

# Revamping SoC Design with RTL Architect and NoC Innovations

Frank Schirrmeister, VP Solutions & Business Development  
Arteris

# Arteris – A Leading SoC System IP Company

ARTERIS IP



Global customer base deploying Arteris interconnect IP and SoC Integration Software

- Silicon-proven IP used in ~3.5 billion+ SoCs shipped to date
- 200+ customers and 725+ SoC design starts to date
- 70-80% market share of automotive ADAS SoC market<sup>1</sup>
- Strong presence in Artificial Intelligence/Machine Learning (AI/ML) system IP
- Broad interoperability - any processor, any IP, any EDA, any foundry
- Innovative technology coupled with expert support results in a 90+% customer retention rate
- 71 patents and 77 patent applications (As of July 31, 2023)
- ISO 9001:2015 Quality Management System (QMS) Certified

## Diversified Customer Base

Subset of Publicly Disclosed Customers



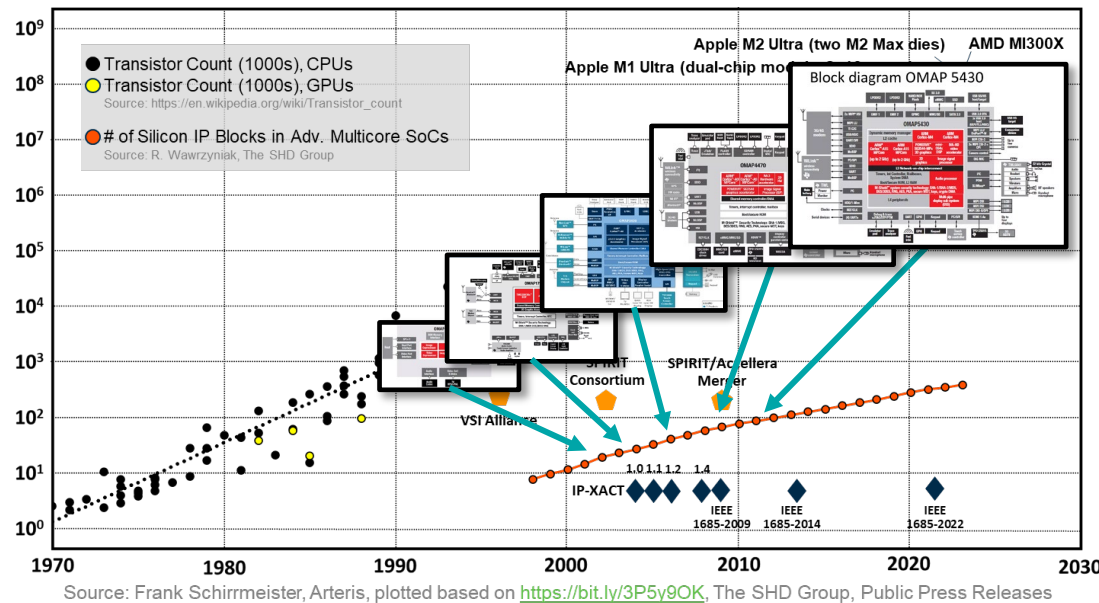
Source: Arteris Q4'23 Earnings Call, <sup>1</sup> Management estimates

# Challenges

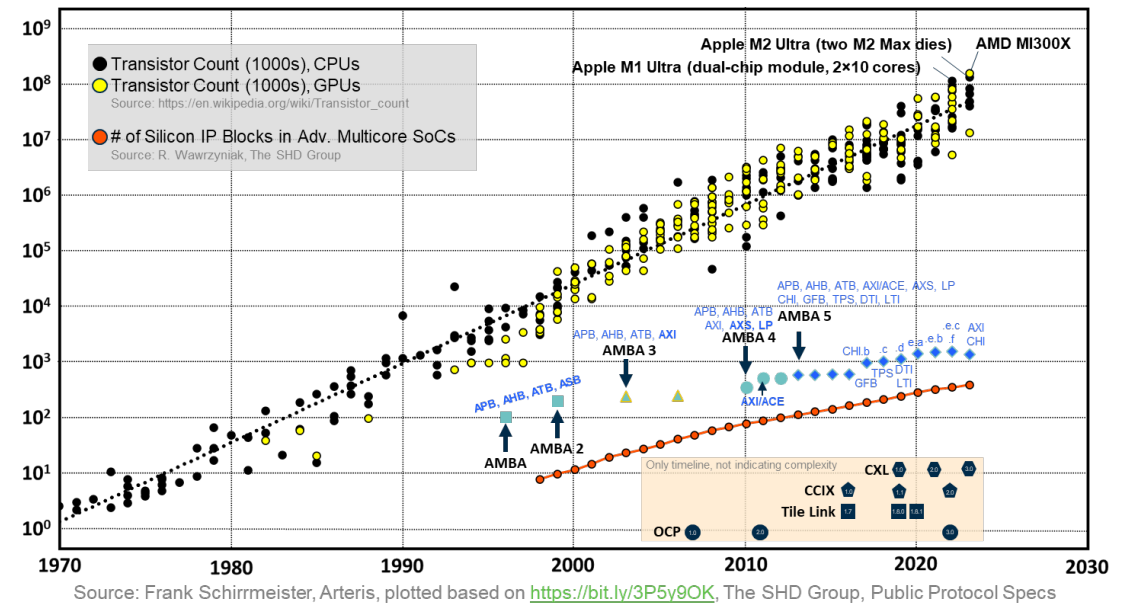
## Challenges with Traditional Flow

# Semiconductor Complexity, SoC Integration and Networks-on-Chips(NoCs)

## Growing SoC Integration Complexity



## Growing Network-on-Chip Complexity

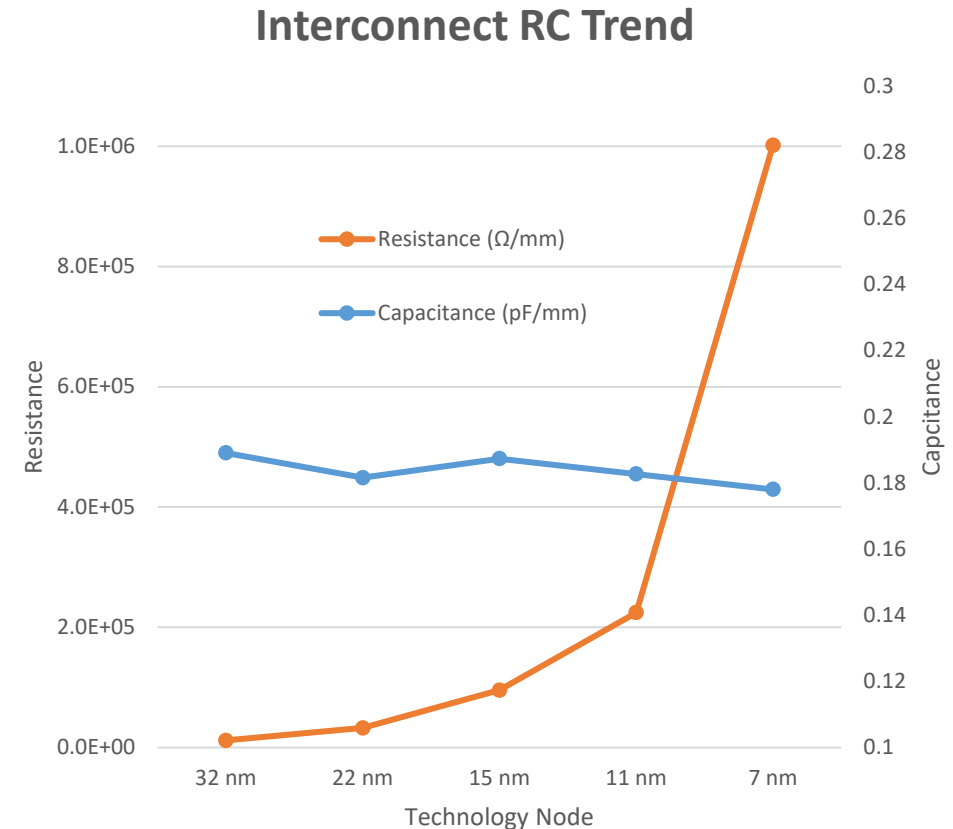


- # of IP Blocks in SoC has grown from 10s to 100s
- Disaggregation offers design version variety
- Standards like IP-XACT need to extend too

- Complexity of NoC protocols have grown 10x (# of pages)
- Variety of NoC protocols has grown
- NoCs evolve into Super-NoCs when split across chiplets

# Interconnect Timing Design Challenges

- No standard methodology for **timing closure** for on-chip IP communications
- Process node advances **add to RC delays for long, cross-chip distances** traveled
- Interconnects that connect different IP blocks span **long distances** and hence **suffer from RC delays**



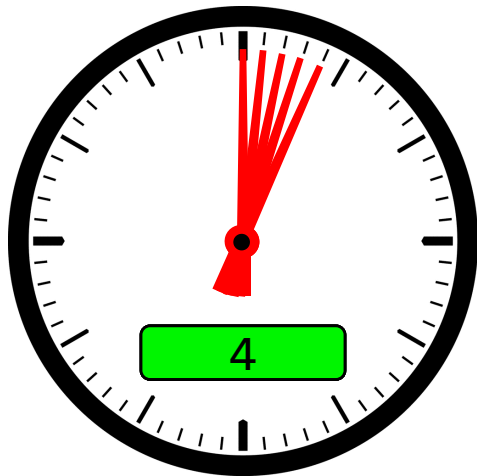
**Interface-Layer Effects and Grain Scattering Impact Resistance Scaling**

Source: Serkan Kinca, et al., "RC Performance Evaluation of Interconnect Architecture Options Beyond the 10-nm Logic Node," IEEE (2014)

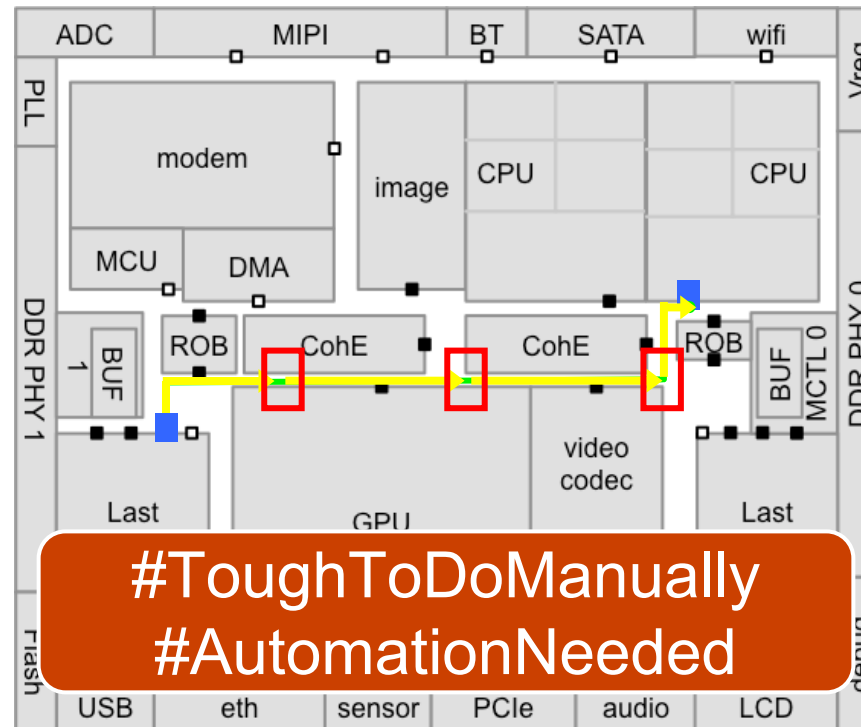
# SoC Timing Issues

Can't Cross Advanced Node SoCs in One Clock Cycle

Physical distance impacts the number of pipeline stages



Clock Cycles



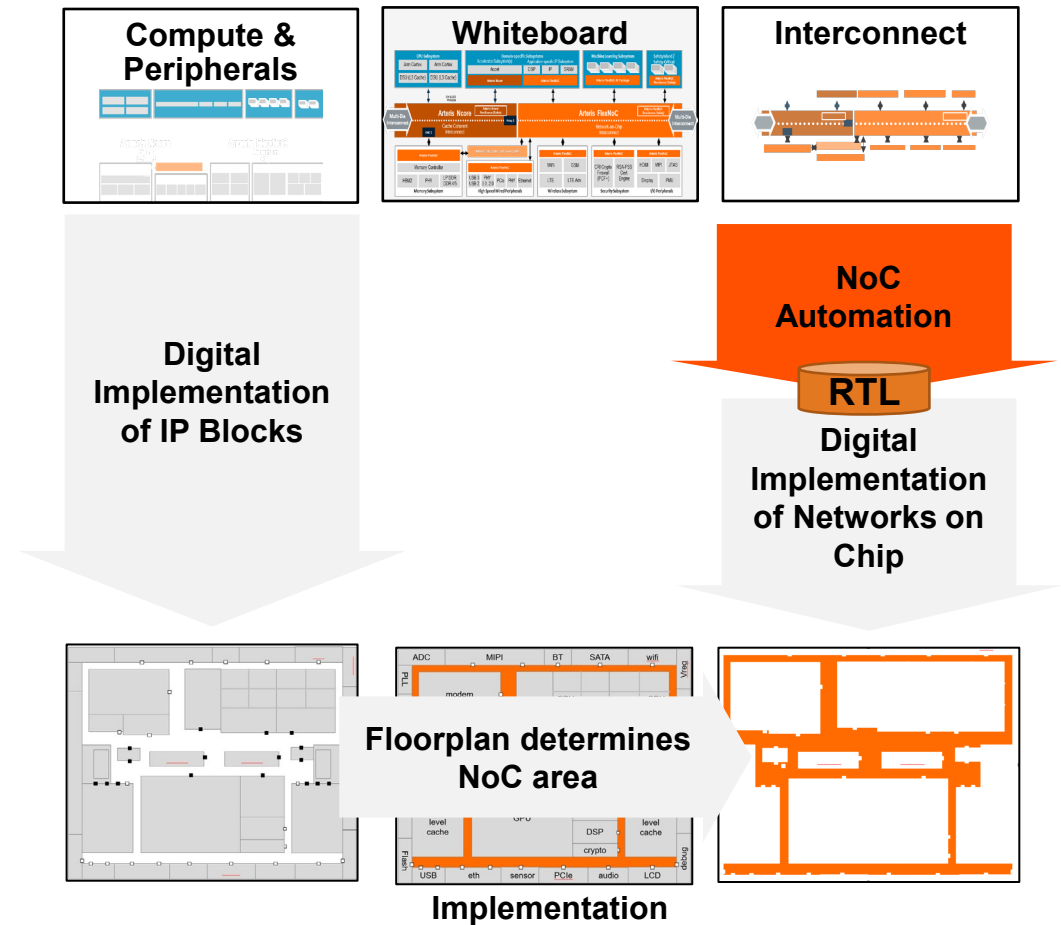
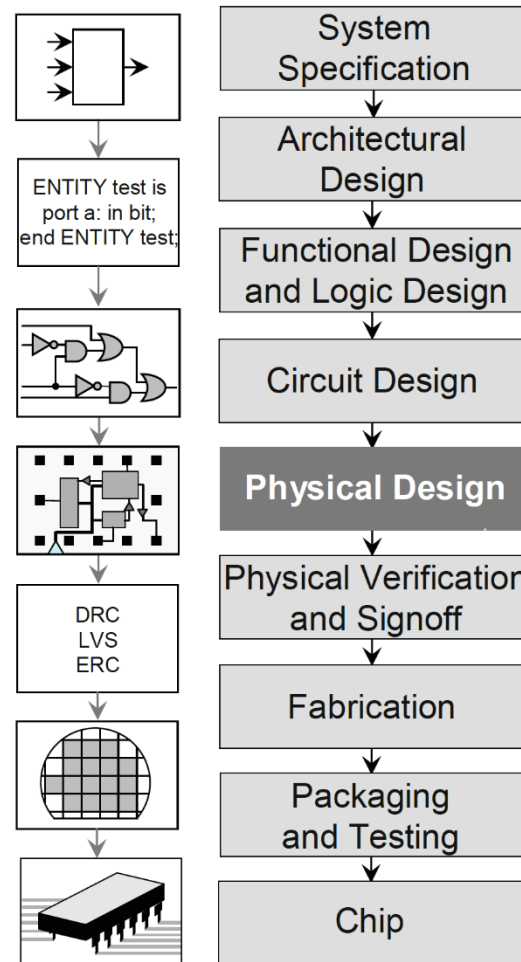
Transport delay =  $F$  (foundry, routing stack, type of driving cell, process voltage, temperature, ...)



