

# Exploring Optimal Cost-Performance Designs for Raw Microprocessors

---

Csaba Andras Moritz

Donald Yeung

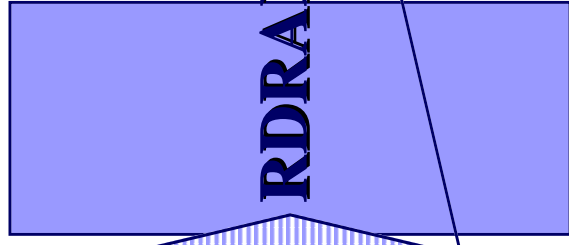
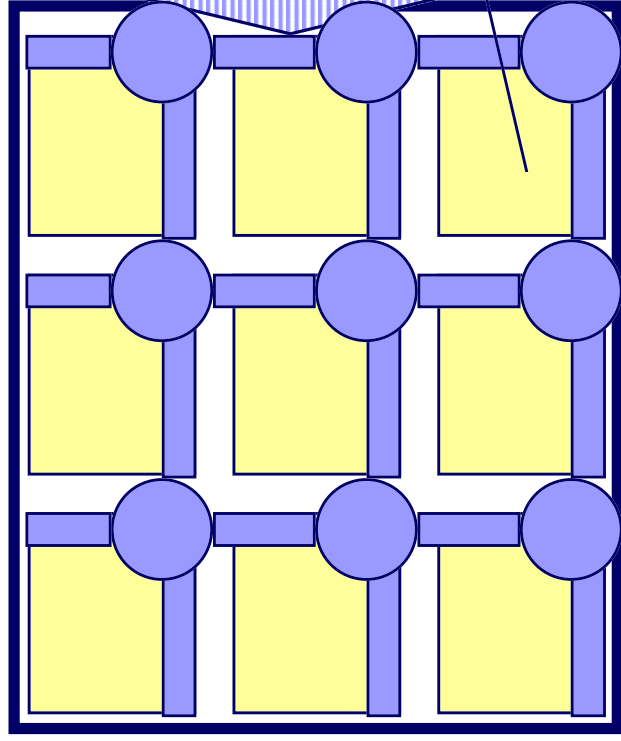
Anant Agarwal

Laboratory for Computer Science  
Massachusetts Institute of Technology  
{andras,piano,agarwal}@lcs.mit.edu

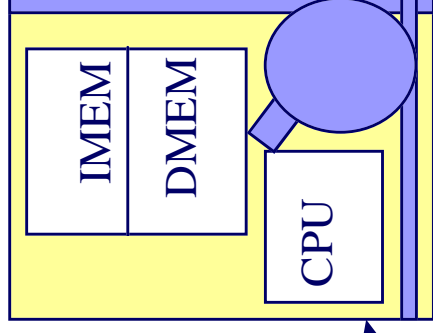
# Raw microprocessor

---

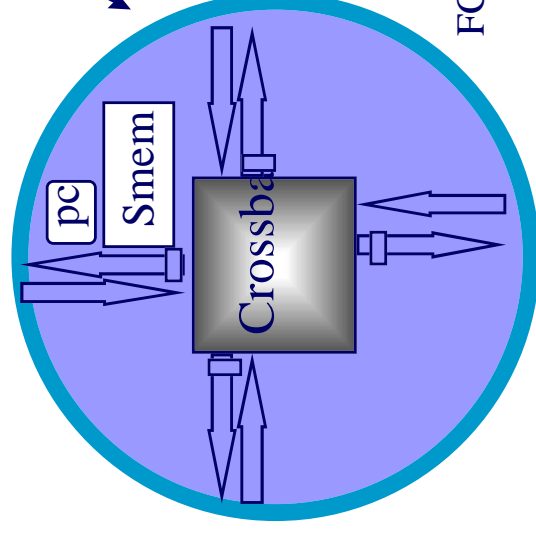
## RAW microprocessor



## Tile



## Switch



# Open challenge: grain & balance

- ◆ Determine the area of each tile: grain size
- ◆ Determine the proportion of area or balance between:
  - memory
  - processing
  - communication
  - chip I/O

# Motivation

---

- ◆ Design Raw systems with optimal performance per unit cost.
  - Analytical framework: optimize grain & balance
- ◆ General Methodology: can be applied for other type of systems, e.g. FPGA devices

# Overview results

---

- ◆ What Raw system should we build given 1 billion transistor area?
- ◆ Our intuition:
  - 1000 tiles
  - single-issue processors
  - 3 words/cycle local com. bandwidth
  - 20 Kbyte memory per tile
  - 30 words/cycle I/O bandwidth per chip

