

# Using RTL Architect for Designing the Ultra-Low Power EdgeVision SoC

Mikail Yayla, Senior Digital IC Design Engineer  
Racyics GmbH

# Session Overview

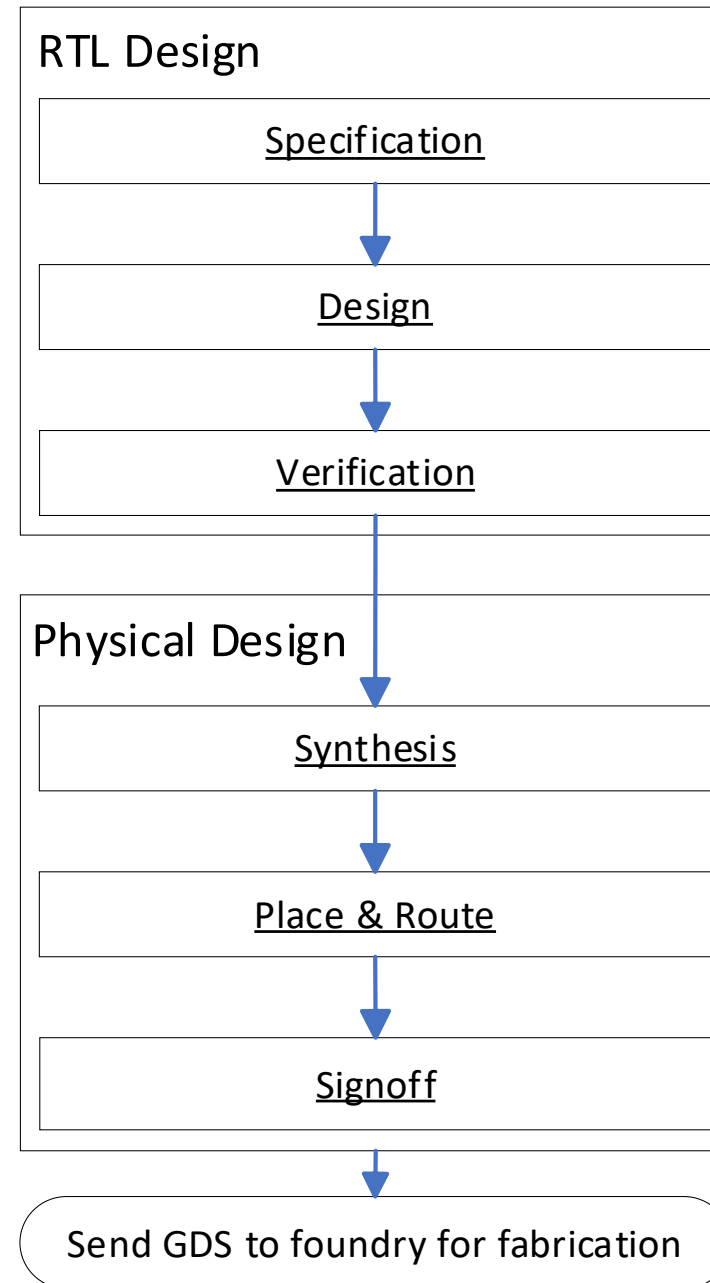


- RTL2GDS Flow
- Design Challenges
- RTL Architect
- Evaluating PPA-Impact of RTL-level changes

# RTL2GDS Flow

An overview of the general flow

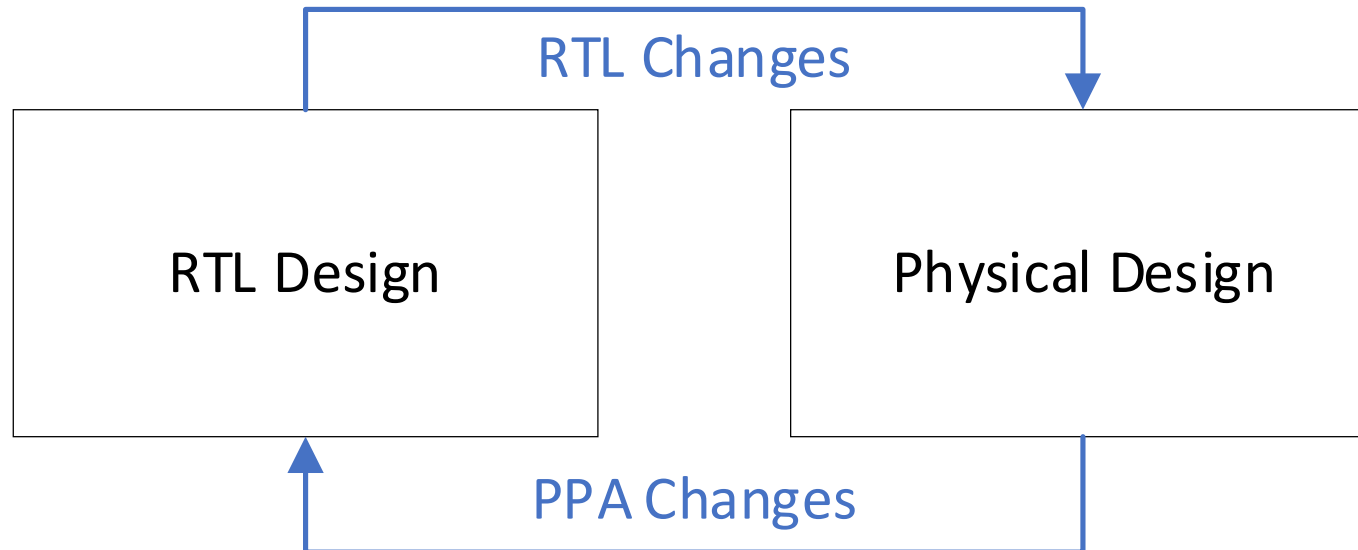
- RTL design
  - Meet with customer, discuss specifications
  - Use HDL to realize functionality
  - Verify HDL
- Physical design
  - Transform HDL to netlist
  - Layout elements of netlist on chip, includes routing
  - Perform signoff checks



# Design Challenges

## RTL2GDS flow reruns

- RTL-level modifications needed in RTL2GDS flow
  - RTL modifications require reruns of the RTL2GDS steps
  - Long runtime of EDA tools
  - Negatively impact project schedule and risk tapeout delay!