# The EDA Flow & NoC Design

Customer uses Arteris tools to design the Network on Chip (NoC)

Customer uses Synopsys tools to implement from RTL to GDSII

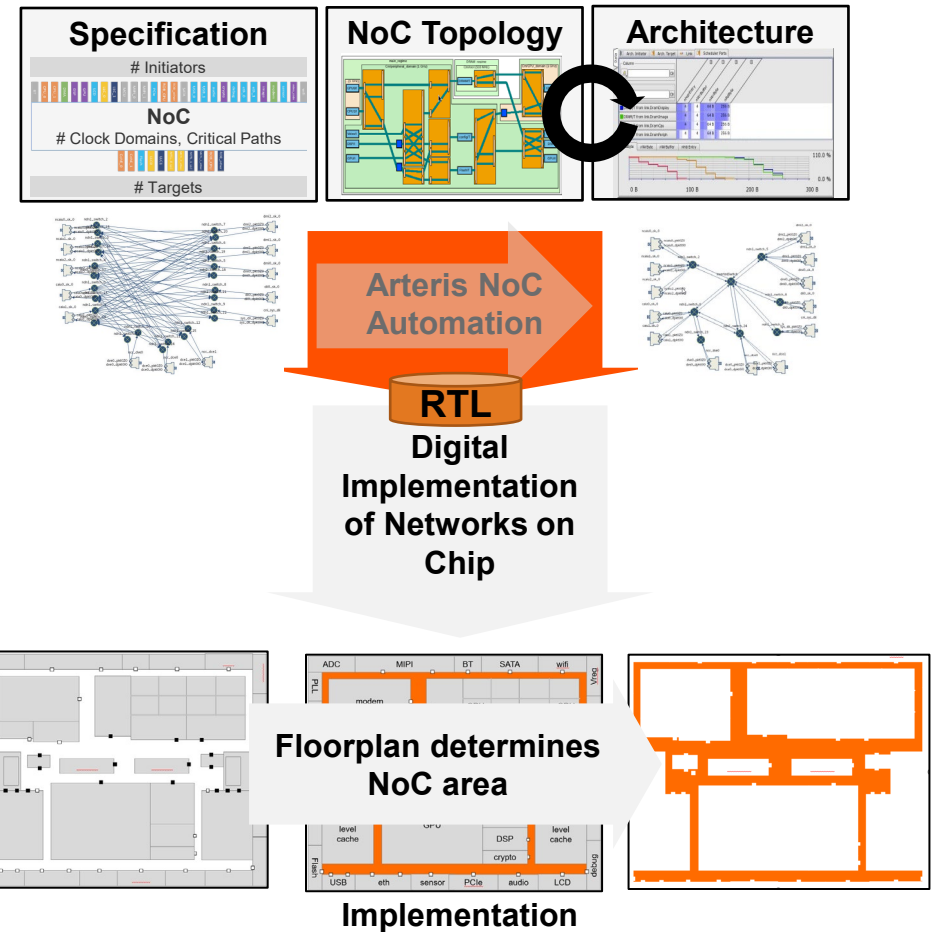
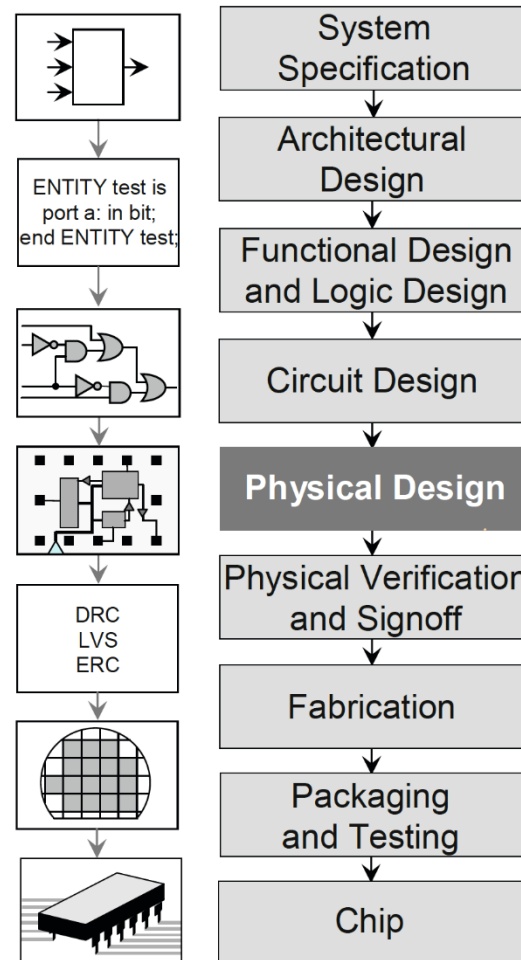


# The EDA Flow & NoC Design

SNPS Platform Architect

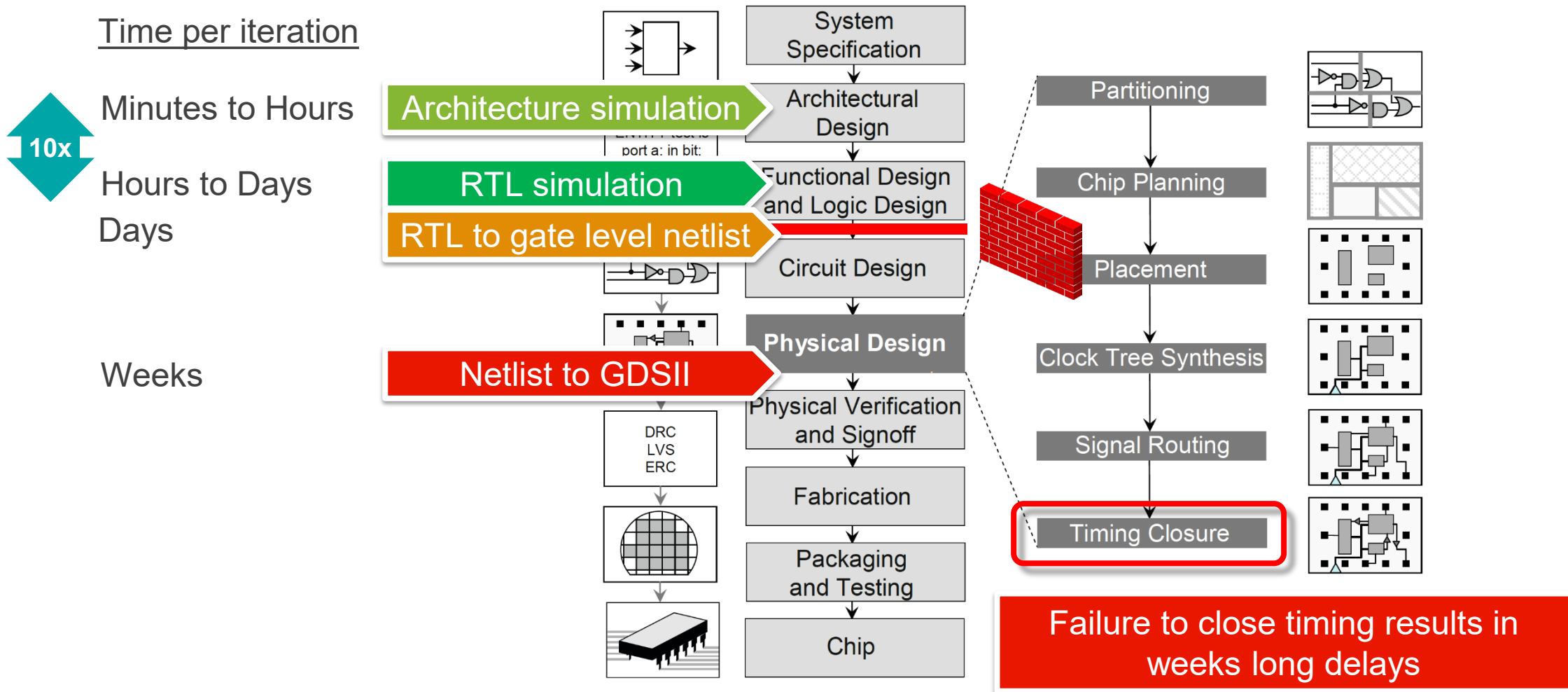
Customer uses Arteris tools to design the Network on Chip (NoC)

Customer uses Synopsys tools to implement from RTL to GDSII





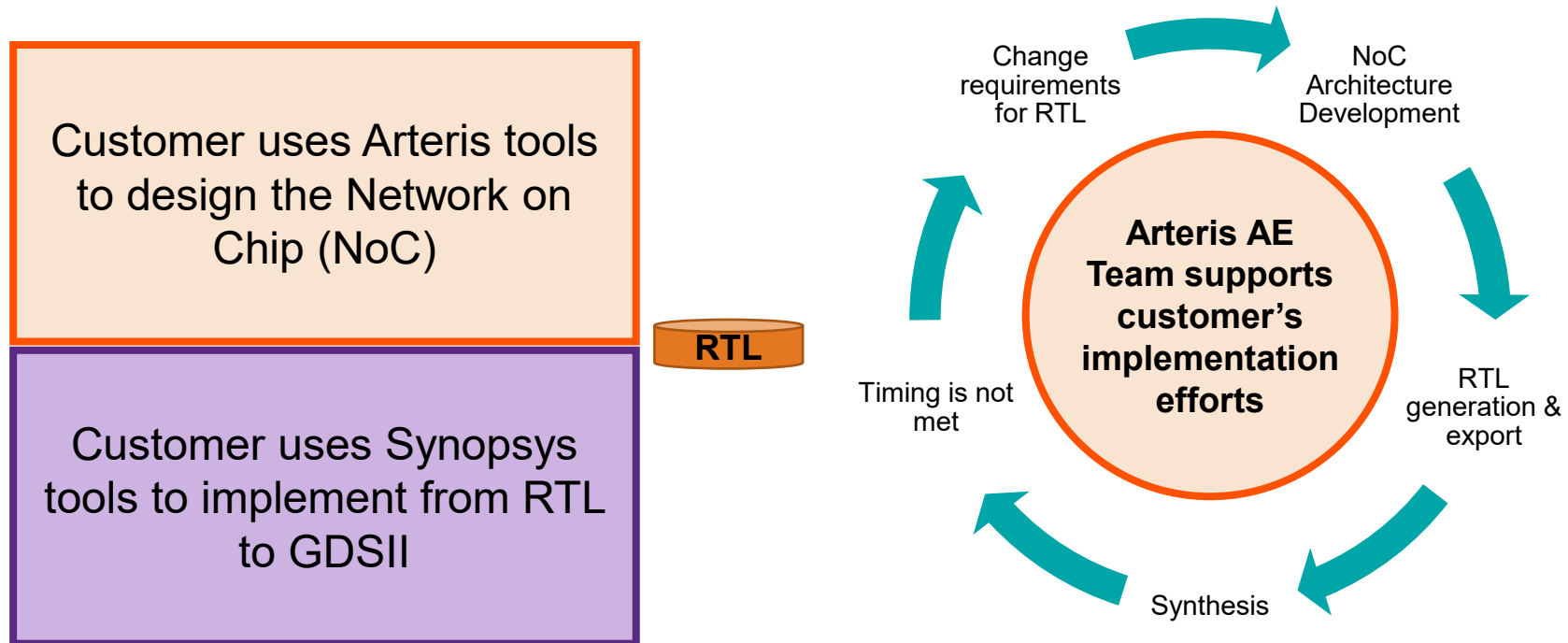
# The EDA Flow & NoC Design



EDA flow diagram source: Andrew B. Kahng, et al., "VLSI Physical Design: From Graph Partitioning to Timing Closure," Springer (2011)