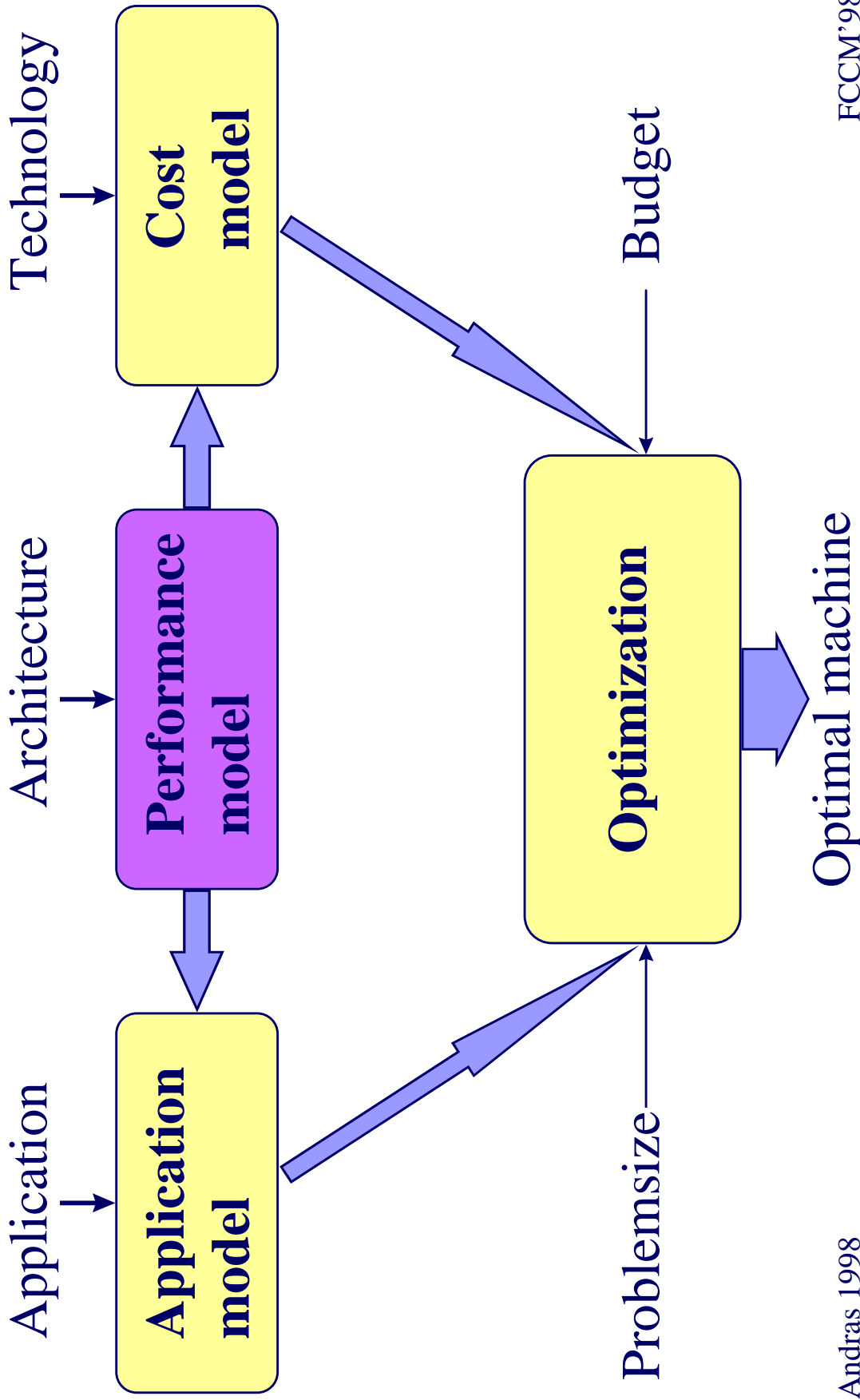
# Outline of presentation

---

- ◆ Analytical framework
- ◆ Models
  - Application model
  - Raw performance model
  - VLSI cost model
- ◆ Optimization problem
- ◆ Example: Jacobi relaxation
- ◆ Conclusions: chip areas, configurations