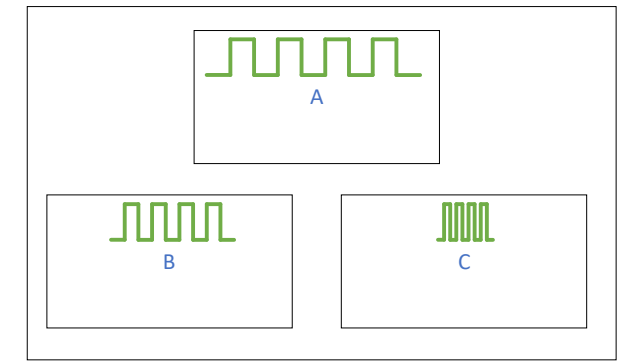
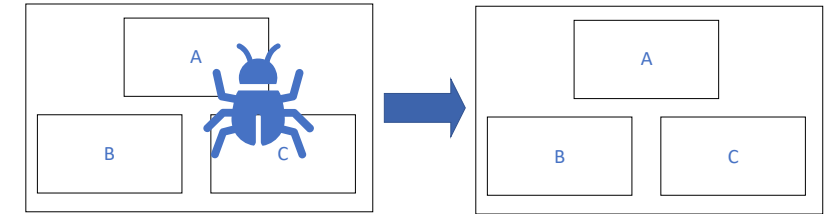
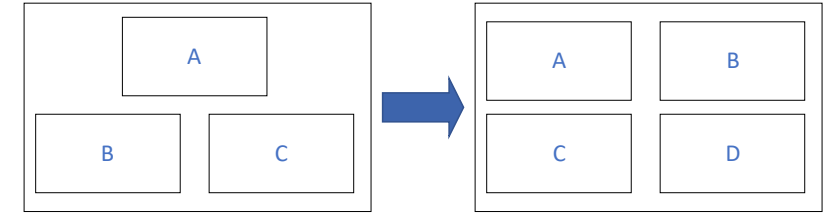


# Examples of Challenges



## RTL-level modifications that require RTL2GDS reruns

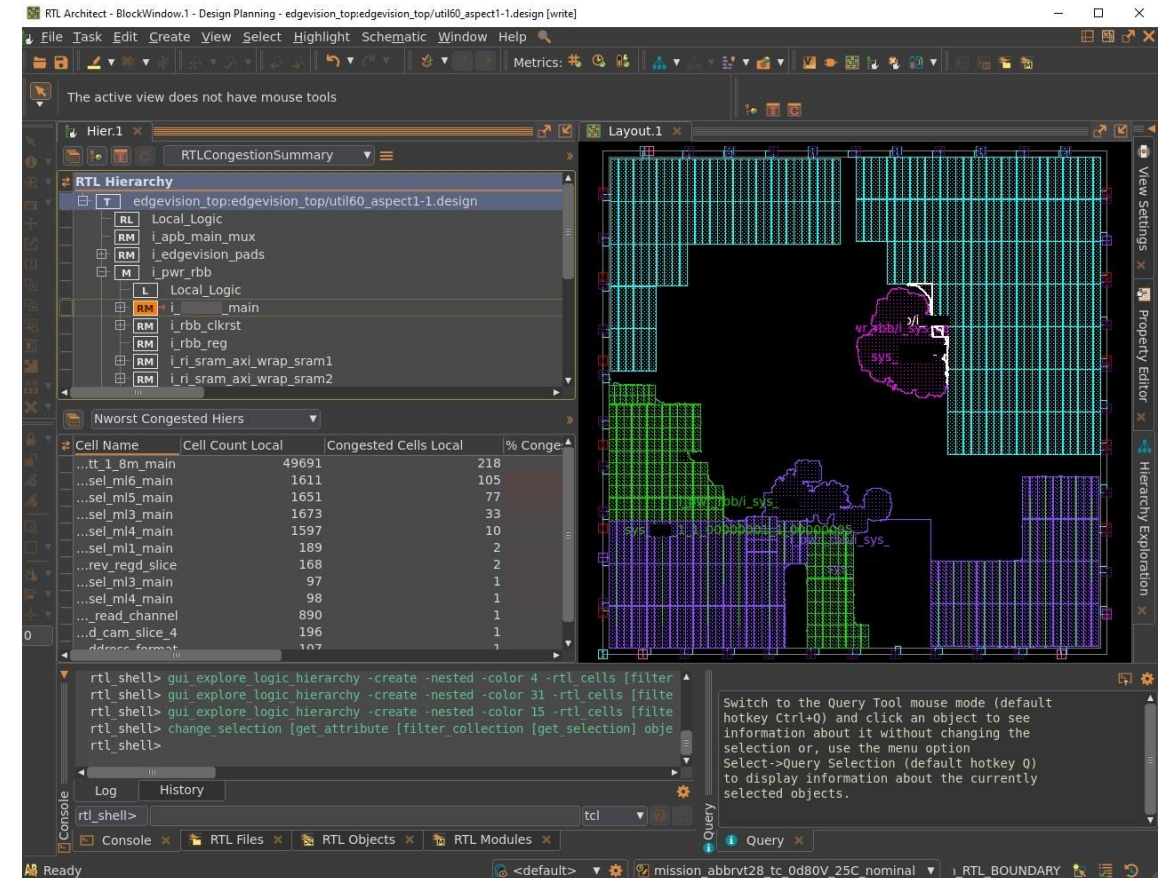
- IP configuration
  - IPs have multiple configuration options
  - Changes in configuration lead to RTL modifications
- Design issues or bugs
  - RTL bugs may become evident in different steps of RTL2GDS flow
  - RTL may not be able to be realized in physical design
- Clock configuration
  - Need to find suitable clock domain frequencies
  - Optimizations in physical design usually needed to meet constraints