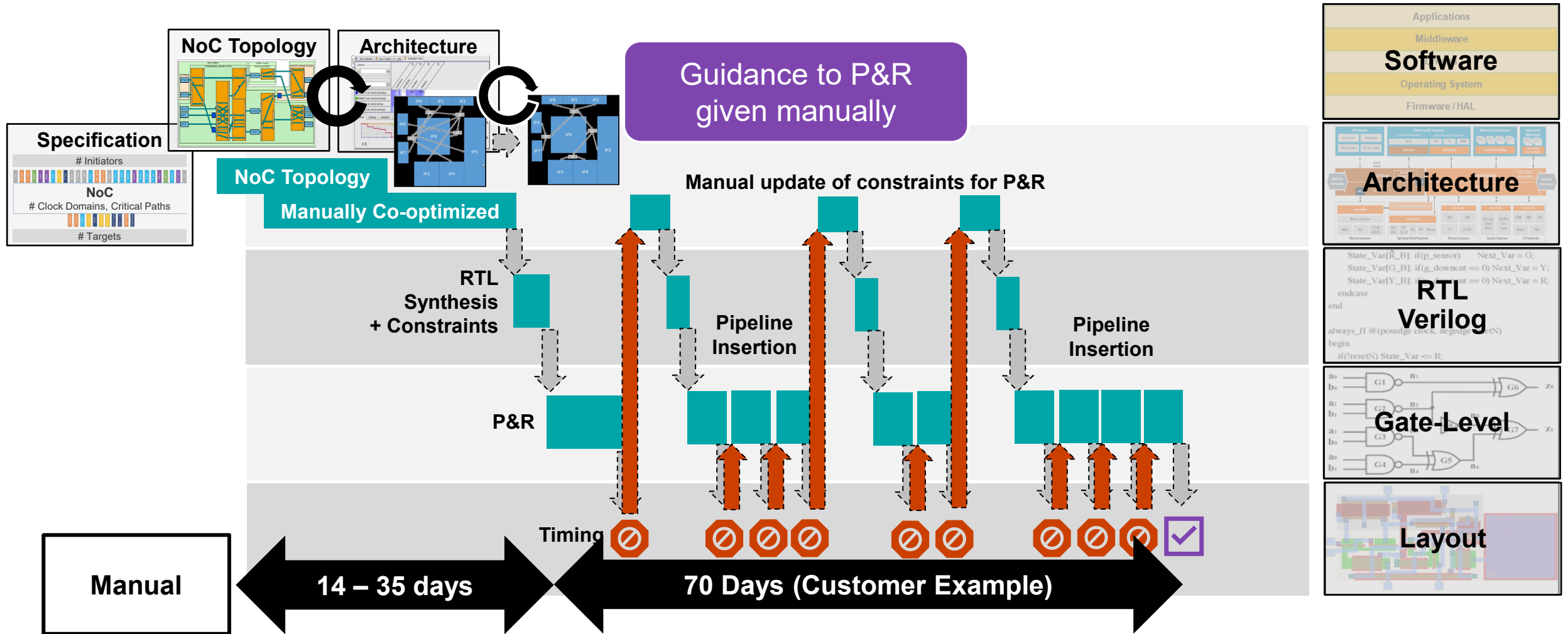
# The Customer PPA Struggle

NoC RTL is different for every refinement



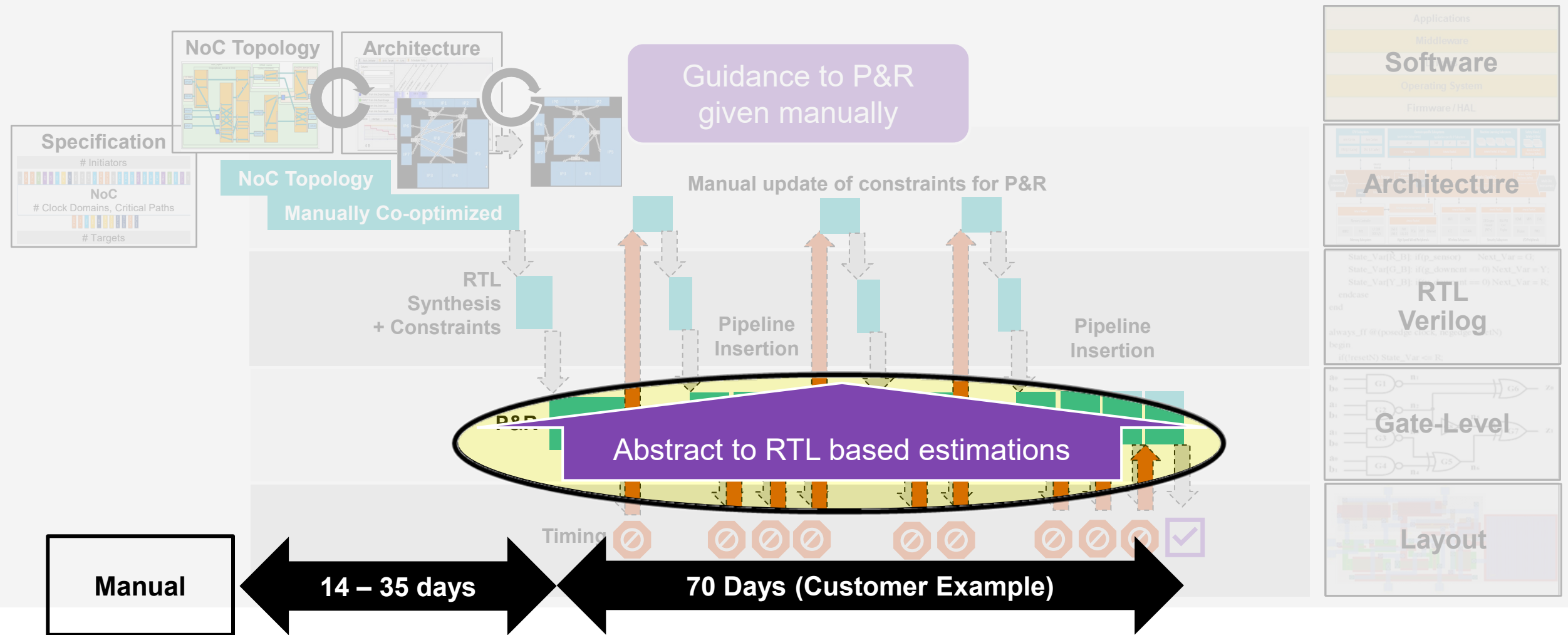
# A Customer Example

## Manual Flow With Some Layout Awareness & Guidance



# Automation Opportunities

Abstract estimations of interconnect delay after layout

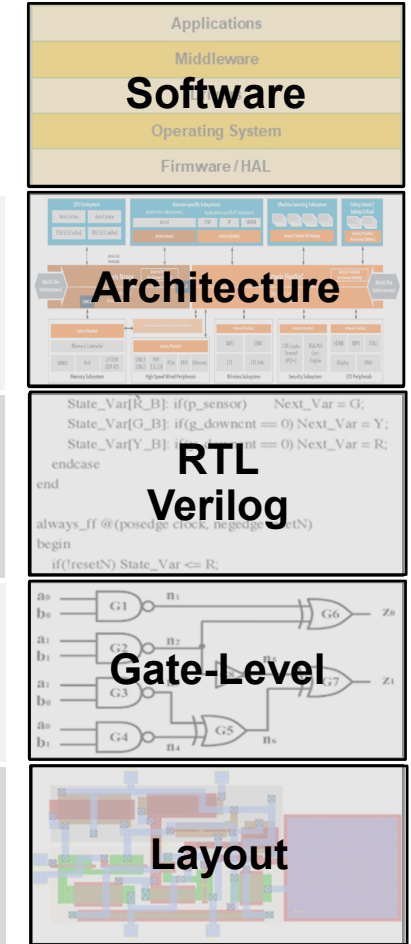
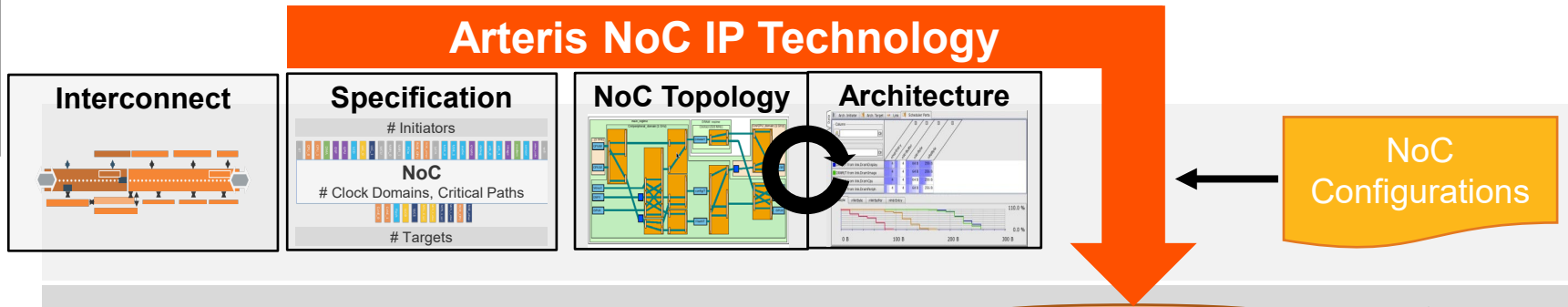
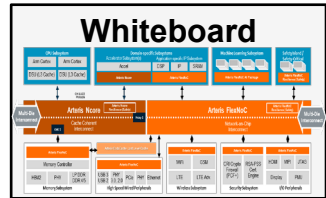


# Physical Exploration with RTL Architect

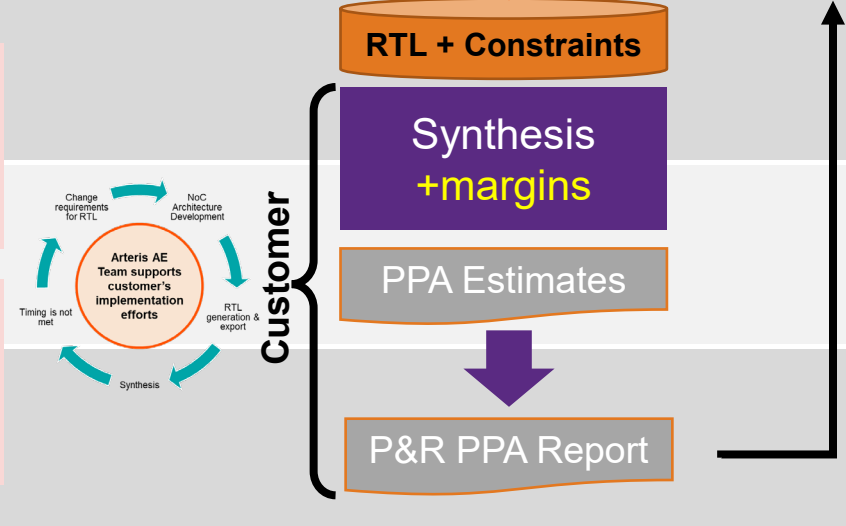
## Limitations of the Existing Flow

# Traditional Flow

Slow to converge with many customer iterations



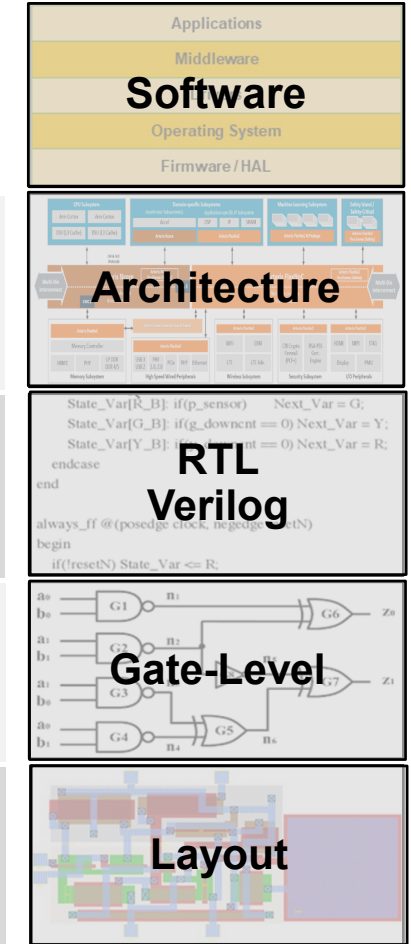
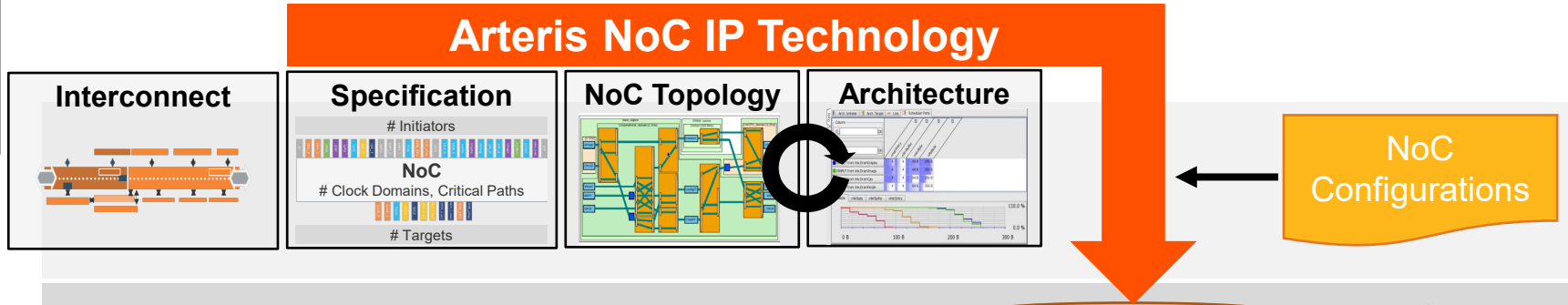
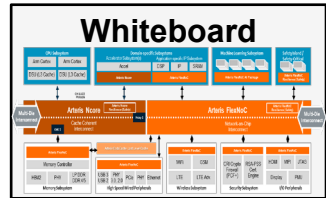
- Problem: The **margins don't predict** all the **physical effects** during implementation
- Customers iterate with Arteris to improve RTL
- Customers provide **gate-centric reports** that don't pinpoint the problem with the RTL.
- It takes **multiple iterations** to converge





# Traditional Flow

Slow to converge with many customer iterations

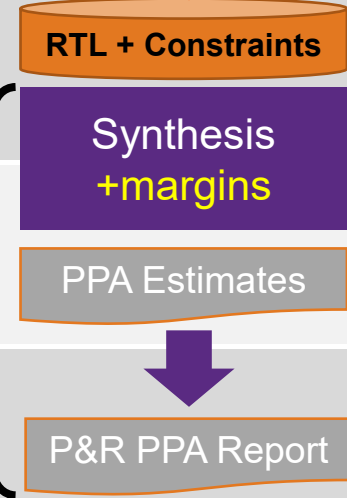


Timing, logic levels, flop count/area numbers determine whether a design can be implemented

DC/DC Topo synthesis to confirm timing with Arteris configurable NoC RTL IP

Design sizes and DC runtimes increase!

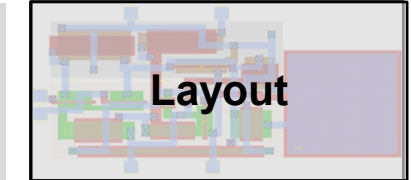
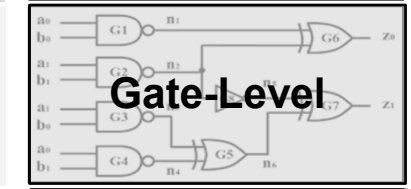
Some customers experience timing closure issues even later, using Fusion Compiler



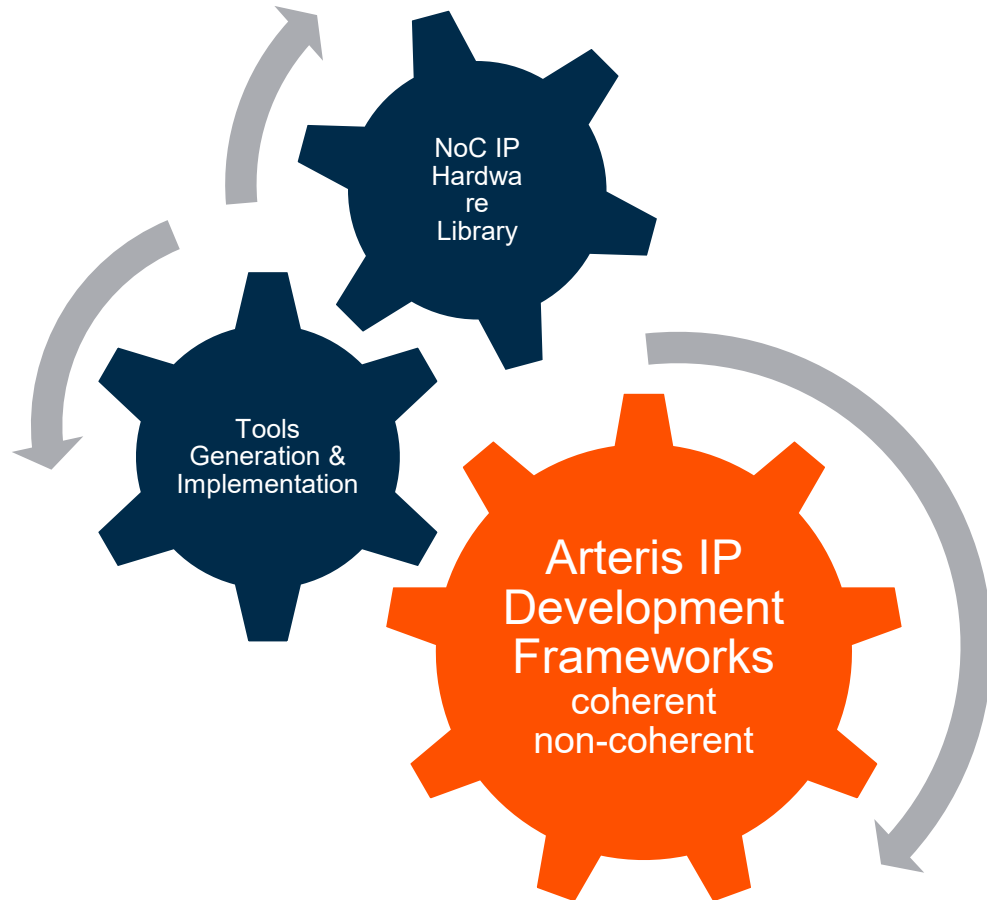
```

State_Var[R_B]: if(p_sensor) Next_Var = G;
State_Var[G_B]: if(g_downcnt == 0) Next_Var = Y;
State_Var[Y_B]: if(y_downcnt == 0) Next_Var = R;
endcase
end
always_ff @(posedge clock, negedge resetN)
begin
if(!resetN) State_Var <= R;

```



# NoC IP Development Frameworks



## NoC Interconnect IP Hardware Library

Protocol converters, switches, rate adaptors, etc

## Tools Generation & Implementation

Generation of Implementation RTL, Collaterals  
Links to EDA Implementation

## Tools Architecture & Validation

Testbench creation  
SystemC models for performance analysis

# Modular Tool Library



Category	Library units
<b>Network Interface Units</b>	<b>ACE-Lite, AXI, AHB, APB, PIF, OCP, others</b>
Transport	mux, demux, serializer, buffer
Domain adapter	clock, power, voltage
Timing	pipeline stage
QoS	bandwidth limiter/regulator
Security	user defined firewalls
Probes	error, statistics, trace
Interchip link	PSI
Resilience/Safety	timeout, safety controller, checkers

# Tool Flow


**Inputs**  
Specification  
 Socket specifications  
Floorplan outline  
 LEF/DEF  
 Visio  
Perf Analysis  
 Traffic Scenarios  
 P&R  
 Timing Reports