# Analytical framework

---

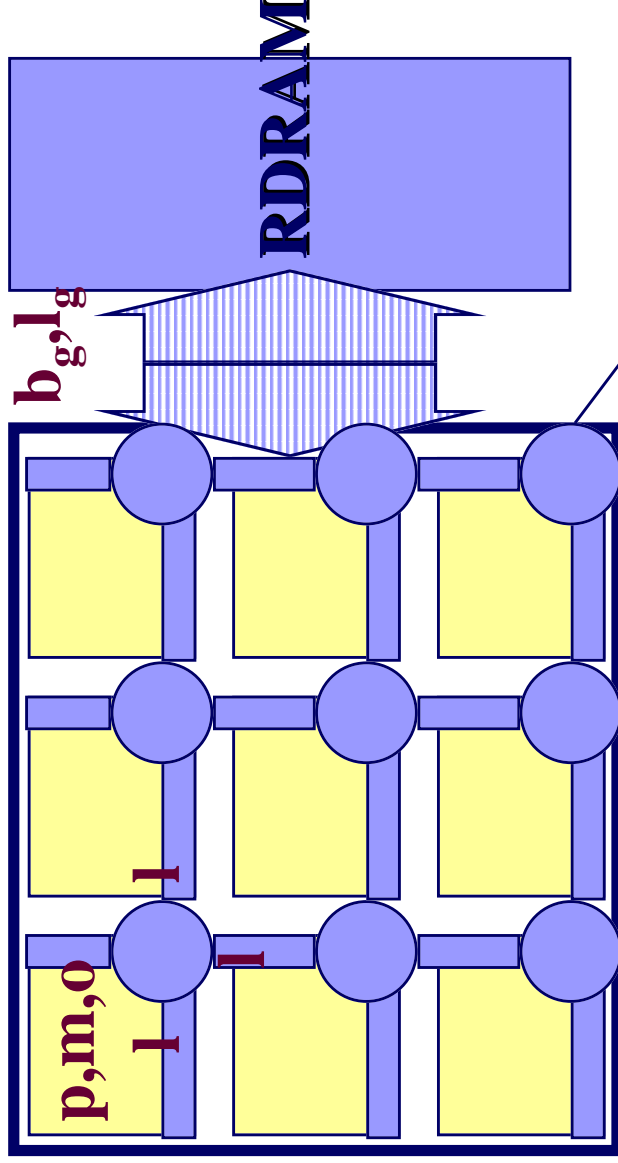


# Performance model

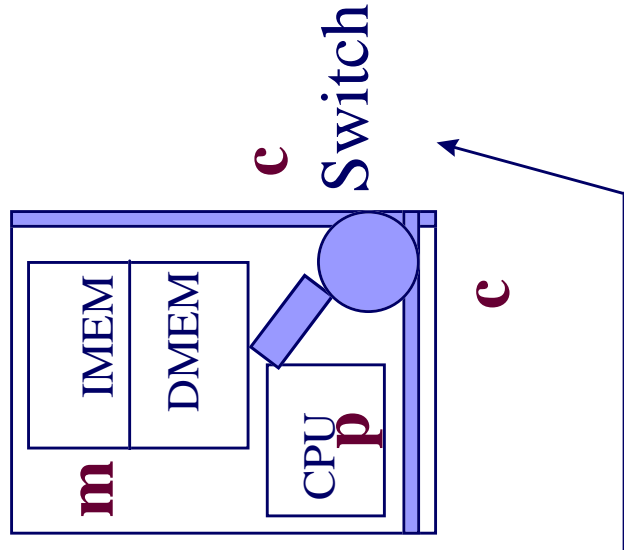
$$P, p, m, c, o, l, b_g, l_g$$

## P tile RAW chip

global communication



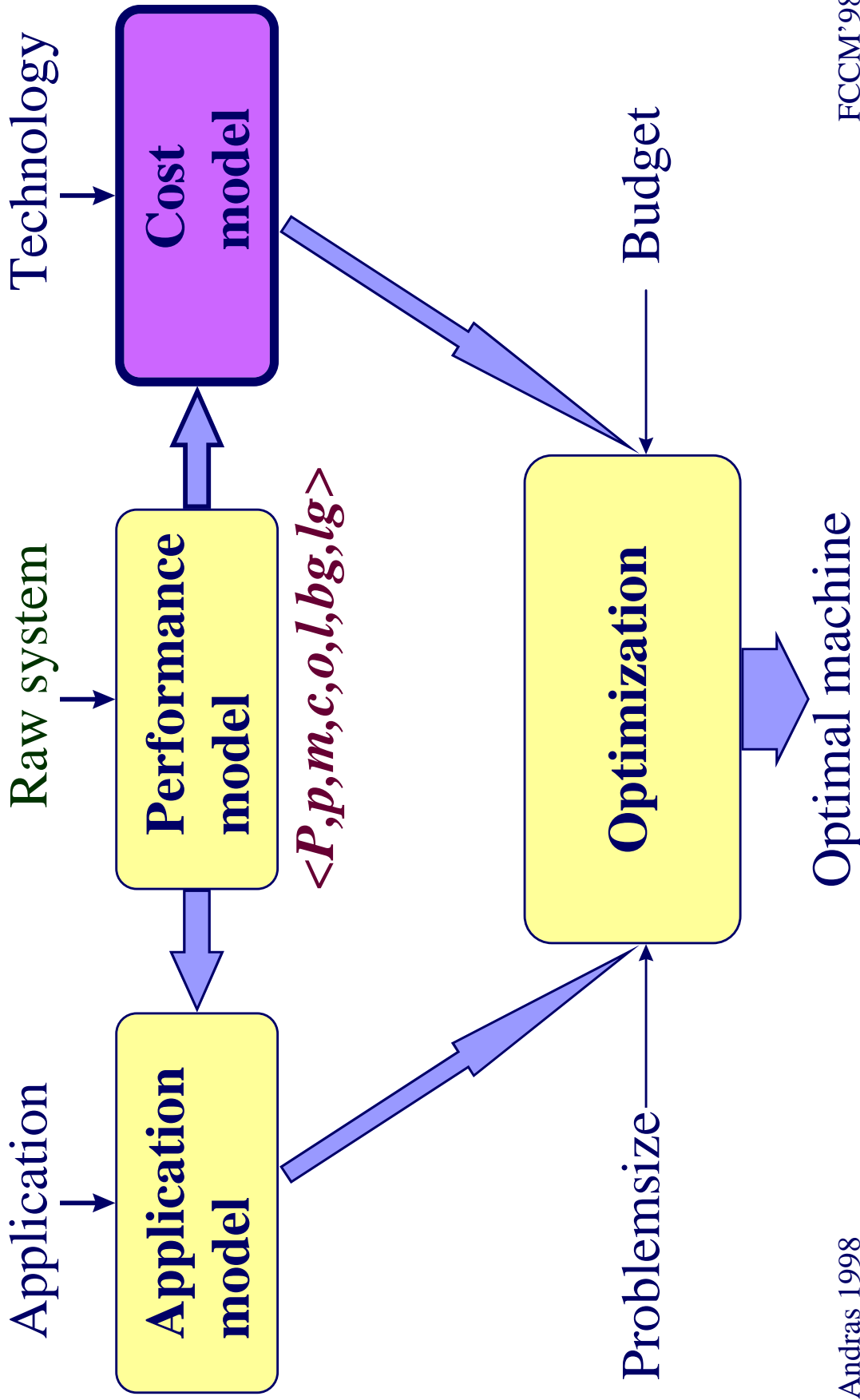
## Tile



a tile or processor with power  $p$   
memory size  $m$  local communication  $c$

