1TB of addressable physical memory
- Performance and power scalability
  - Implementation options include smartphone, tablet and server power envelopes
  - System coherency with ACE
  - big.LITTLE processing with Cortex-A7 and CCI-400
- IP available now



Cortex-A15 is the high-performance engine for your highly-connected device

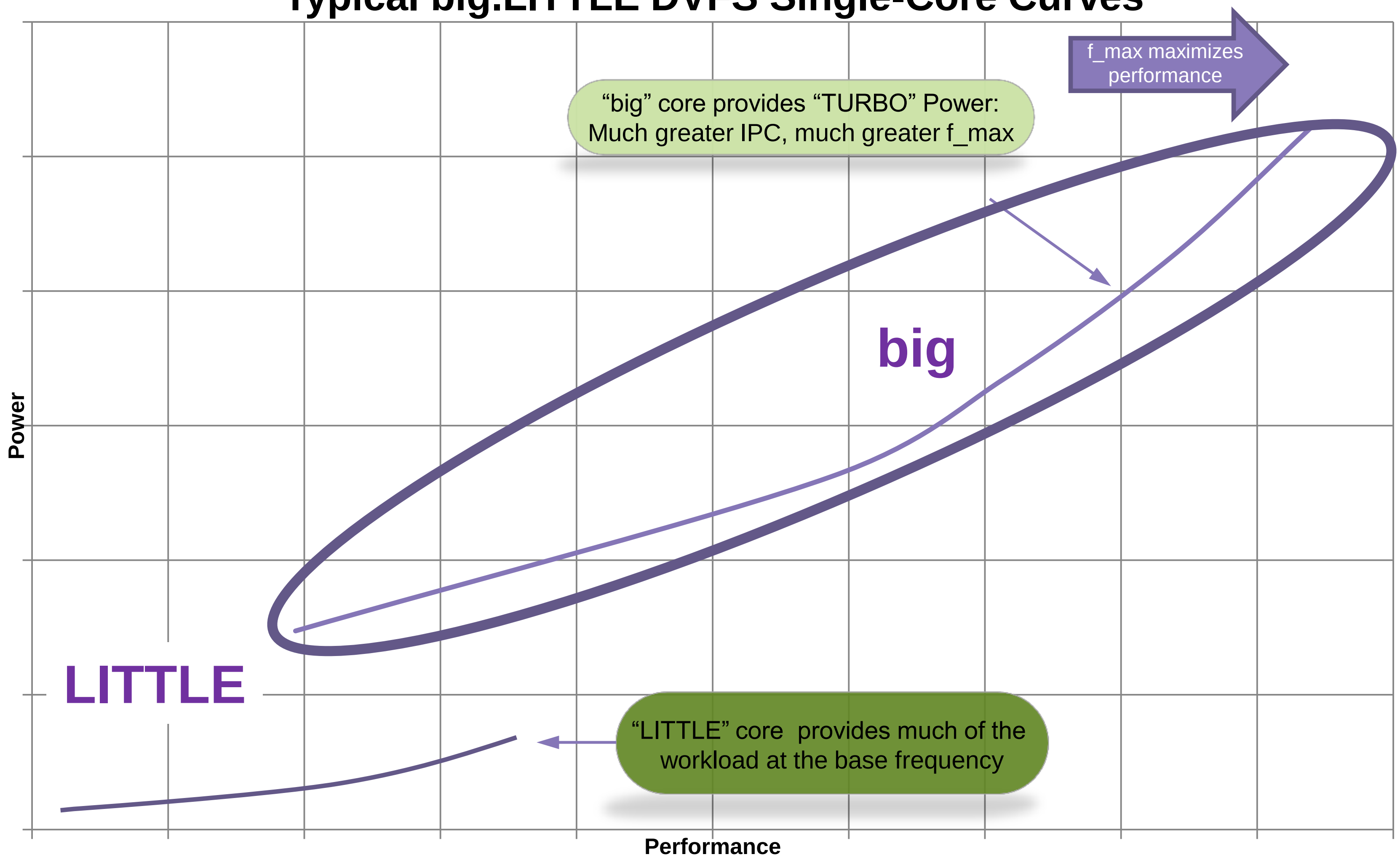
Implementation Targeting for a big.LITTLE System-on-Chip

- Big cluster: Cortex-A15 processor
  - Choose aggressive frequency target
  - Power is mitigated ~50% with MP software
- LITTLE cluster: Cortex-A7 processor
  - Choose high efficiency target
  - Very small area for quad core!
- CoreLink™ CCI-400 Cache Coherent Interconnect
  - Implement to favor performance
  - Do not starve the big cluster
- GIC-400
  - Provides transparent virtualized interrupt control
  - Implement to favor performance



Performance and Energy-Efficiency

Typical big.LITTLE DVFS Single-Core Curves



Collaboration Expanded

To Deliver Optimized Methodologies For ARM Cortex Processors

Synopsys Users Group

ARM and Synopsys Expand Collaboration to Optimize Power and Performance, and Accelerate Design and Verification for ARM Technology-based SoCs

CAMBRIDGE, United Kingdom and MOUNTAIN VIEW, Calif., Aug. 28, 2012

ARM and Synopsys Collaborate to Deliver Optimized Reference Implementations for ARM Processors

Optimized Methodologies for ARM's Cortex-A15, Cortex-A7 and CCI-400 Solutions Help Designers Achieve Processor Performance and Power Objectives Faster

CAMBRIDGE, UK, and MOUNTAIN VIEW, Calif. Mar. 21, 2013

SYNOPSYS® Accelerating Innovation

Collaboration Objectives

Optimal Starting Point For Cortex-A15 Processor Implementation

snug
Synopsys Users Group

QOR

- Meet power target while optimizing for best timing within power budget, best area within power and timing budgets
- Target market requires a power centric implementation

Schedule

- Develop Cortex-A15 quad core flow quickly for stand-alone or big.LITTLE
- Enable ARM and Synopsys customers timely access

Flow

- RTL through Route
- Repeatable, robust, easily modifiable scripts

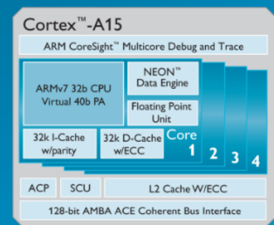
Documentation

- Guidelines for joint customers to follow when targeting a different configuration
- Best practices and pitfalls

Primary Deliverables: Reference Implementations (RI)
with real, repeatable results

ARM + Synopsys Collaboration

- Cortex-A15 dual core processor
- TSMC 28HPM process
- ARM POP™ IP: core optimized standard cells and fast cache instances



+

Galaxy

- PrimeTime
- Design Compiler
- IC Compiler
- StarRC

Synopsys Engineering and Low Power Expertise

Reference Implementation for an ARM Cortex-A15 MPCore processor optimized for balanced timing and power

Today's Session

ARM-Synopsys Project Introduction	Bernard Ortiz de Montellano
Engineering Trade-Offs in the Implementation of a High Performance Cortex-A15 Dual Core Processor	Joe Walston

Engineering Trade-offs

For a Cortex-A15 Dual Core Processor

Synopsys Users Group

Setup	Standard Cell Libraries	Core Configuration	
Flow Development	Flow Setup	Implementation Strategy	Trade-offs
Future Improvements	Multisource CTS	Data Flow Analysis	
Conclusions	Results	Top 10 Best Practices	

SYNOPSYS® Accelerating Innovation

Engineering Trade-offs

For a Cortex-A15 Dual Core Processor

Synopsys Users Group

Setup	Standard Cell Libraries	Core Configuration	
Flow Development	Flow Setup	Implementation Strategy	Trade-offs
Future Improvements	Multisource CTS	Data Flow Analysis	
Conclusions	Results	Top 10 Best Practices	

Standard Cell Libraries

ARM Artisan® Logic for TSMC 28HPM Overview

snug
Synopsys Users Group

- Technology Details
  - TSMC 28HPM process
  - 10 layer metal (1p10m_5x2y2z)
  - ARM POP™ IP libraries
    - Fast-Cache Instance RAMs
    - 12T high-speed cells
- PVT Configuration - 4 corners
  - Setup (OC_WC) : SSG / 0.81v / 0c
  - Hold (OC_BC): FF / 1.05v / 125c
  - Power (OC_LEAK): TT / 0.9v / 85c
  - IR (OC_IR): FFG / 1.0v / 125c
- Three transistor channel lengths available
  - CS = short (faster, more power)
  - CM = medium (standard)
  - CL = long (slower, less power)
  - Same cell footprint across all channel lengths

Standard Cell Selection
(Multiple Vt / Channel variants)

Vt Class	Channel Variant	Cell Family
ULVT	CS	
	CM	
LVT	CS	
	CM	
	CL	
SVT (RVT)	CS	
	CM	
	CL	
HVT	CM	
UHVT	CM	

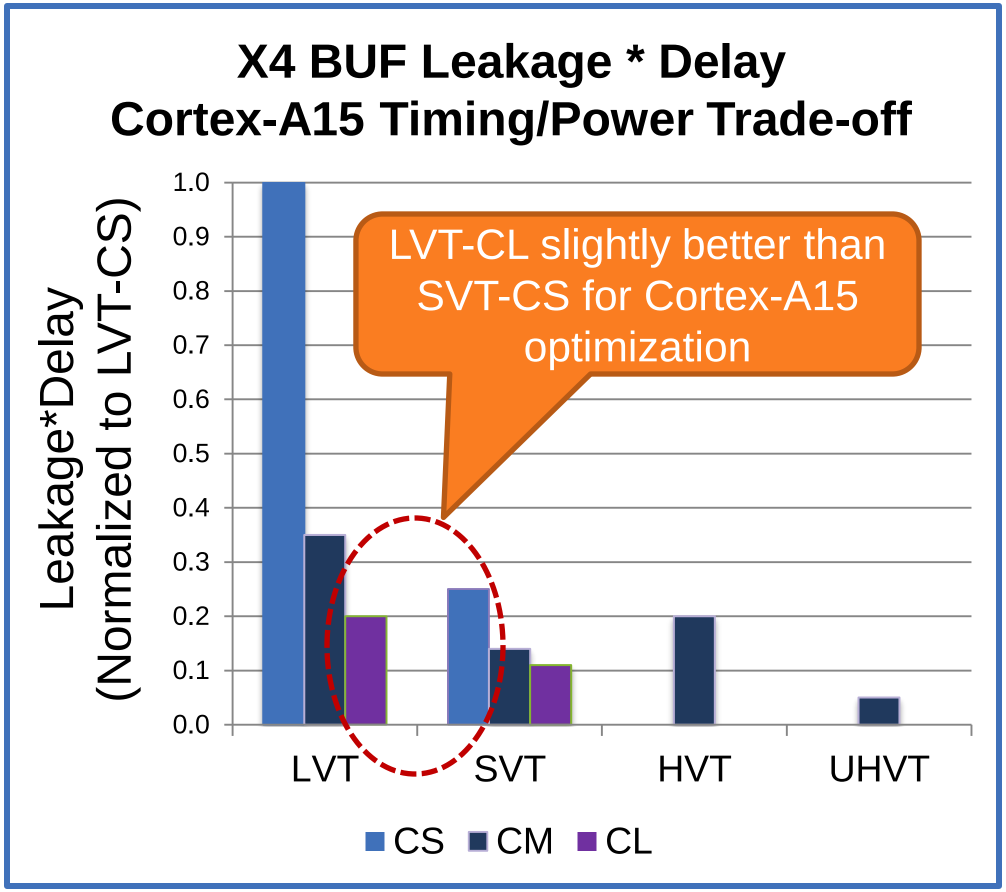
- 8 different optimization classes
- ULVT not used for this project
- **CL** channel has monetary cost

SYNOPSYS® Accelerating Innovation

Standard Cell Libraries

Leakage/Timing Trade-off: Family Comparison

- Comparison for Cortex-A15 leakage/timing trade-off
- Plotting product of leakage and delay for X4 BUF
  - Larger values indicate more leakage cost for a given delay
- **Conclusions for Cortex-A15 with leakage/timing trade-off:**
  - LVT-CS high leakage cost
  - LVT-CL slightly better trade-off than SVT-CS
  - SVT-CL better low-leakage option than HVT



