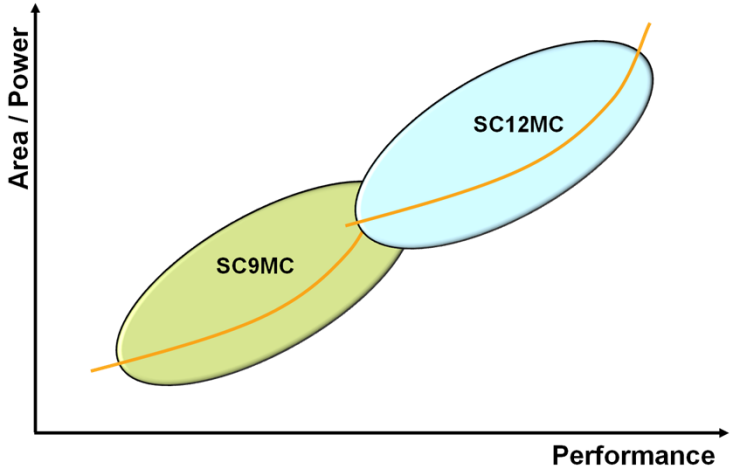
Standard Cell Benchmarking: Avoiding 5 Common Pitfalls

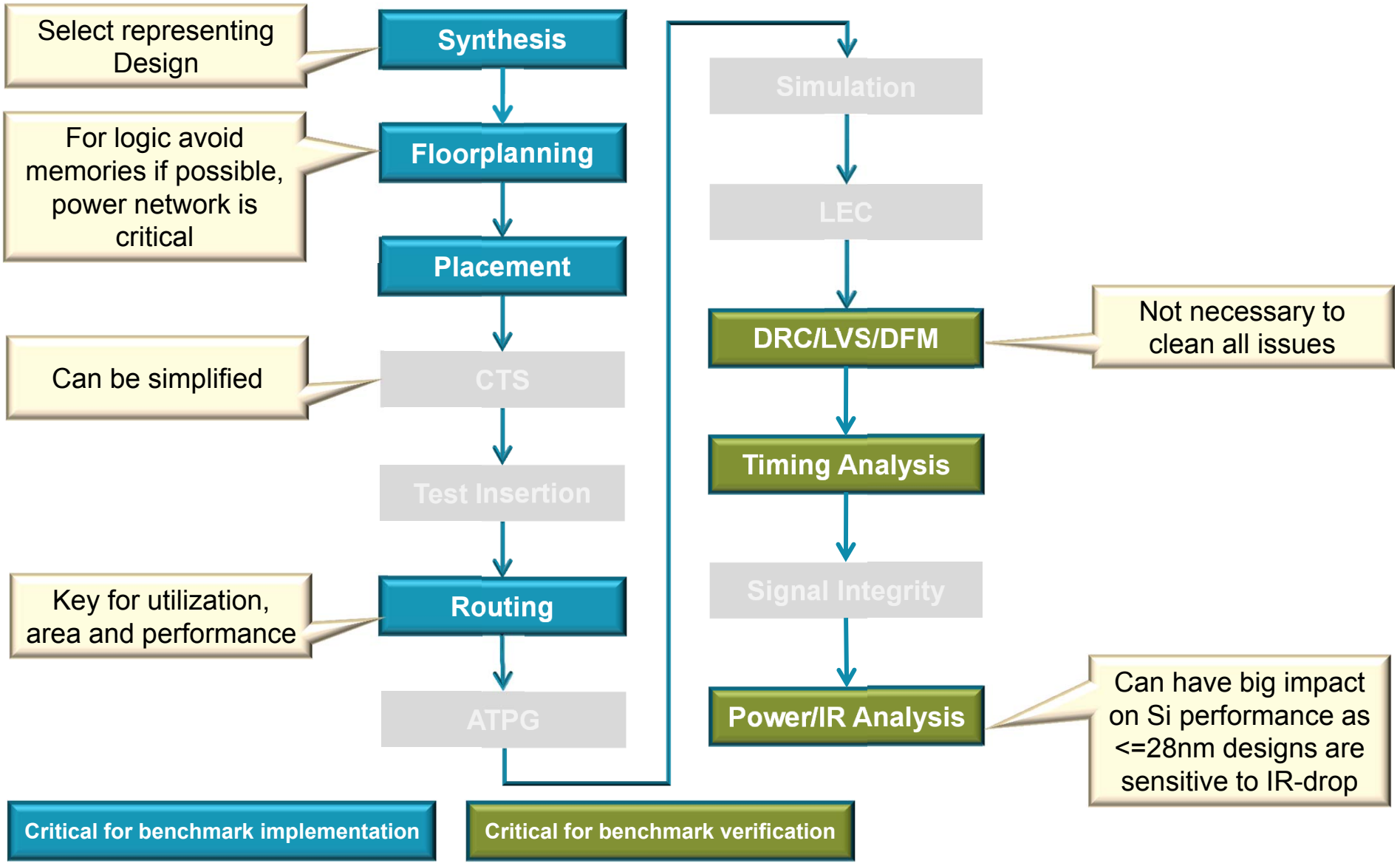
2012/2013

Benchmarking Standard Cells

- Benchmarking is an important component of library selection
- Proper evaluation yields an appropriate selection of standard cell libraries
  - Optimize power, performance and area for your design
  - Compare “like” to “like”
- Pitfalls in benchmarking can lead to erroneous results
- What are these common pitfalls?
- How can they be avoided?



Critical Logic Benchmark Steps



Pitfall #1: Proper Constraints

Set proper constraints

- Control maximum transitions
  - ARM standard cells are clean to foundry guidelines for electromigration (EM)
  - EM is ensured by controlling edge rates, also referred to as transition times
  - Set in synthesis: 2/3 of the Liberty value
- Control maximum wire length
  - Wires have significant resistance variations in small geometries
- Control IO constraints
  - Relax if not important to prevent penalizing high frequency libraries
- Controlling transition and wire lengths have other benefits
