



# Heterogeneous Systems Design

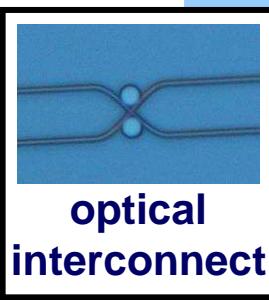
*F. Frantz, I. O'Connor*

# Research interests

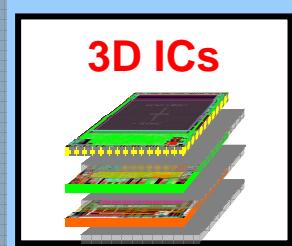
nanoprocessors with orders of magnitude improvements in

- # operations /  $\mu\text{m}^3 \cdot \text{s} \cdot \text{W}$
- flexibility
- fault tolerance

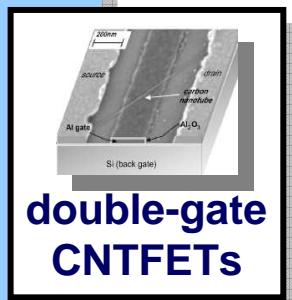
Emerging technologies in future computing architectures



optical interconnect



3D ICs

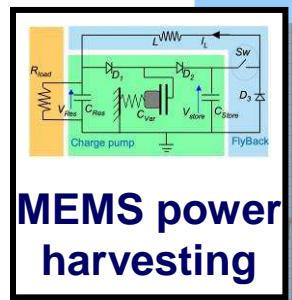


double-gate CNTFETs

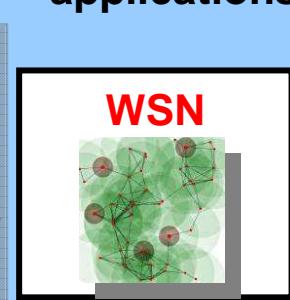
biomedical applications

- lab-on-chip
  - imaging for dosage monitoring
- wireless sensor networks
- energy harvesting and damping
  - energy-aware reconfiguration

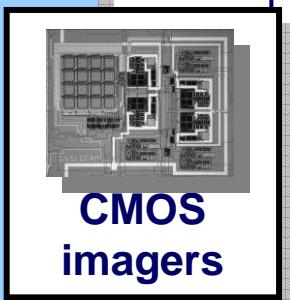
Smart sensors and systems for emerging applications



MEMS power harvesting

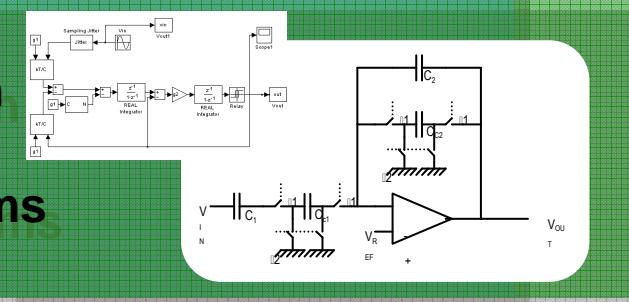
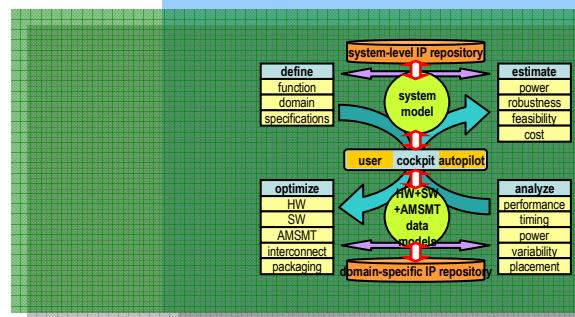


WSN



CMOS imagers

Modeling and design methods for heterogeneous systems



# **Problem statement**

---

- synthesis and partitioning of complex integrated 3D heterogeneous systems using various criteria (cost, performance, thermal distribution ...)

# Problem statement

- synthesis and partitioning of complex integrated 3D heterogeneous systems using

- automated design, using optimization, experience (DB), from specifications to sized system

- mapping sets of system functions to hardware groups (tiers), with limited interconnect and tier resources

- stacking chips in the Z-dimension
- increased functionality and complexity
- thermal and mechanical issues