# Analytical framework

---



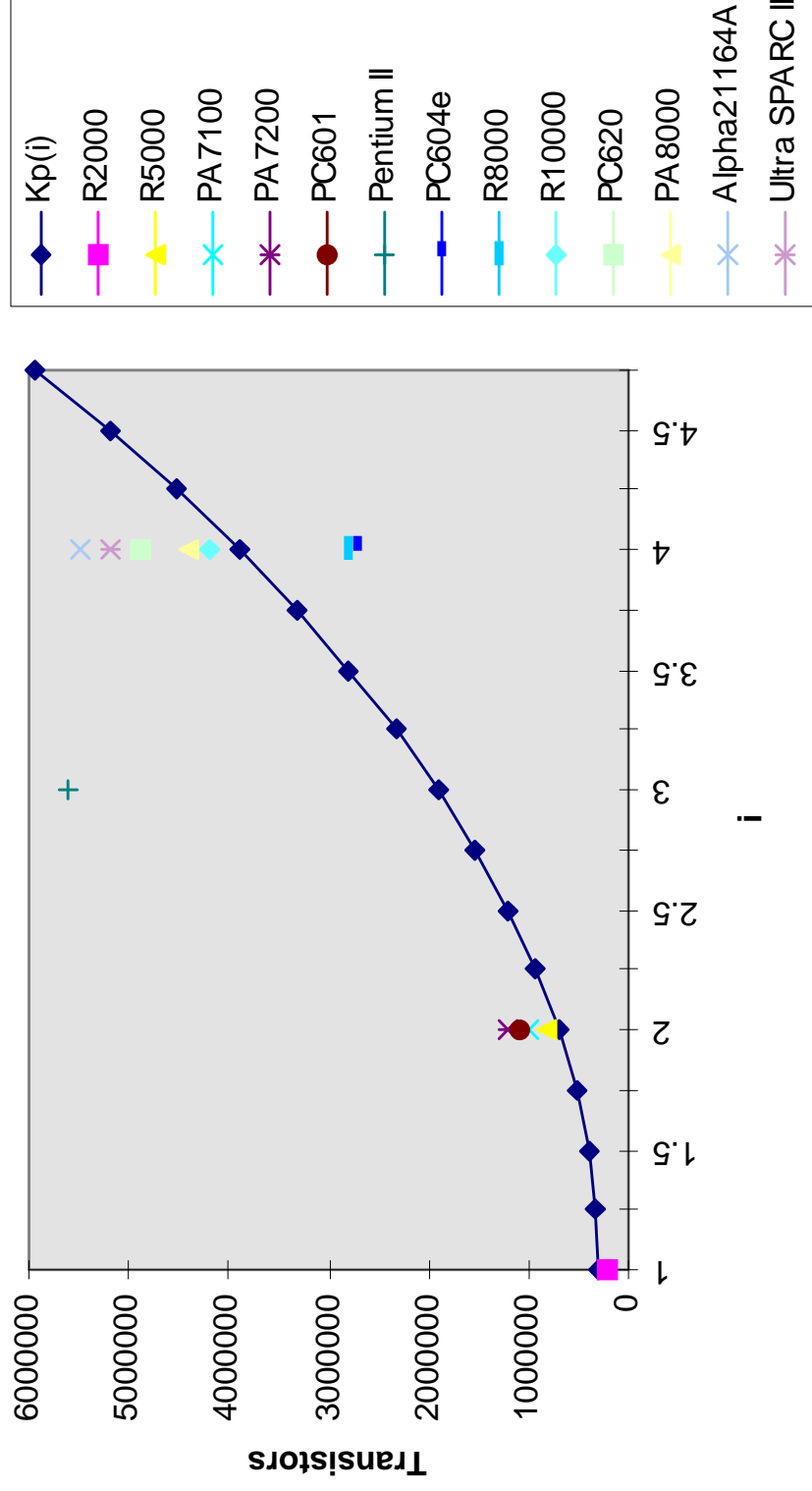
# Cost model

---

- ◆ Silicon area expressed in SRAM bits
- ◆ Empirical study superscalars, routers,...
- Processor cost model:
  - » area dedicated processing in function of  $p$
- Switch cost model:
  - » switch area in function of local com. bandwidth
- Memory cost model:
  - » memory areas depending on size&type
- Chip I/O cost model: cost of pins
  - » cost of providing chip I/O bandwidth