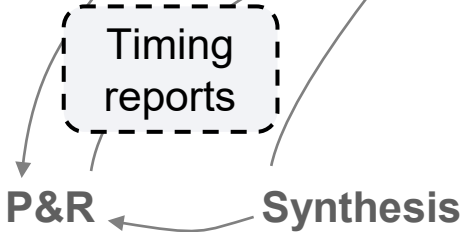
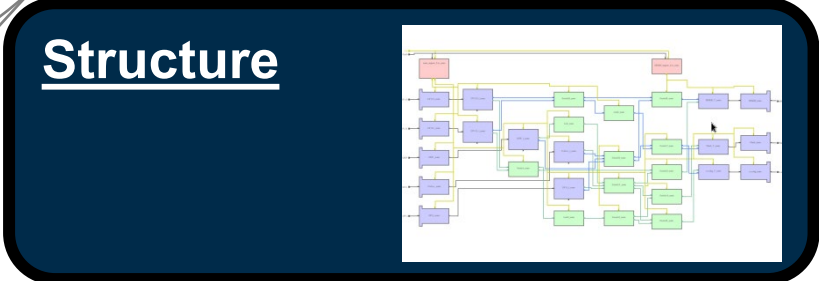
**Specification**

Column	Row	Modifications	Defer	clock	bus type	IP type	width	width/height	align/locate	compression	grouping
Initiator											
B CPU0	mainRegime/Cm/root	AXI	Initiator	...	[x0]	...	...	...	Separated		
B CPU1	mainRegime/Cm/root	AXI	Initiator	...	[x0]	...	...	...	Separated		
B DSP	mainRegime/Cm/root	AXI	Initiator	...	[x0]	...	...	...	Separated		
B GPU	mainRegime/Cm/root	AXI	Initiator	...	[x0]	...	...	...	Separated		
B Video	mainRegime/Cm/root	AXI	Initiator	...	[x0]	...	...	...	Separated		
Target											
B DRAM	mainRegime/Cm/root	AXI	Target	...	[x0]	...	...	...	Separated		
B Flash	mainRegime/Cm/root	AXI	Target	...	[x0]	...	...	...	Separated		
B config	mainRegime/Cm/root	AXI	Target	...	[x0]	...	...	...	Separated		

**Outputs**  
Structure  
 RTL  
 SystemC  
 IP-XACT  
 Scripts / constraints  
 Gate Estimates  
 Verification testbenches

**Physical Awareness & Architecture**  
 auto pipe, placement  
  
 switch topology  
 QoS, buffers  
 serialization  


**Performance Analysis**  
 use case based performance  




# Architecture Refinements

## Physical and Performance Co-design

Element specDemo.archDemo

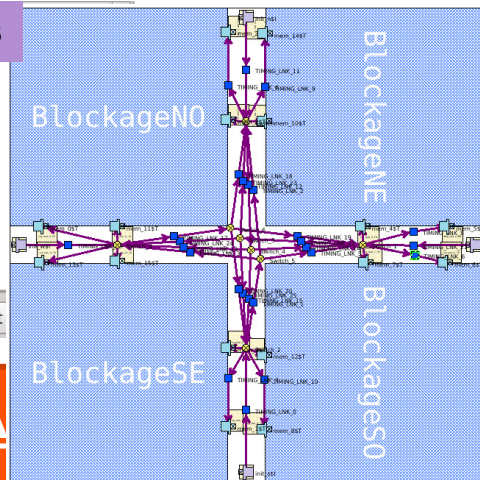
**40 meters**

Info :

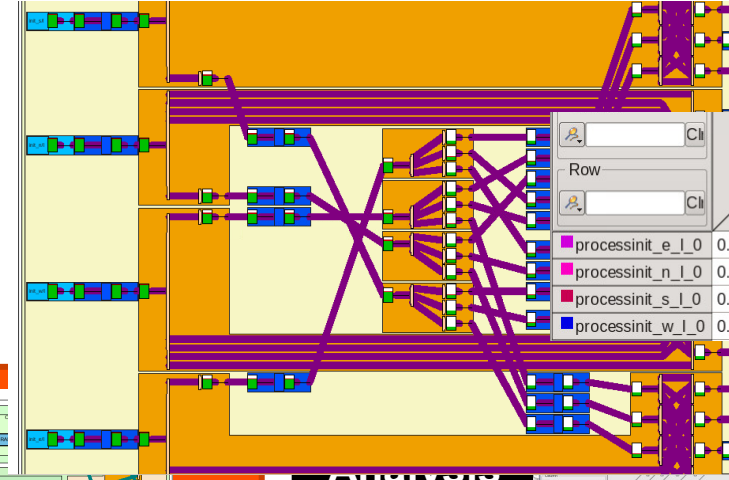
- Wire Length 40704.00000 mm
- Length 106.00000 mm
- Number of Sw... 16
- Number of Links 52
- Number of Clo... 0
- Number of Pa... 0
- Latency 6.81250 ns
- Maximum Lat... 9.00000 ns

Network : datapa...  
Wire Length 13977.60000 mm

Refresh Topology Infos Export



Topology Full



Row	Required	Efficiency	Satisfaction	Throughput	State
processinit_e_i_0	0.0 %	87.4 %	PASS	28.2 GB/s	Green
processinit_n_i_0	0.0 %	89.8 %	PASS	29.0 GB/s	Green
processinit_s_i_0	0.0 %	89.5 %	PASS	28.2 GB/s	Green
processinit_w_i_0	0.0 %	89.9 %	PASS	28.7 GB/s	Green

**Physical Architecture**  
auto pipe,  
placement

Element specDemo.archLow...

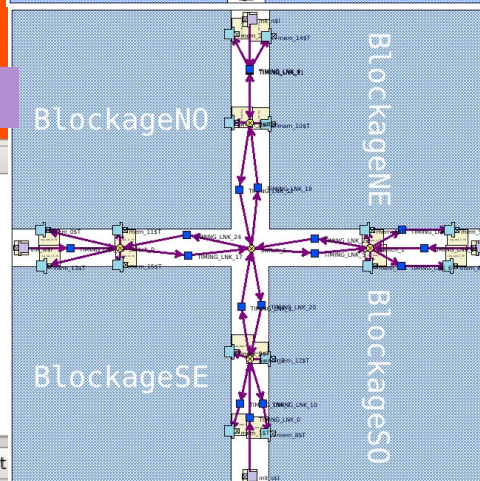
**20 meters**

Info :

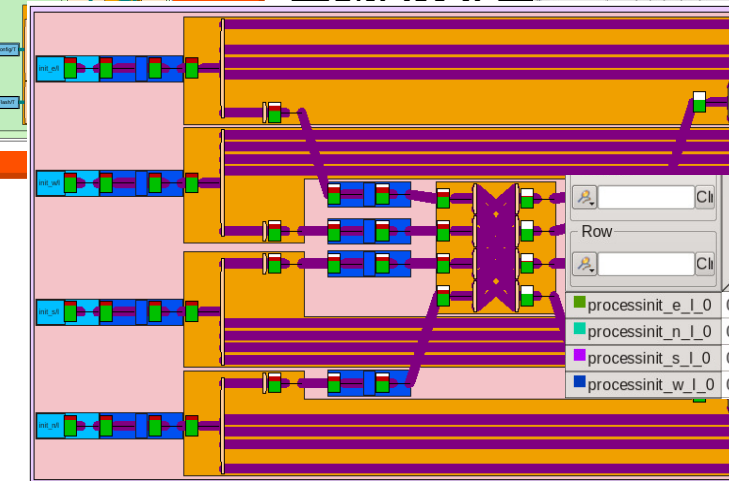
- Wire Length 19837.44000 mm
- Length 51.66000 mm
- Number of Sw... 10
- Number of Links 36
- Number of Clo... 0
- Number of Pa... 0
- Latency 7.25000 ns
- Maximum Lat... 10.00000 ns

Network : datapa...  
Wire Length 10024.32000 mm

Refresh Topology Infos Export



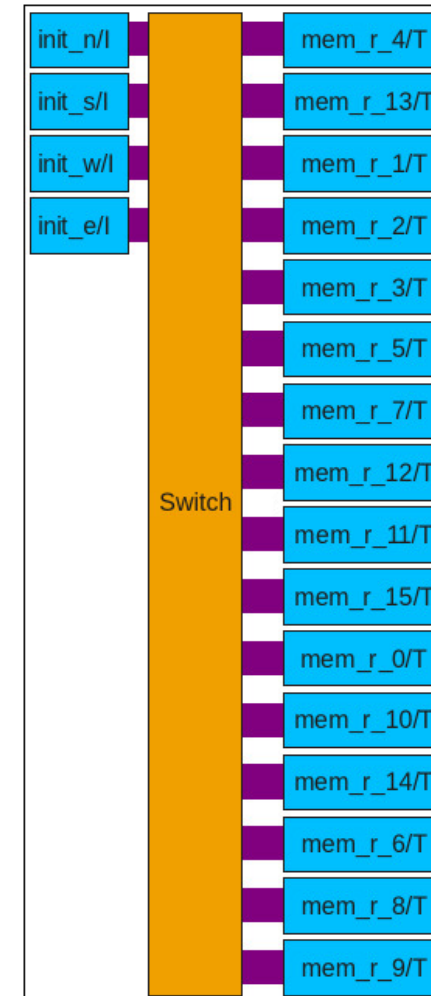
Topology Cost



Row	Required	Efficiency	Satisfaction	Throughput	State
processinit_e_i_0	0.0 %	72.5 %	PASS	23.4 GB/s	Green
processinit_n_i_0	0.0 %	76.3 %	PASS	24.7 GB/s	Green
processinit_s_i_0	0.0 %	79.2 %	PASS	25.1 GB/s	Green
processinit_w_i_0	0.0 %	76.0 %	PASS	24.1 GB/s	Green

# Signal Processing Design Example

- Use case: Signal processing as part of a larger Communications SoC.
- The design includes x4 DMAs that need to access 'scratchpad' SRAMs.
- A crossbar approach is a good starting point.





# Specify Network Interface Units, Connectivity and Address Maps



specDemo

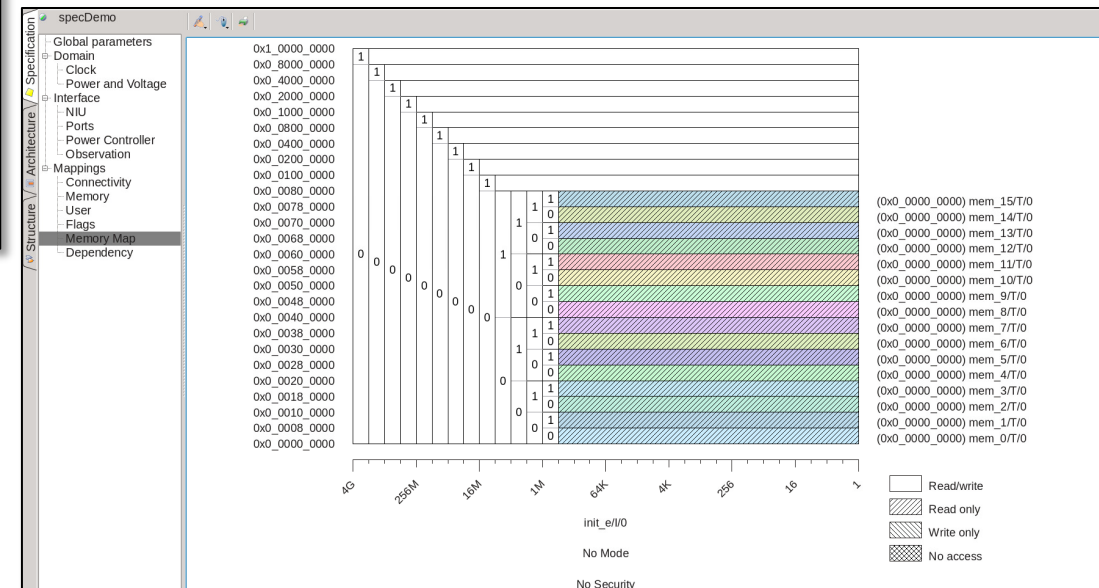
Global parameters  
Domain  
Clock  
Power and Voltage  
Interface  
NIU  
Ports  
Power Controller  
Observation  
Mappings  
Connectivity  
Memory  
User  
Flags  
Memory Map  
Dependency

Column	Row	clock	protocol	reference	niUType	power	watchDogs	additionalClocks	conversion	clockGating	synthesisInfo	responseForWarding	comment
Initiator													
init_e	sram_tile_noc_data_clk_regime/Cm/root	REFERENCE	pdAXI	Initiator	...	[x0]	...	...	Separated	Partial			
init_n	sram_tile_noc_data_clk_regime/Cm/root	REFERENCE	pdAXI	Initiator	...	[x0]	...	...	Separated	Partial			
init_s	sram_tile_noc_data_clk_regime/Cm/root	REFERENCE	pdAXI	Initiator	...	[x0]	...	...	Separated	Partial			
init_w	sram_tile_noc_data_clk_regime/Cm/root	REFERENCE	pdAXI	Initiator	...	[x0]	...	...	Separated	Partial			
Target													
mem_0	sram_tile_noc_data_clk_regime/Cm/root	REFERENCE	pdAXI	Target	...	[x0]	...	...	Separated	Partial			
mem_1	sram_tile_noc_data_clk_regime/Cm/root	REFERENCE	pdAXI	Target	...	[x0]	...	...	Separated	Partial			
mem_2	sram_tile_noc_data_clk_regime/Cm/root	REFERENCE	pdAXI	Target	...	[x0]	...	...	Separated	Partial			
mem_3	sram_tile_noc_data_clk_regime/Cm/root	REFERENCE	pdAXI	Target	...	[x0]	...	...	Separated	Partial			
mem_4	sram_tile_noc_data_clk_regime/Cm/root	REFERENCE	pdAXI	Target	...	[x0]	...	...	Separated	Partial			
mem_5	sram_tile_noc_data_clk_regime/Cm/root	REFERENCE	pdAXI	Target	...	[x0]	...	...	Separated	Partial			
mem_6	sram_tile_noc_data_clk_regime/Cm/root	REFERENCE	pdAXI	Target	...	[x0]	...	...	Separated	Partial			
mem_7	sram_tile_noc_data_clk_regime/Cm/root	REFERENCE	pdAXI	Target	...	[x0]	...	...	Separated	Partial			
mem_8	sram_tile_noc_data_clk_regime/Cm/root	REFERENCE	pdAXI	Target	...	[x0]	...	...	Separated	Partial			
mem_9	sram_tile_noc_data_clk_regime/Cm/root	REFERENCE	pdAXI	Target	...	[x0]	...	...	Separated	Partial			
mem_10	sram_tile_noc_data_clk_regime/Cm/root	REFERENCE	pdAXI	Target	...	[x0]	...	...	Separated	Partial			
mem_11	sram_tile_noc_data_clk_regime/Cm/root	REFERENCE	pdAXI	Target	...	[x0]	...	...	Separated	Partial			
mem_12	sram_tile_noc_data_clk_regime/Cm/root	REFERENCE	pdAXI	Target	...	[x0]	...	...	Separated	Partial			
mem_13	sram_tile_noc_data_clk_regime/Cm/root	REFERENCE	pdAXI	Target	...	[x0]	...	...	Separated	Partial			
mem_14	sram_tile_noc_data_clk_regime/Cm/root	REFERENCE	pdAXI	Target	...	[x0]	...	...	Separated	Partial			
mem_15	sram_tile_noc_data_clk_regime/Cm/root	REFERENCE	pdAXI	Target	...	[x0]	...	...	Separated	Partial			