Only available classes/variants plotted

Engineering Trade-offs

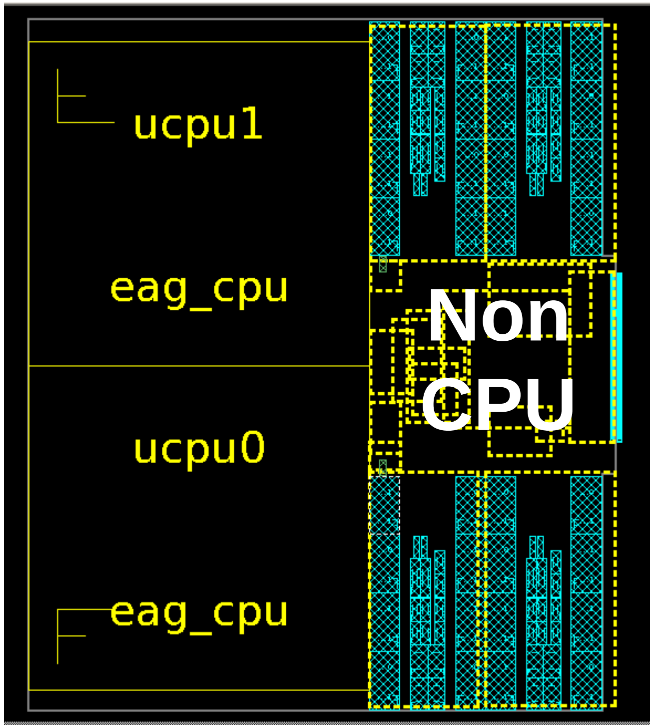
For a Cortex-A15 Dual Core Processor

Setup	Standard Cell Libraries	Core Configuration	
Flow Development	Flow Setup	Implementation Strategy	Trade-offs
Future Improvements	Multisource CTS	Data Flow Analysis	
Conclusions	Results	Top 10 Best Practices	

Core Configuration

Cortex-A15 Dual Core Processor

Configurable Feature	Selected Value
# Cores	2
L2 cache size	1MB
L2 tag RAM register slice	0
L2 data RAM register slice	0
L2 arbitration register slice	Not Included
L2 logic idle gated clock	Included
Regional gated clocks	Included
ECC/parity support	Include Parity/ECC in L1 and ECC in L2
NEON	Included
VFP	Included
Generic Interrupt Controller	Included
Shared Peripheral Interrupts	128
DFT Strategy	Scan compression
UPF/Power Strategy	Shut-down w/ isolation



Engineering Trade-offs

For a Cortex-A15 Dual Core Processor

Setup	Standard Cell Libraries	Core Configuration	
Flow Development	Flow Setup	Implementation Strategy	Trade-offs
Future Improvements	Multisource CTS	Data Flow Analysis	
Conclusions	Results	Top 10 Best Practices	

Synopsys' Core Optimization Collateral

Built on Galaxy Tool RMs

Reference Implementations (RIs)

- Leverages HPC
- Core and technology library specific
- Includes scripts, floorplan, constraints

Lynx Plug-Ins

- RI scripts instrumented for Lynx environment

Hi-Performance Core (HPC) Methodology

- Leverages RMs, tuned for high perf cores
- Core and technology library independent

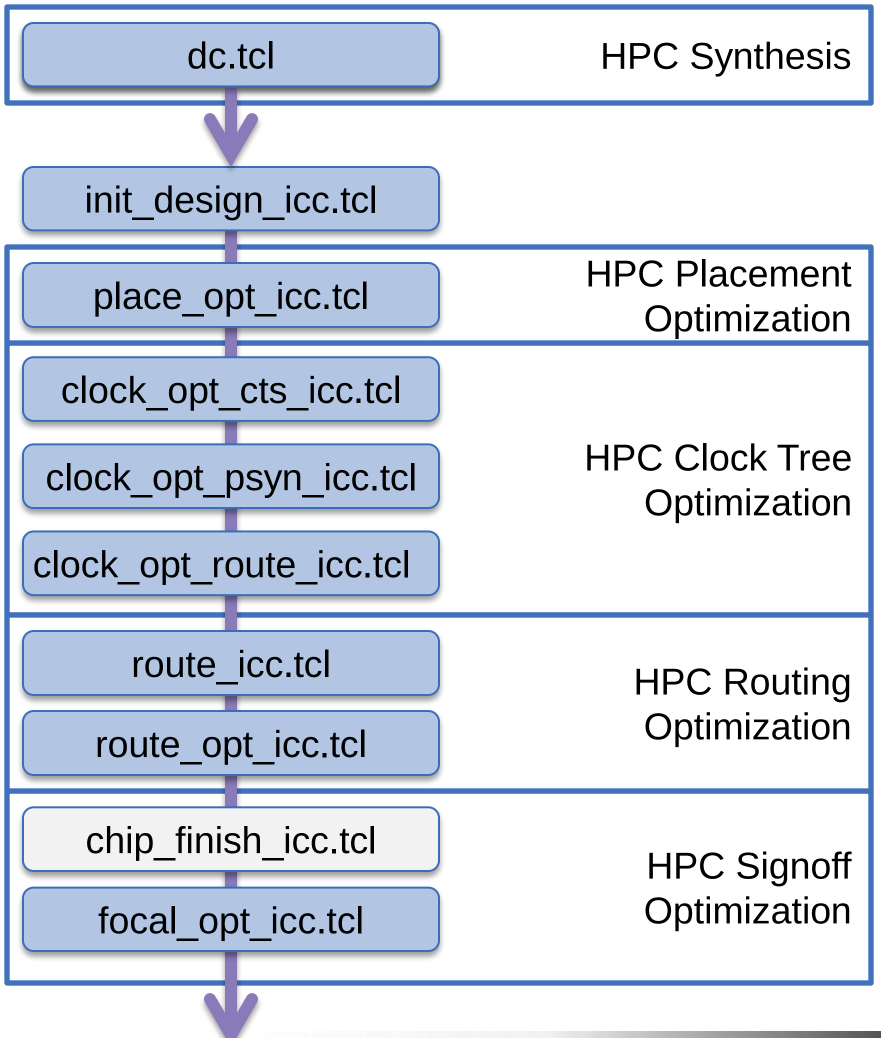
Reference Methodologies (RMs)

- Tool- and release-specific scripts
- Core and technology library independent

More design/technology specific

Reference Implementation Flow Development

- Basic RM-style flow with HPC-related add-ons
- Each flow step has driver script
- HPC adds customizations like:
  - ICG handling
  - Path group weighting
- Chip finishing not part of collaboration framework



Engineering Trade-offs

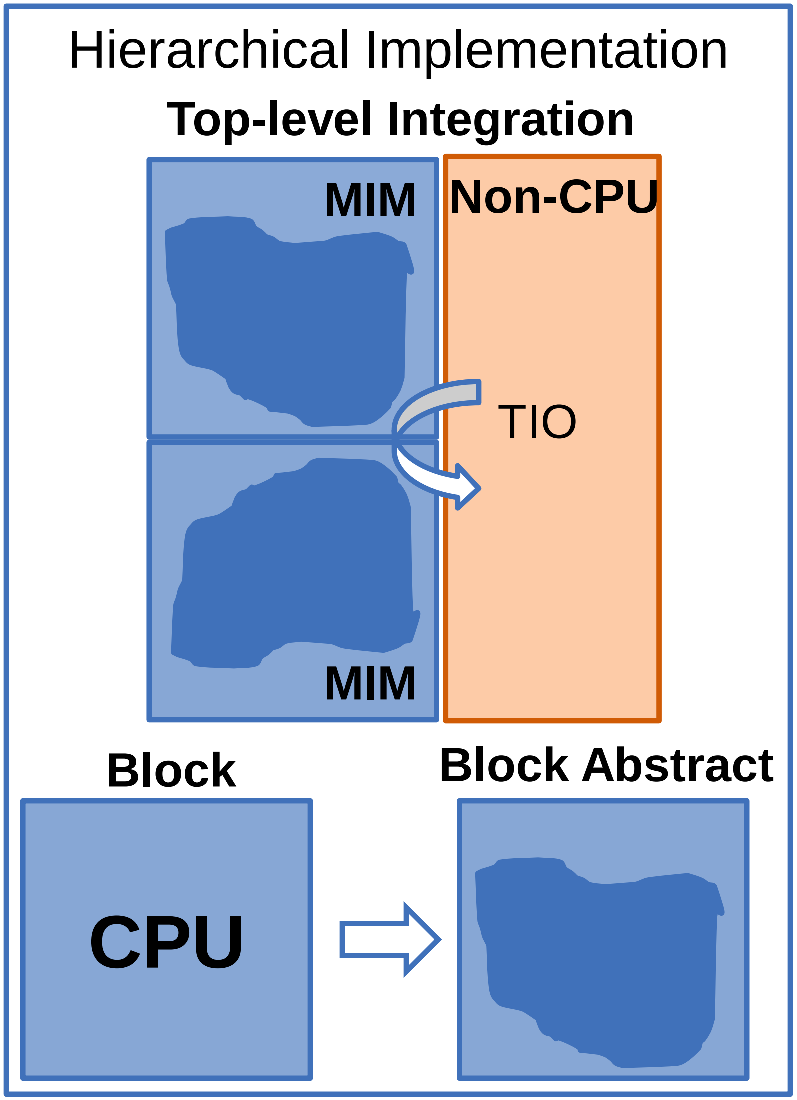
For a Cortex-A15 Dual Core Processor

Synopsys Users Group

Setup	Standard Cell Libraries	Core Configuration	
Flow Development	Flow Setup	**Implementation Strategy**	Trade-offs
Future Improvements	Multisource CTS	Data Flow Analysis	
Conclusions	Results	Top 10 Best Practices	

Implementation Strategy

- Hierarchical implementation
  - 2 CPUs instantiated as *multiple instantiated modules* (MIMs)
  - Top-level includes Non-CPU
- Block-level
  - *Block abstracts (BA)* created for top-level closure
  - BAs reduce memory footprint and runtime in hierarchical flow
- Top-level
  - Used *Transparent Interface Optimization* (TIO) for top-level timing closure



Engineering Trade-offs

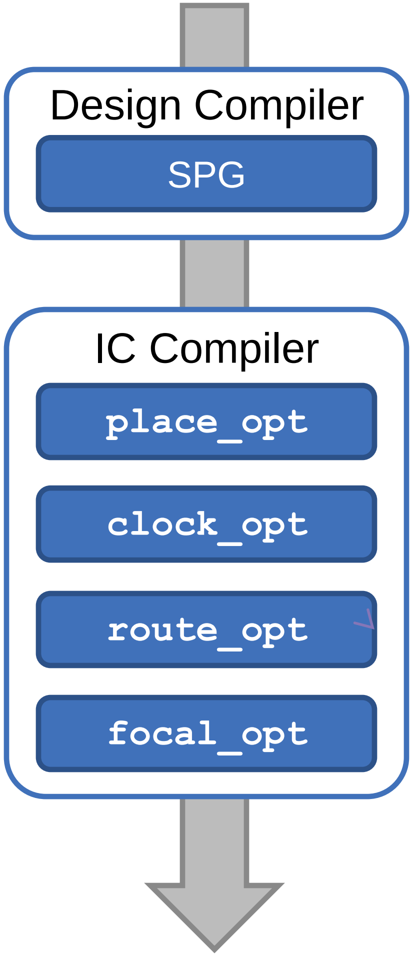
For a Cortex-A15 Dual Core Processor

Synopsys Users Group

Setup	Standard Cell Libraries	Core Configuration	
Flow Development	Flow Setup	Implementation Strategy	**Trade-offs**
Future Improvements	Multisource CTS	Data Flow Analysis	
Conclusions	Results	Top 10 Best Practices	