  - Reduces long term reliability issues
  - Prevents timing closure issues during STA
  - Timing closure issues can arise due to wide wire resistance variation
    - Causes extrapolation outside tables
    - Seen at fast process corners and high temperature
- Match constraints for all benchmarks

Pitfall #1: Proper Constraints, “dont_use” Lists

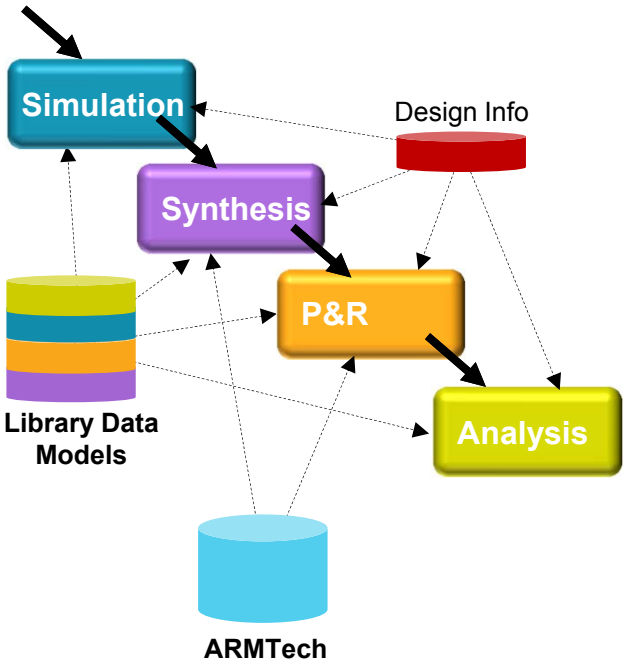
Set proper constraints

- “dont_use” lists should vary with design targets
- High Performance targets
  - Don’t use low drive cells, except low drive BUF/INV for hold fixing
  - Don’t use cells targeting low power or high density
  - Use any cell types with optimized timing arcs
- High Density / Low Power targets
  - Use low drive cells
  - Don’t use cells with high pin densities
  - Don’t use cells targeting high performance (ex: cells with optimized timing arcs)
- All benchmarks
  - Don’t use delay cells, except for hold fixing
  - Don’t use specialty cells with same function as general cells (ex: latch-free clock gate)
  - Don’t use non-scan flops, unless required by design