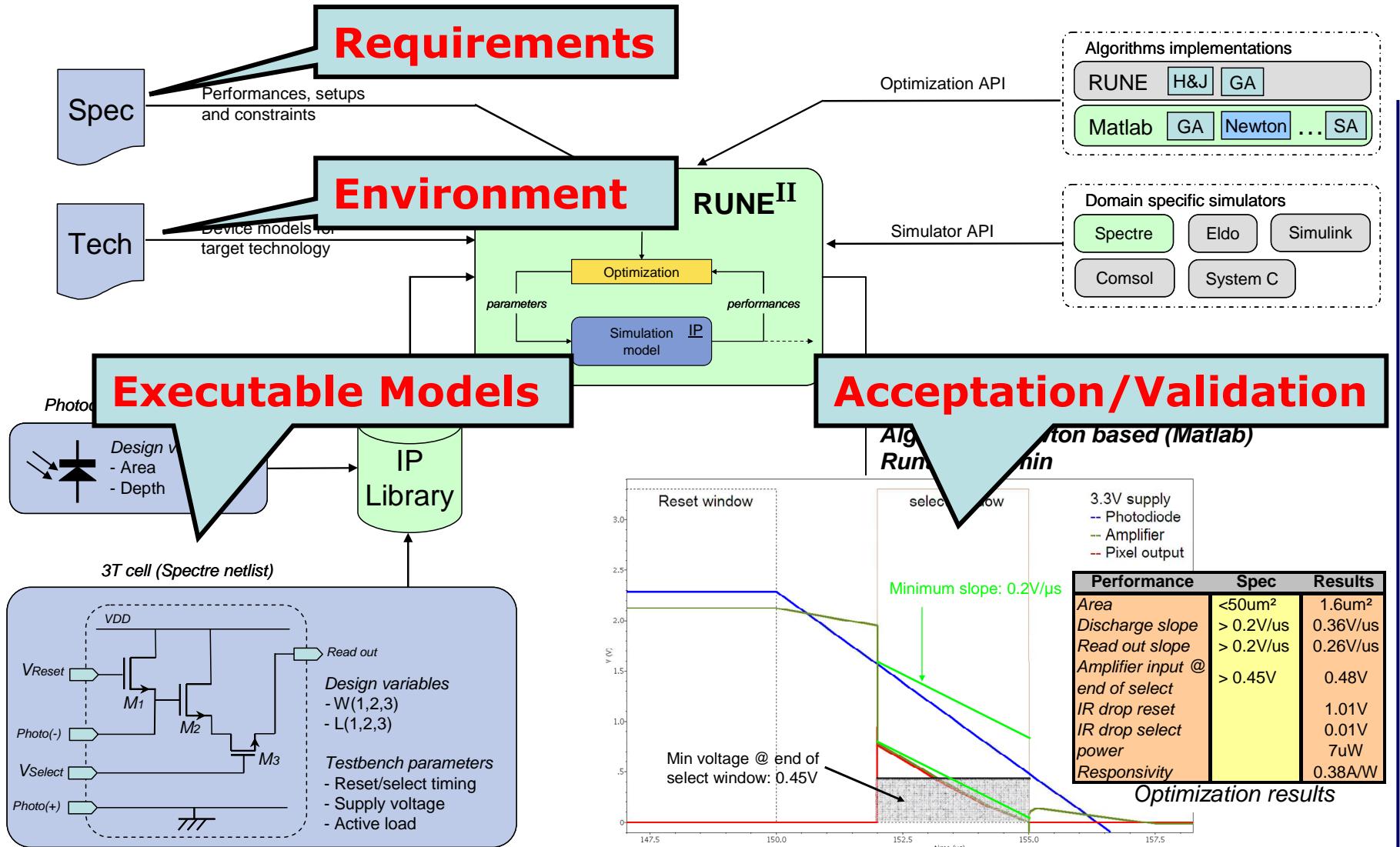
- physical heterogeneity (more than one physical domain)
- technology heterogeneity (more than one technology in stack)
- abstraction heterogeneity (more than one level of description)

