Uri Weiser
Professor of Engineering Technion

Memory driven architecture: flipping the inequality computing vs. memory

The talk covers research done by: Prof. Y. Etsion, Dr. Z. Guz, Prof. I. Keidar, Prof. A. Kolodny, S. Kvatinsky, Prof I. Keslassy, T. Zidenberg, Prof. A. Mendelson, Y. Nacson, Prof E. Friedman, Prof. U. Weiser

This conference's message

"The large energy consumption associated with the ever increasing internet use and the lack of efficient renewable energy sources to support it"

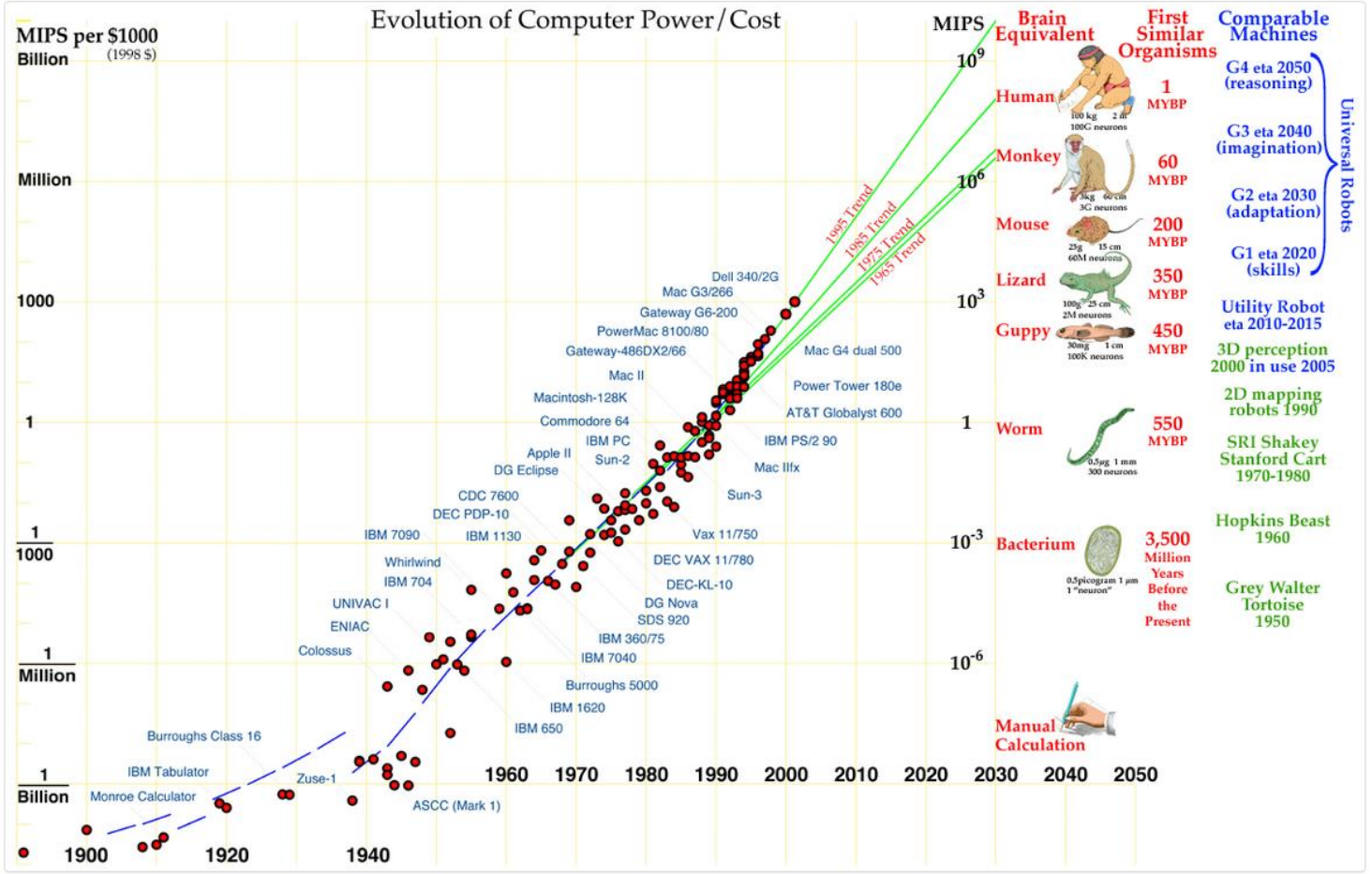
***Energy problems in data-com systems**