Pitfall #2: Parasitics

Account for parasitics

- WLM refers to the wire load models included in the Liberty (.lib) model
- Very inaccurate for most design
- Why? Wiring parasitics can dominate, especially at small geometries



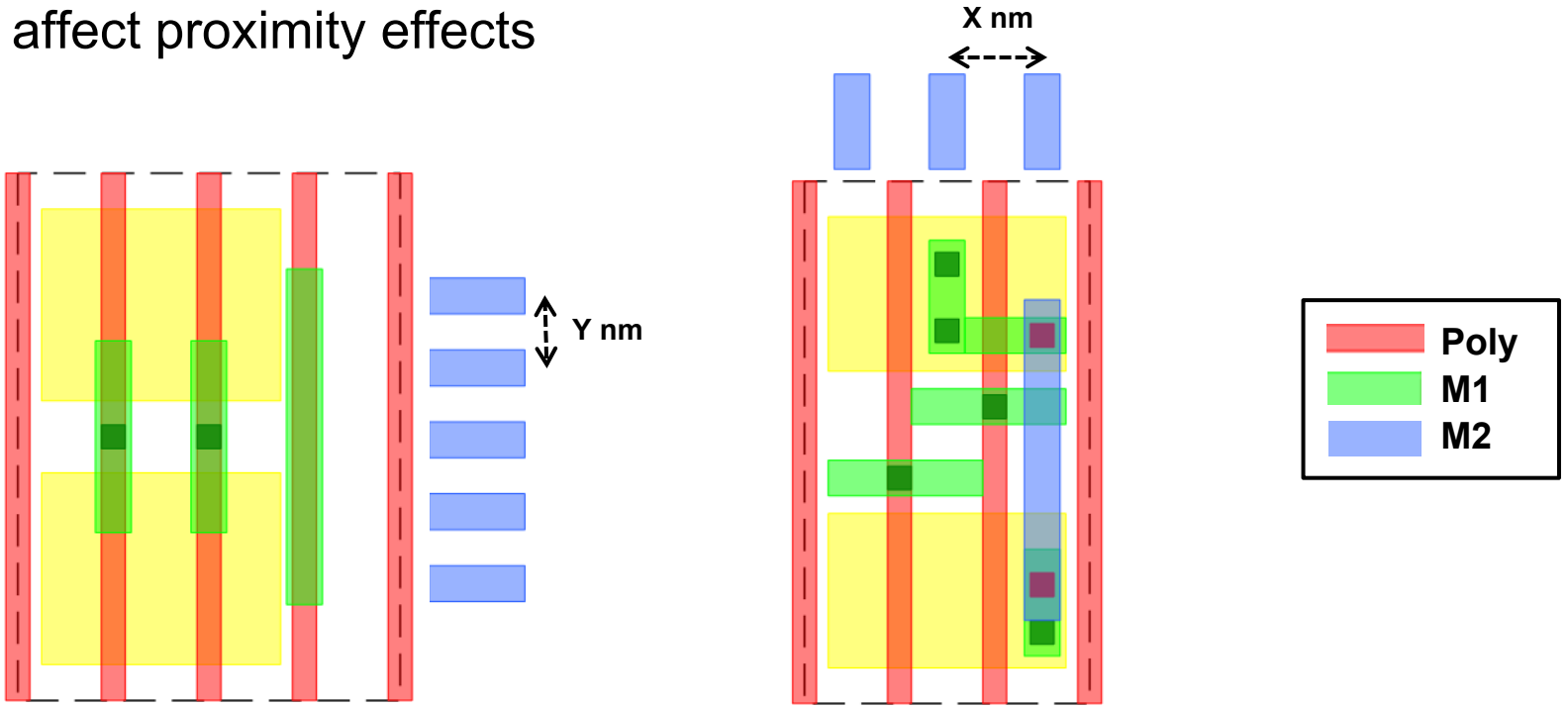
- If WLMs must be used, the SAME WLMs should be used for all vendors

- Use parasitic tables and physical synthesis
  - Physical synthesis supported by synthesis tools
  - Uses parasitic tables
    - Included in the ARM Routing Technology Kit or
    - Provided by the foundry

Pitfall #3: Power Network

Align power network to the architecture

- Architecture affects routing
  - Determines metal direction(s), best VIAs, pin accessibility
  - Direct relationship to best power grid
  - May affect proximity effects



Different architectures with different metal directions.
Ideal power networks will vary.