# Problem statement

---

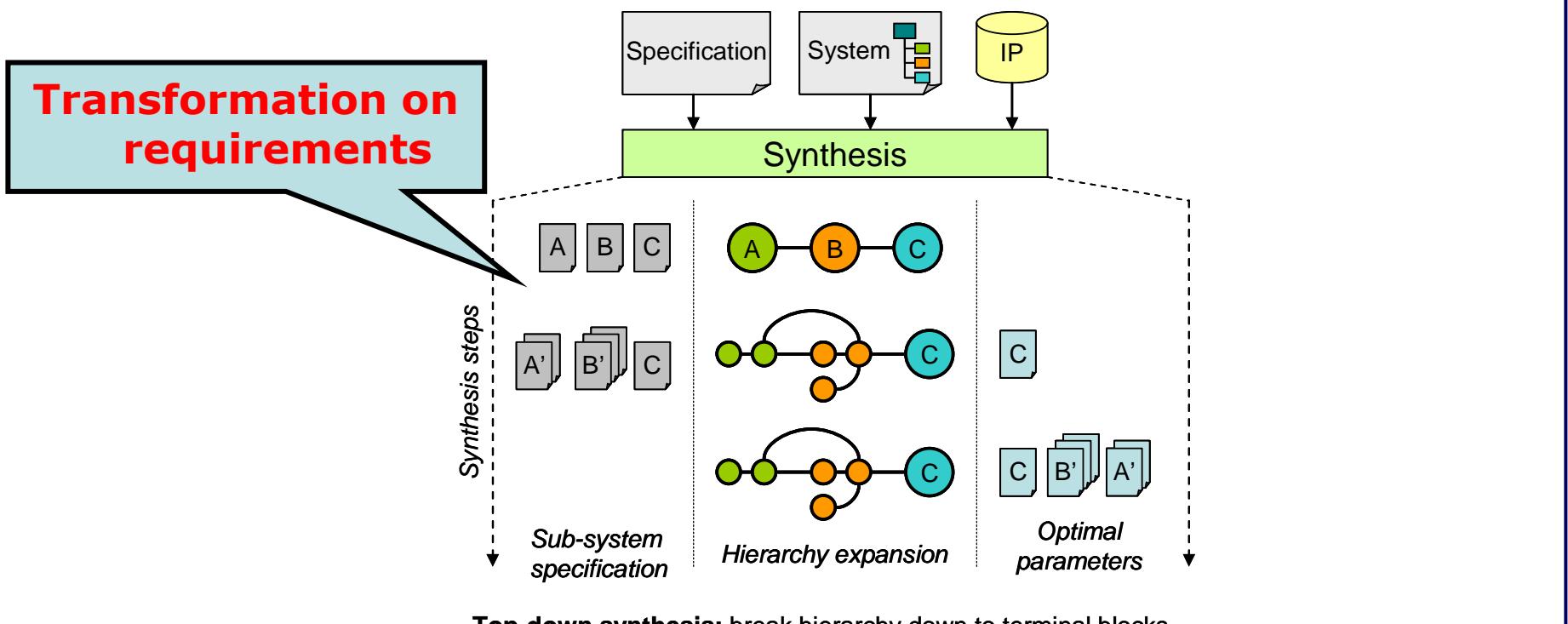
- synthesis and partitioning of complex integrated 3D heterogeneous systems using various criteria (cost, performance, thermal distribution ...)
- **fundamental issue 1:** how to efficiently represent the various multi-physics phenomena at abstraction levels suited to the complexity of the problem to be solved
- **fundamental issue 2:** how to model and optimize the synthesis process with many individual, interdependent but potentially parallelizable design tasks