Engineering Trade-offs

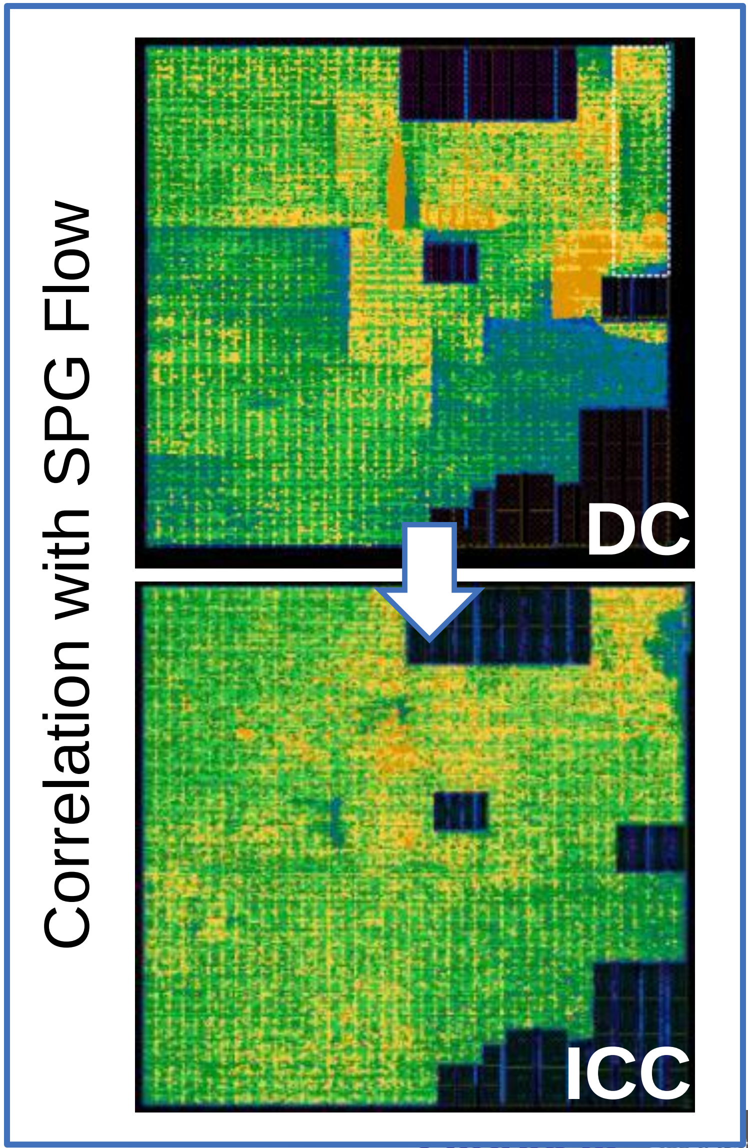
1. Synopsys Physical Guidance (SPG)
2. Placement Bounds
3. Managing Uncertainty
4. Power Managed Flow
5. CTS Customization
6. Crosstalk Mitigation



1. Synopsys Physical Guidance

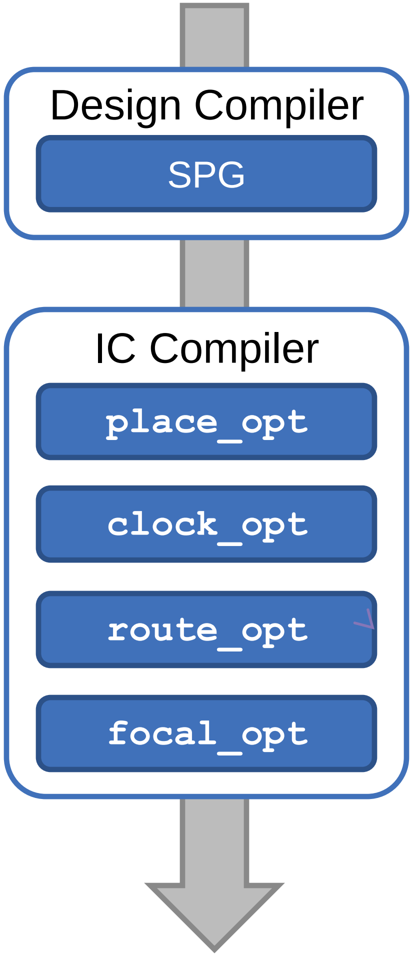
Improved Correlation & QoR for Place & Route

- DC-Graphical Synopsys Physical Guidance (SPG) improves timing correlation between synthesis and placement
- More QoR exploration possible early in the flow
- Placement bounds used during DC placement, removed for ICC
- SPG synthesis TNS needs to be consistent to **`place_opt`** TNS



Engineering Trade-offs

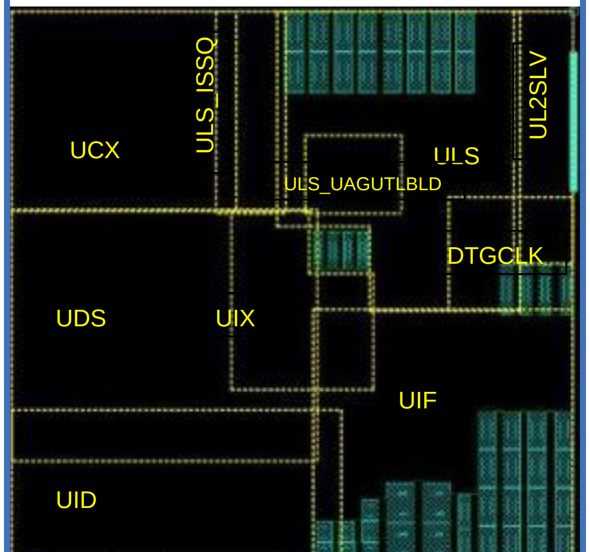
1. Synopsys Physical Guidance (SPG)
2. Placement Bounds
3. Managing Uncertainty
4. Power Managed Flow
5. CTS Customization
6. Crosstalk Mitigation



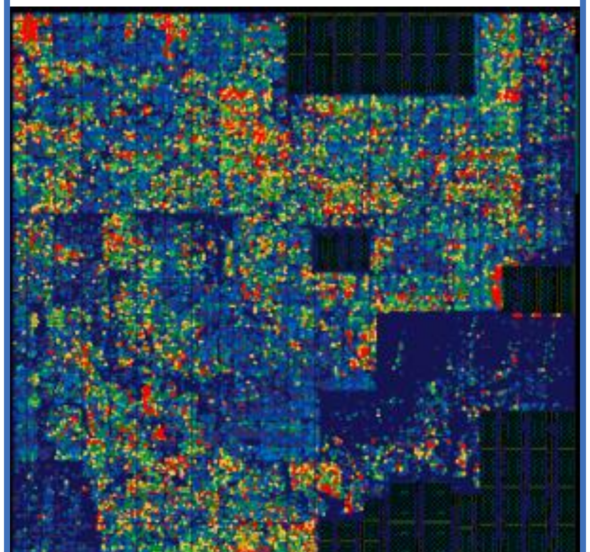
2. Placement Bounds
Use During Synthesis

- QoR improves with synthesis placement bounds
  - Bounds are created from data flow and unbounded placement
    - Cortex-A15 CPU has known data flow which bounds should reflect
  - Examine DC placement with bounds
  - Tune/add bounds for better placement QoR

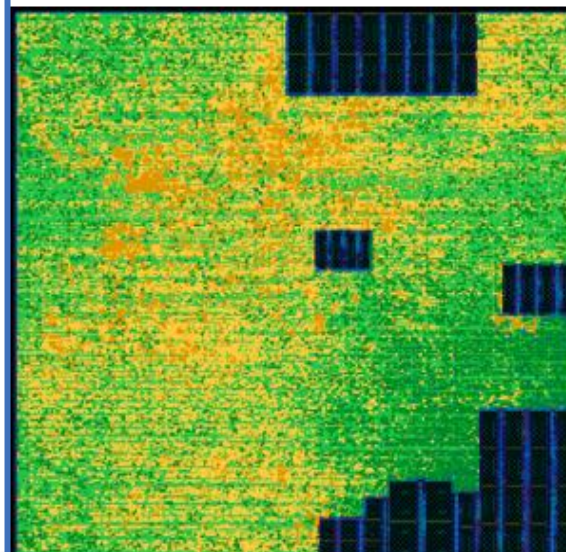
Placement Bounds in DC



Cell Density Map in DC



Cell Density Map in ICC

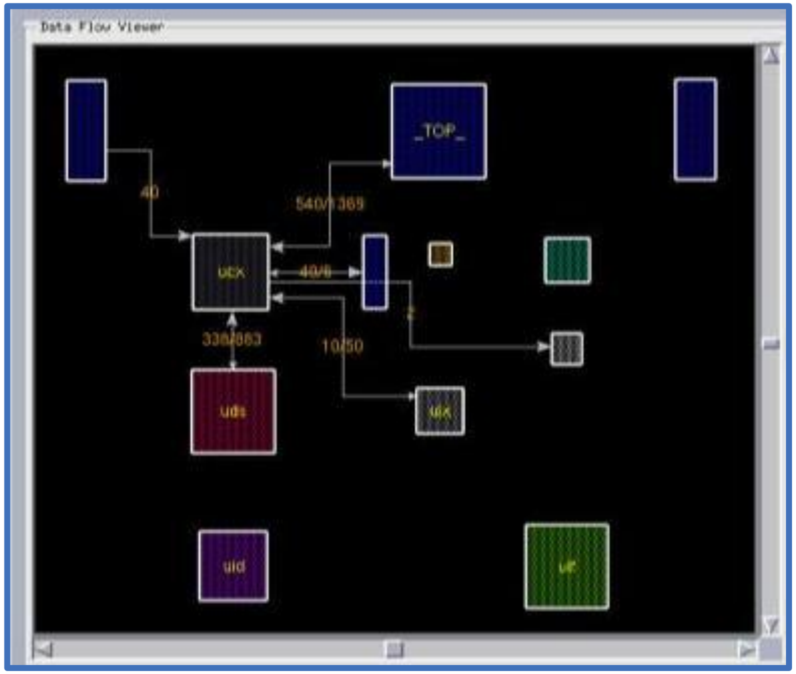


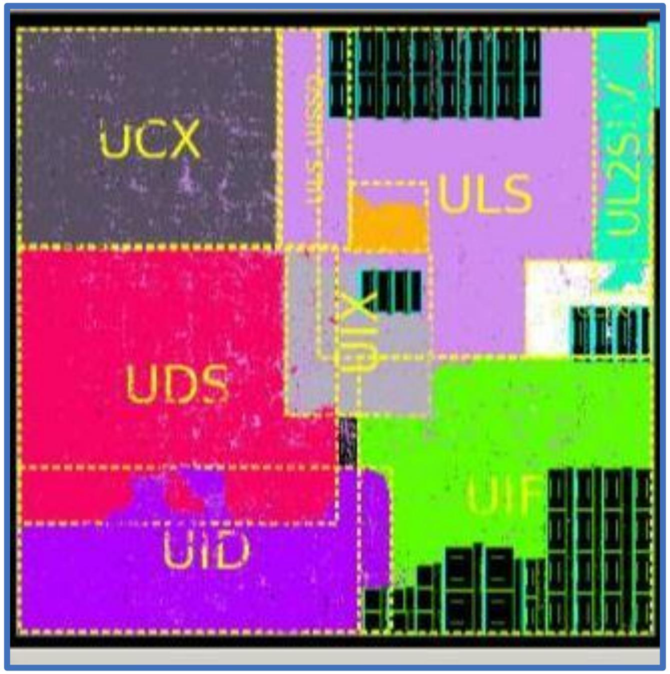
2. Placement Bounds

Use Data Flow Analysis (DFA) to Qualify Bounds

- DFA confirms placement guidance and bounds
- Visualize interconnect count and proximity
- No major issues found

Data Flow Analysis On CPU Block



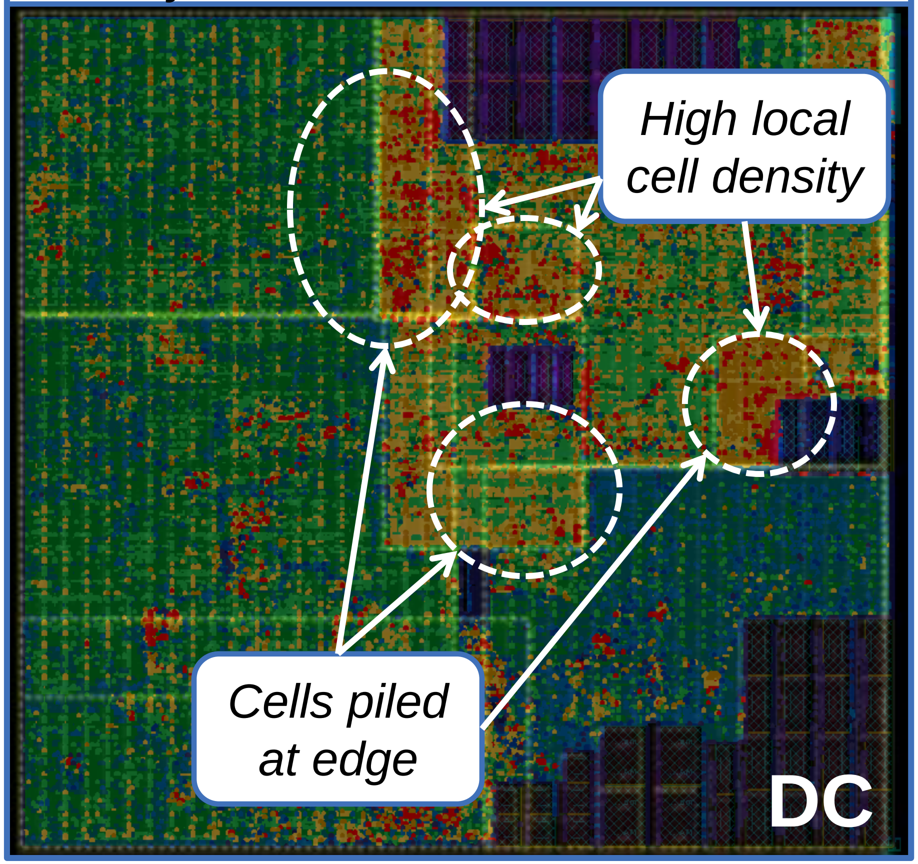


2. Placement Bounds

Bounds Awareness

- Poor bounds cause problems
  - Increased TNS
  - High local cell utilization
- Review bounds quality to reduce these effects
  - Examine placement after DC
  - Identify bounds w/ high utilization
  - Identify bounds w/ cells at edge
- Resize or move bounds to produce more evenly distributed DC placement
- **Always validate bounds quality**