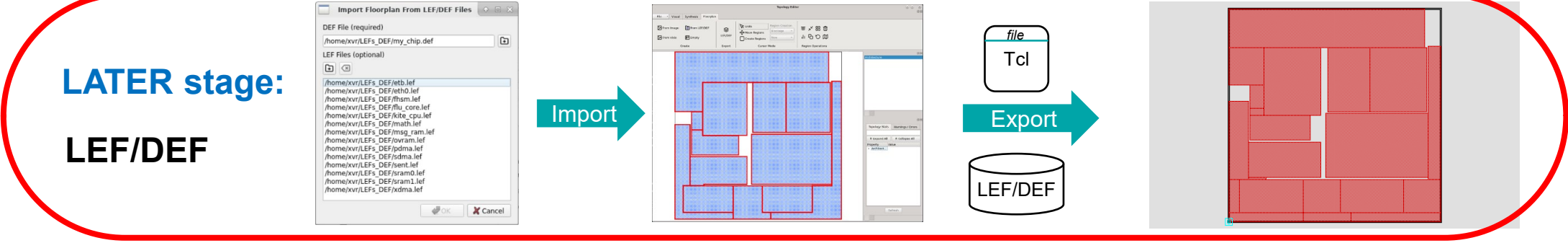
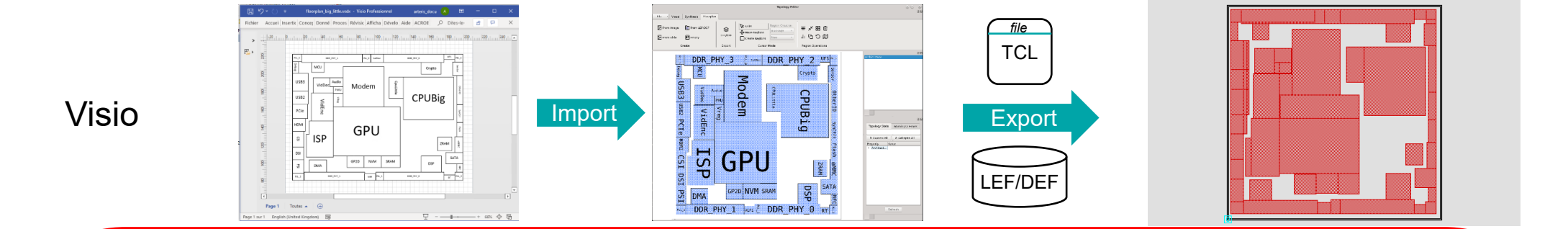
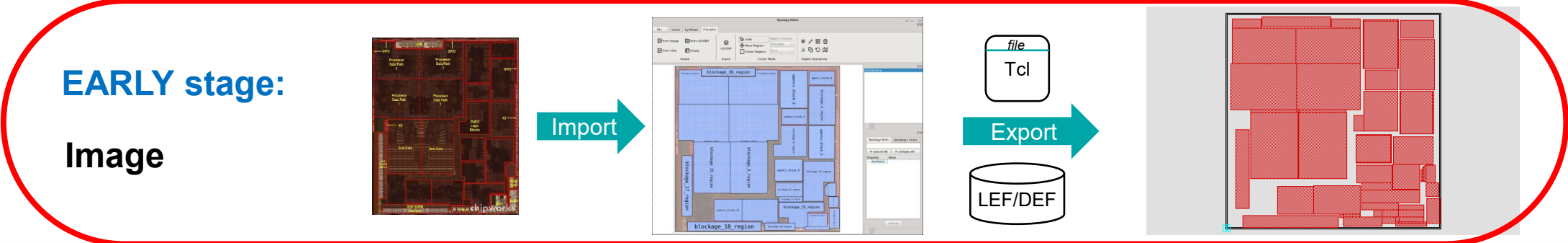
specDemo

Global parameters  
Domain  
Clock  
Power and Voltage  
Interface  
NIU  
Ports  
Power Controller  
Observation  
Mappings  
Connectivity  
Memory  
User  
Flags  
Memory Map  
Dependency

Column	Row	mem_0/TIO	mem_1/TIO	mem_2/TIO	mem_3/TIO	mem_4/TIO	mem_5/TIO	mem_6/TIO	mem_7/TIO	mem_8/TIO	mem_9/TIO	mem_10/TIO	mem_11/TIO	mem_12/TIO	mem_13/TIO	mem_14/TIO	mem_15/TIO
Initiator																	
init_e/I/O		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
init_n/I/O		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
init_s/I/O		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
init_w/I/O		✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓	✓
Target																	



# Consider Floorplan Early

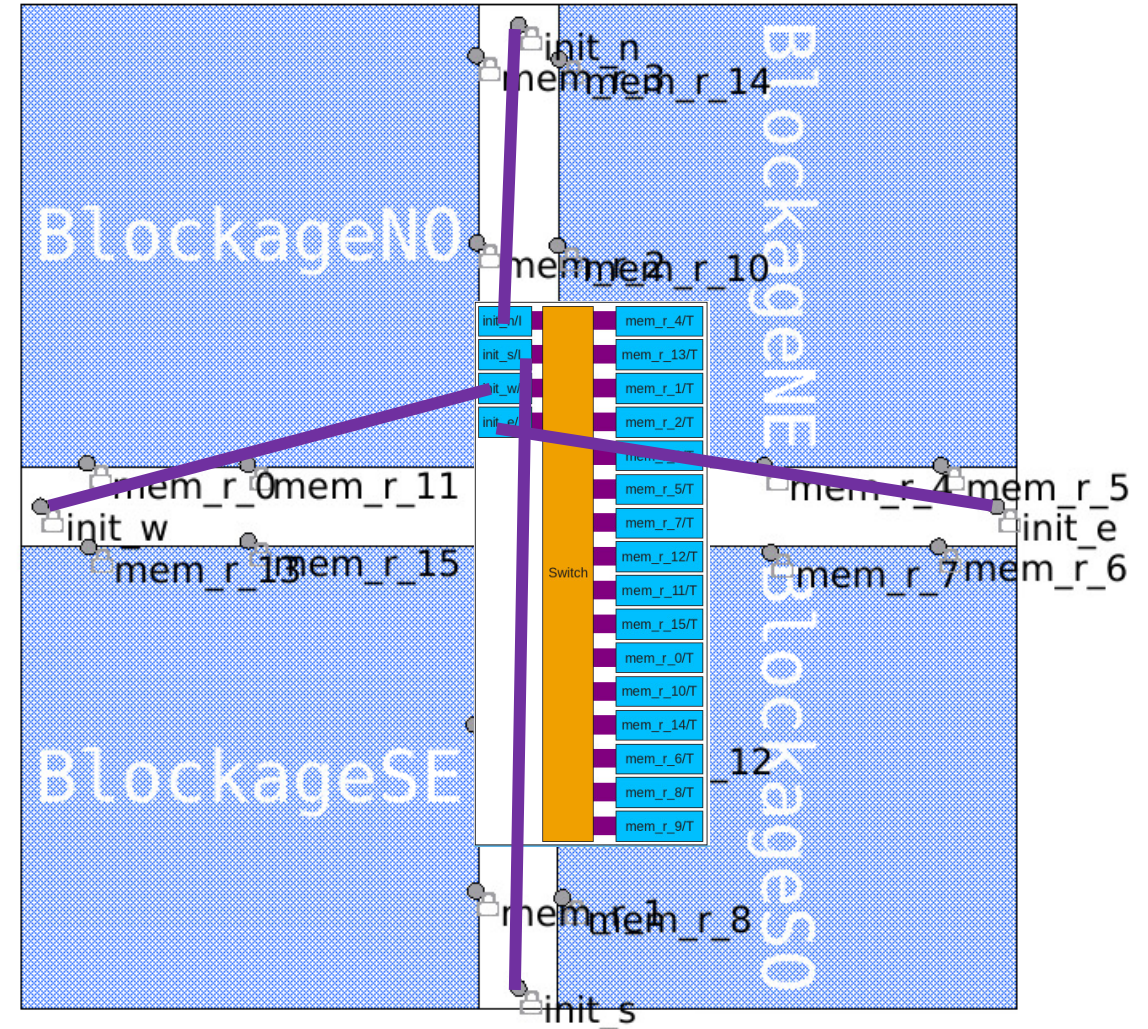
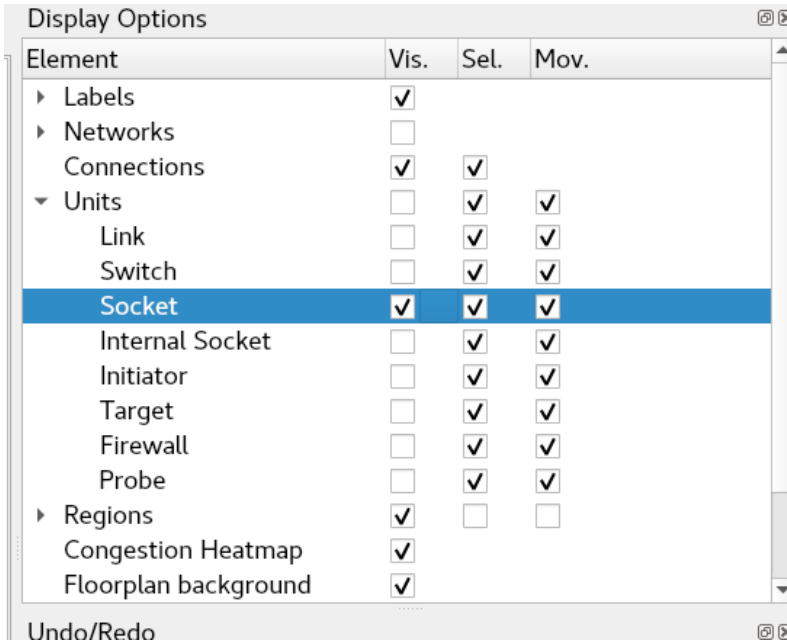




# Example Case

## Physical Socket Placement

- Initial constraints
  - Blockages on Floorplan View
  - Socket Positions

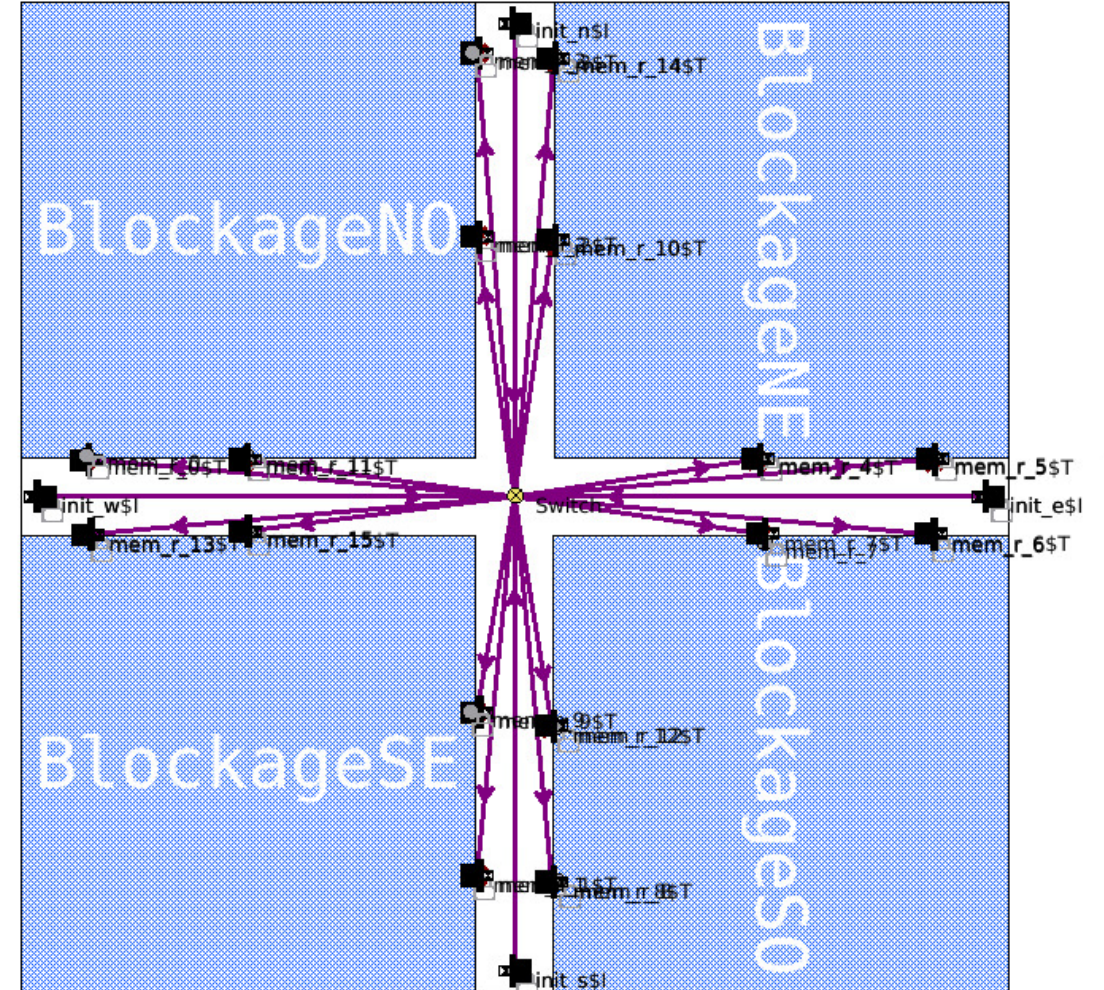
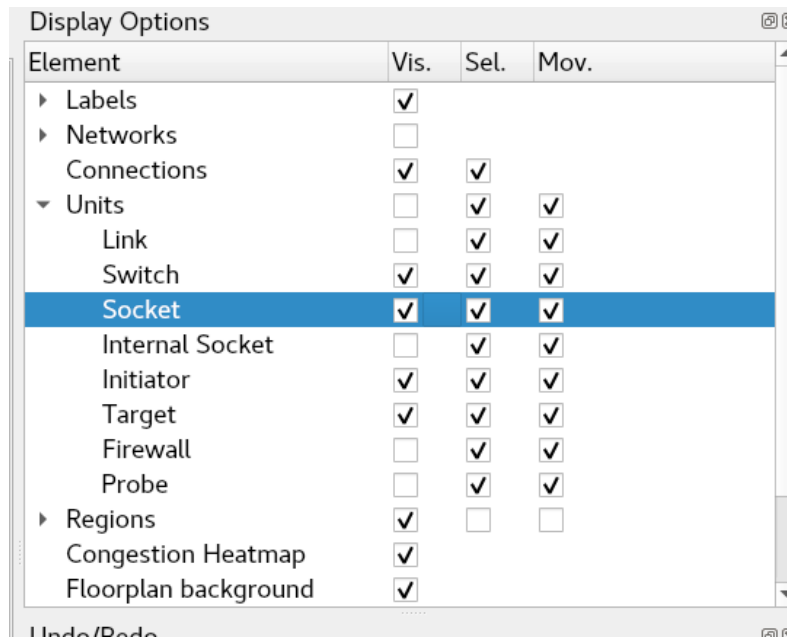




# Example Case

## Initial Crossbar Design

- Place the initial topology
  - Automated starting point
  - Note the possible central congestion issue