# RTL Architect

By Synopsys

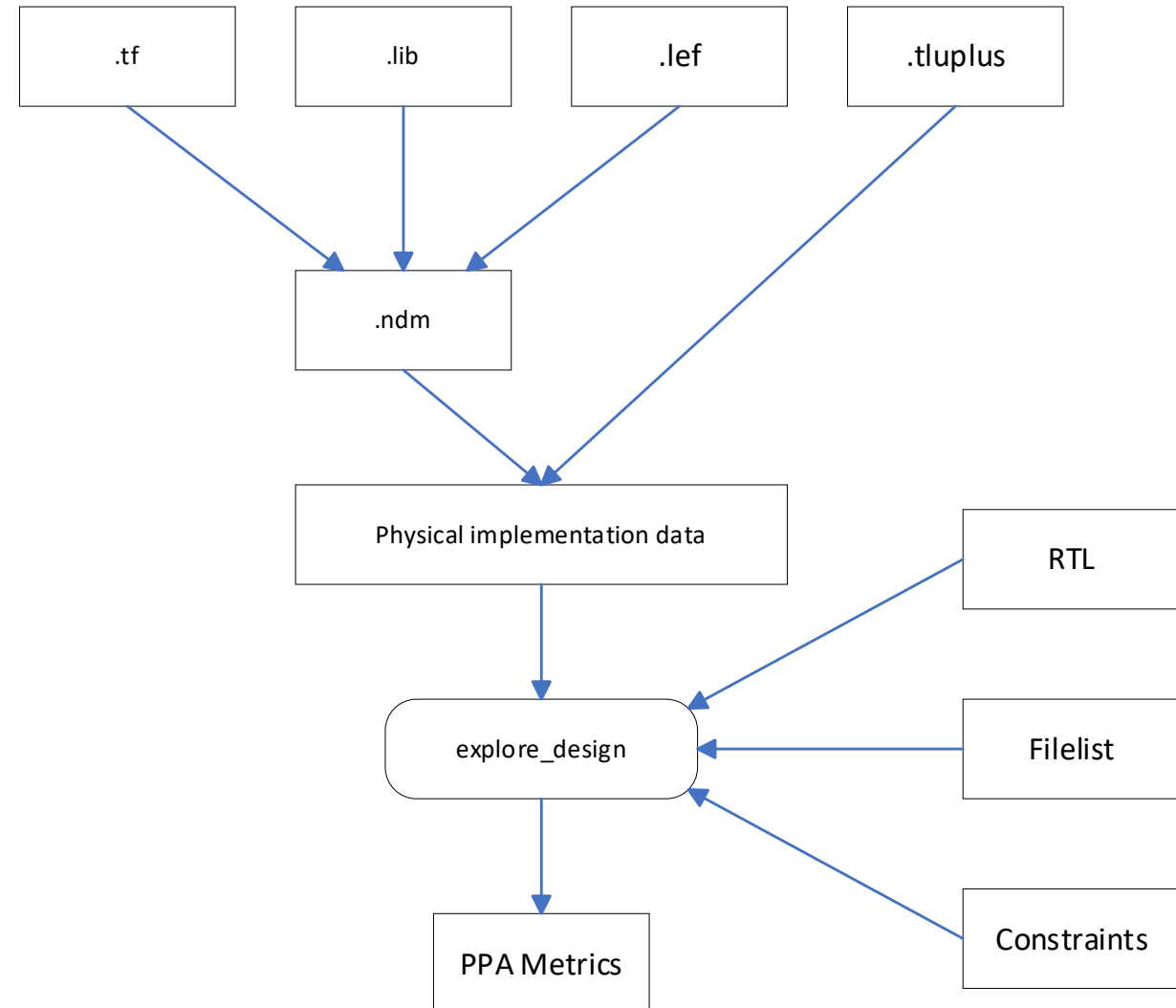
- Addresses challenges in RTL2GDS flow
  - RTL, clock, power, etc. modifications
  - High-level view for RTL engineers
- Benefits of RTL Architect
  - Enables assessing PPA-impact of RTL changes
  - RTL engineers can perform high-level DSE
  - Avoids costly feedback loop in RTL2GDS flow
  - Fast and accurate