# RUNE II : Specification driven optimization



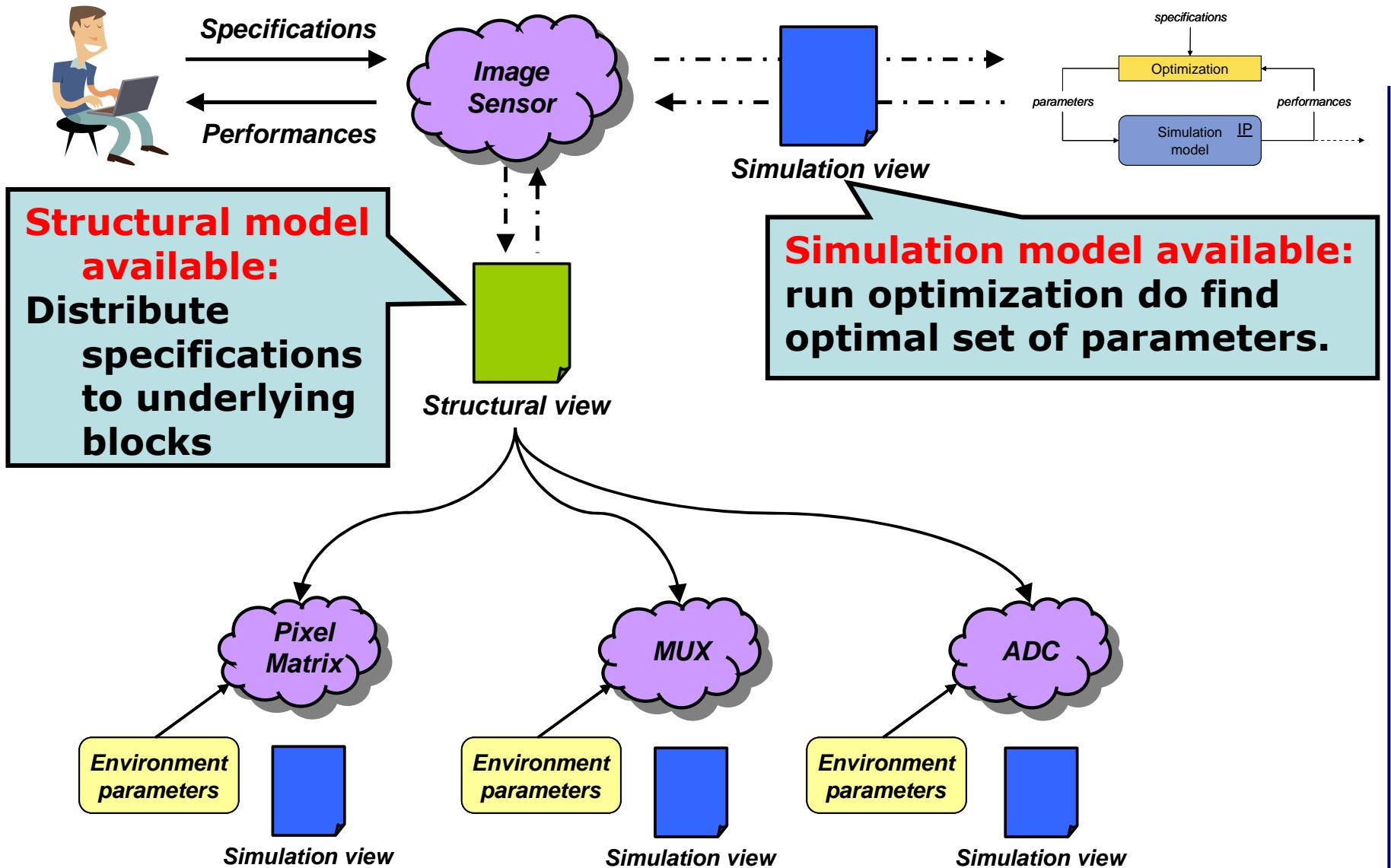
# Need for abstraction

- As synthesis progresses, hierarchy is expanded and requirements are refined.

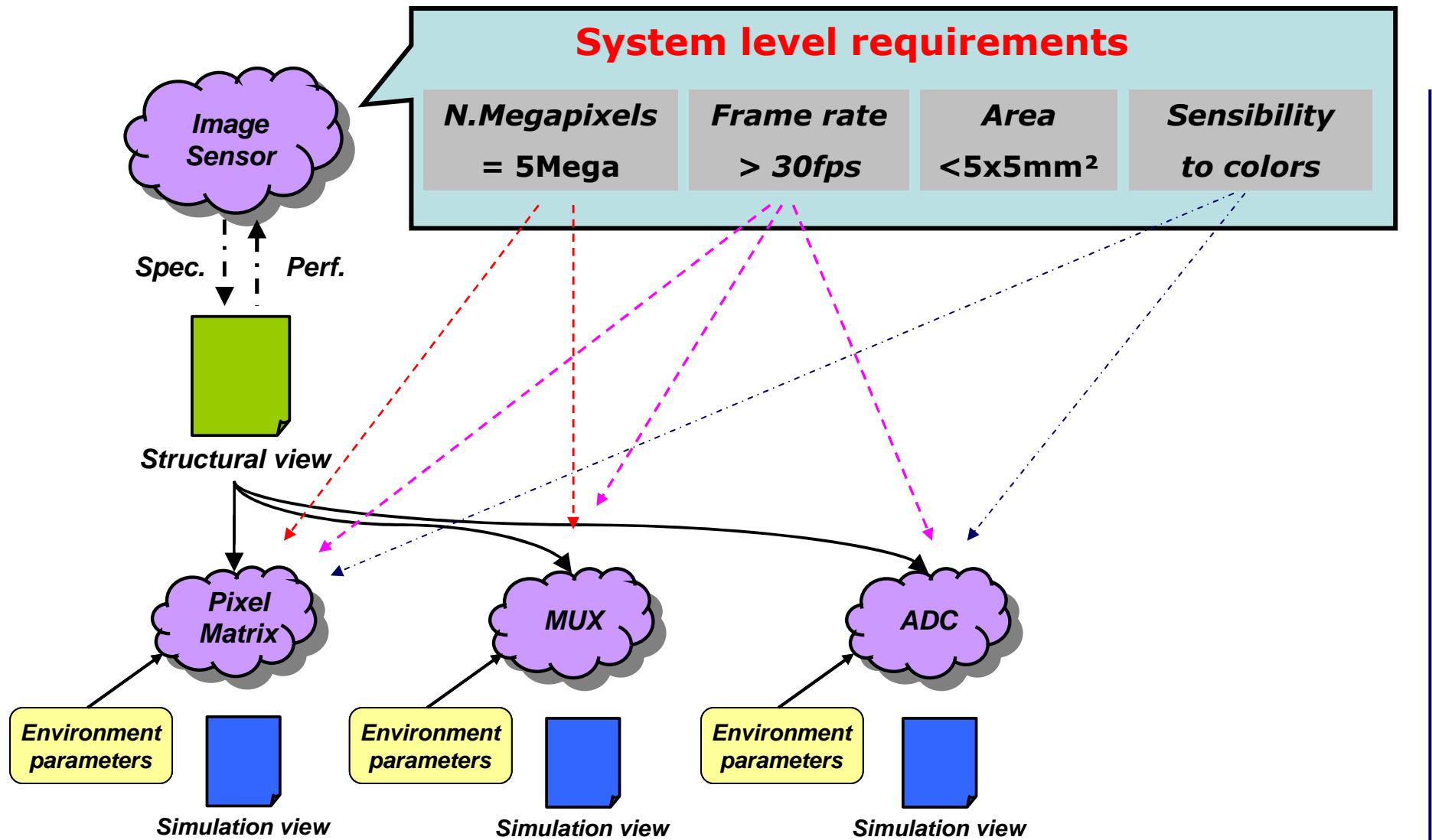


- BUT this is a priori and therefore done in the presence of uncertainty
  - of the impact of specifications (e.g. speed on power consumption)
  - of the impact of environment (technology, temperature, variability)