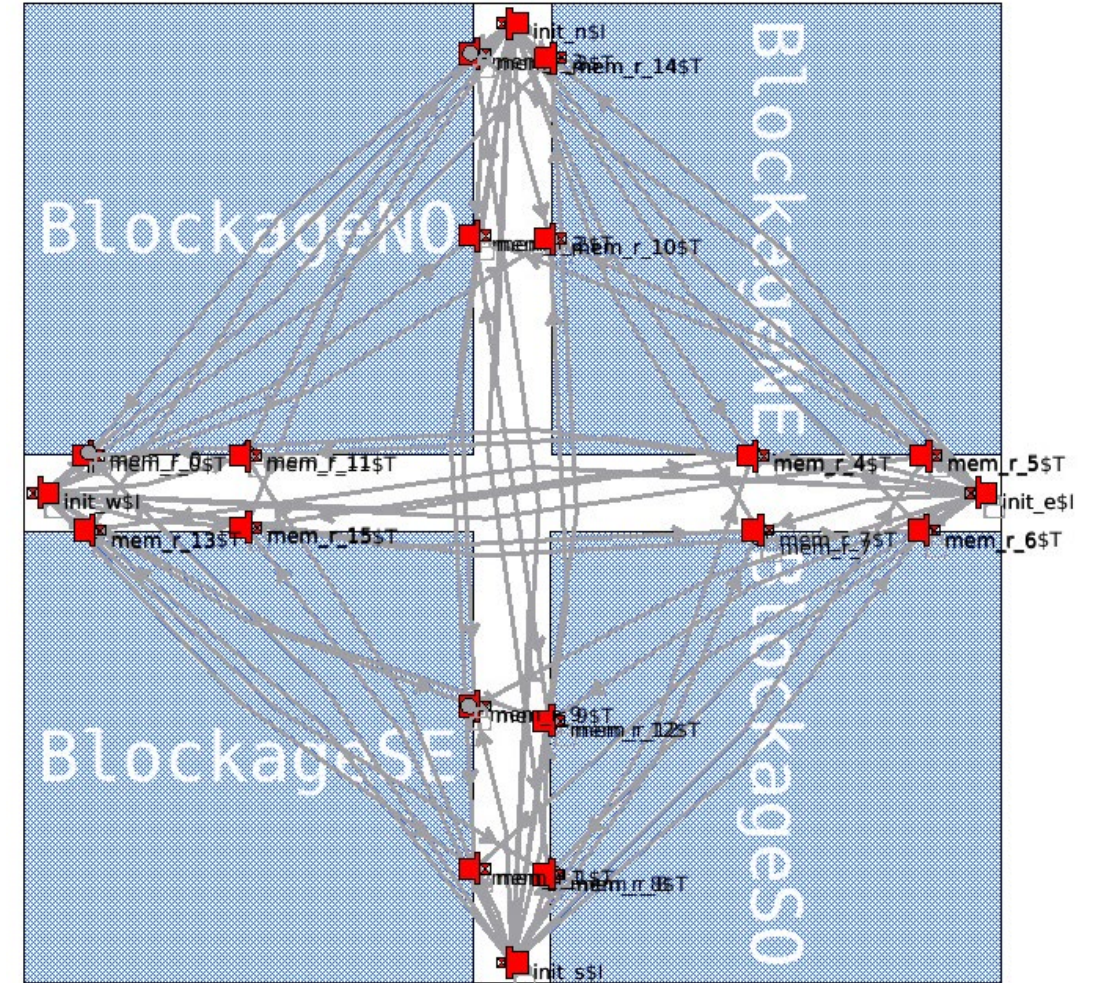




# Example Case

## Considering Options – Step 1

- Reset the initial topology
  - Users can create a more optimal version using the floor plan as a guide

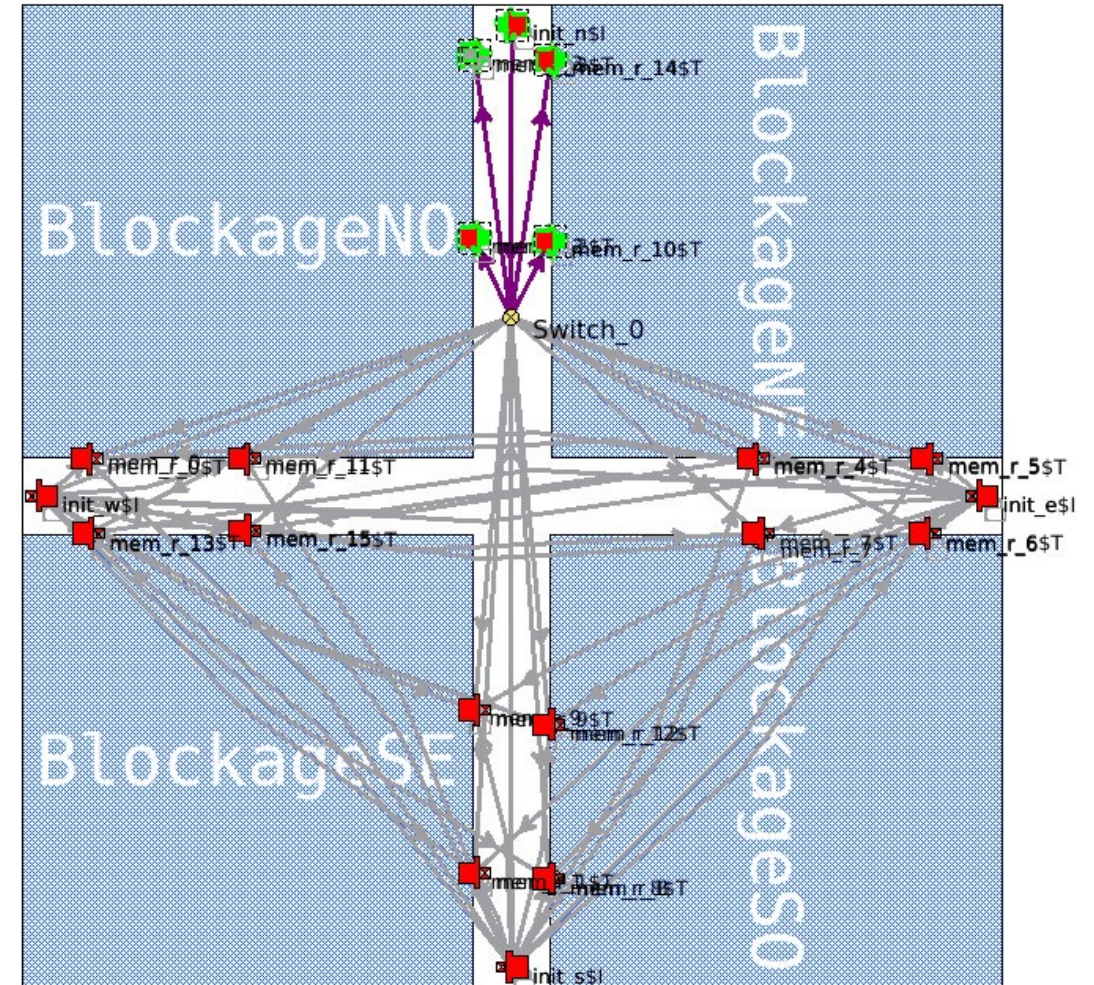




# Example Case

## Considering Options – Step 2

- Add a switch
  - Selecting 5 connections to be connected to a new switch as a group

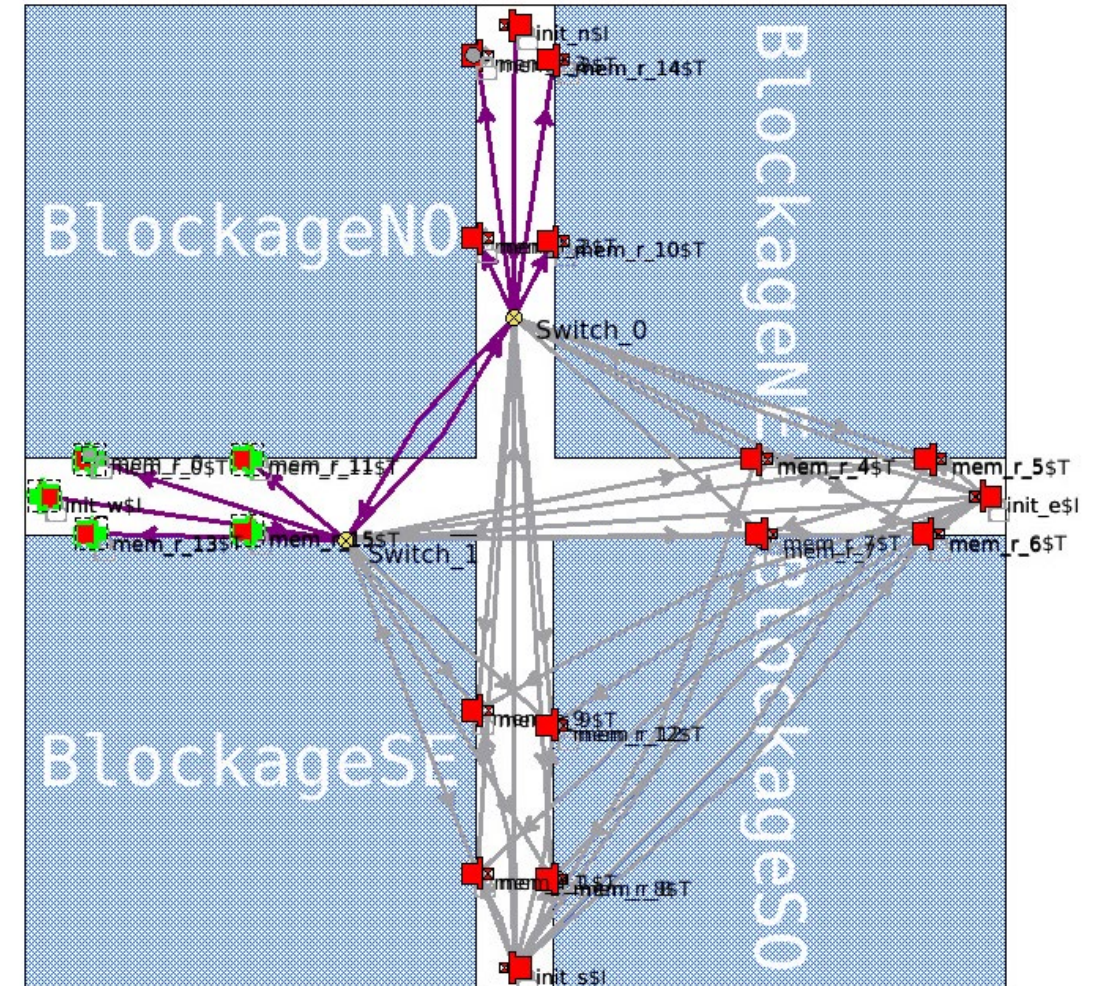




# Example Case

## Considering Options – Step 3

- Add another switch
  - Selecting another 5 connections to be connected to a new switch as a group

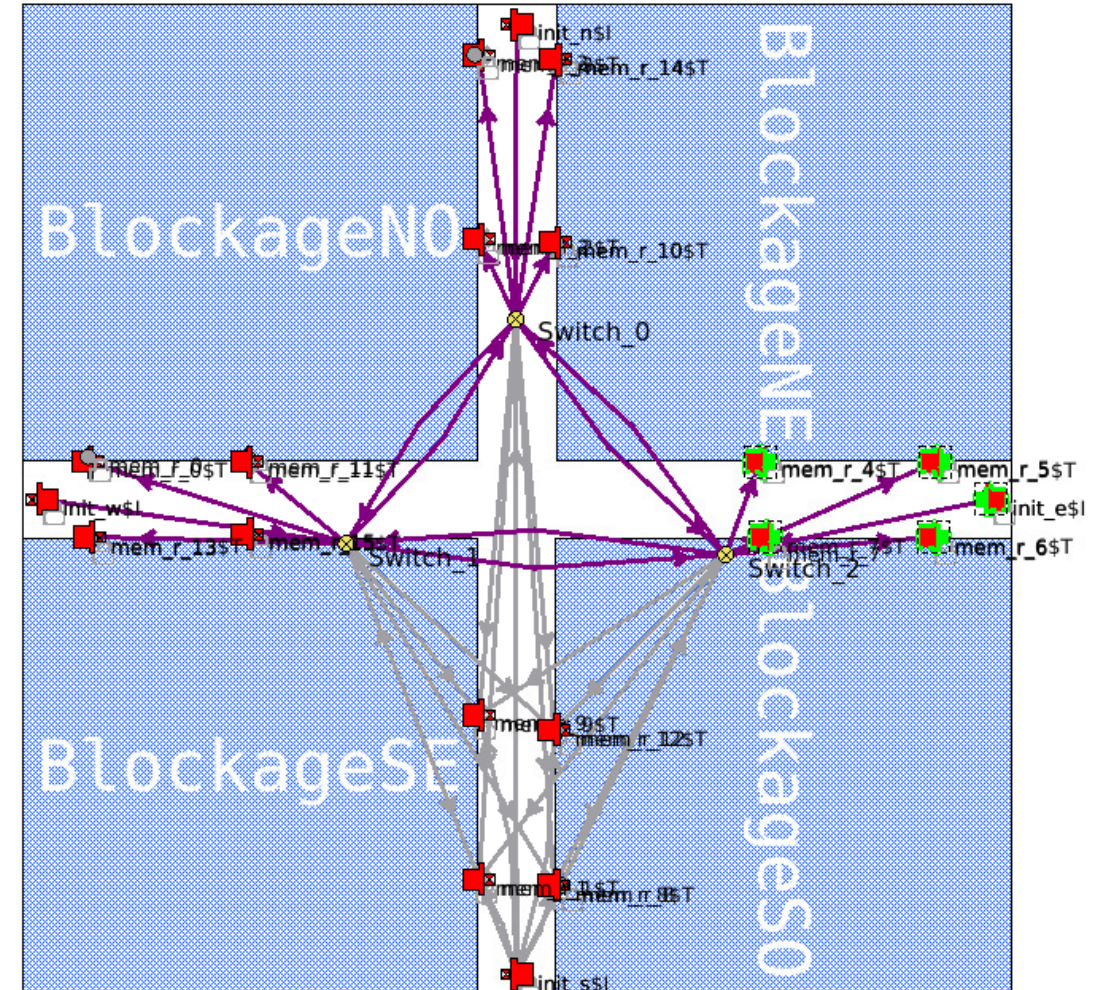




# Example Case

## Considering Options – Step 4

- Add another switch
  - Selecting another 5 connections to be connected to a new switch as a group