# Processor cost model

---

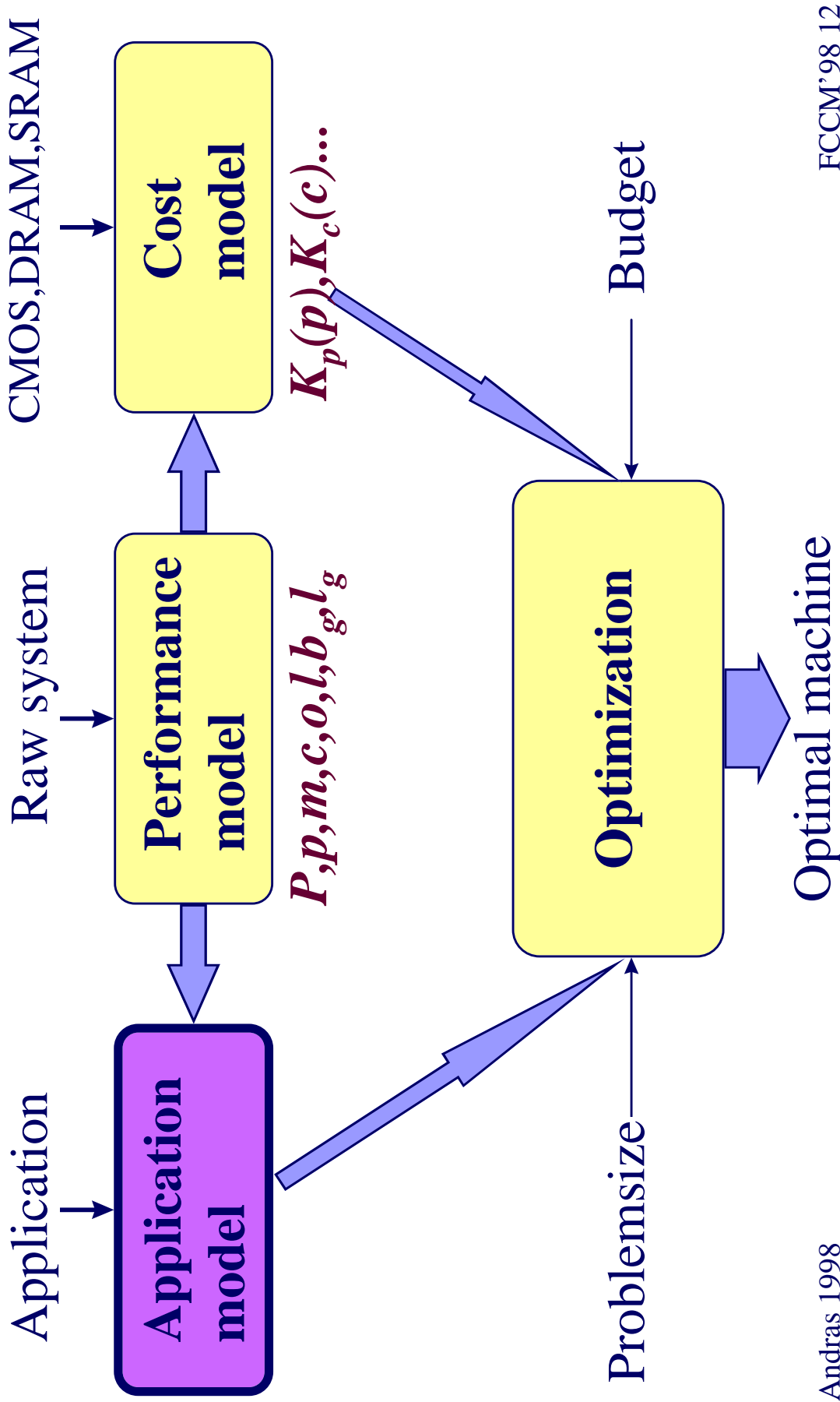


Processor cost function  $K_p(i)$  and cost of logic areas (no cache),

$i$  = issue slots,  $p = \sqrt{i}$

# Analytical framework

---



# Application model

---

- ◆ Subproblems
  - loaded/stored from external RDRAM
  - all subproblems are visited in sequence
- ◆ Requirements :  $R_i, i$  in  $\{p, m, c, o, l, b, g, lg\}$ 
  - Each application has specific requirements of processing, communication, memory etc.
- ◆ Run-time:  $T = \max(T_p, T_c, T_g)$
-

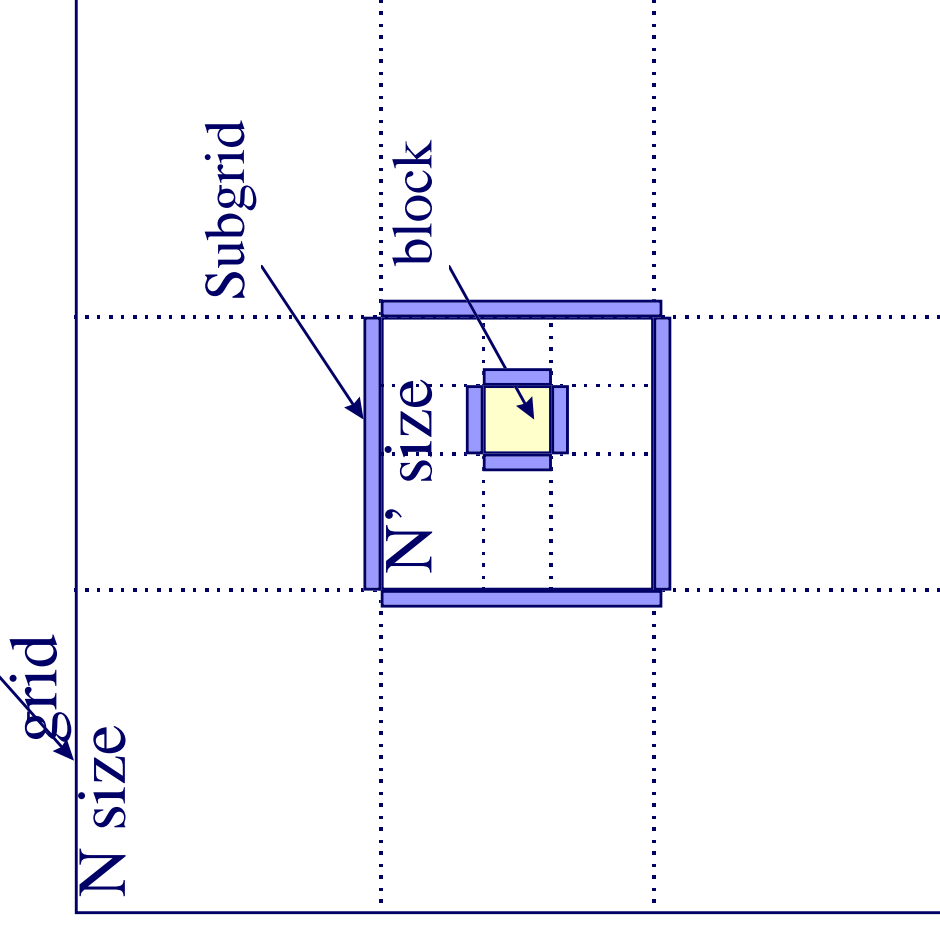
# Jacobi Relaxation 2D

---

- ◆ Iterative algorithm
- ◆ Each step replaces the value at each node of a grid with the average of the values of its nearest neighbors