Placement From Early CPU Synthesis Run With Bounds

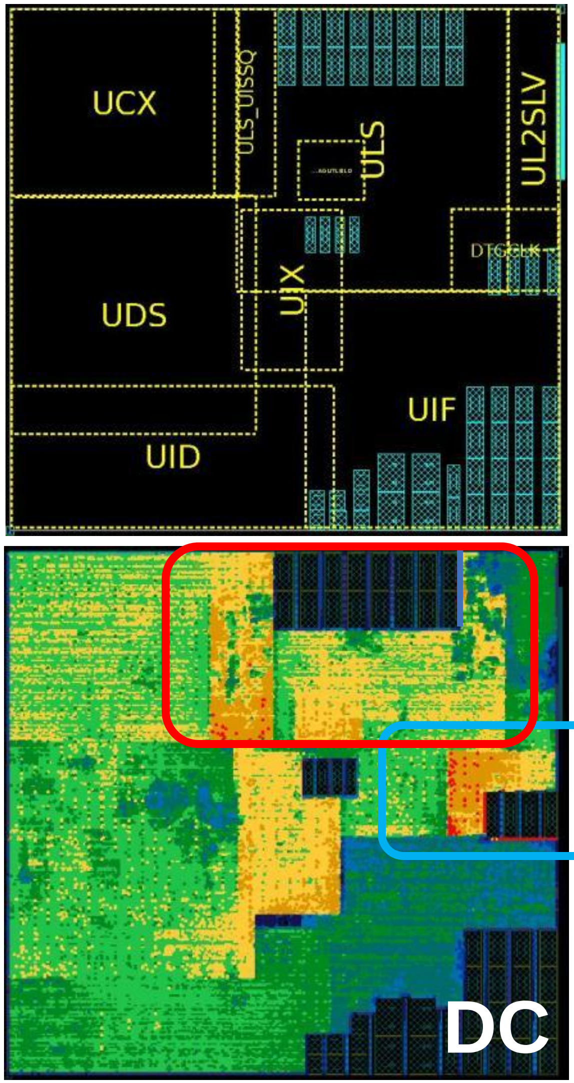


2. Placement Bounds

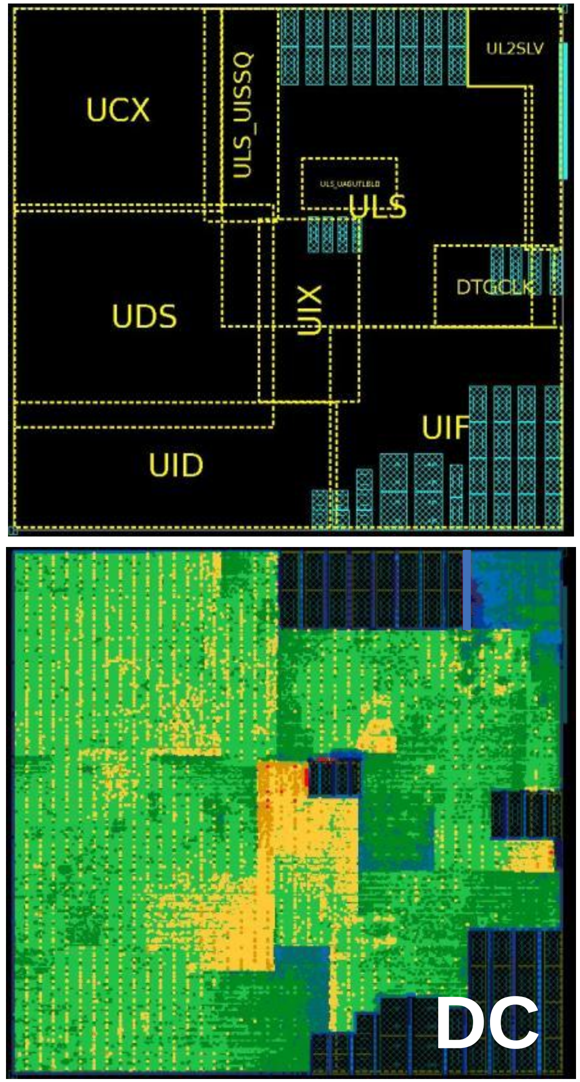
Bounds Refinement

Synopsys Users Group

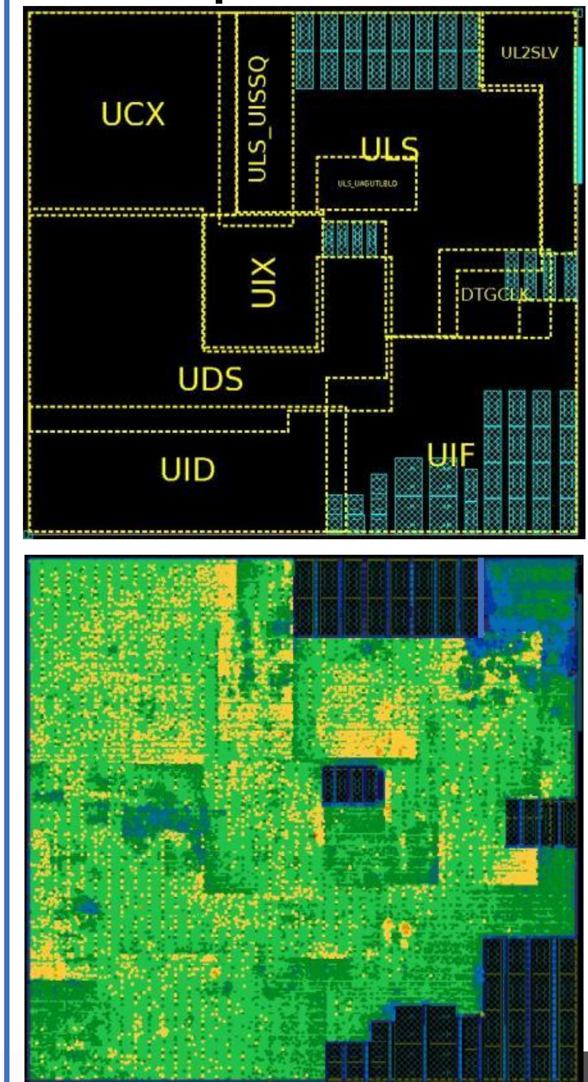
Reference



Experiment 1



Experiment 2

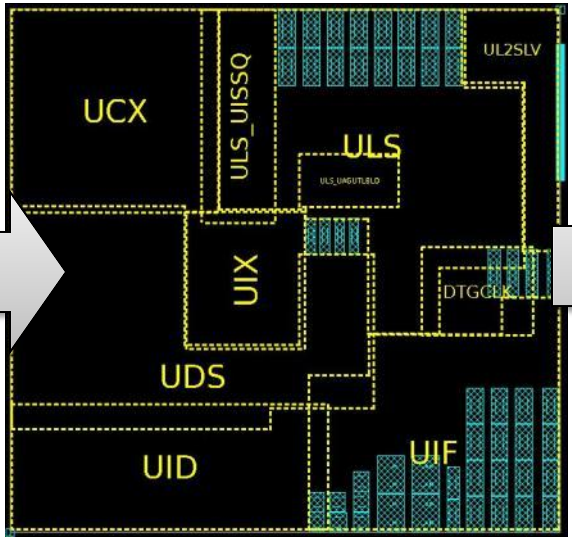


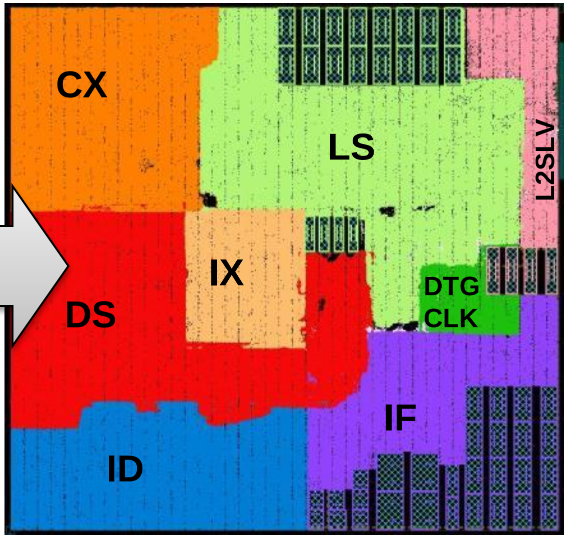
2. Placement Bounds

Bounds Refinement - Leveraged ARM Experience

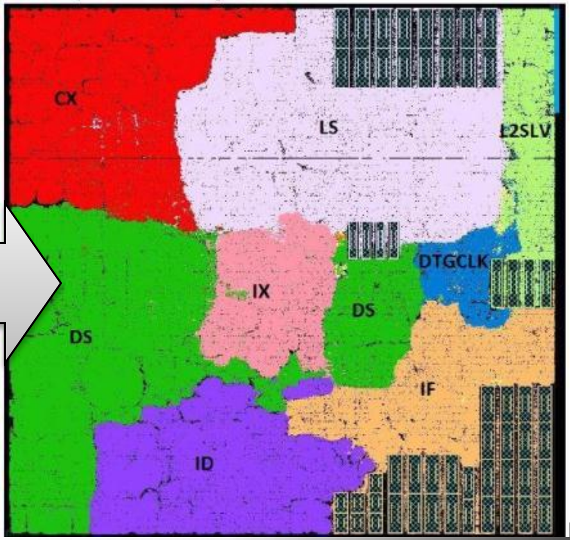
Experiment 2

- Shifted UIX to little left of tag ram
- UDS bound reshaped as per ARM white paper
- UIF bound reshaped to improve density



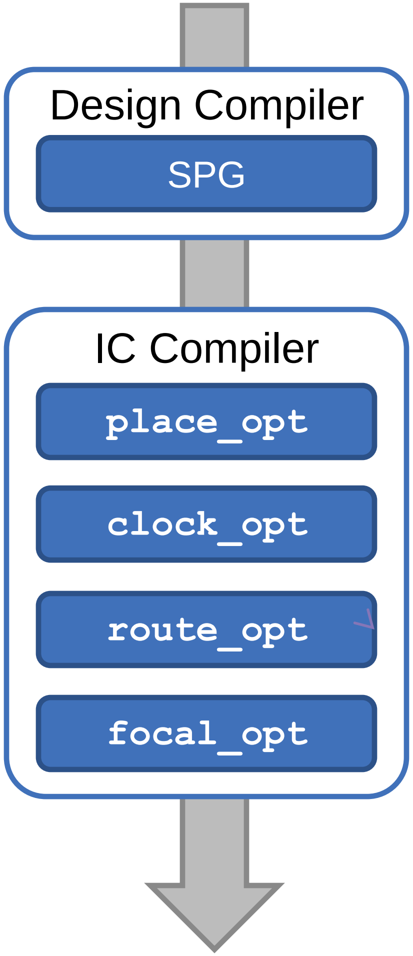


ARM's White Paper
Placement using suggested placement bounds



Engineering Trade-offs

1. Synopsys Physical Guidance (SPG)
2. Placement Bounds
3. Managing Uncertainty
4. Power Managed Flow
5. CTS Customization
6. Crosstalk Mitigation



3. Managing Uncertainty

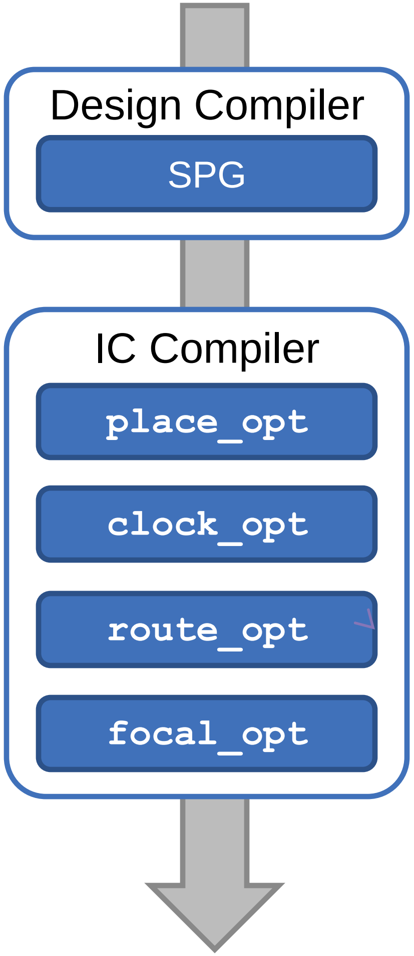
TNS Consistency Has To Be Managed

- SPG adds wire delay estimation to synthesis
  - Timing consistency from DC to ICC
- **No extra margin needed for synthesis**
  - Traditional synthesis adds guard band for wire delay
  - SPG does not need guard band
- Clock uncertainty is reduced during flow development
  - Help flow work when TNS is large (compared to impact of clock tree)
  - Uncertainty in DC and `place_opt` set to correlate TNS