Pitfall #3: Power Network and Routability

Align power network to the architecture

- The power network is an important part of a benchmark
  - Poor cell design can result in a significant area hit with a power grid in place
  - Poor power grid can unnecessarily block routing resources
- ARM architects standard cells for optimal use with a power grid
  - Layouts consider a lower 2D grid on M2/M3 for libraries using M2 rails
- A proper power grid
  - Minimizes IR drop and EM
  - Allows for fast di/dt (switching) response
  - Uses narrow evenly spaced straps placed frequently
  - Places power and ground straps alternately
- A benchmark without a power grid does not account for library routability

Power Supply Overview

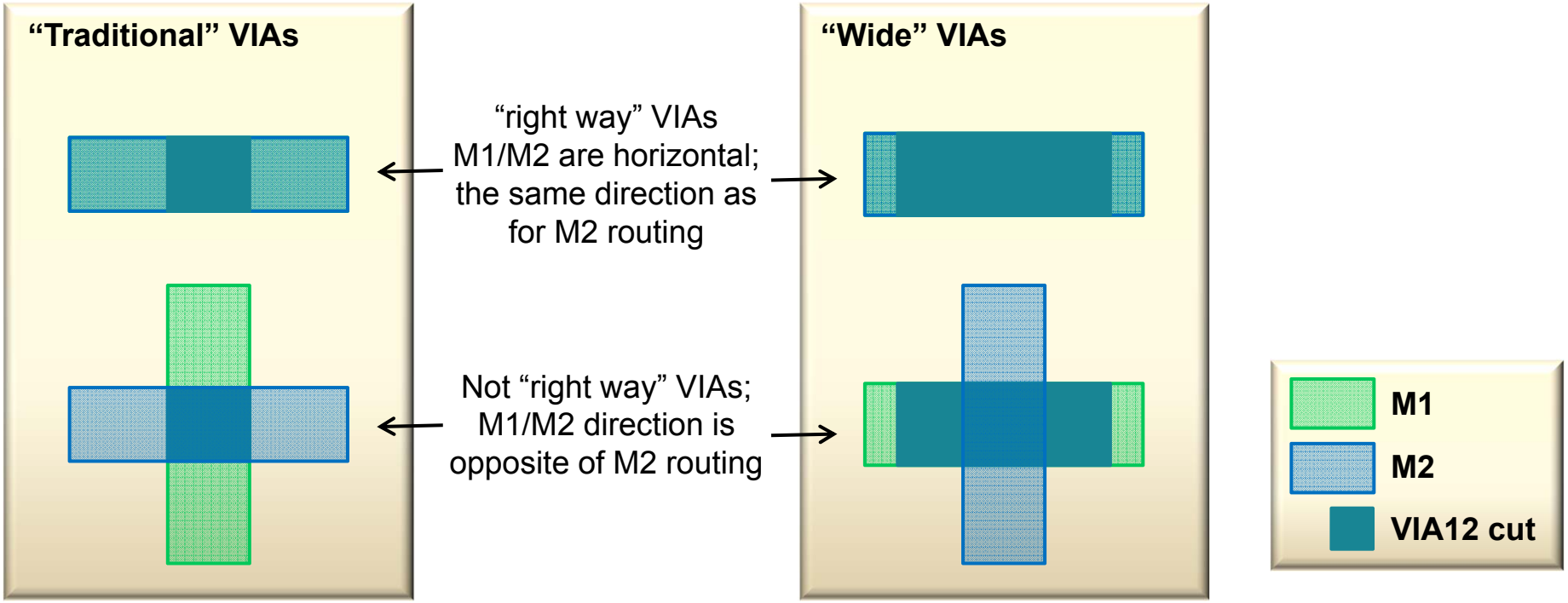
Recommendations & Considerations

Standard Cell Designer (IP Vendor)	Chip/block Design (Customer)	Reason