# Jacobi 2D

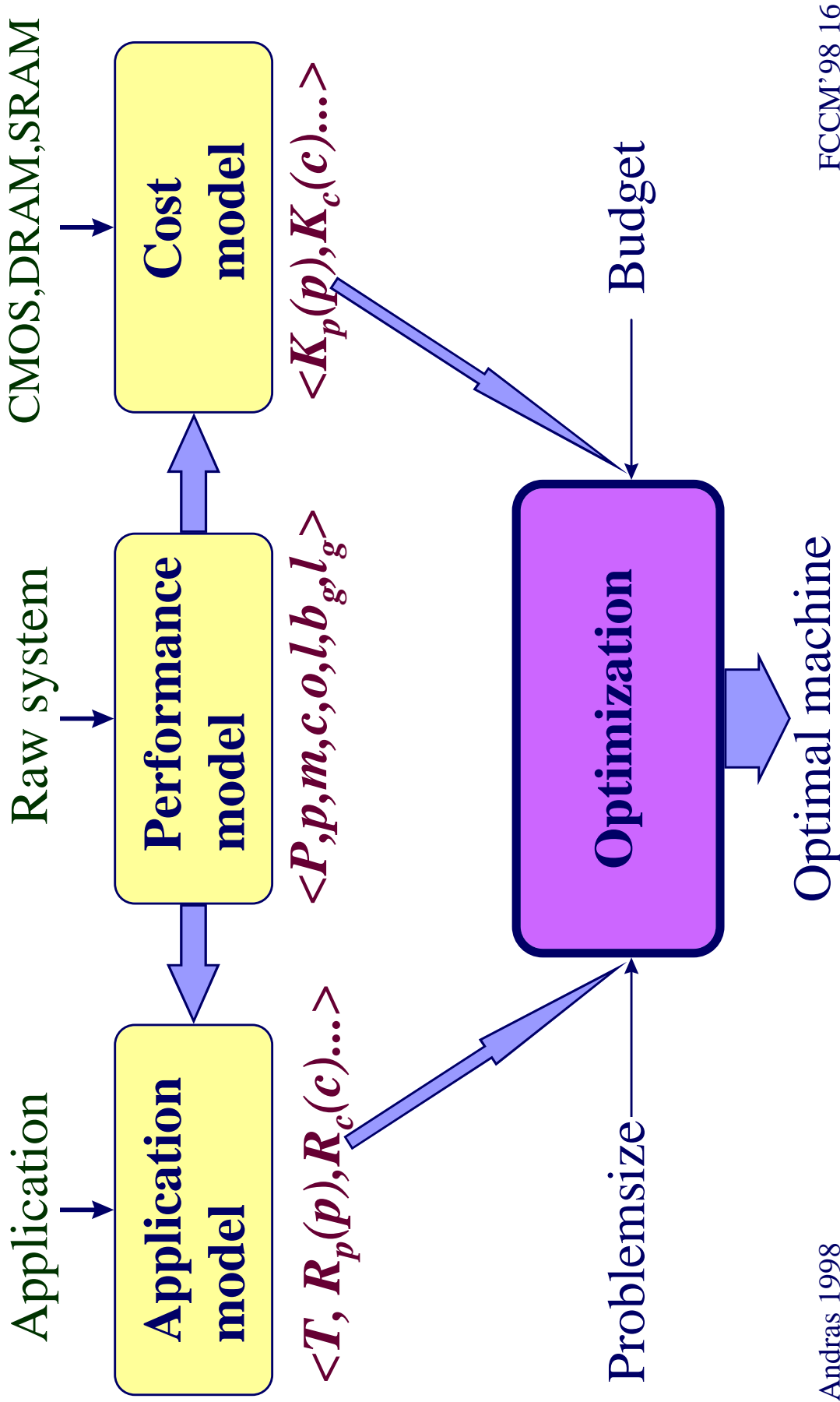
---



- ◆ Grid partitioned in subgr
- ◆ Blocking algorithm
  - Each block executed on a tile
- ◆  $R_p = 4 N^3 / P$  for  $N^2$  iterations,  $T_p = R_p / p$ .
- ◆  $R_{bg} = 2 N ( N^2 / \text{sqrt}(N) )$
- ◆  $R_{lg} = R_{bg} / N'$
- ◆  $R_c = N^2 8 \text{sqrt}(N' / P) N / N'$

# Analytical framework

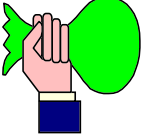
---



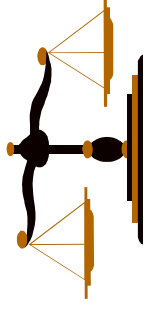
# Optimization Problem

---



- Is a nonlinear constrained based optimization:
- ◆ Given:  $B$  budget and an application (Jacobi, matrix-x,..)
  - ◆ Objective: find machine  $\langle P, p, m, c, o, l, b_g, l_g \rangle$  and subproblem that minimizes run-time for the application
  - ◆ Constraints:
    - $B \geq K$  (cost) 
    - Balance statements