Cortex-A15 CPU	Clock Setup Uncertainty (ps)		
Flow Step	Initial	Flow Dev.	Final
DC SPG	150	0	120
`place_opt`	150	50	120
`clock_opt`	100	0	60
`route_opt`	100	50	50
`focal_opt`	50	50	50
Signoff = 50ps			

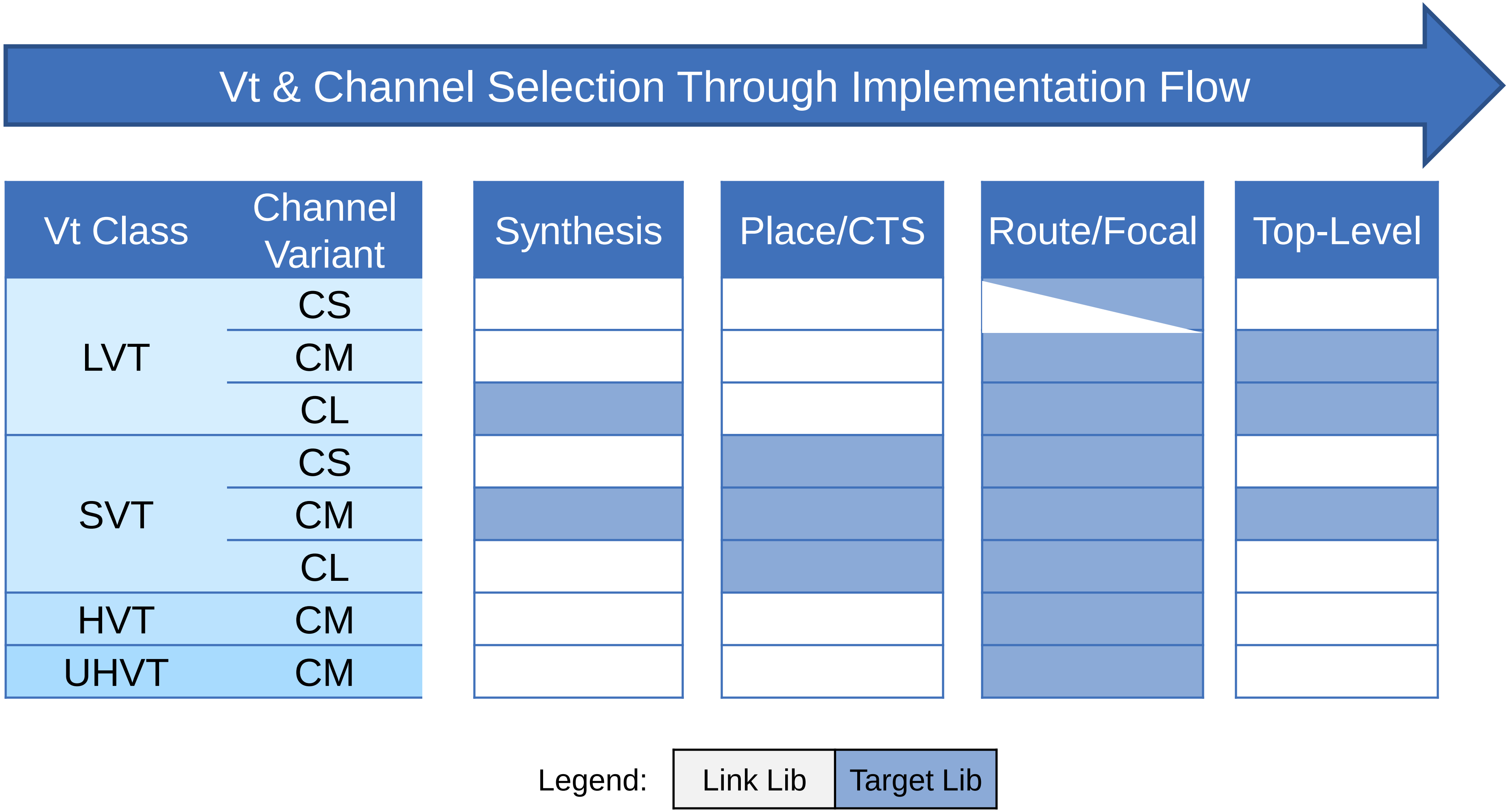
Engineering Trade-offs

1. Synopsys Physical Guidance (SPG)
2. Placement Bounds
3. Managing Uncertainty
4. Power Managed Flow
5. CTS Customization
6. Crosstalk Mitigation



4. Power Managed Flow

Vt & Channel Selection @ Each Implementation Stage



4. Power Managed Flow
Vt & Channel Selection During Synthesis

snug
Synopsys Users Group

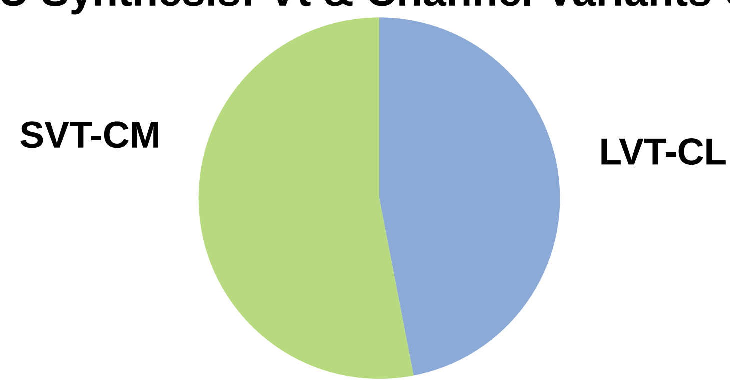
- Started with LVT-CM for flow development
  - Leakage was too high
- Tried SVT-CM in synthesis to keep leakage low
  - Caused high TNS/utilization and FP growth
- Added LVT-CL to synthesis flow
  - Final flow uses SVT-CM and LVT-CL for synthesis

Vt Class	Channel Variant	Synthesis
LVT	CS	
	CM	
	CL	
SVT	CS	
	CM	
	CL	
HVT	CM	
UHVT	CM	

Legend:

Link Lib	Target Lib

CPU Synthesis: Vt & Channel Variants Usage

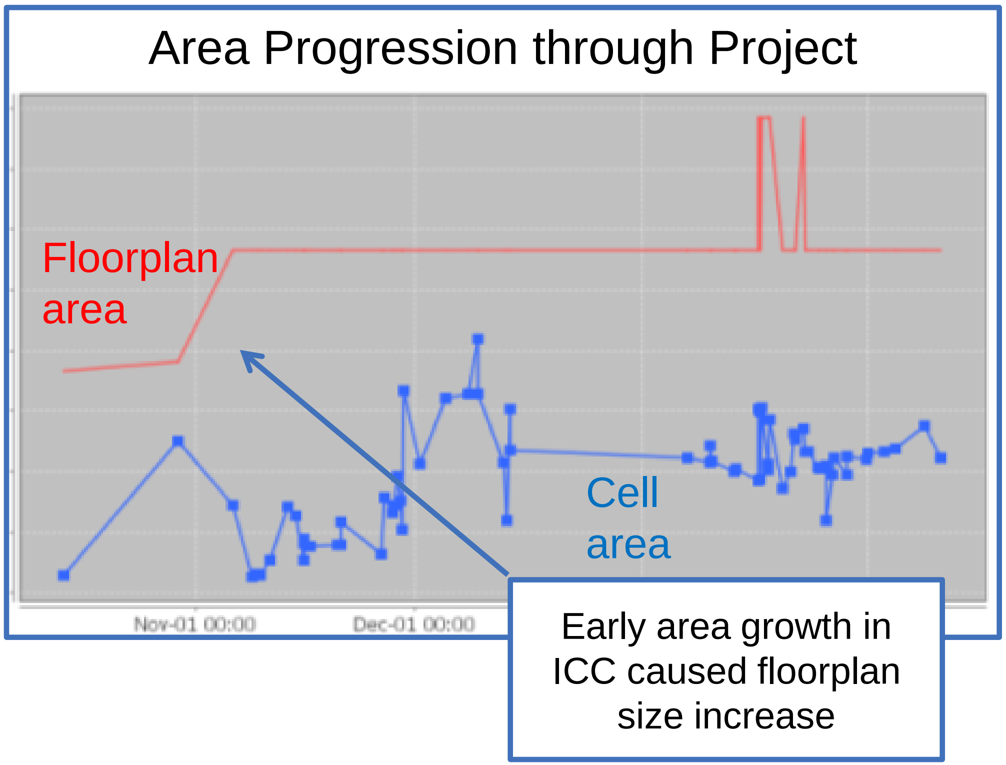


4. Power Managed Flow

Vt & Channel Selection During Placement/CTS

- **SVT-only placement and CTS**
  - Keep leakage power low through the flow

- Watch for area growth early in project



Vt Class	Channel Variant	Place/CTS
LVT	CS	
	CM	
	CL	
SVT	CS	
	CM	
	CL	
HVT	CM	
UHVT	CM	

Legend: Link Lib | Target Lib

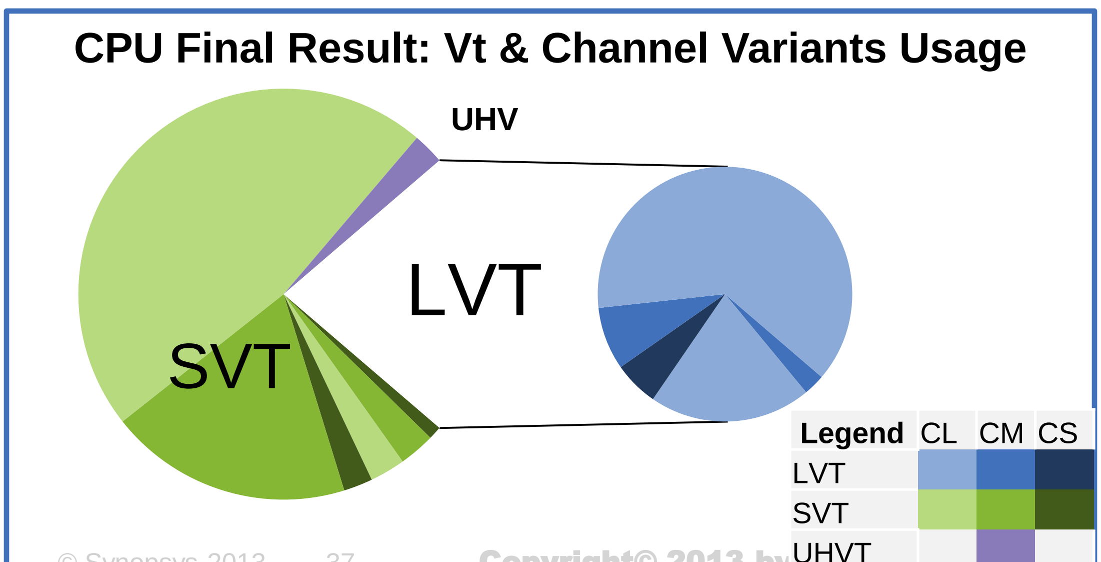
SYNOPSYS® Accelerating Innovation

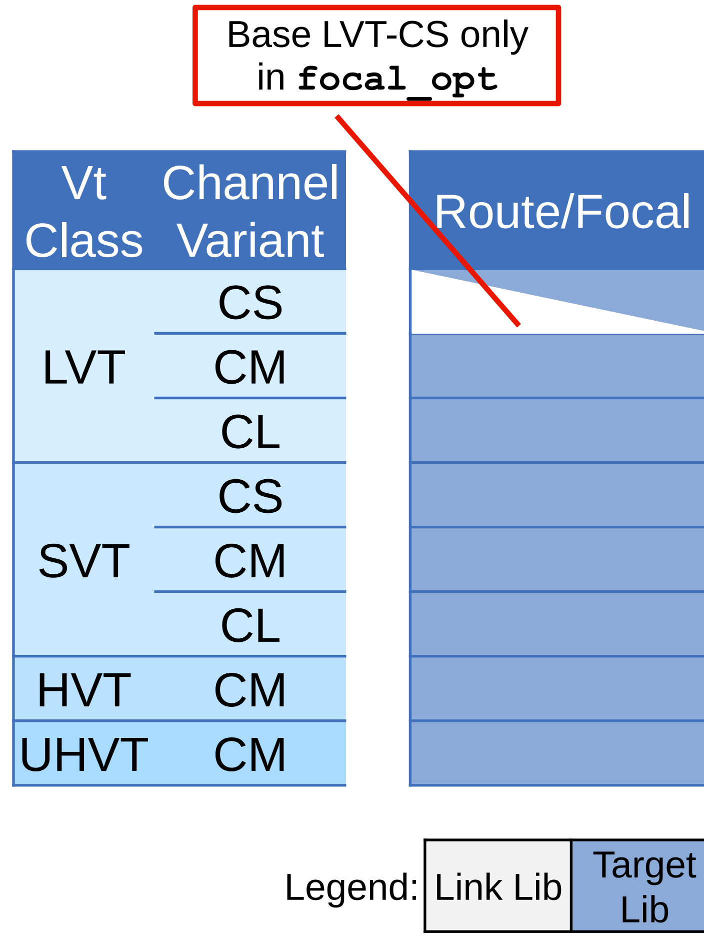
4. Power Managed Flow

Vt & Channel Selection During Route/Focal_opt

- Introduction of crosstalk increases TNS in route_opt
  - Use LVT cells available to reduce crosstalk