# RTL Architect Prerequisites and Flow

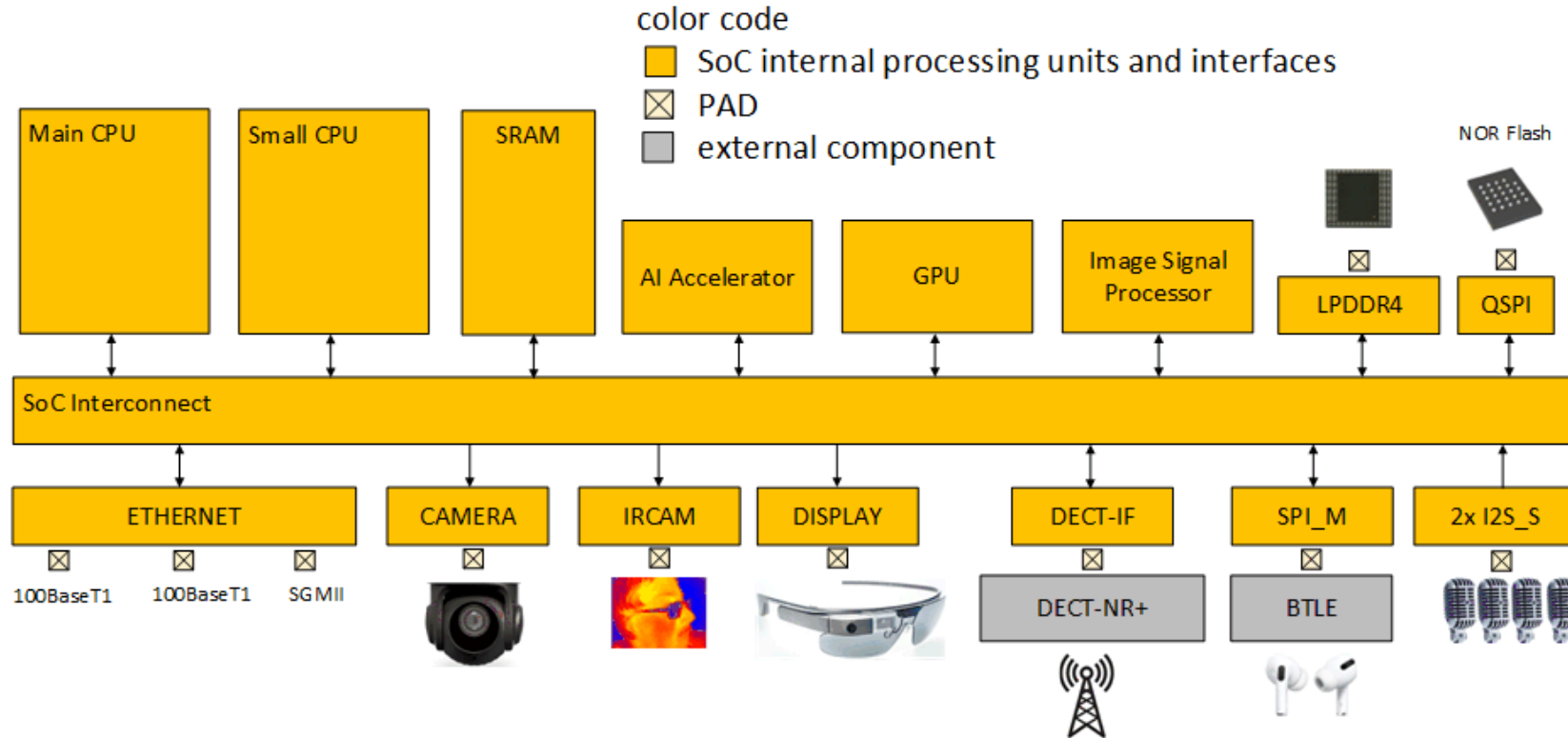


- Physical implementation data
  - .tf: Technology file
  - .lib: Reference library
  - .lef: Abstracted physical layout info
  - .tluplus: Info for calculating parasitics
- RTL and filelists
  - Include functionality
- Constraints
  - Clock, power, area, etc.



# EdgeVision SoC

Developed by Racyics and partners



- Using RTL Architect in the design process of our EdgeVision SoC
  - Early decisions for the design regarding RTL changes and clock constraints
  - Assess impact of RTL-level modifications on PPA



# Experiment Scenarios and Settings

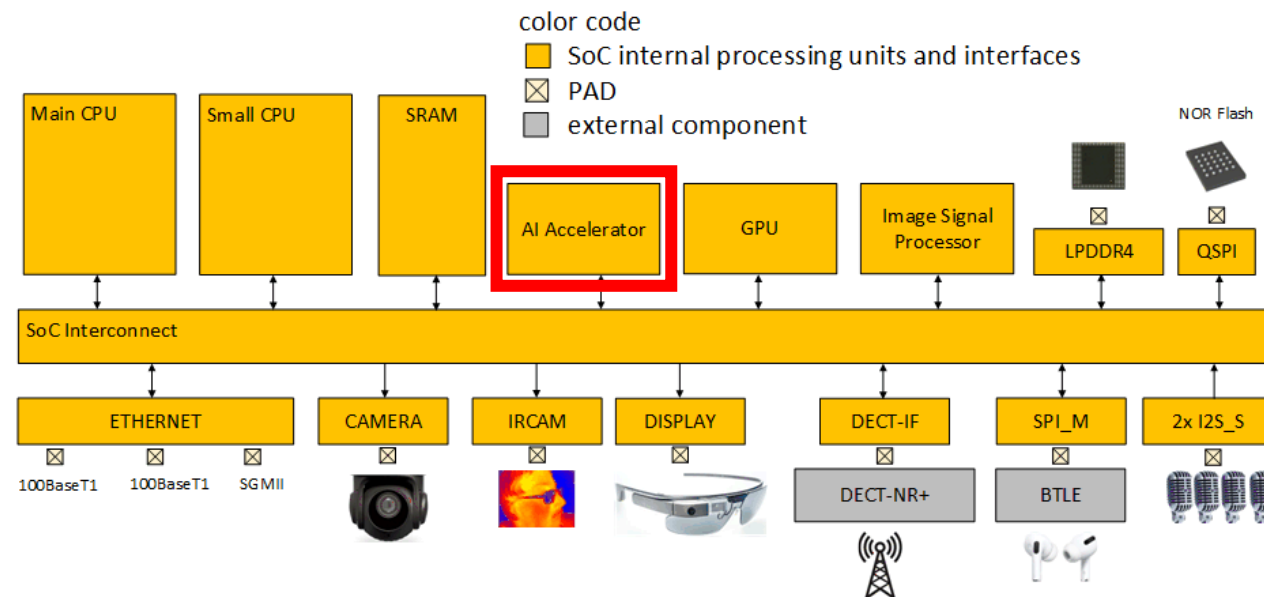
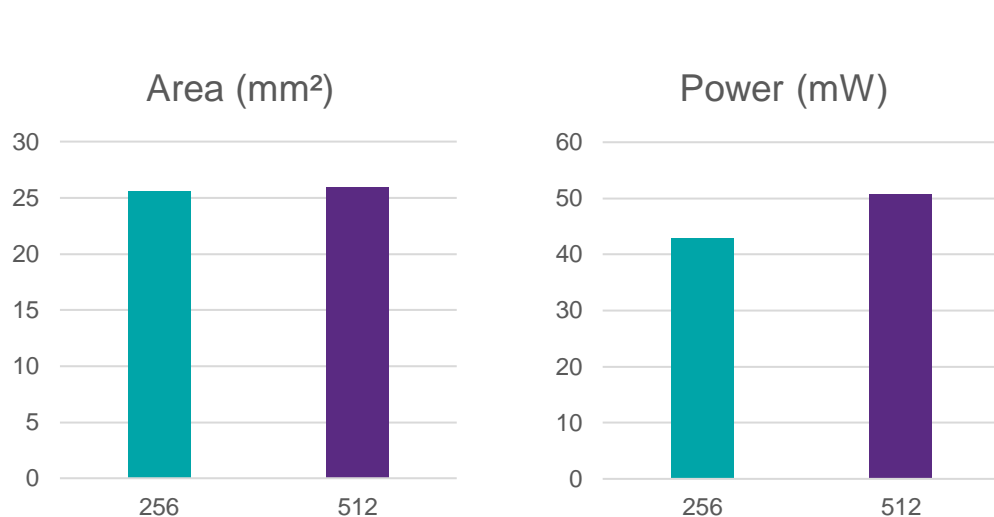