Largest cell supply rail possible		▪ Reduces IR drop ▪ Improves EM characteristics ▪ Allows for greater redundancy of supply VIAs (IR, EM, yield)
	Lower 2D supply grid ▪ M2/M3 grid for libraries w/M2 rails ▪ M1/M2 grid for libraries w/M1 rails	▪ Reduces di/dt ▪ Reduces resistance ▪ Reduces on-chip inductance to local decoupling capacitance ▪ Reduces VIA resistance versus using only an upper supply grid
	Upper 2D supply grid (top 2 metals)	▪ Reduces IR drop ▪ Handles global supply distribution ▪ Improves routing by freeing lower planes for signal routing

Pitfall #4: Tech file for routing

Use the appropriate tech file for routing

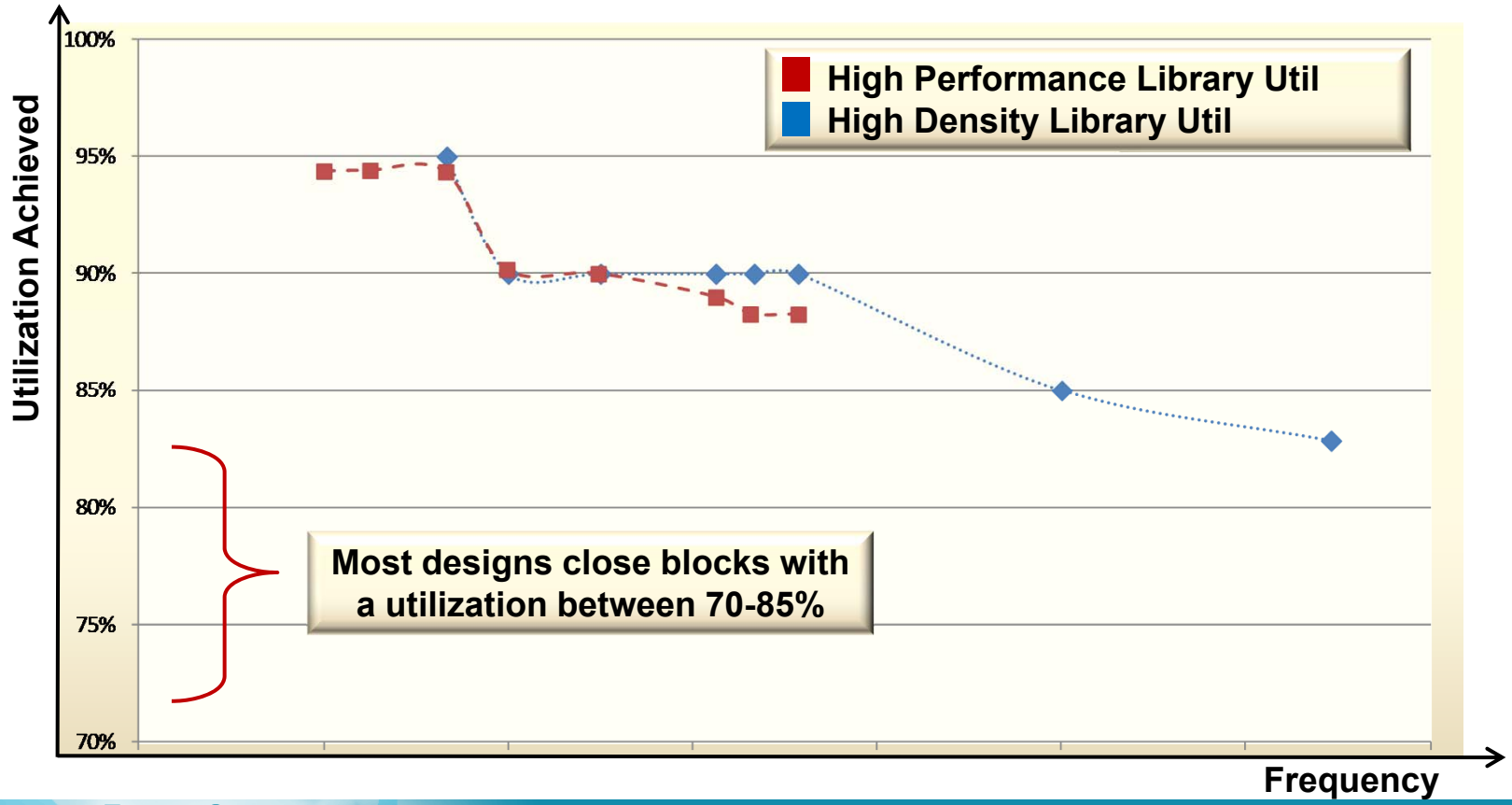
- Architecture affects routing
  - Different architectures may optimize for different metal pitch and VIA connections
  - Different vendor tech files may include different VIAs
  - Use vendor supplied tech files!