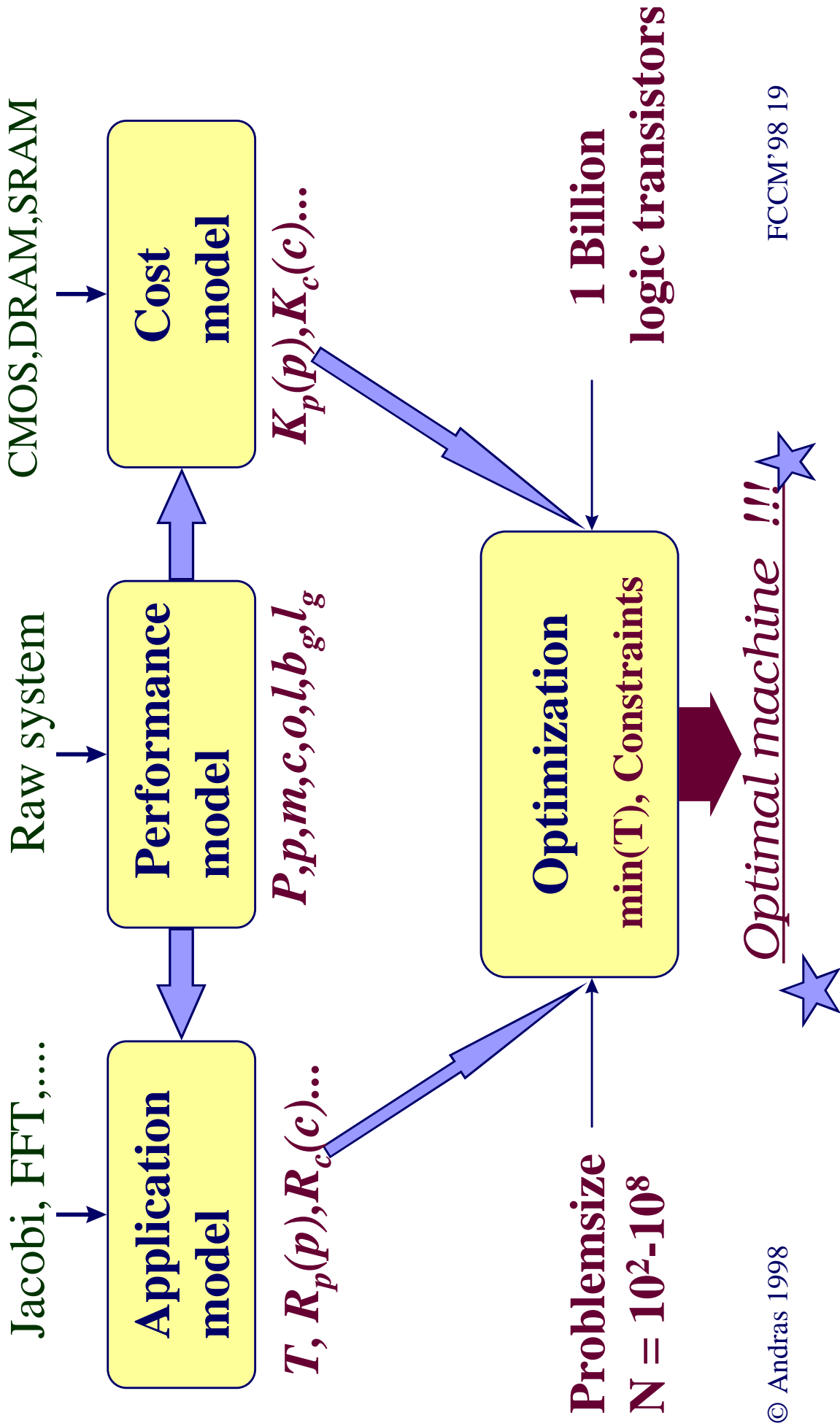
# Balance statements



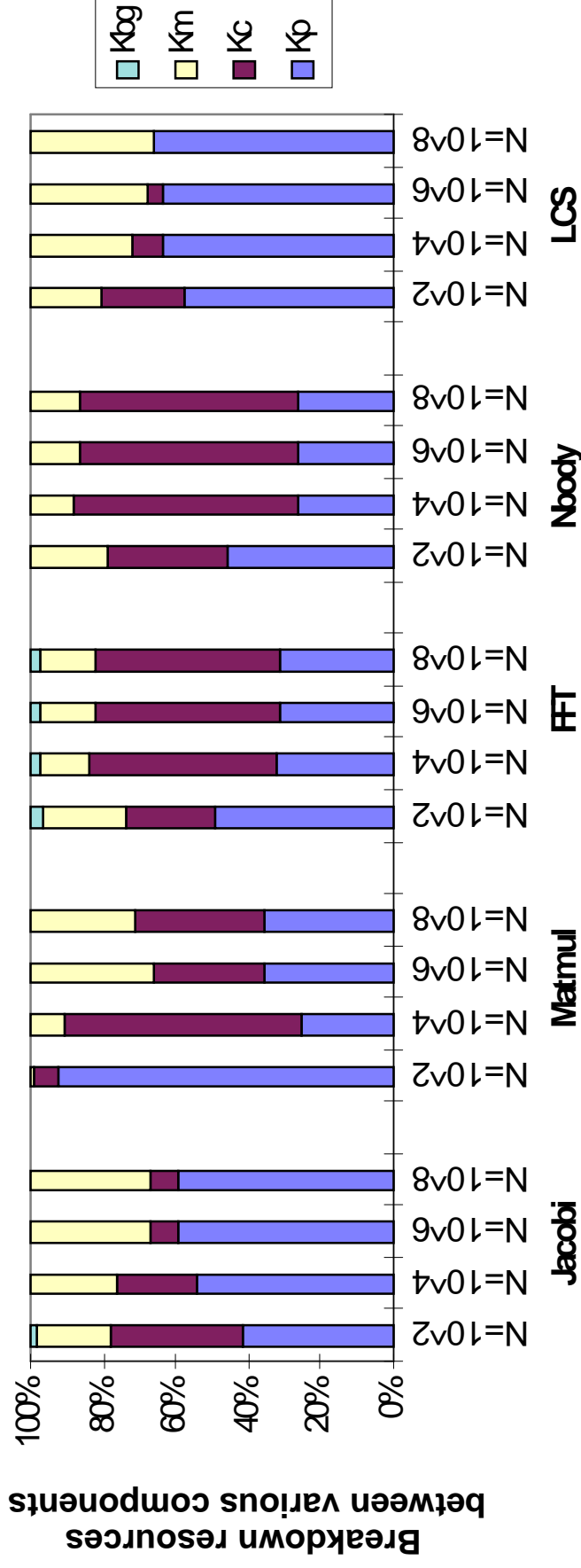
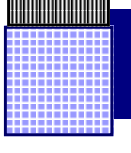
- ◆ Just a trick!!! Give optimal resource utilization, reduce search space  $\Rightarrow$  *Balanced configurations.*
- ◆ 3 Statements:
  - 1. *Fit subproblem on chip + memory for communication overlapping*
  - 2. *Local communication  $T =$  computation  $T$*
  - 3. *Global communication  $T =$  computation  $T$*