- AI accelerator configurations
  - Number of MACs in accelerator IP
  - PPA-impact of 256 vs. 512 MACs
- SRAM memory block size
  - Partners requested 1 to 2 MB SRAMs
  - PPA-impact of 1MB and 2MB SRAMs
- Clock constraints
  - Usually defined empirically
  - PPA-impact of 100, 200, 400 MHz
- Experiment setup
  - Two CPU cores and 100 GB RAM
  - Run experiments in parallel
- Common experiment settings
  - Two power domains
  - 1:1 aspect ratio of die
  - Utilization factor of 0.6
  - Typical corner

# AI Accelerator Configs: Area and Power



## 256 and 512 8x8 MAC Units

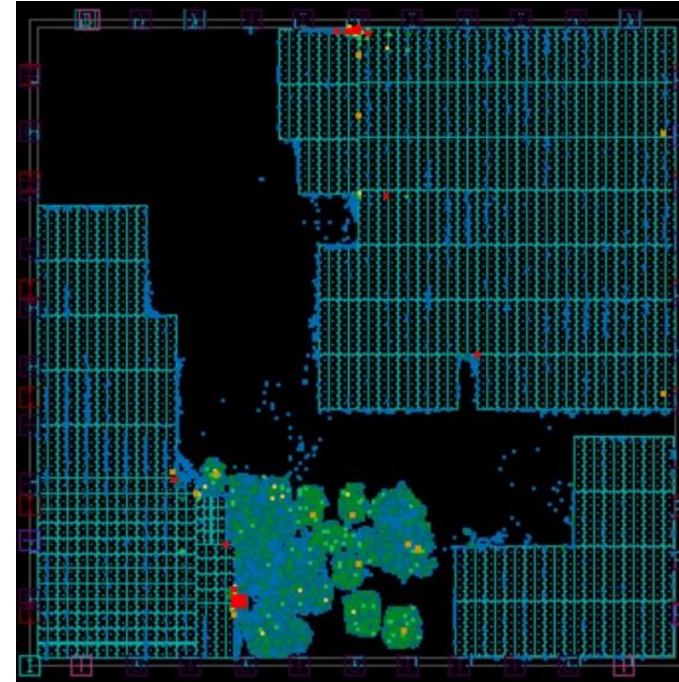
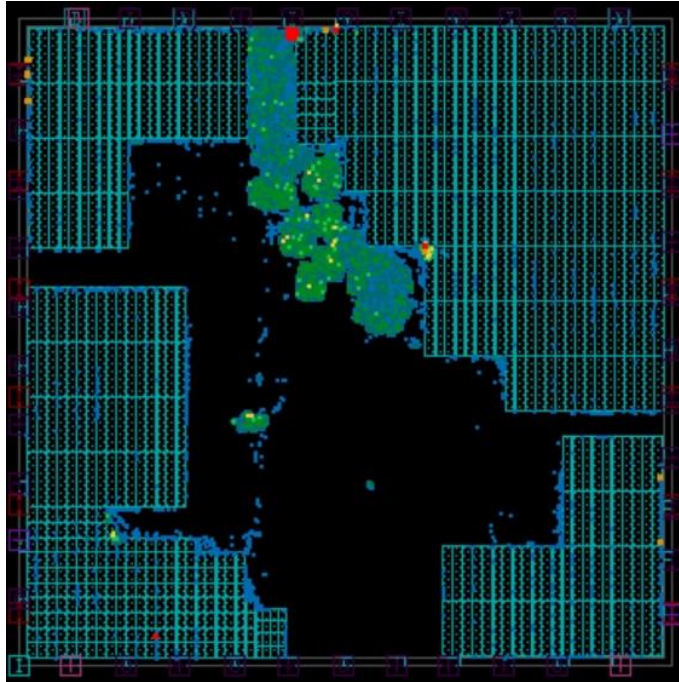


- AI accelerator experiment conclusion

- Almost no impact on die area (1%)
- Increase in power usage (18%)
- We plan to use 512 8x8 MAC units

# AI Accelerator Configs: Floorplan

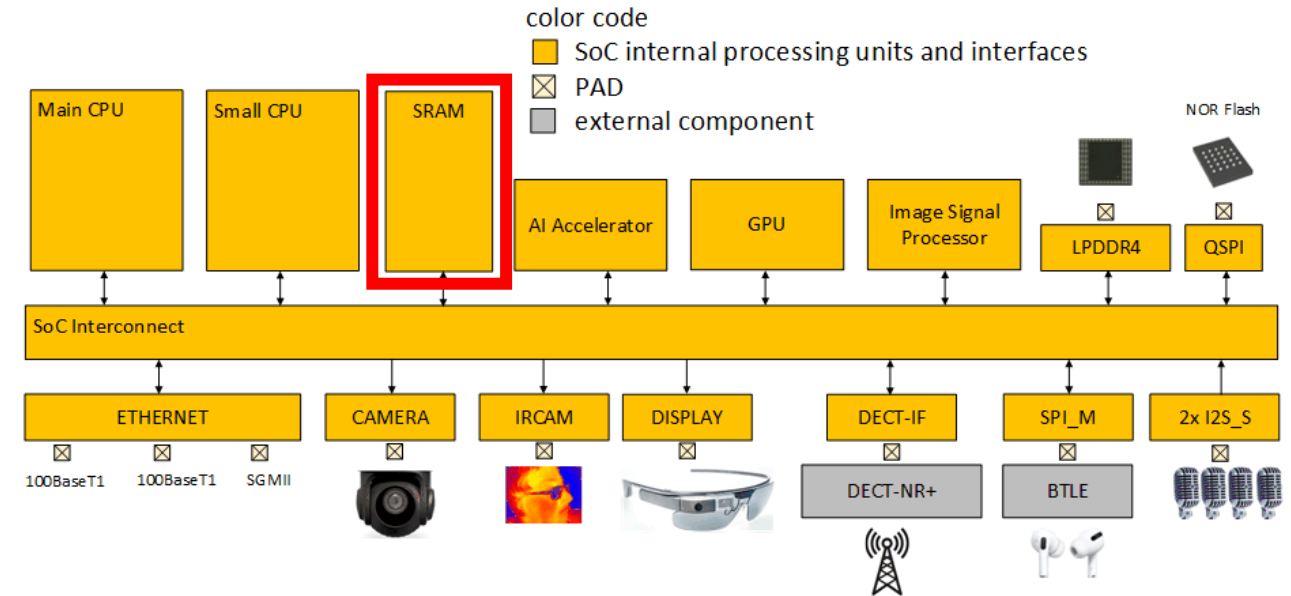
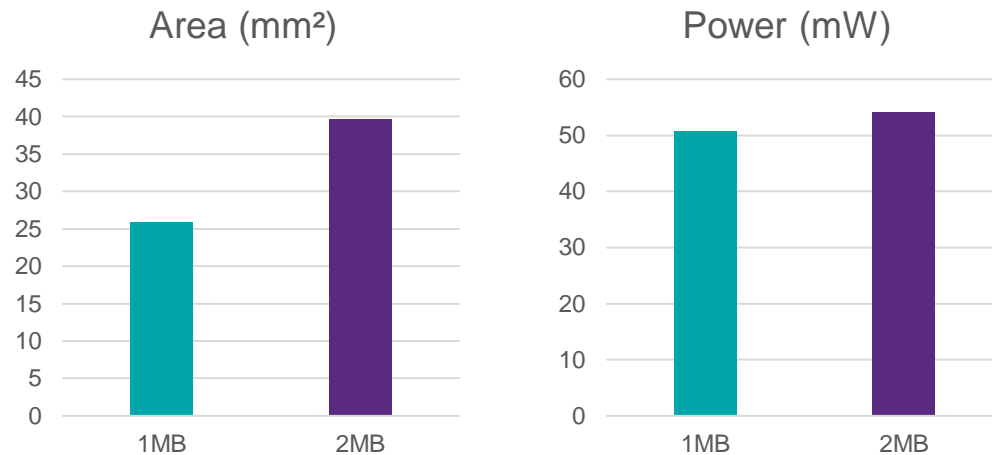
256 and 512 8x8 MAC Units