- **`Focal_opt` uses all Vt & channel**
  - Improves both timing and power
  - Net reduction in leakage even with 22% LVT





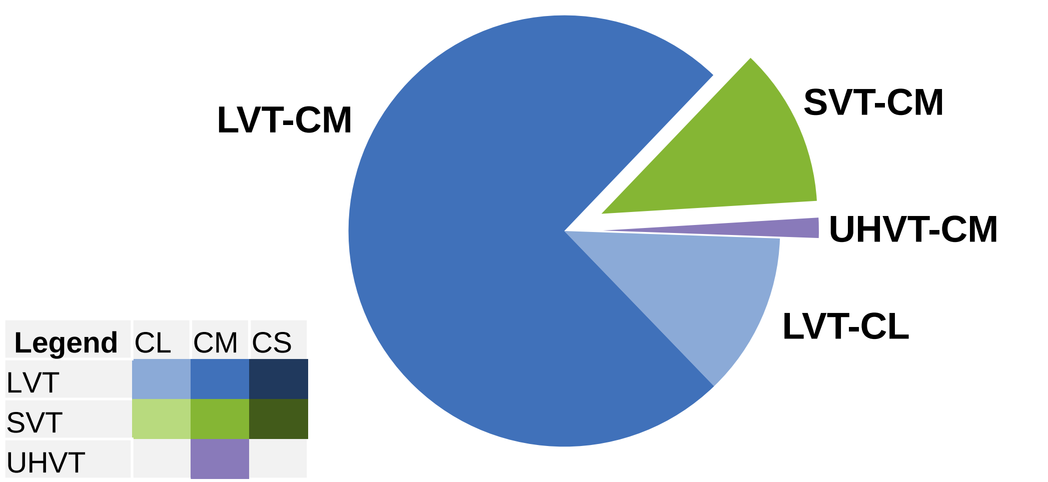
4. Power Managed Flow

Vt & Channel Selection @ Top-Level

Synopsys Users Group

- Top-level is more timing-challenged
  - Crosstalk in L2 cache RAM channels
  - ICG enable timing more critical
  - Higher connectivity in central area
  - Non-CPU sensitive to area growth
- **Added LVT-CM to keep TNS and utilization down**

Non-CPU Placement: Vt & Channel Variants Usage



Vt Class	Channel Variant	Top-level
LVT	CS	
	CM	
	CL	
SVT	CS	
	CM	
	CL	
HVT	CM	
UHVT	CM	

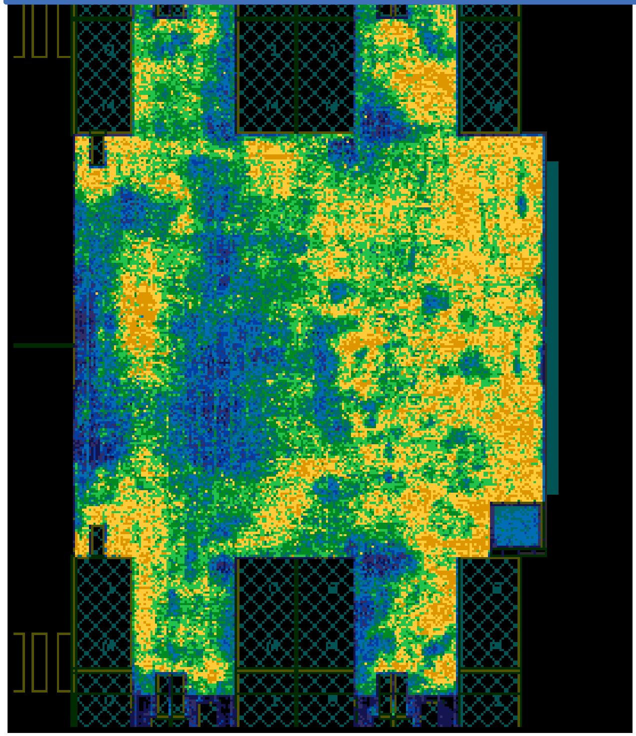
Legend: Link Lib | Target Lib

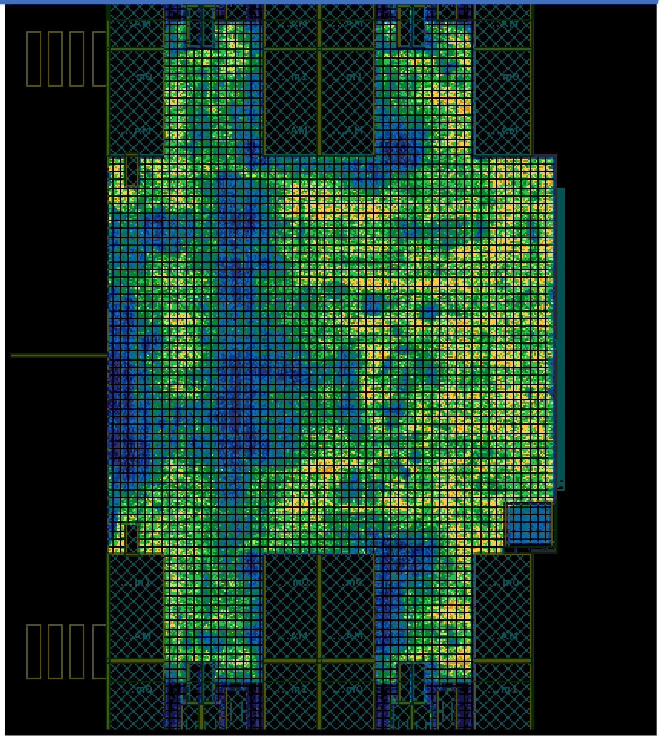
SYNOPSYS® Accelerating Innovation

4. Power Managed Flow

`Place_opt`: Cell Density & Power

Non-CPU Cell Density Maps After `place_opt`





Use of LVT-CM in Non-CPU synthesis provided the expected utilization

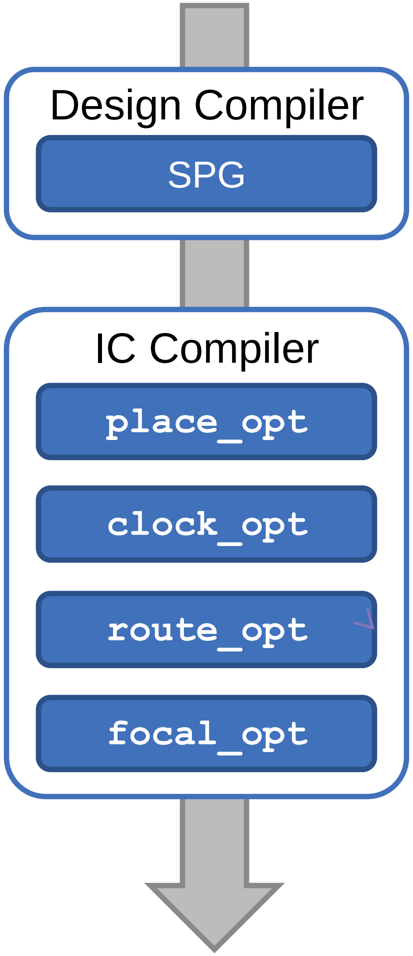
	SVT-CM + LVT-CL	SVT-CM + LVT-CM/CL
Core Area Utilization	67%	61%
Cell Leakage Power	86% target	271% target

Utilization too high

3X increase in leakage

Engineering Trade-offs

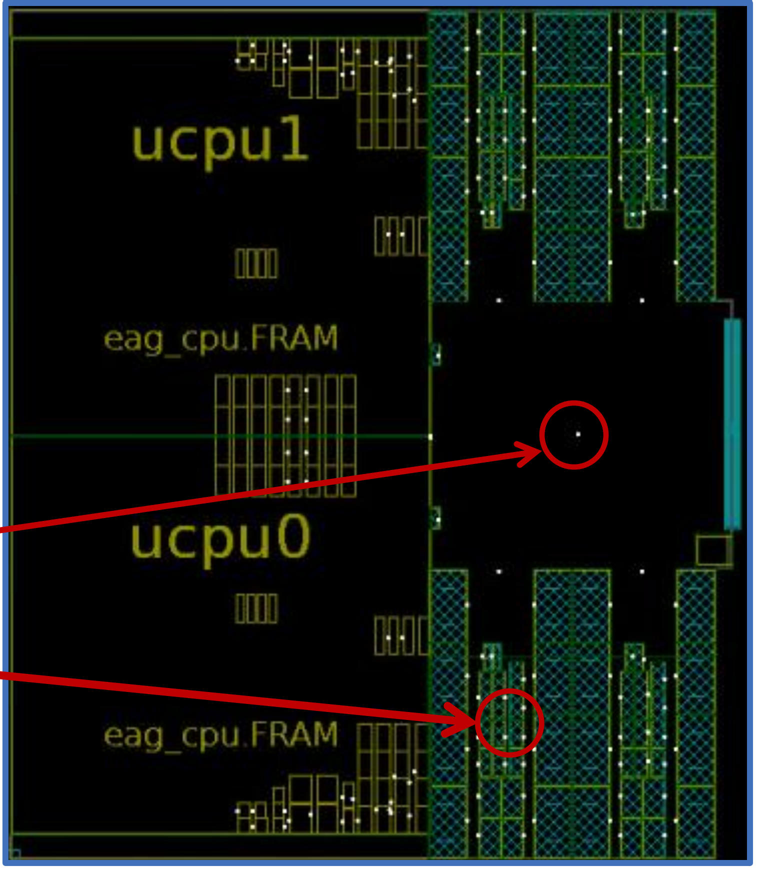
1. Synopsys Physical Guidance (SPG)
2. Placement Bounds
3. Managing Uncertainty
4. Power Managed Flow
5. CTS Customization
6. Crosstalk Mitigation



5. CTS Customization

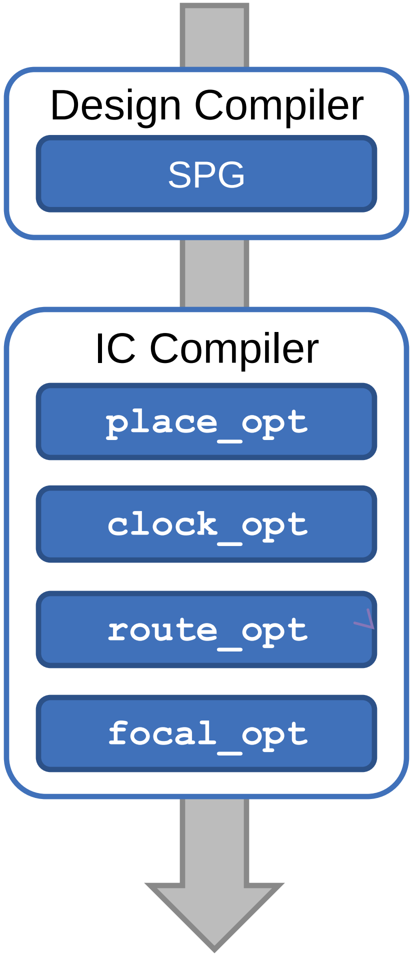
- General
  - **LVT-CS cells for clock drivers**
  - Back-annotated computed latency for integrated clock gating cells (ICGs) to synthesis
- CPU
  - Specify early clock on 2 architectural ICGs
- Non-CPU
  - Fix architectural ICG location in DC
  - Magnet place RAM ICGs
  - Delay clock to RAM ICGs
- Debug Tip: **Validate ICG counts**