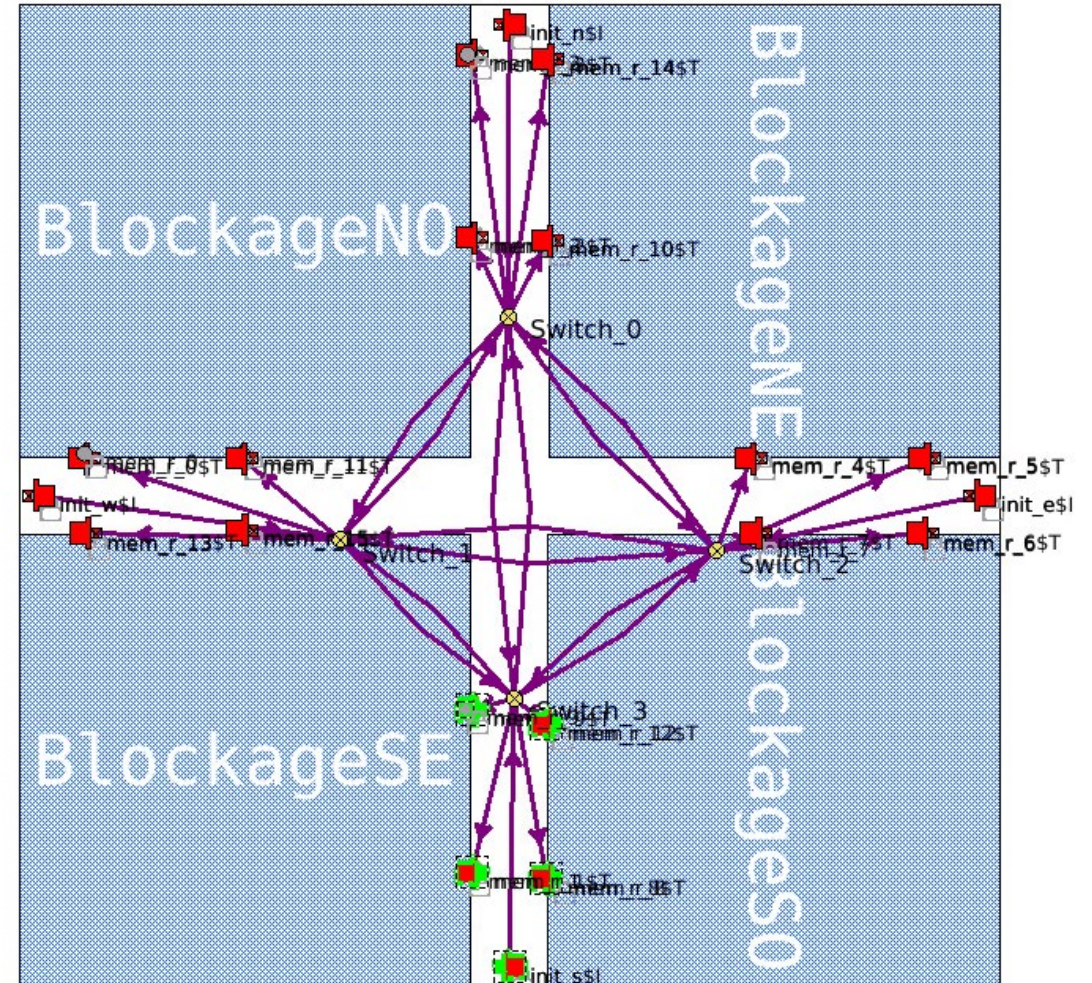




# Example Case

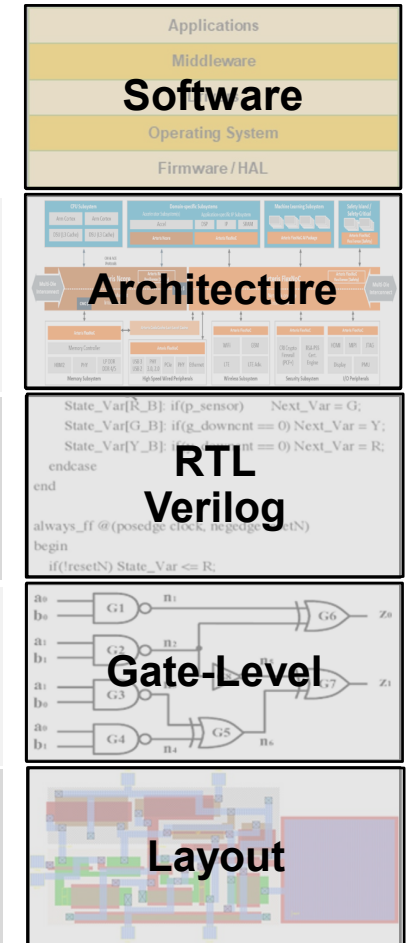
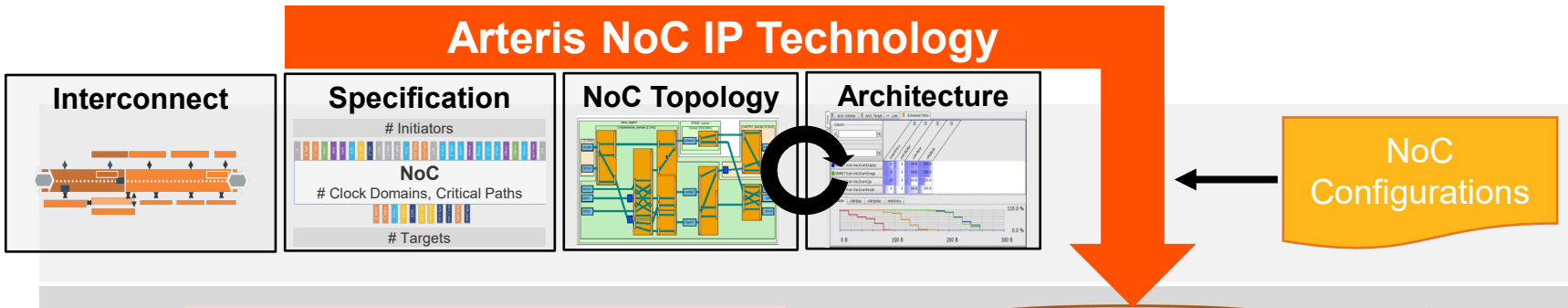
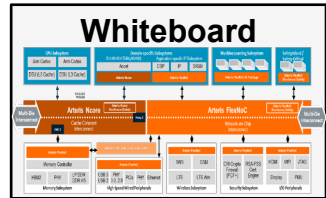
## Considering Options – Step 5

- Add another switch
  - Selecting another 5 connections to be connected to a new switch as a group



# RTL Architect Exploration

Converge faster with less customer iterations



Timing, logic levels, flop count/area numbers determine whether a design can be implemented

RTLA synthesis to confirm timing with Arteris configurable NoC RTL IP  
better PPA and faster runtimes

RTL + Constraints  
RTL Architect +aligned margins  
PPA Estimates

RTL Architect predicts implementation PPA more accurately  
Customers arrive at implementable configurations faster

# PPA Review with RTL Architect

- RTL Architect
  - Provides reliable timing
  - Allows assessments early
  - ... to fix high logic levels
- Designers review violations
  - Possible timing bug
  - Possible recoding
  - Allowing ULVT
  - Adding a pipe stage
  - Choosing faster memory



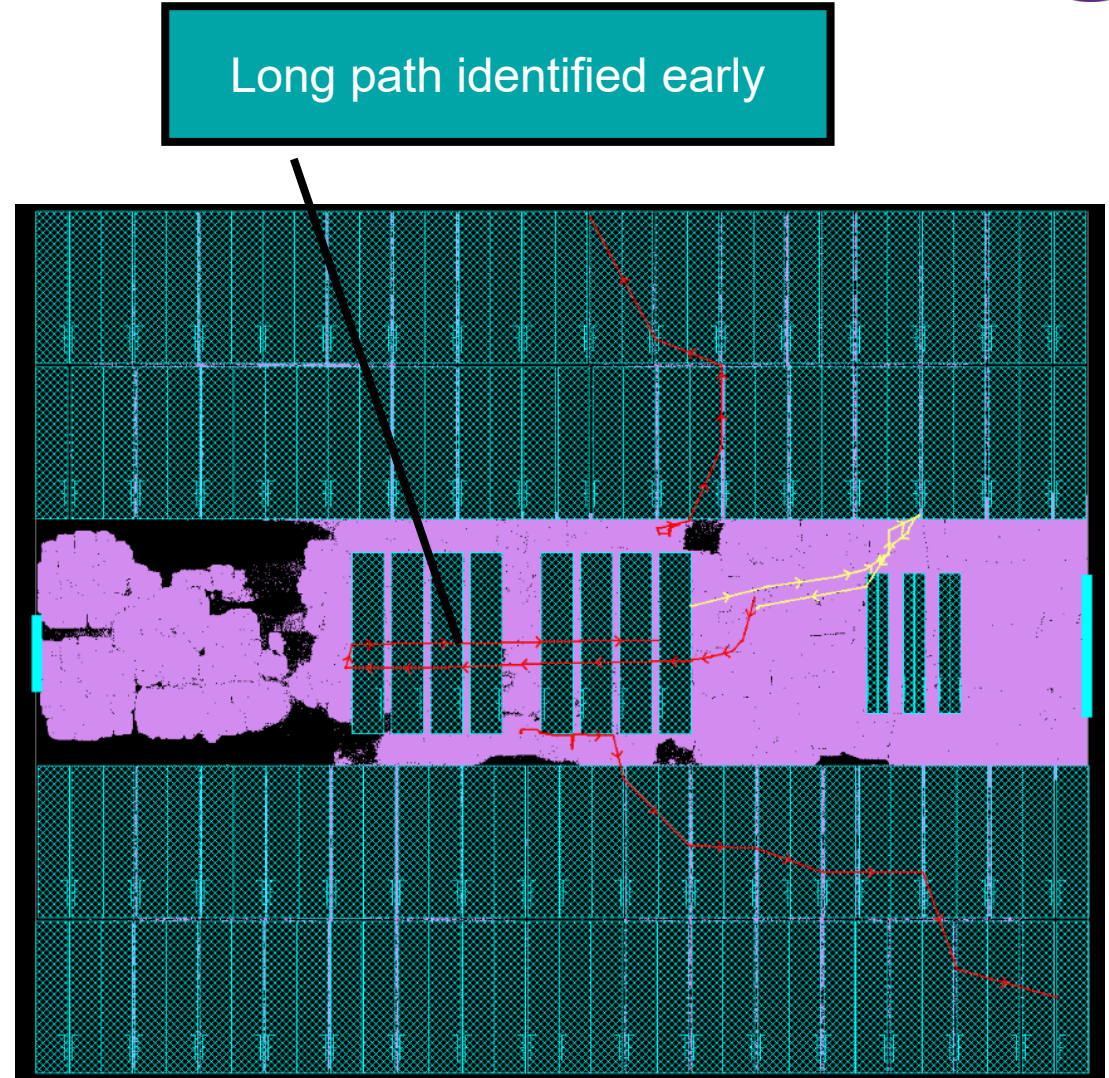
Floorplanned Blocks		Wed Mar 8 19:24:44 2023									
Block	Flop2Flop Paths				Macro2Flop	Flop2Macro	Errors & Warnings	Bucketizer Link			
	WNS (ns)	TNS (ns)	NVP	LOL	WNS	WNS					
aiu_top_a	-0.0116	-0.99	149	36				<a href="#">Bucketizer</a>			
aiu_top_b	0.1731	0.00	0	19	0	0					
aiu_top_c	0.2067	0.00	0	21	0	0					
aiu_top_d	0.1083	0.00	0	19	0	0					
config_dii_a	0.0186	0.00	0	25	0	0					
csr_network_a	0.0328	0.00	0	10	0	0					
dce_a	0.0019	0.00	0	19	0.1519	0.0095					
dii_top_a	0.0103	0.00	0	25	0	0					
dmi_a	-0.0188	-1.87	431	19	-0.0585	-0.0509		<a href="#">Bucketizer</a>			
dmi_b	-0.0105	-0.95	251	35	-0.0530	-0.0403		<a href="#">Bucketizer</a>			
dmi_c	0.2524	0.00	0	16	1.3984	0.3572					
dmi_d	0.2524	0.00	0	16	1.3984	0.3572					
dn_a	0.0008	0.00	0	8	0	0					
dve_a	0.0259	0.00	0	3	0	0					
fsc_a	0.1764	0.00	0	4	0	0					
gen_wrapper	-0.0115	-0.36	178	5	-0.0701	-0.0915		<a href="#">Bucketizer</a>			
grb_a	0.1342	0.00	0	6	0	0					
ioaiu_top_a	0.0026	0.00	0	4	0.0680	0.0090					
ioaiu_top_b	0.0003	0.00	0	20	0.0622	0.0005					
ioaiu_top_c	0.0024	0.00	0	37	0	0					
ioaiu_top_d	0.2839	0.00	0	19	0	0					
ndn1_a	0.0083	0.00	0	3	0	0					
ndn2_a	0.0367	0.00	0	6	0	0					
ndn3_a	0.0722	0.00	0	4	0	0					

Source: Internal Reference Design, 5 nm



# PPA Review

- RTL Architect allows early review of placement issues
- Allows to address well before actual layout runs are done
  - Refining Floorplan
  - Exploring with placement bounds
  - Exploring rtl recoding



Source: Internal Reference Design, 5 nm

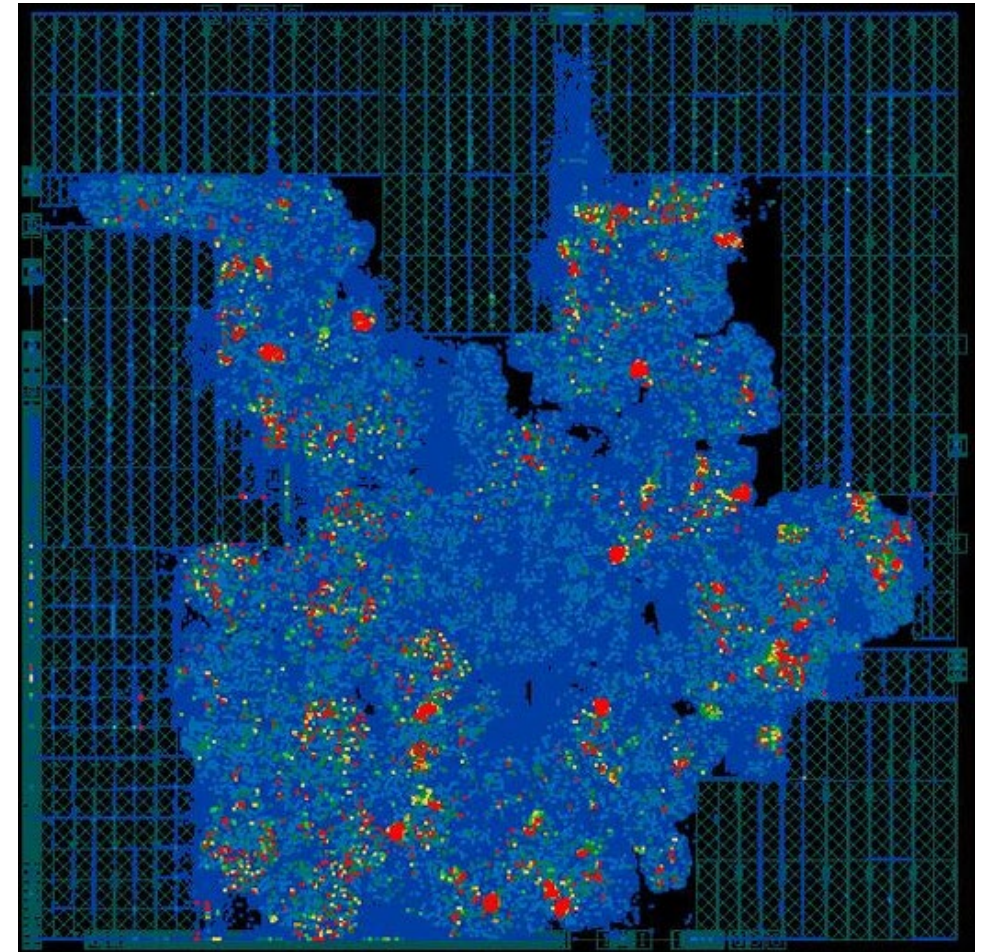


# PPA Review

- Using RTL Architect in early congestion analysis with PG
  - Review Cell Density map & Hot spots
  - Review Utilization
  - Make floorplan changes if any

```
Both Dirs: Overflow = 133855 Max = 140 (1 GRCs) GRCs = 56061 (0.18%)  
H routing: Overflow = 50390 Max = 140 (1 GRCs) GRCs = 17718 (0.11%)  
V routing: Overflow = 83465 Max = 90 (1 GRCs) GRCs = 38343 (0.24%)  
1
```