- Observations
  - Small area increase (1%), but 20% more FFs for 512 MACs
  - Memory needs larger area compared to logic

# Memory Configs: Area and Power

1 or 2 MB memory per SRAM module



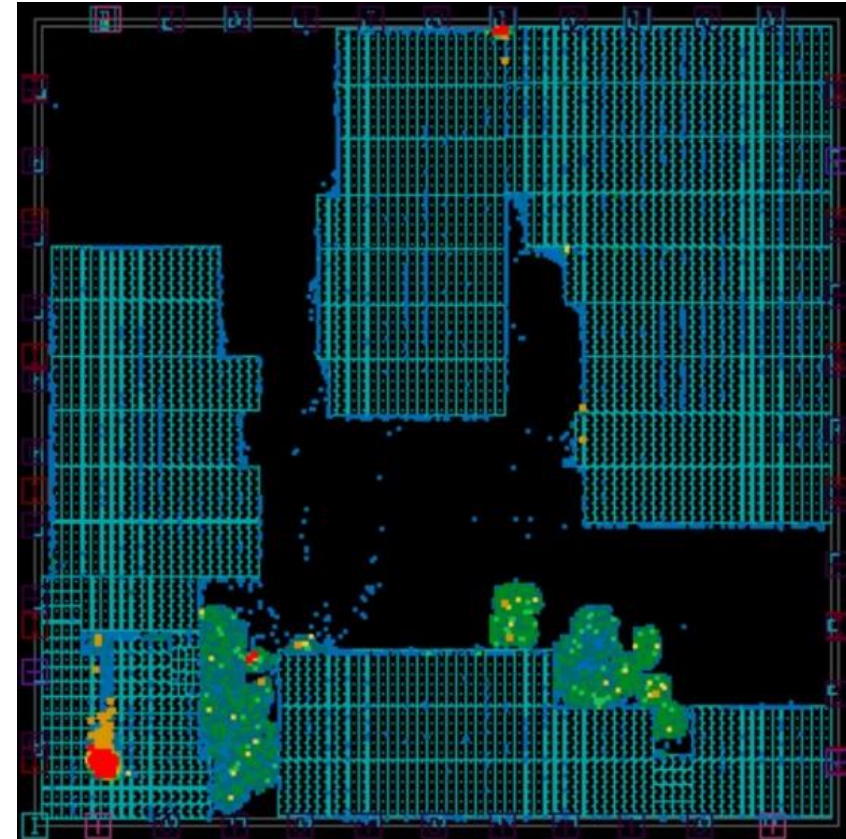
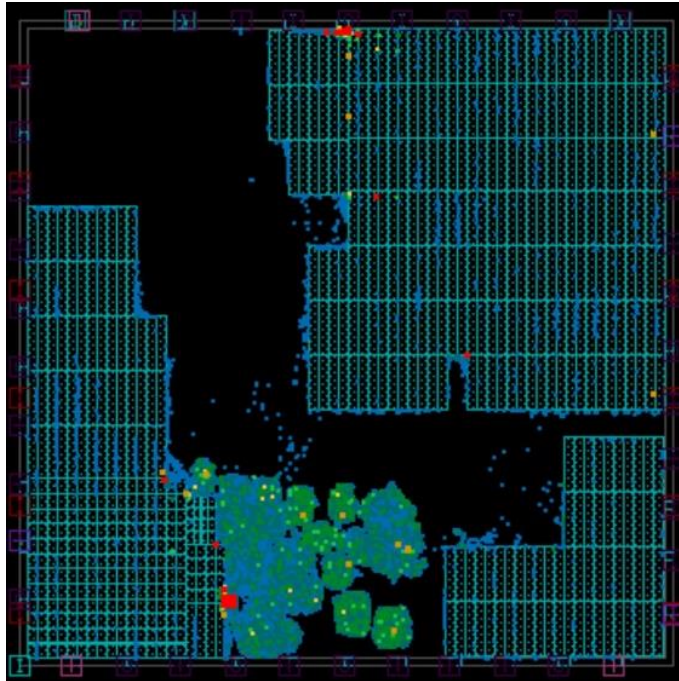
- Memory experiments conclusion

- Larger die size needed (53%), with die size package size can be decided early
- Relatively small impact on power (7%)
- We use 1 MB per SRAM module