Non-CPU ICG Placement



Engineering Trade-offs

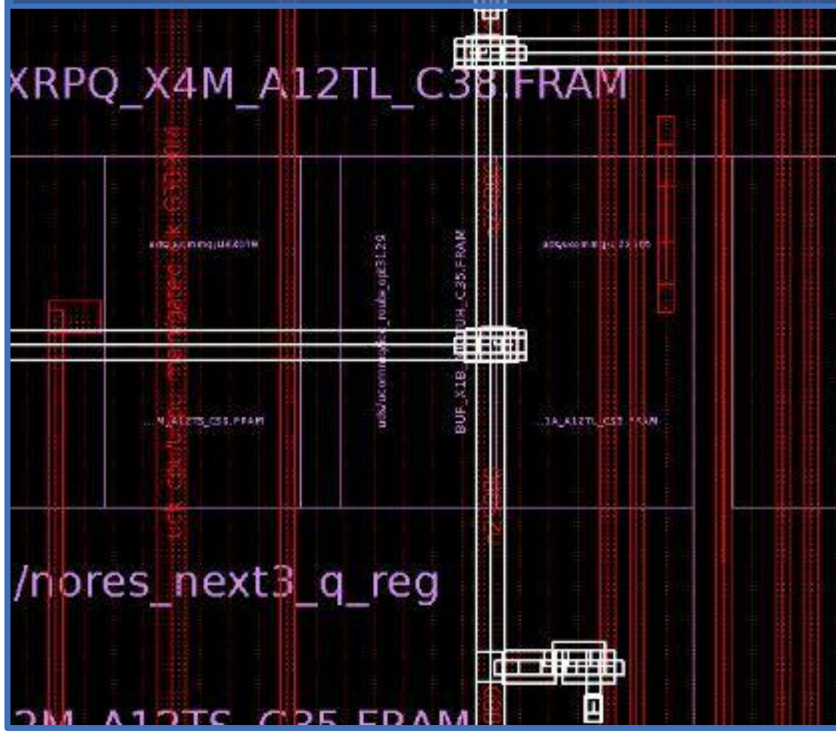
1. Synopsys Physical Guidance (SPG)
2. Placement Bounds
3. Managing Uncertainty
4. Power Managed Flow
5. CTS Customization
6. Crosstalk Mitigation



6. Crosstalk Mitigation

- Crosstalk impact at 28nm is large
  - Non-CPU routing density can be a problem
  - RAM channels were widened to accommodate power routing, switches and clock NDRs
- Clock net rules for crosstalk
  - **CPU used 3X spacing rule**
  - **Non-CPU used shielding + 3X spacing**
- Crosstalk Timing Optimization:
  - Open all LVT classes to allow footprint-compatible swapping

3X Clock Net Spacing In CPU



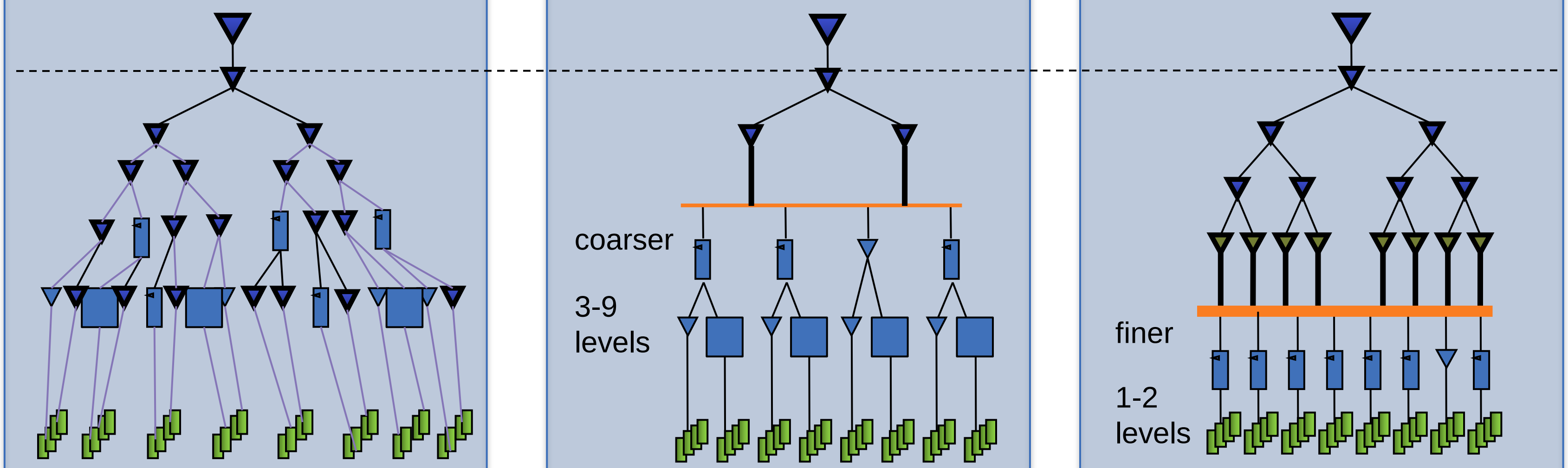
Engineering Trade-offs

For a Cortex-A15 Dual Core Processor

Setup	Standard Cell Libraries	Core Configuration	
Flow Development	Flow Setup	Implementation Strategy	Trade-offs
Future Improvements	**Multisource CTS**	Data Flow Analysis	
Conclusions	Results	Top 10 Best Practices	

Multisource (MS) CTS

Better Skew And OCV Robustness Than Conventional CTS



Traditional CTS

Skew/OCV	✓
Power	✓✓✓

- Automated synthesis of complex clock relationships

Multisource CTS

Skew/OCV	✓✓
Power	✓✓

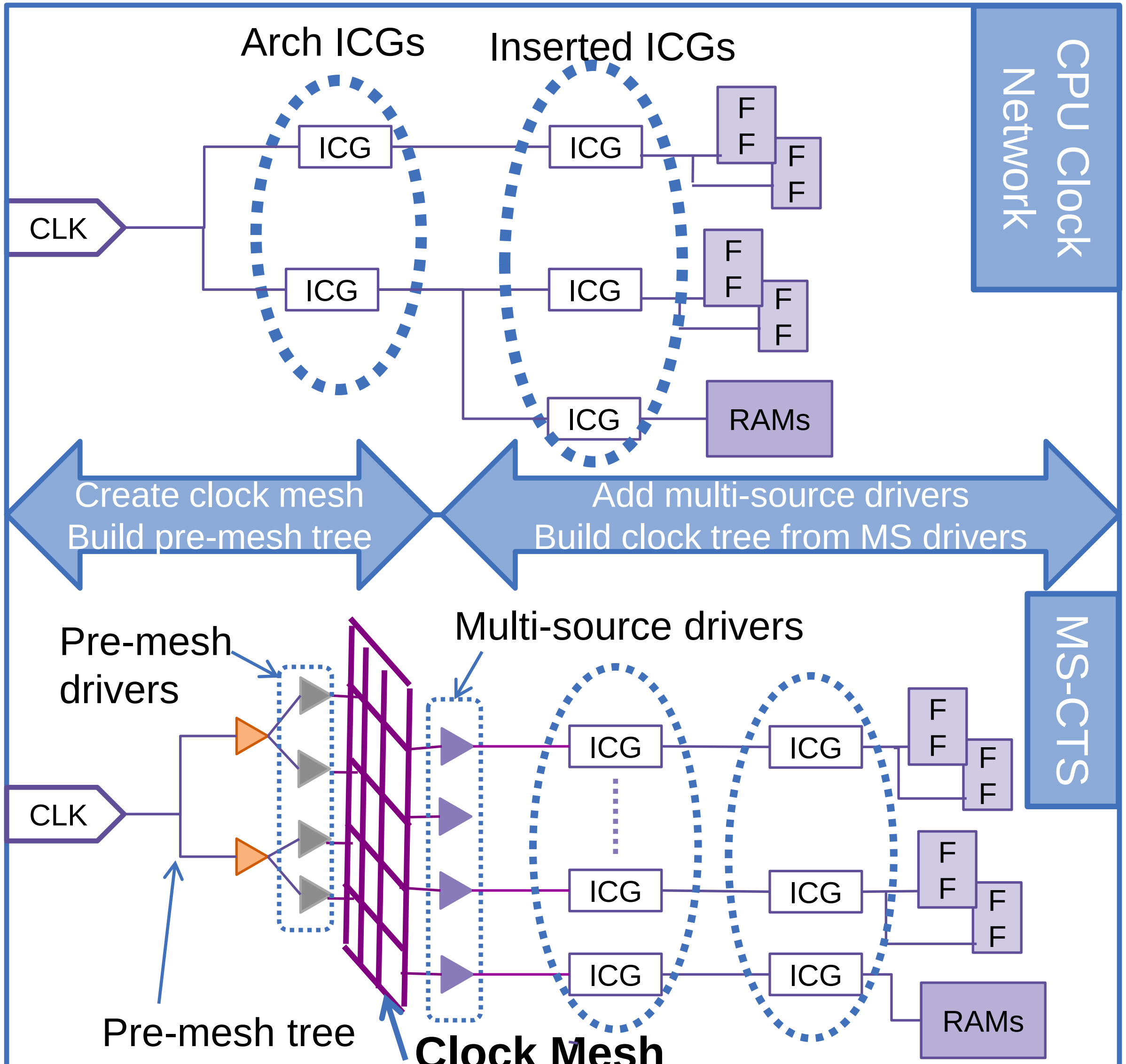
- Coarse mesh uses less power
- More flexible topology

Clock Mesh

Skew/OCV	✓✓✓
Power	✓

- Fine mesh - more power
- Offers lowest skew, highest OCV tolerance

MS-CTS Implementation- CPU



Overall Metrics	Trad. CTS	MS-CTS
Sinks	155K	155K
CTBuffers	4.9K	5K
BufferArea	11K	12K
Global Skew	61 ps	32 ps
Local Skew	52 ps	26 ps
Latency	863 ps	768 ps

__Note__
Not used in current implementation due to lack of SPICE models

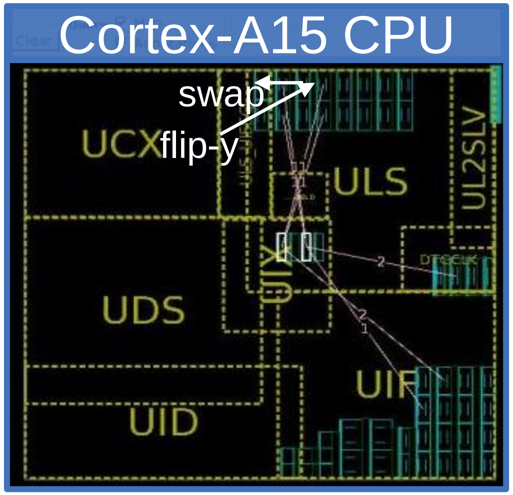
Engineering Trade-offs

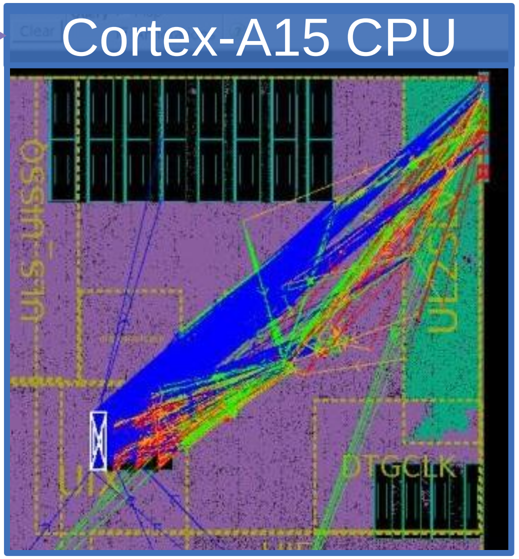
For a Cortex-A15 Dual Core Processor

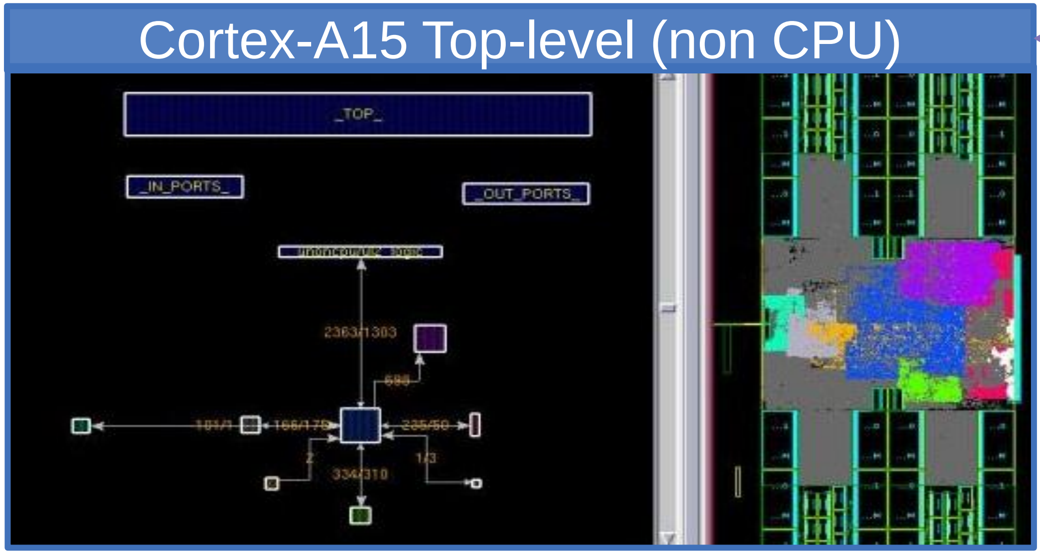
Setup	Standard Cell Libraries	Core Configuration	
Flow Development	Flow Setup	Implementation Strategy	Trade-offs
Future Improvements	Multisource CTS	**Data Flow Analysis**	
Conclusions	Results	Top 10 Best Practices	