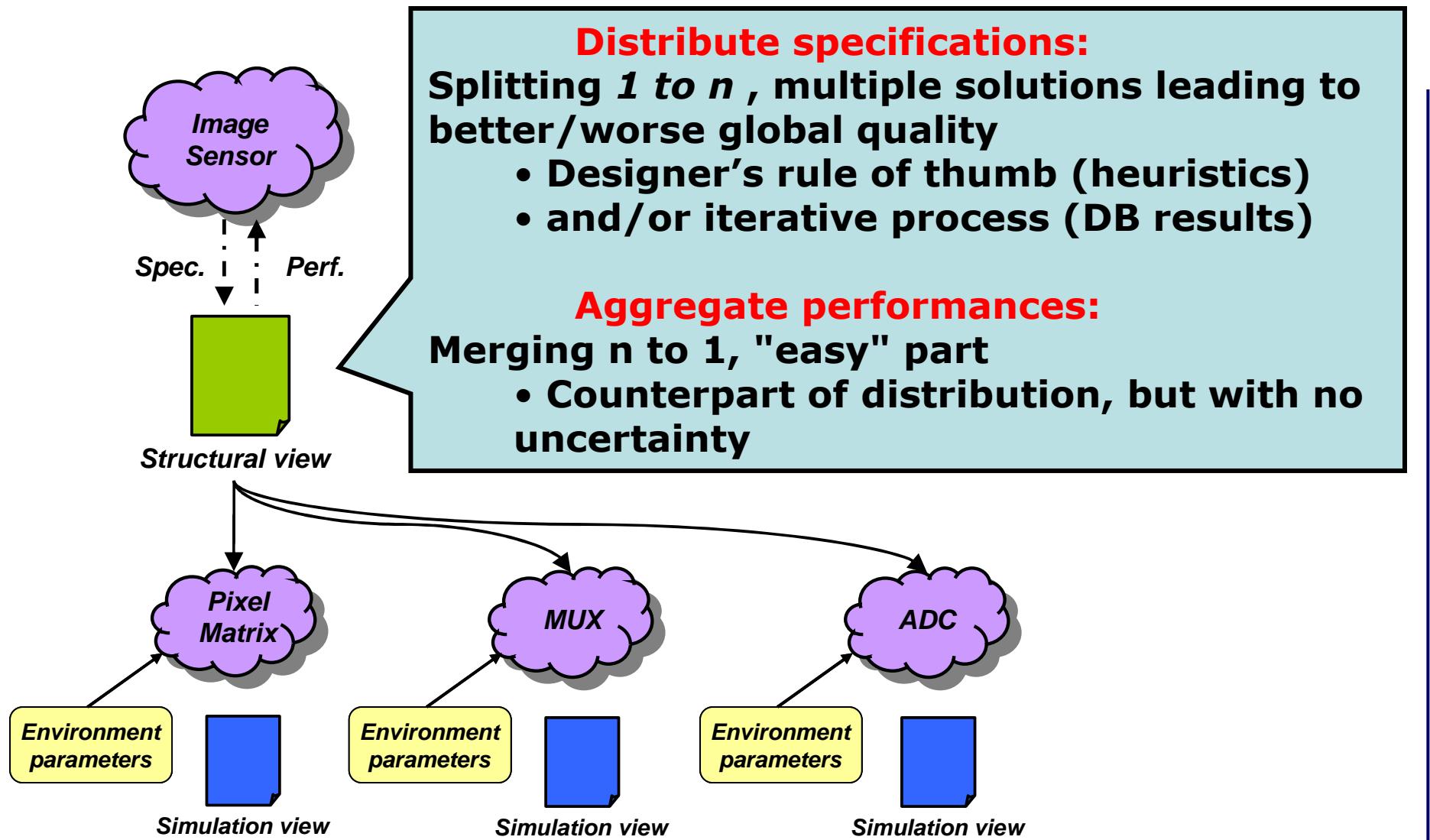
# Synthesis, simulation and models



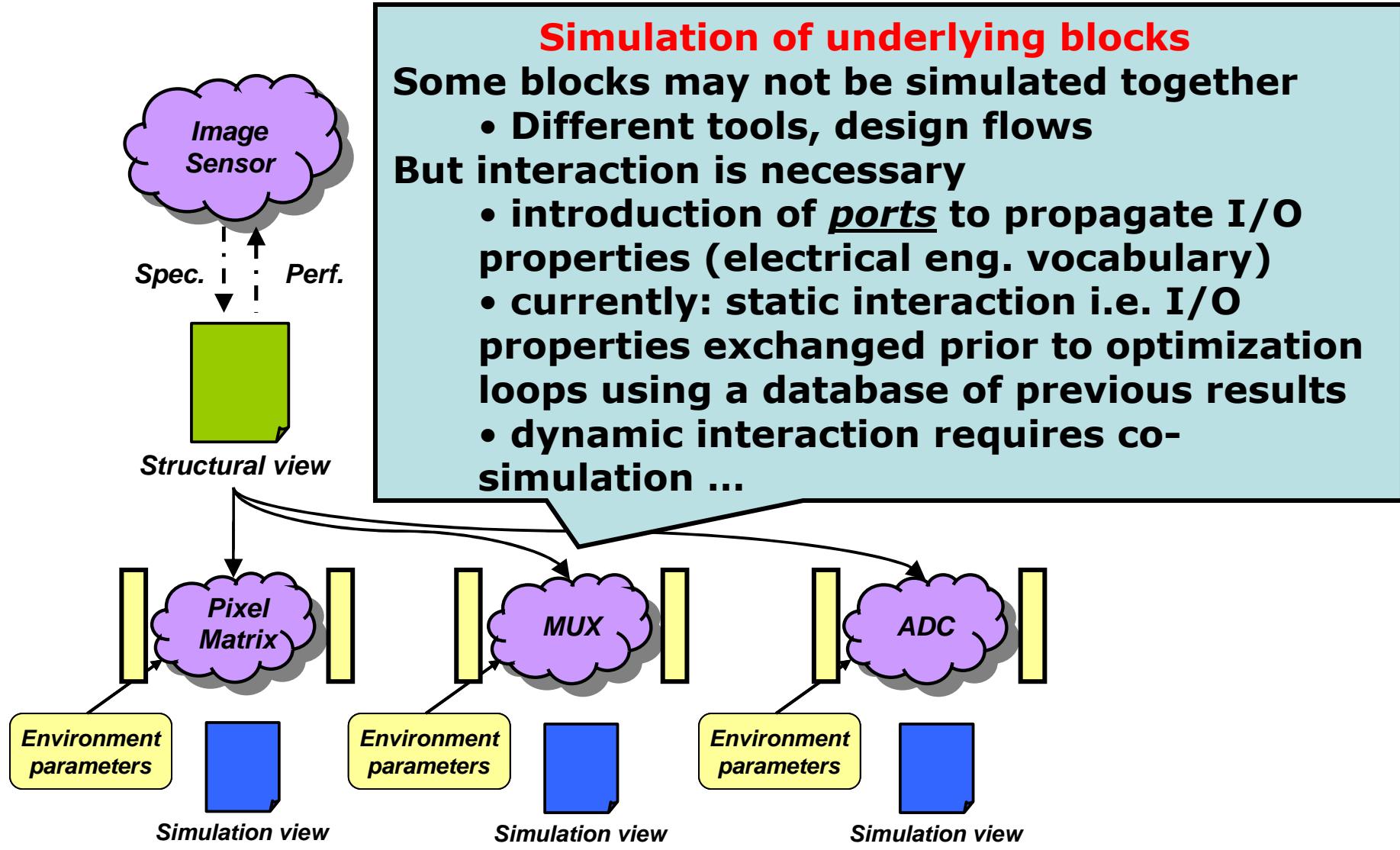
# Synthesis, simulation and models (2)