# Memory Configs: Floorplan

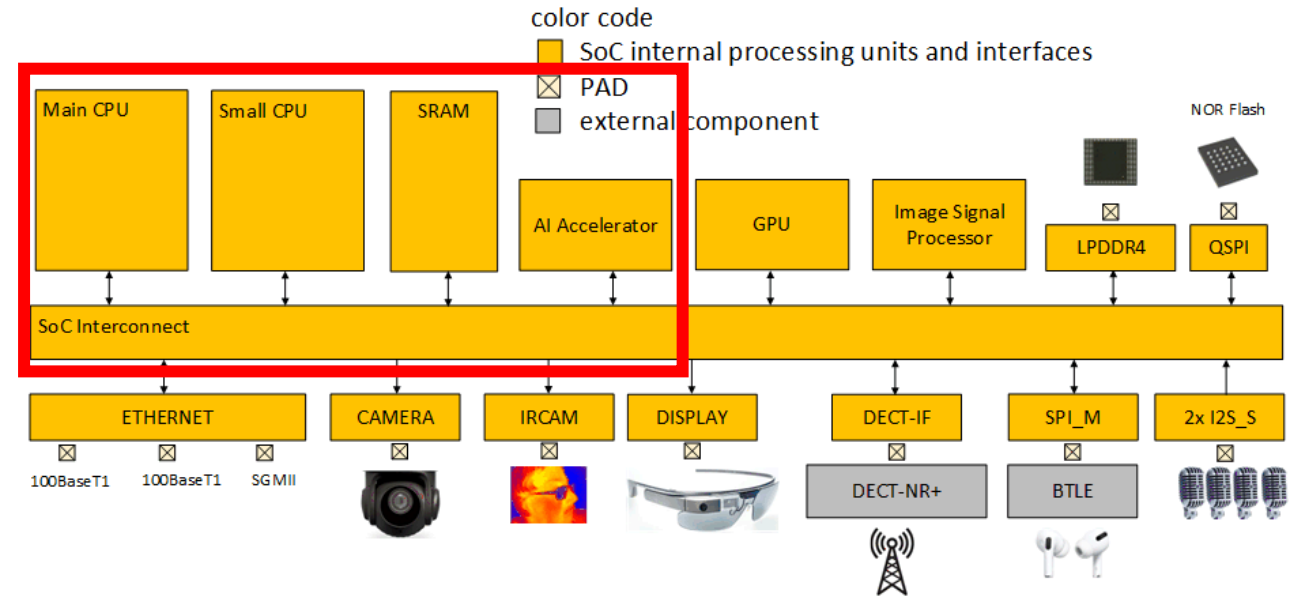
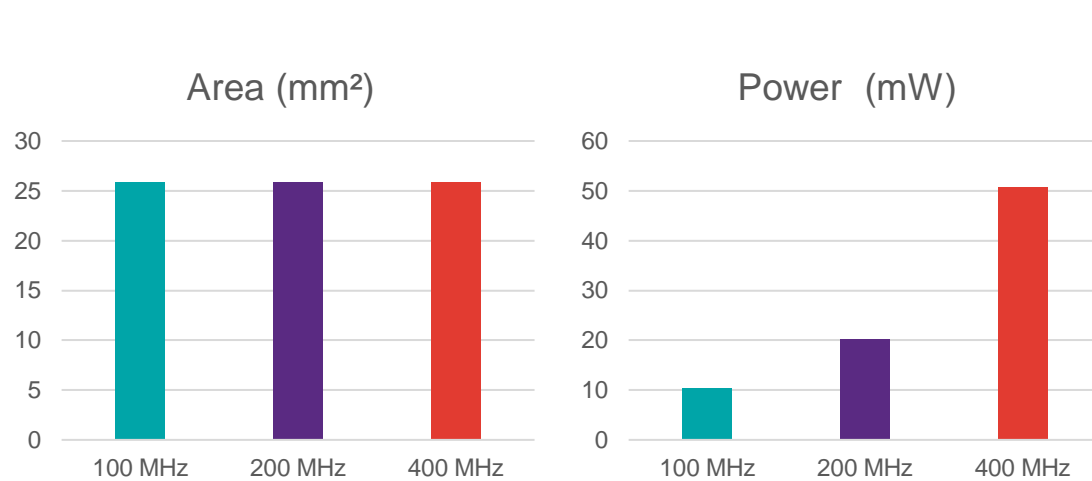
1 or 2 MB memory per SRAM module



- Observations
  - 53% larger die size needed
  - Memory consumes lots of area compared to logic