Source: Internal Reference Design, 5 nm

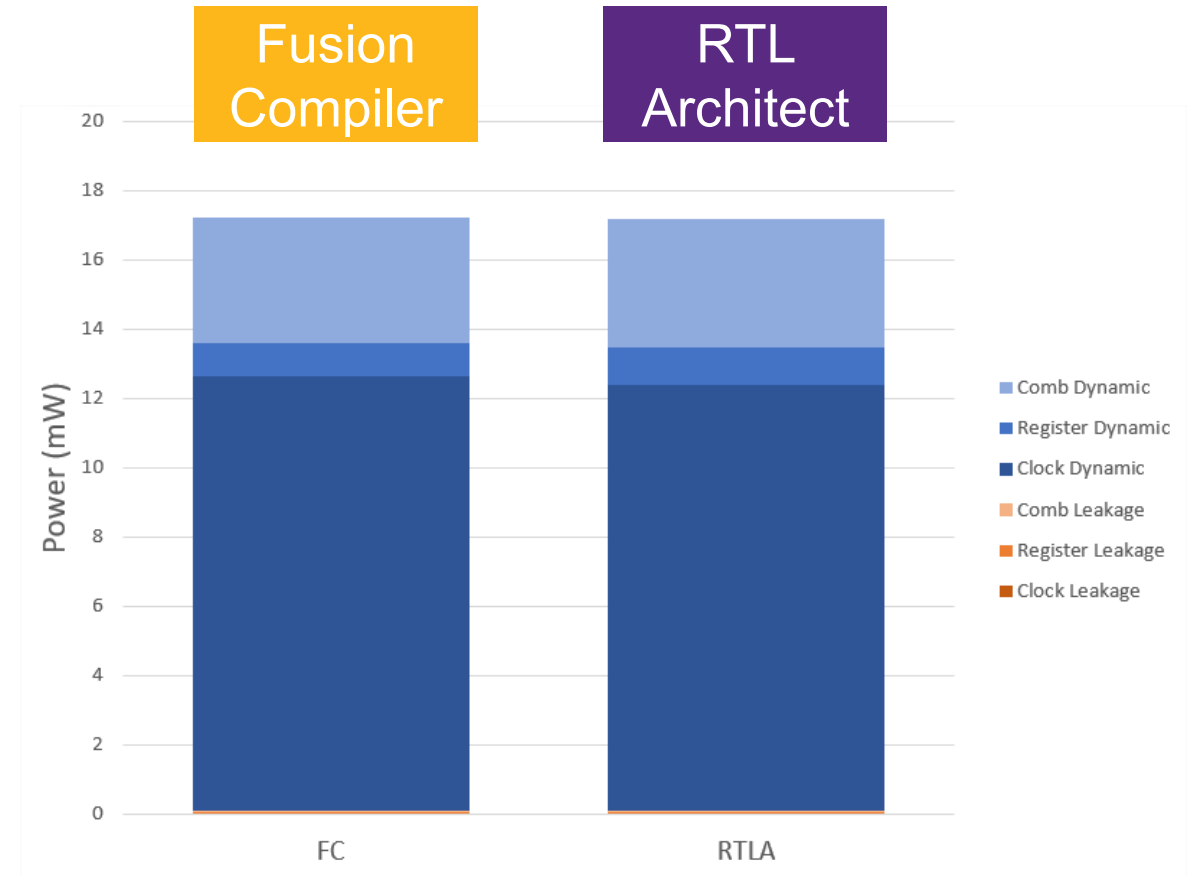


Source: Internal Reference Design, 5 nm

# Early Power Estimation

- RTL Architect provides reliable power estimates

Run	Clock Leakage	Register Leakage	Comb Leakage	Clock Dynamic	Register Dynamic	Comb Dynamic	Total
FC	5.12E-04	5.00E-02	4.25E-02	1.25E+01	9.71E-01	3.61E+00	1.72E+01
RTL	5.10E-04	5.27E-02	4.07E-02	1.23E+01	1.08E+00	3.72E+00	1.72E+01
Deviation	-0.4%	5.4%	-4.2%	-2.1%	10.9%	2.9%	-0.2%

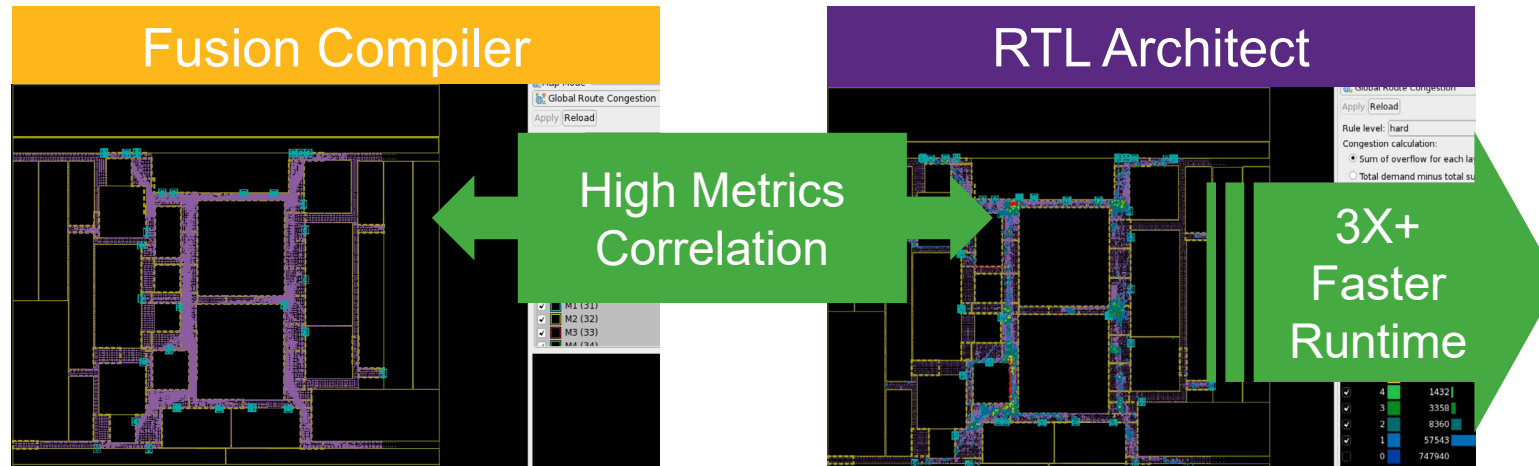
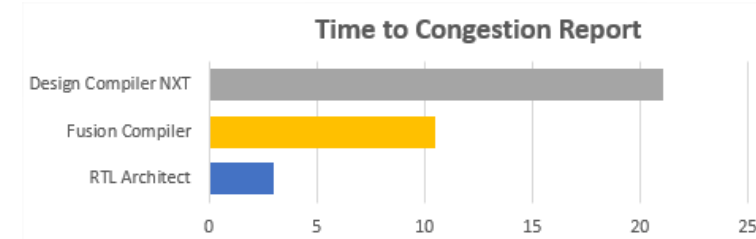


Source: Internal Test Design, 16nm

# NoC Implementation with RTL Architect

- RTL Architect QoR results highly correlated to Fusion Compiler
  - Area within 2%
  - Timing within 10%
  - Power within 5%

- RTL Architect runtime 3X+ faster than Fusion Compiler & 6X+ faster than DCNXT



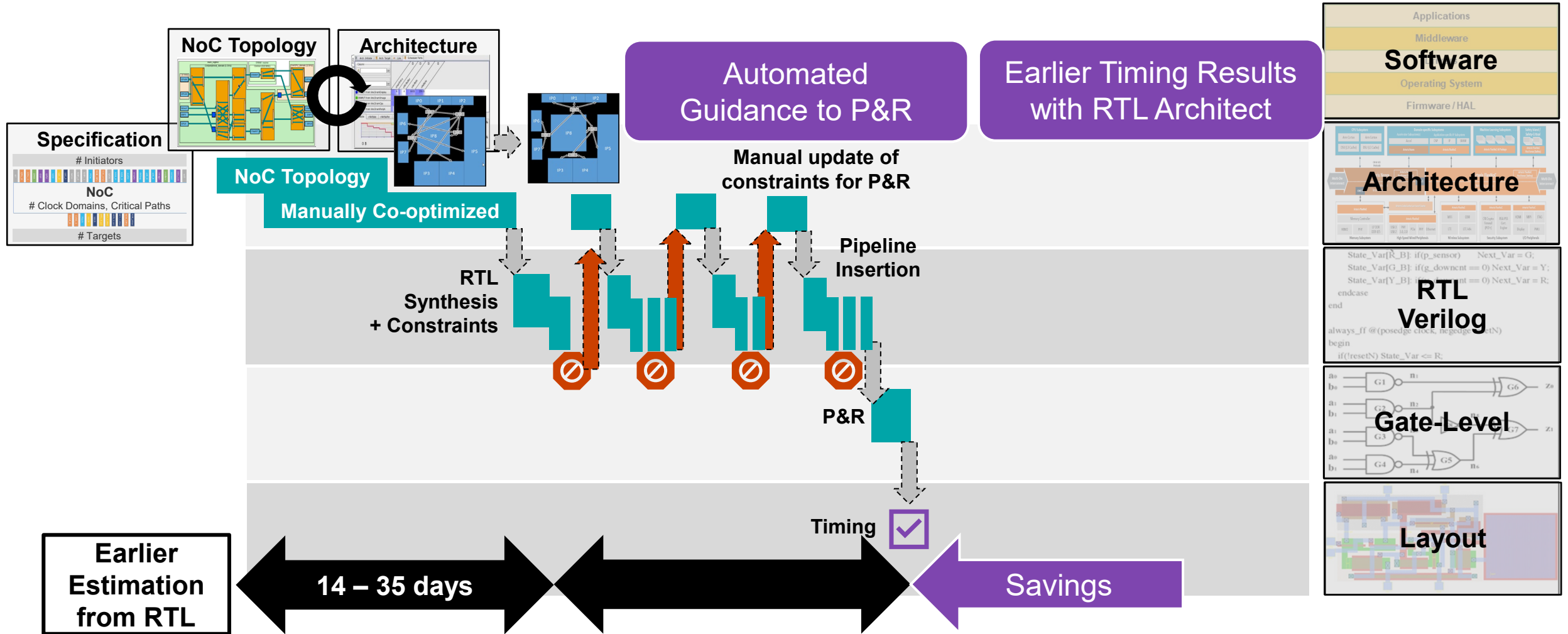
**Improved RTL quality with RTL Architect  
Better RTL delivering better PPA**

# Applying RTL Architect Earlier Estimations

ARTERIS IP



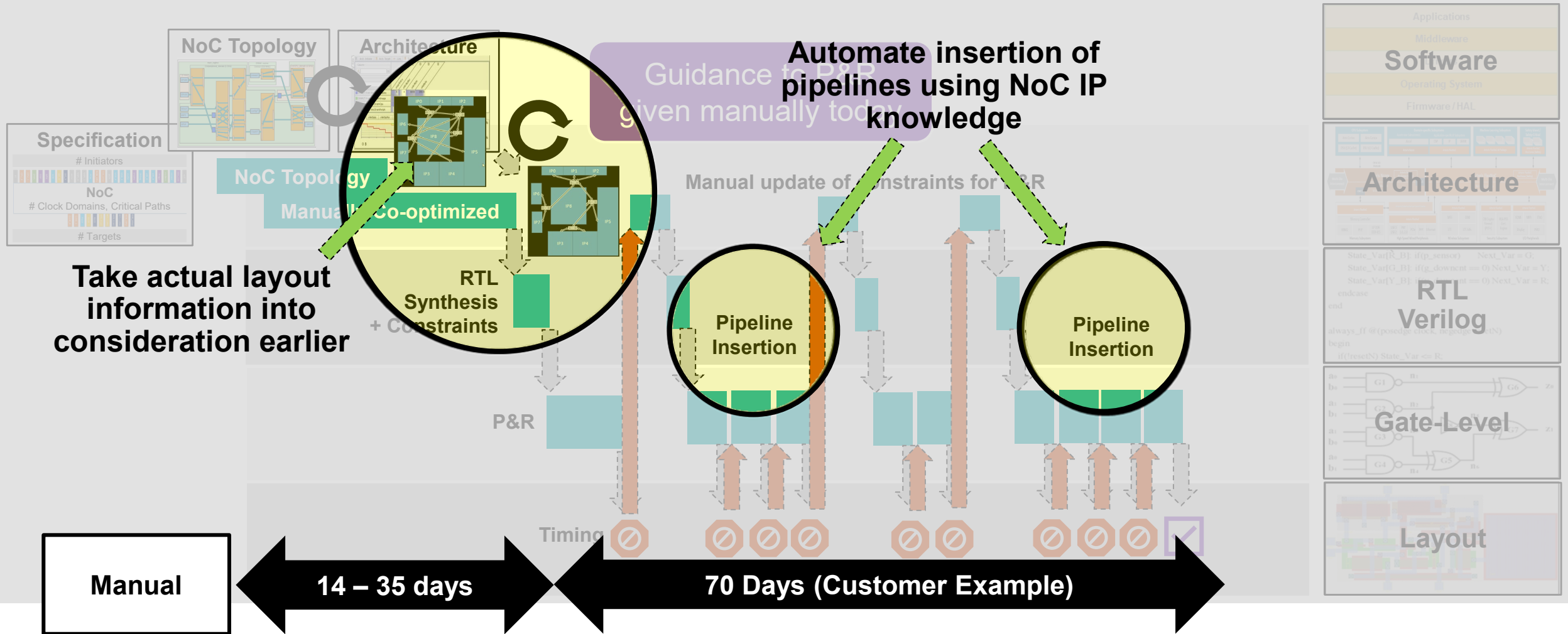
Accelerate Turn-Around-Time





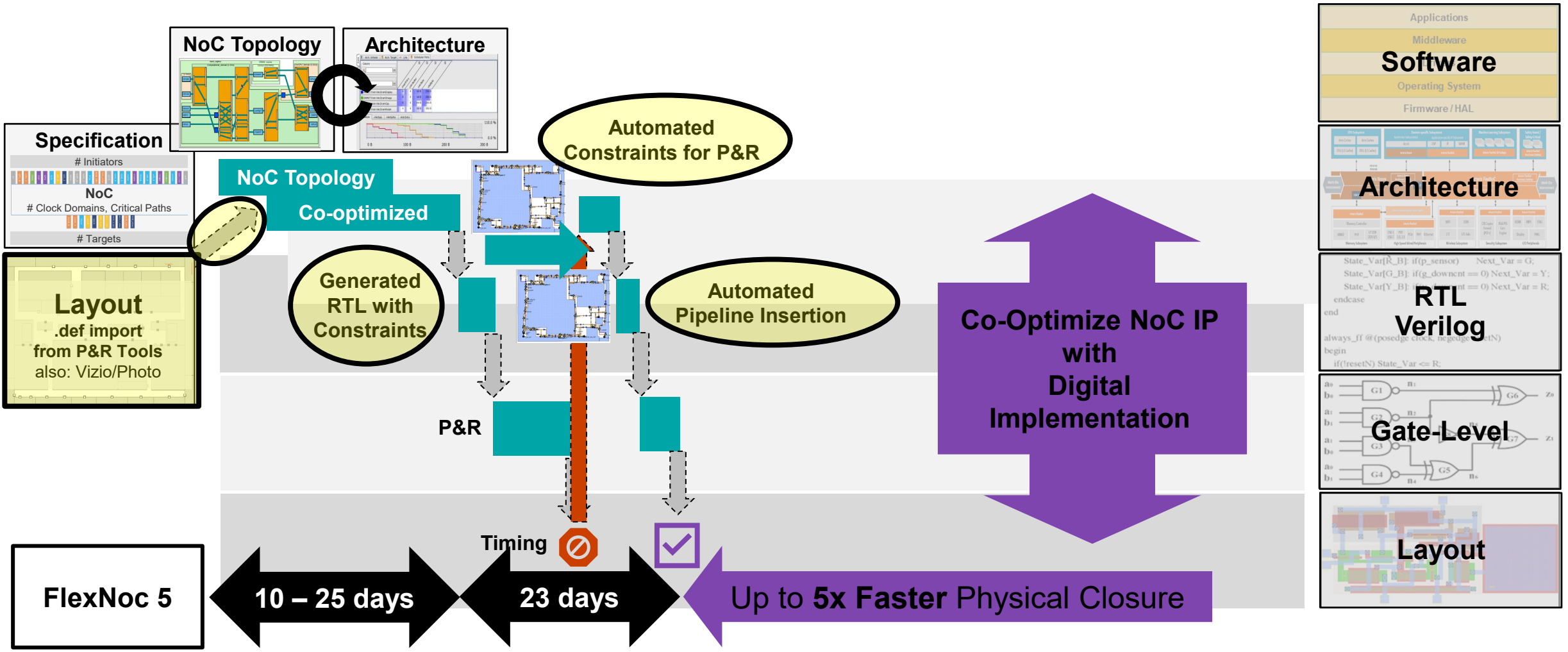
# Further Automation Opportunities

Apply further abstraction



# FlexNoC 5

Announced February 2023



Applications  
Middleware  
**Software**  
Operating System  
Firmware / HAL

Architecture

```

State_Var[R_B] if (p_sensor) Next_Var = G;
State_Var[G_B] if (g_downcount == 0) Next_Var = Y;
State_Var[Y_B] if (f_downcount == 0) Next_Var = R;
endcase
end
always_ff @(posedge clock, negedge rstn)
begin
  if (resetN) State_Var <= R;

```

Gate-Level

Layout

# Conclusions

# Conclusion

- Fast RTL Exploration Solution
  - Built on implementation and signoff engines for improved convergence
  - Will benefit our customers and easy interaction
- Collaboration and Next Steps
  - Arteris IP to update physical constraints generation to drive floor plan refinement based on RTL Architect's timing and power estimation
  - Expand support of abutted vs. channel-based implementation

➤ **Synopsys and Arteris are collaborating to improve NoC implementation flows for tough floorplans**



***THANK YOU***

***YOUR  
INNOVATION  
YOUR  
COMMUNITY***