# Analytical framework

---



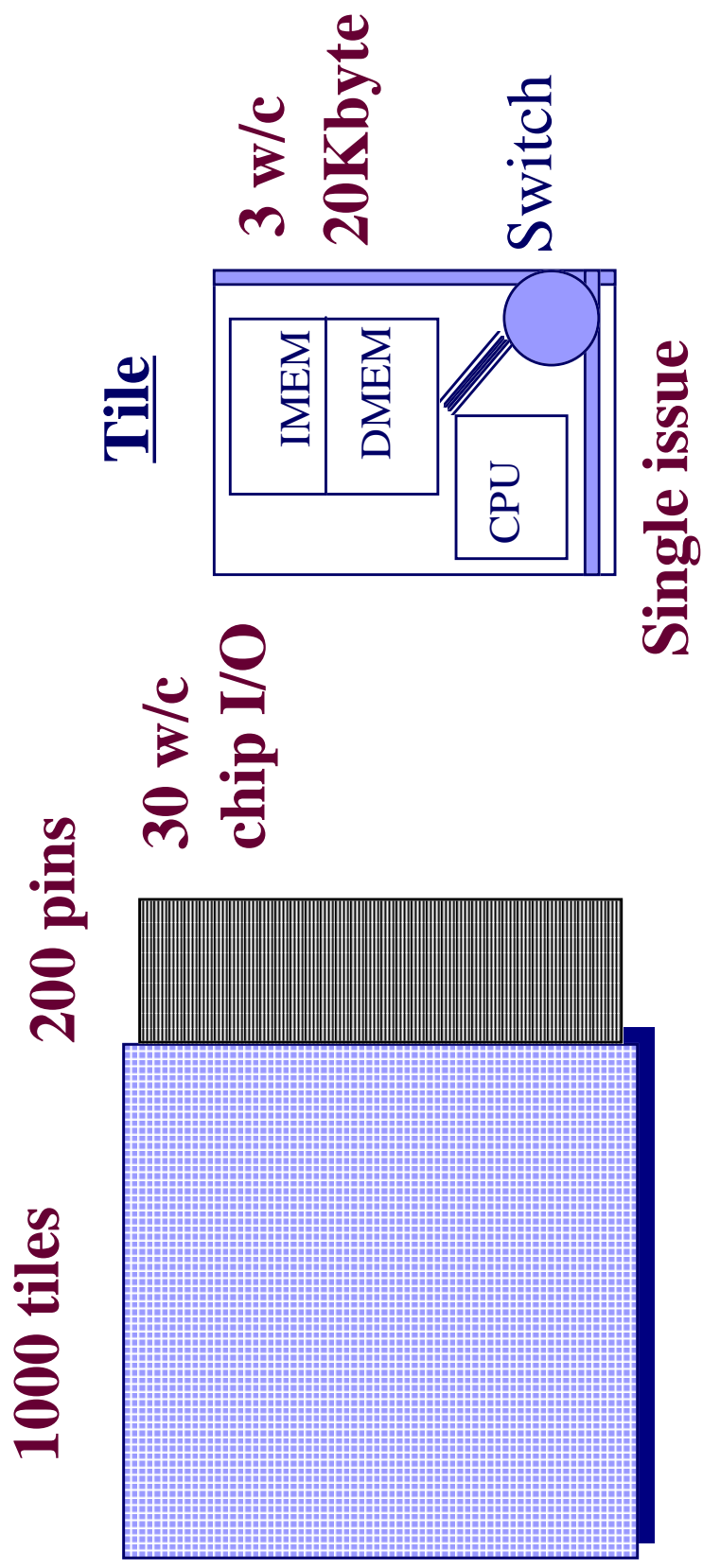
# Chip areas optimal machine



Breakdown of chip areas: processing, memory, local and global com  
1 billion logic transistor area.

# Raw chip for 1 billion transistors

---



# Conclusions

---

- ◆ Areas
  - 40% of area to processing
  - 35% to communication
  - 25% memory
- ◆ Design comparison: DRAM vers SRAM
  - significant speedup (up to 3 times)