Data Flow Analyzer (DFA)

- Advanced Flyline Analysis
  - Debug macro placement in CPU
  - Analyze IO Critical Paths in CPU
- Data Flow Analysis
  - Confirm Placement guidance and bounds in Non-CPU







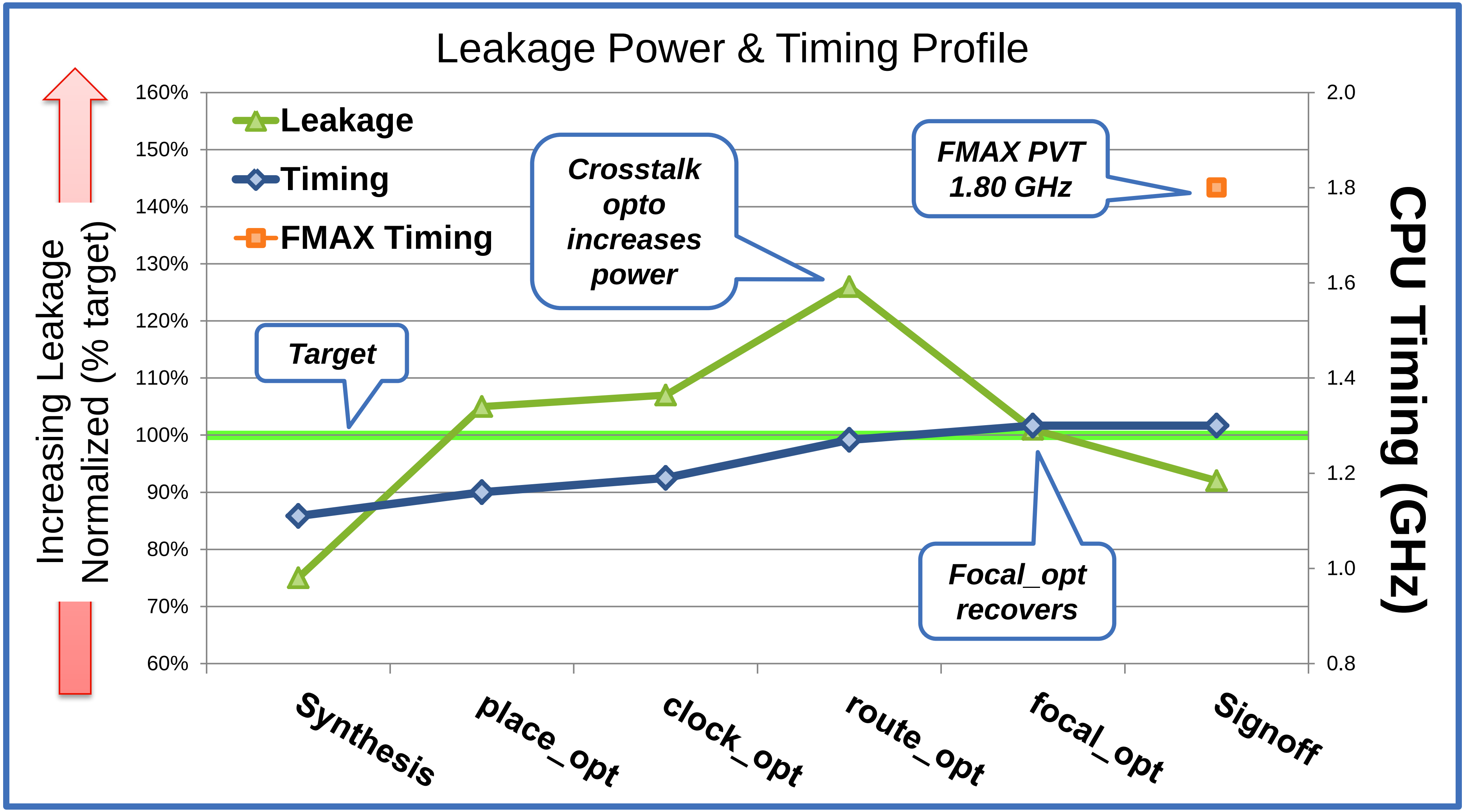
Engineering Trade-offs

For a Cortex-A15 Dual Core Processor

Setup	Standard Cell Libraries	Core Configuration	
Flow Development	Flow Setup	Implementation Strategy	Trade-offs
Future Improvements	Multisource CTS	Data Flow Analysis	
Conclusions	**Results**	Top 10 Best Practices	

Results

Consistent Power/Timing @ Each Step Vs Spec Target



Engineering Trade-offs

For a Cortex-A15 Dual Core Processor

Synopsys Users Group

Setup	Standard Cell Libraries	Core Configuration	
Flow Development	Flow Setup	Implementation Strategy	Trade-offs
Future Improvements	Multisource CTS	Data Flow Analysis	
Conclusions	Results	**Top 10 Best Practices**	

Top 10 Best Practices

For A Power & Timing Balanced Implementation

1 Beat leakage target in synthesis

- Restrict Vt classes and reduce uncertainty to keep power low

2 Review placement bounds for quality

3 Do not over constrain timing in DC or ICC

- No extra margin needed for DC SPG
- Keep ICC uncertainty and derating low to keep power down

4 Timing/Power consistency throughout the flow

- Use SPG flow in DC and ICC

5 Manage Vt class and channel length throughout the flow

- Introduce leakier classes only to keep timing/area in check
- Judicious use of LVT (try not to use LVT-CS until focal_opt)

Top 10 Best Practices

For A Power & Timing Balanced Implementation

6 Watch cell density, because it impacts power/timing

- Areas can rise early in project due to constraints/bounds
- Grow floorplan, use LVT or reduce margin to keep area in check

7 Manage for power at each flow step

- Understand each rise in power through the flow
- Use SVT in placement and CTS to reduce power

8 Use -power & enable power-aware optimization

- Enabled by HPC flow variable

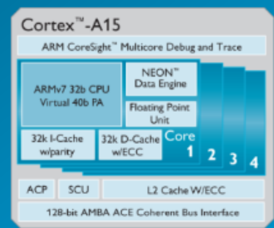
9 Use aggressive clock NDR or shielding for crosstalk prevention

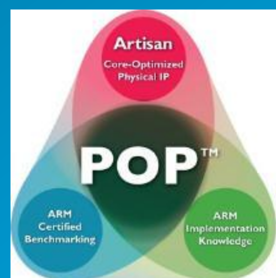
10 **Aggressive power and timing tradeoff is possible!**

ARM + Synopsys Collaboration

snug
Synopsys Users Group

- Cortex-A15 dual core processor
- TSMC 28HPM process
- ARM POP™ IP: core optimized standard cells and fast cache instances





+

Galaxy

PrimeTime

Design Compiler

IC Compiler

StarRC

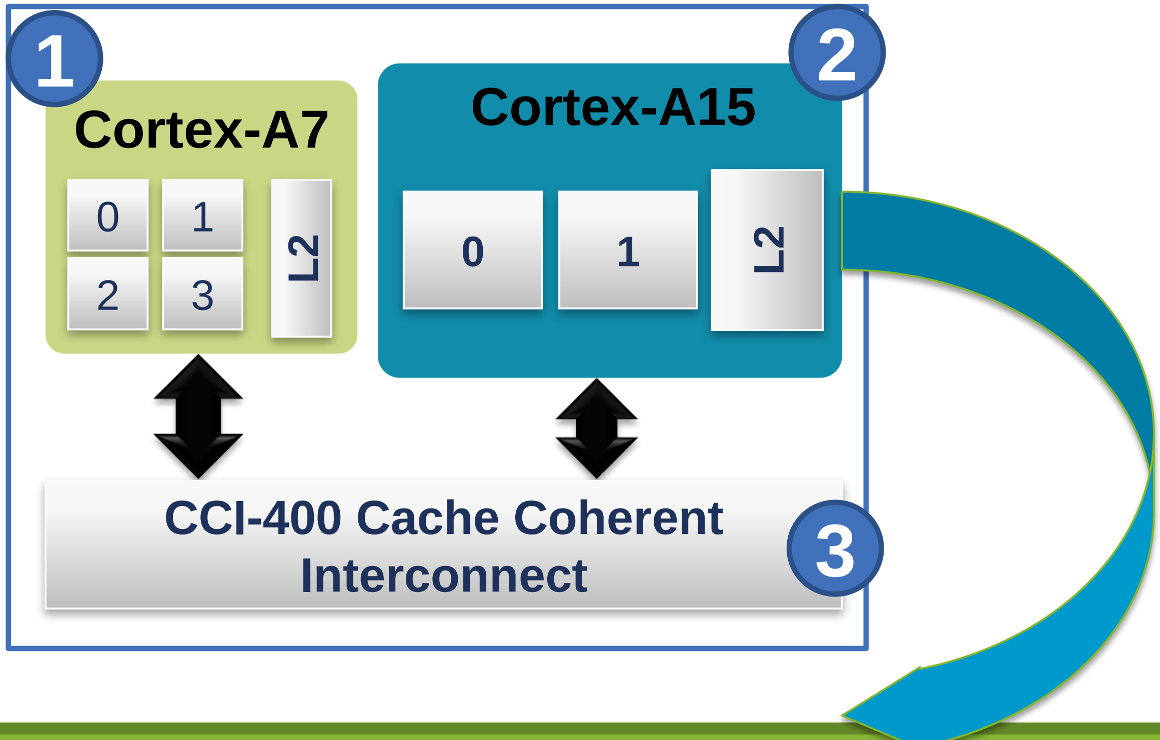
High Performance Core (HPC) scripts + Timing/Power Trade-off Expertise

Reference Implementation for an ARM Cortex-A15 Processor
Optimized for low power and performance
Available Through SolvNet To Joint Customers Today!

Reference Implementation

Collateral & Availability (1/2)

- Available for key components of the ARM big.LITTLE system



Reference Implementation for the ARM Cortex-A15 Processor
Your best starting point for optimized implementation!

Reference Implementation

Collateral & Availability (2/2)

- ARM & Synopsys joint customers can download RI scripts & documentation from:

www.synopsys.com/ARM-Opto

- For other processor cores, contact Synopsys technical support to help you configure and deploy HPC scripts
- For further optimization and customization support, contact Synopsys Professional Services

Reference Implementation for the ARM Cortex-A15 Processor
Your best starting point for optimized implementation!

High-Perf. Core Implementation

Sessions of Interest - Tuesday, March 26th

Presenters	Time	Session
Synopsys Lunch & Learn	12:00 PM to 1:30 PM	1. Optimization Exploration of ARM® Cortex™ Processor-Based Designs with the Lynx Design System
ARM & Synopsys Joint Tutorial	1:30 PM to 3:30 PM	2. Power-centric Timing Optimization of an ARM® Cortex™-A7 Quad Core Processor 3. Engineering Trade-Offs in the Implementation of a High Performance ARM® Cortex™-A15 Dual Core Processor
Broadcom MediaTek Samsung STMicroelectronics Customer Panel	4:15 PM to 5:15 PM	4. Achieving Optimum Results on High Performance Processor Cores

snug
Synopsys Users Group