# Synthesis, simulation and models (3)



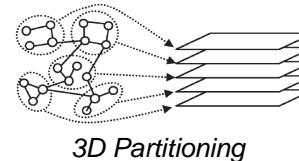
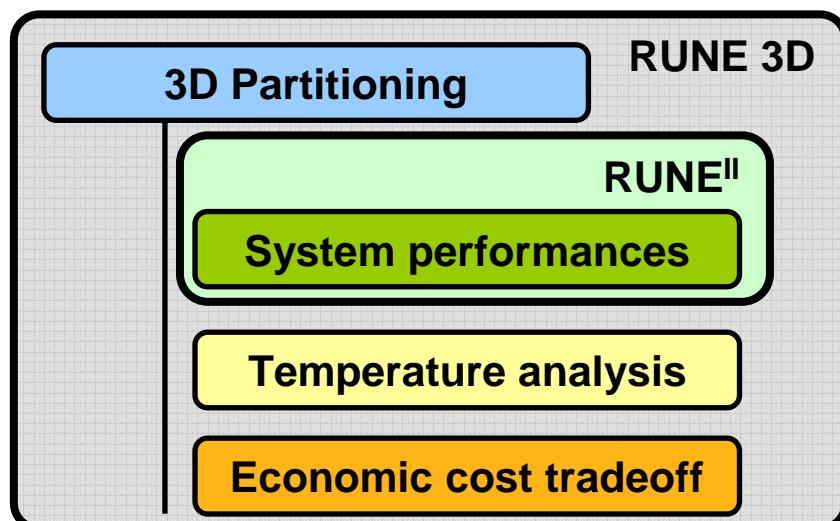
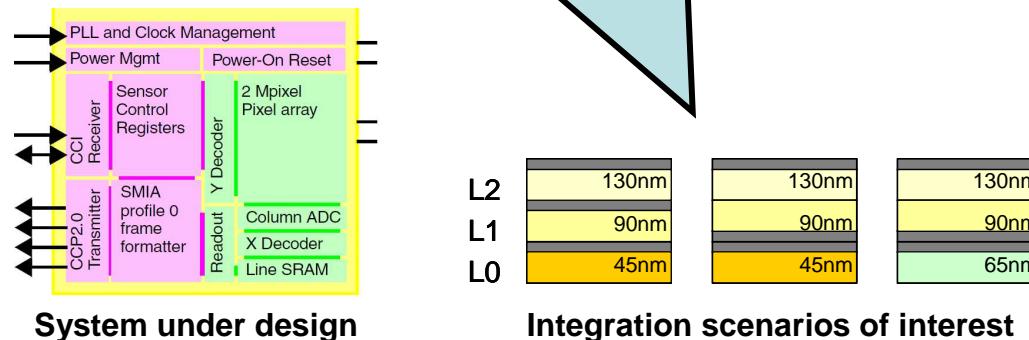
# Synthesis, simulation and models (4)



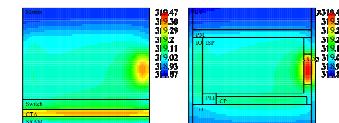
# RUNE 3D – What changes from 2D?

- Exploring the 3D more dimensions:
  - One system, several integration scenarios**

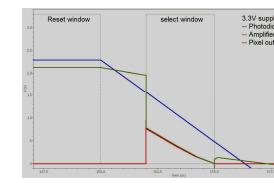
Cases set by the user, but impact is evaluated automatically



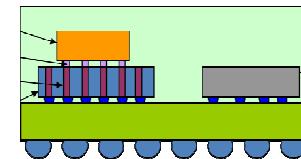
3D Partitioning



Thermal simulation



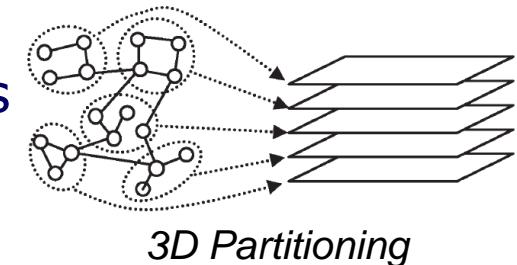
Performance evaluation



Economic cost

# 3D Partitioning

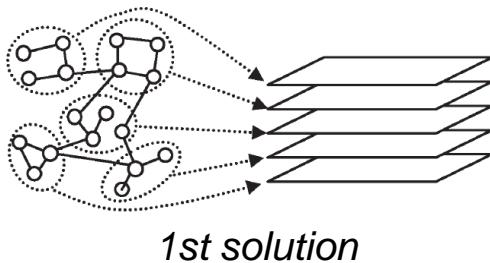
- Assign technologies, orientation, and block to each tier
  - Minimize production cost (\$), footprint (area), power
  - Maximize performances (ex.: BW)
  - Respect constraints (ex.:  $T^\circ$ ) and bounds
  - Interlayer model selection
    - TSV, TPV, Wire Bounding, etc
- Floorplan like optimization problem with:



$$\text{Objective} = \$\text{(integration)} + \$\text{(dies)} + A_{\max} + T^\circ_{\max} + P_{\text{total}} + BW_{\text{slack}}$$

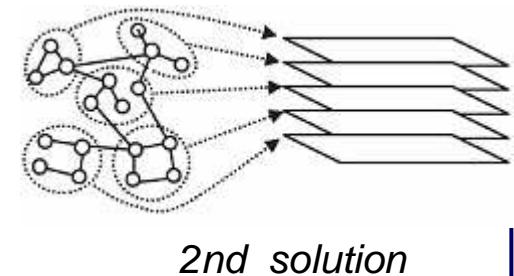
| Objective function factor   | Required information                        |
|-----------------------------|---------------------------------------------|
| <i>Integration cost</i>     | Layer orientation, integration technology   |
| <i>Die cost</i>             | Layer area (floorplan), technology node     |
| <i>Max Area (footprint)</i> | Layer area (floorplan)                      |
| <i>Temperature</i>          | Floorplan, power per block                  |
| <i>Total power</i>          | Power per block, technology scaling factor  |
| <i>Performance (BW)</i>     | Model, Temperature, wire length (floorplan) |