# Clock Frequencies: Area and Power

100, 200, and 400 MHz



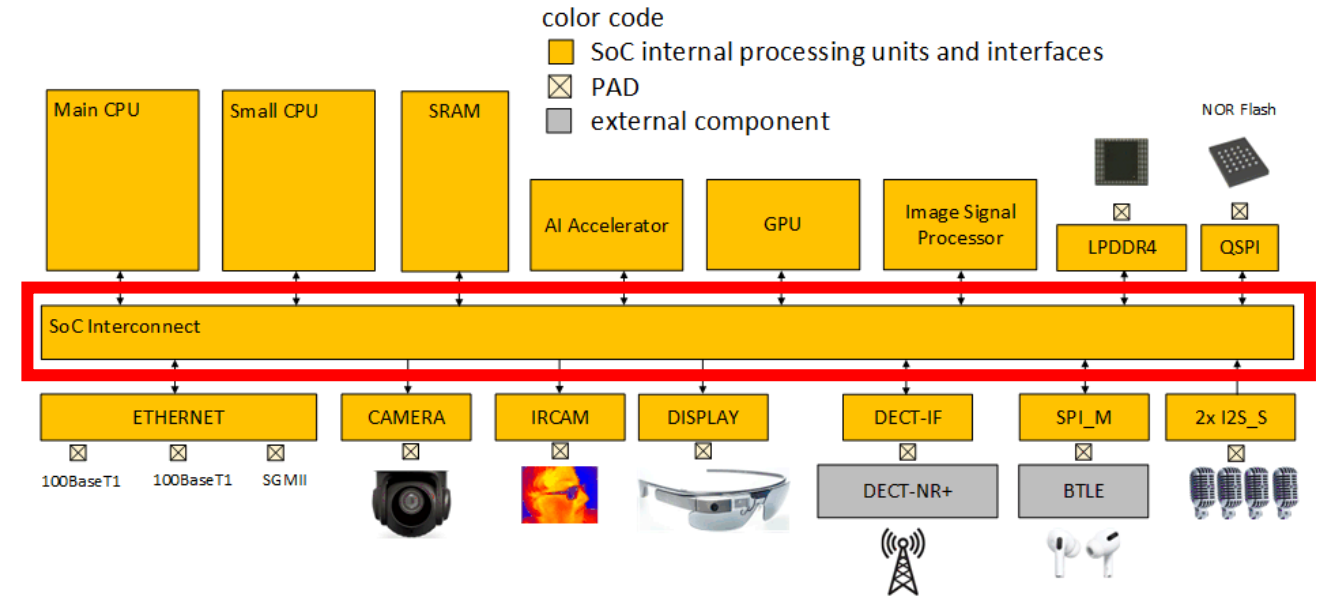
## • Clock experiments conclusion

- Clock speed has almost no impact on area
- Power increase from 200 to 400 MHz is 150%
- We use 400 MHz for now, will be optimized later

# Extra: Evaluation for SoC Interconnect

Can we achieve timing closure for entire interconnect?

- Current scenario
  - Connected: CPUs, AI accelerator, memory
  - Here timing closure is achieved
- Is timing closure achievable when all IPs are connected?
  - Run RTL Architect on interconnect module
  - Target frequency: 400 MHz
  - Compare timing reports
- Results for interconnect module
  - (old design) report\_timing: **0.39** ns slack
  - (new design) report\_timing: **0.46** ns slack
  - Conclusion: No further action required

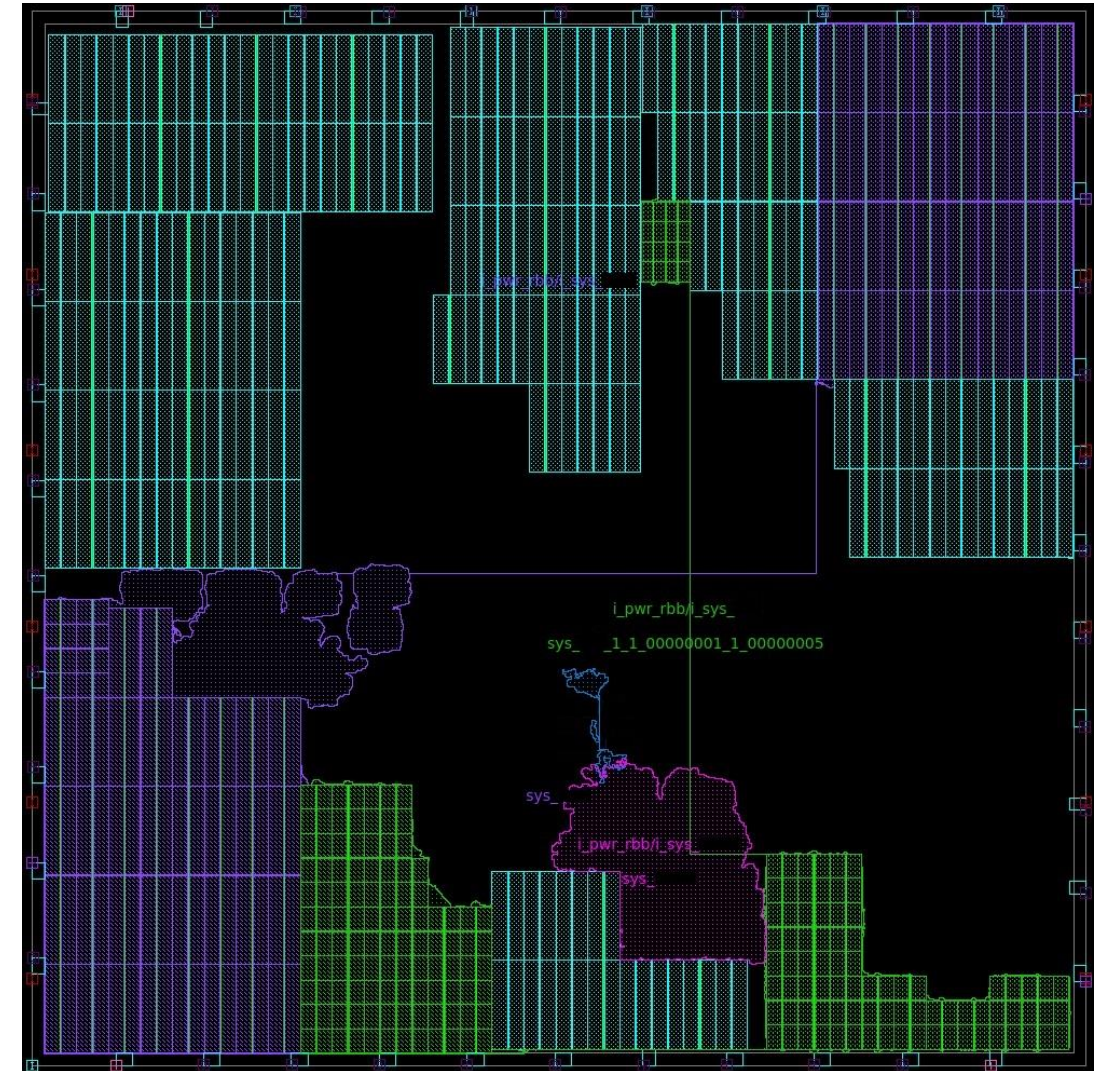


Inter-connect	Std. cells	FFs	Ports	Area
old	31889	6970	3257	24057
new	141424	26199	15841	101538



# Conclusion

- Key observations
  - Accelerator mainly impacts power
  - Memory uses large portion of area
  - Clock frequency range determined
- Preliminary decisions
  - 512 MACs for AI accelerator (small power cost)
  - 1 MB per SRAM module (high area cost)
  - 400 MHz clock frequency (high power cost)
- Advantages/challenges
  - Collaboration between RTL and physical design
  - Enables fast PPA studies
  - Resource consumption needs to be considered







***THANK YOU***

Our  
Technology,  
Your  
Innovation™