Pitfall #5: Optimize Utilization

Optimize utilization for the architecture and your design

- Architecture and power grid can affect utilization
  - Local routing congestion can limit utilization
  - Dial up utilization during benchmark



GETTING RESULTS FOR YOUR DESIGN

Determining the Fastest Library

- Run synthesis and place and route (SP&R) on the RTL at smaller and smaller cycle times
  - Use a pin-and-aspect-ratio floorplan – sets relative pin locations and aspect ratio
  - Include power grid: lower grid may differ across libraries
  - A rough initial cell utilization of 75% or higher is suggested for benchmarking
    - After CTS, routing, etc. this will be higher

- When the worst negative slack (WNS) starts increasing
  - You have found the theoretical fastest speed one can implement that library/design combination
  - **The library that produces the fastest design is the fastest library: the winner**
  - The area/routability/power consumption is ***not*** being measured here

Remember…

#1 The x and y of the floorplan and relative pin positions of the slowest library
#2 The maximum frequency of the slowest library

Determining the Lowest Area* Library

*Area is the area of the final design size,
not the area of all the cells within the floorplan

Remember…

#1 The x and y of the floorplan and relative pin positions of the slowest library
#2 The maximum frequency of the slowest library

- The question now : For a given speed & lowest area, which library will still route?
- Method – take the floorplan **#1** from previous slide
  - The power routing should ***not*** be retained, only the x-y size and relative pin position
  - Power routes should be implemented per library as before
- SP&R both libraries at the same target frequency in this floorplan: Frequency **#2** from previous slide
- Reduce area 5% at a time until libraries fail to route, or the achieved frequency starts to fall: either is considered the end point
- **The library that routes in the smallest area at the target frequency wins**