# 3D partitioning – Constraining design

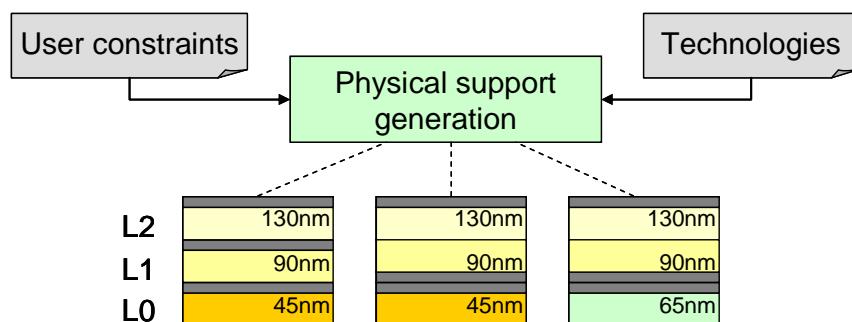


**Local modifications:**  
operations on blocks (swap, rotate, resize)

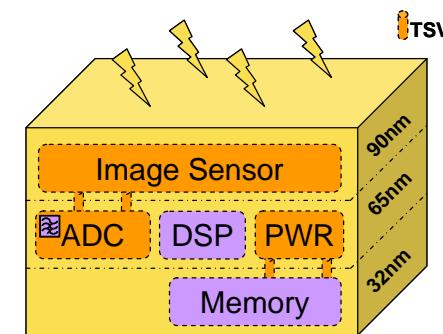
**Global modifications:**  
operations on layers (flip, change technology)



- Large number of degrees of freedom
  - **User** knowledge to constraint design space (realistic choices)
  - Run parallel problems with **global choices fixed**
    - Layer orientation, layer technology node



**Physical support:** define orientation, order and technology node of layers prior to partitioning optimization



**Location constraint:** pixel matrix on top!

# 3D partitioning – Optimization loop

- Move / Sizing
- Evaluate

**Input:**

- Graph

**Output:**

- Area, position
- Wirelength
- Wireload (+TSV)

**Parameters:**

- Material properties

**Input:**

- Block position
- Block power

**Output:**

- $T^\circ(x,y,z)$ ,  $T^\circ_{\text{max}}$

**Input:**

- Wireload (+TSV)
- Block  $BW$

**Output:**

- OK/NOK
- How far?

**Parameters**

- Package
- Technology nodes

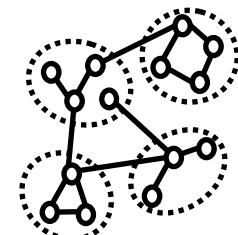
**Input:**

- Wafer number
- Area/layer
- Nature/layer

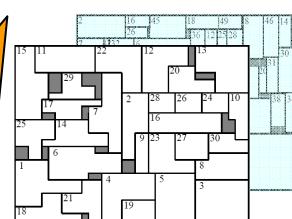
**Output:**

- \$

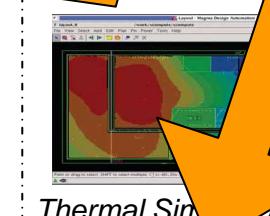
$$f(A, \$, T^\circ, P_{\text{total}}, BW) = f(A, \$, T^\circ_{\text{max}}, P_{\text{total}})$$



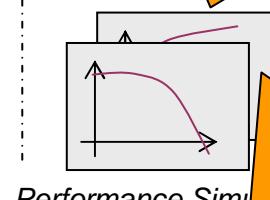
Partitioning candidate solution



Floorplan packing



Thermal Simulation



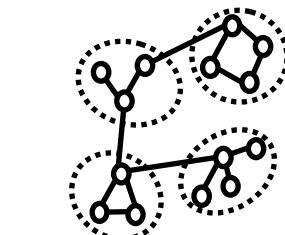
Performance Simulation

$$\text{Objective function} \\ f(A, \$, T^\circ, P_{\text{total}}, BW)$$

Economic cost (\$)

Move blocks

Evaluate objective function



Partitioning candidate solution

Move blocks

# In a nutshell

---

- Executable models exist:
  - Different abstraction levels
  - Different languages
  - No "model generation" (code generation)
- Experience is available in the form of:
  - Database of previous results (Response Surface Model)
  - Designer knowledge (rule of thumb)
- Objective: manage complexity
  - Design space exploration runtime
  - Model previous experience (how to use it?) to help to converge to a good quality solution
- Problem: organize how/when information is used and updated (dependency) during the requirement refinement