Determining the Lowest Power Library

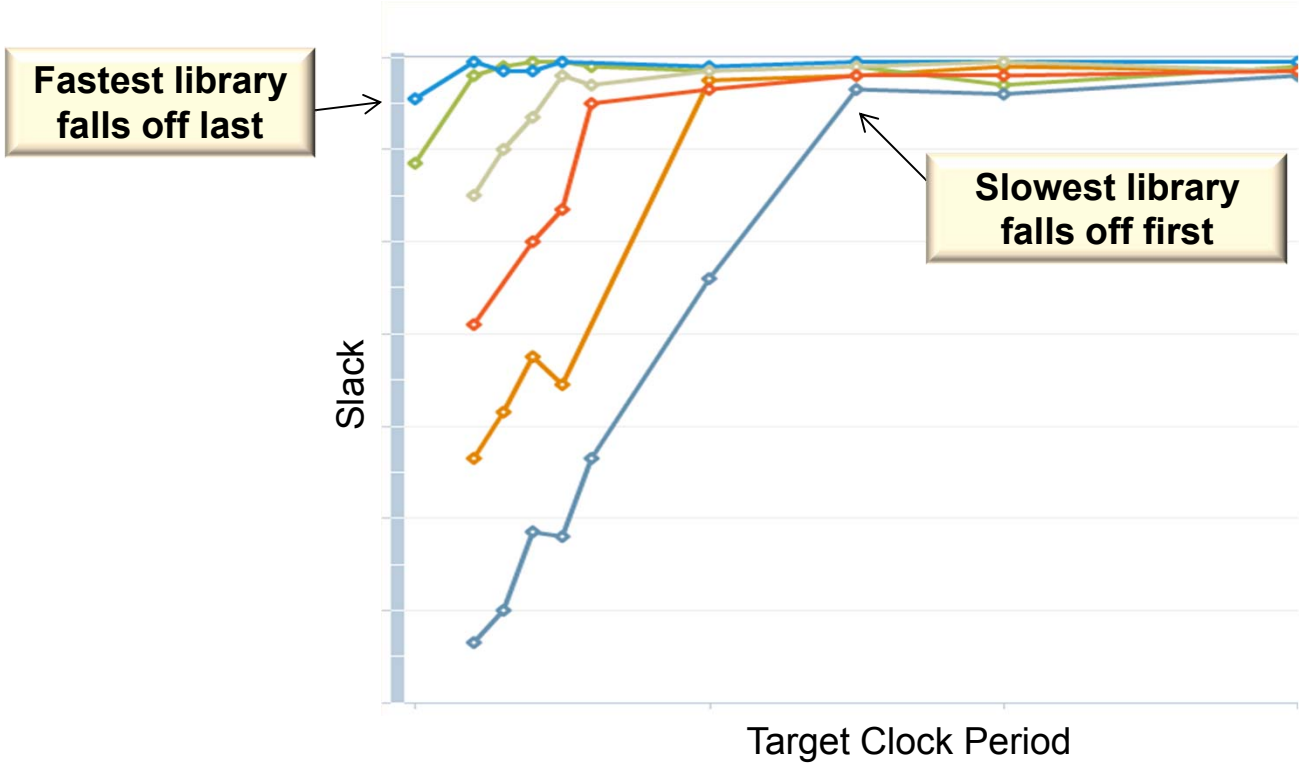
- Method
  - Take the final design from the previous step for each library
  - Take the last run before the end point
- Measure the dynamic and static powers for each of the 2 designs
- **The winner is the one with the lowest power**

ARM Internal Benchmarks

- Target a large design without large run times
- Understand critical path and logic levels
- Know constraints
  - max_transition
  - Utilization target
- Perform full flow
  - Synthesis, P&R with a power grid, CTS, hold fixing
- Use current EDA tools for library

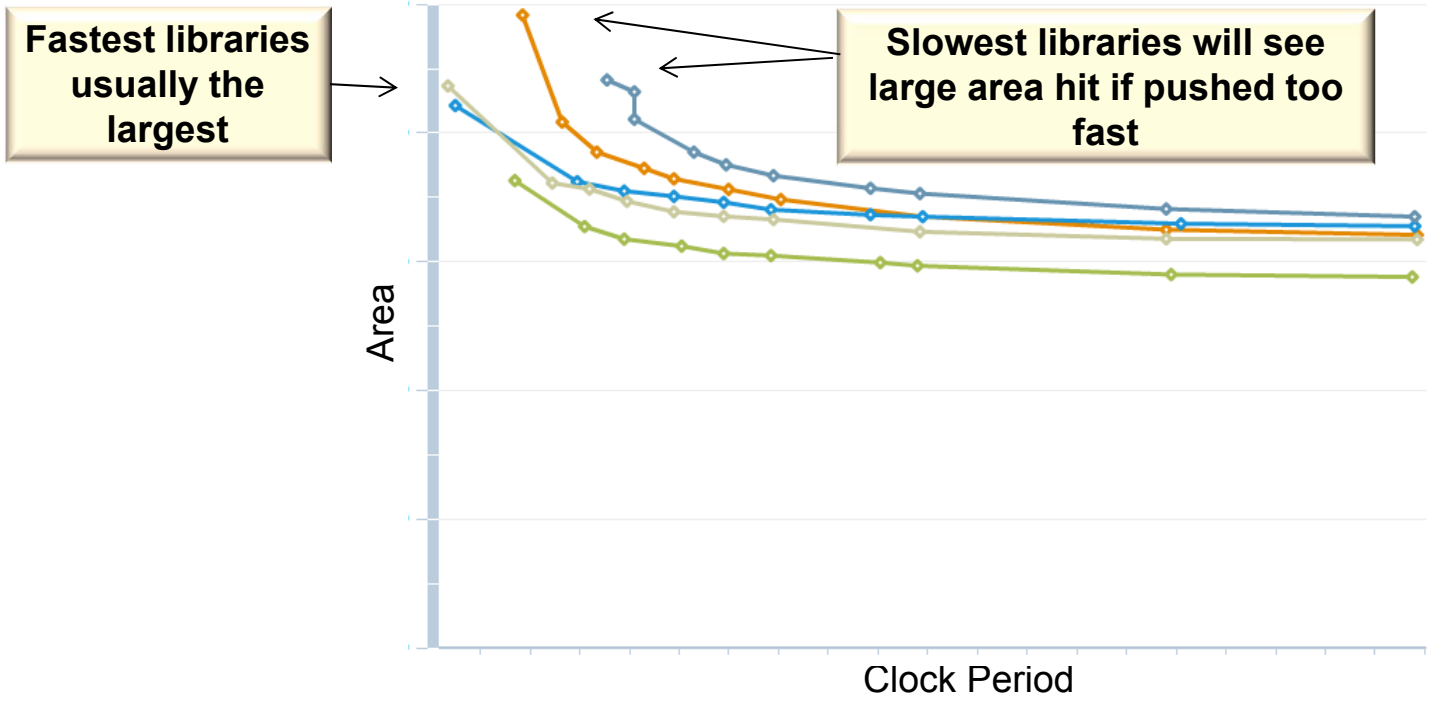
Slack vs Target Clock Period

- Sample libraries
- Multiple clock constraints
- Watch where slack goes from zero to negative



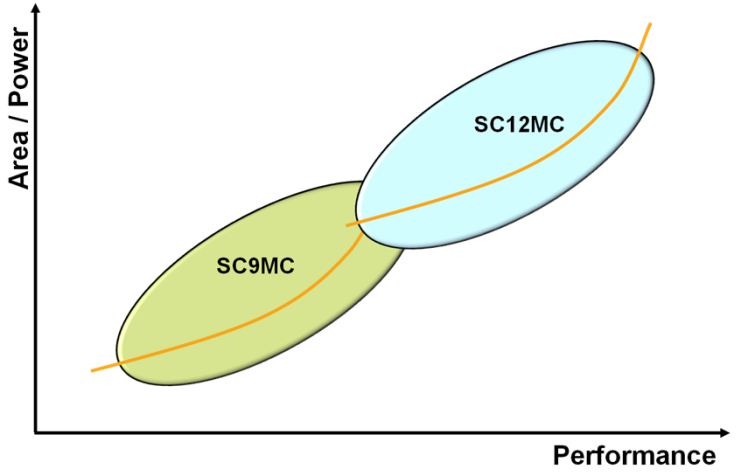
Slack vs Target Clock Period

- Sample libraries
- Multiple clock constraints
- Watch where slack goes from zero to negative
- As slack goes negative, area increases



Benchmarking Standard Cells

- Benchmarking is an important component of library selection
- Proper evaluation yields an appropriate selection of standard cell libraries
  - Optimize power, performance and area for your design
  - Compare “like” to “like”
- Avoid the common pitfalls
  1. Set proper constraints
  2. Account for parasitics
  3. Align power network to the architecture
  4. Use the appropriate tech file for routing
  5. Optimize utilization



5 Common Pitfalls

1. Set proper constraints
   - Select target PPA priority and conduct benchmark
   - Optimize cell set with “dont_use” lists
   - Edge rates & wire lengths
2. Account for parasitics
   - Use the same parasitic tables for all vendors
3. Align power network to the architecture
   - Watch for blocked resources
4. Use the appropriate tech file for routing
   - VIA selection can be important
   - Metal pitch can change with architecture
5. Optimize utilization
   - Different libraries will have different utilization limits