***Energy problems in computers: from systems to the chip level**

***Advanced solar energy harvesting**

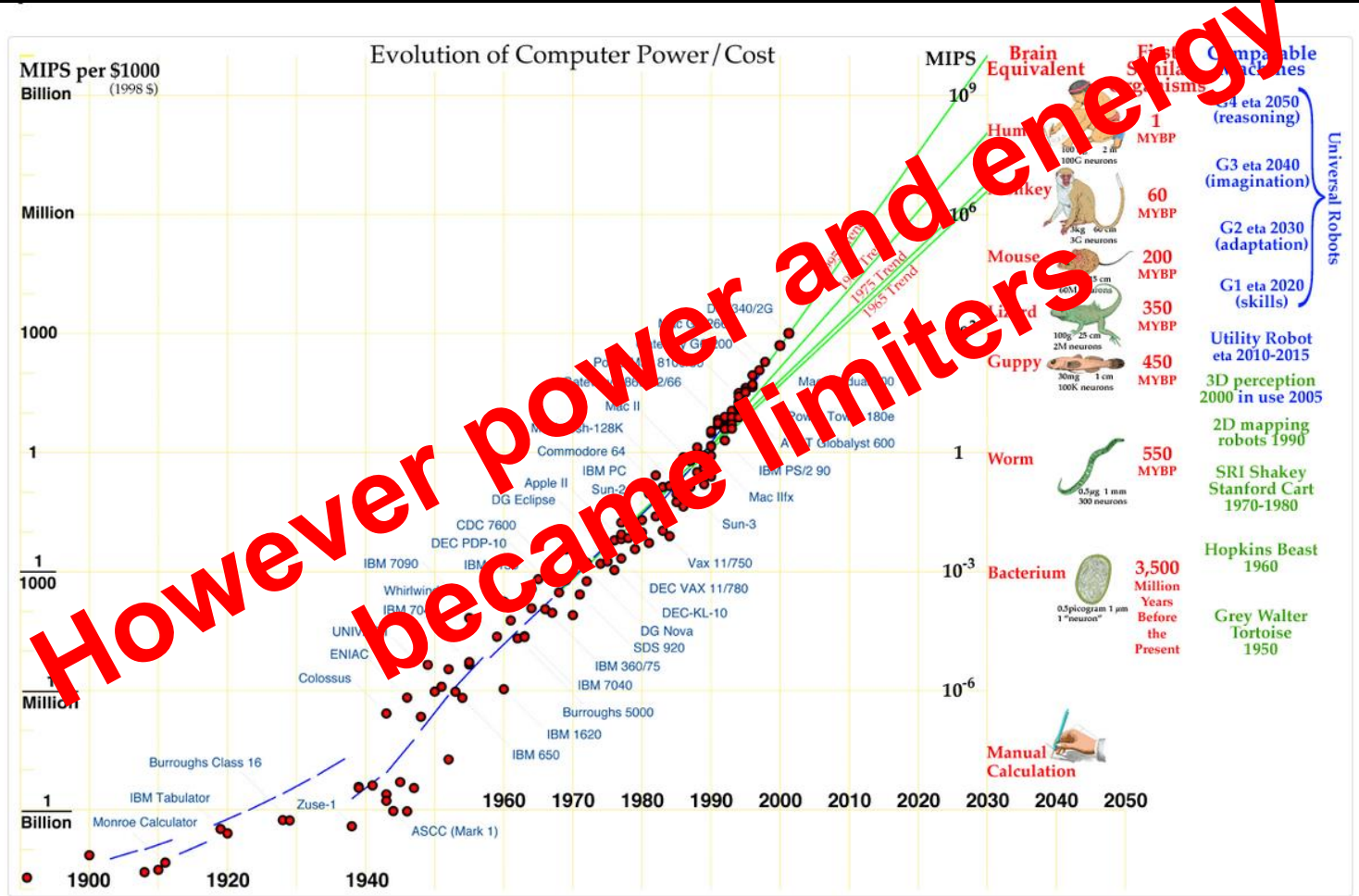
Scent of Solutions?

The Trend Our Customers Expect



From:

The Trend Our Customers Expect



However power and energy became limiters

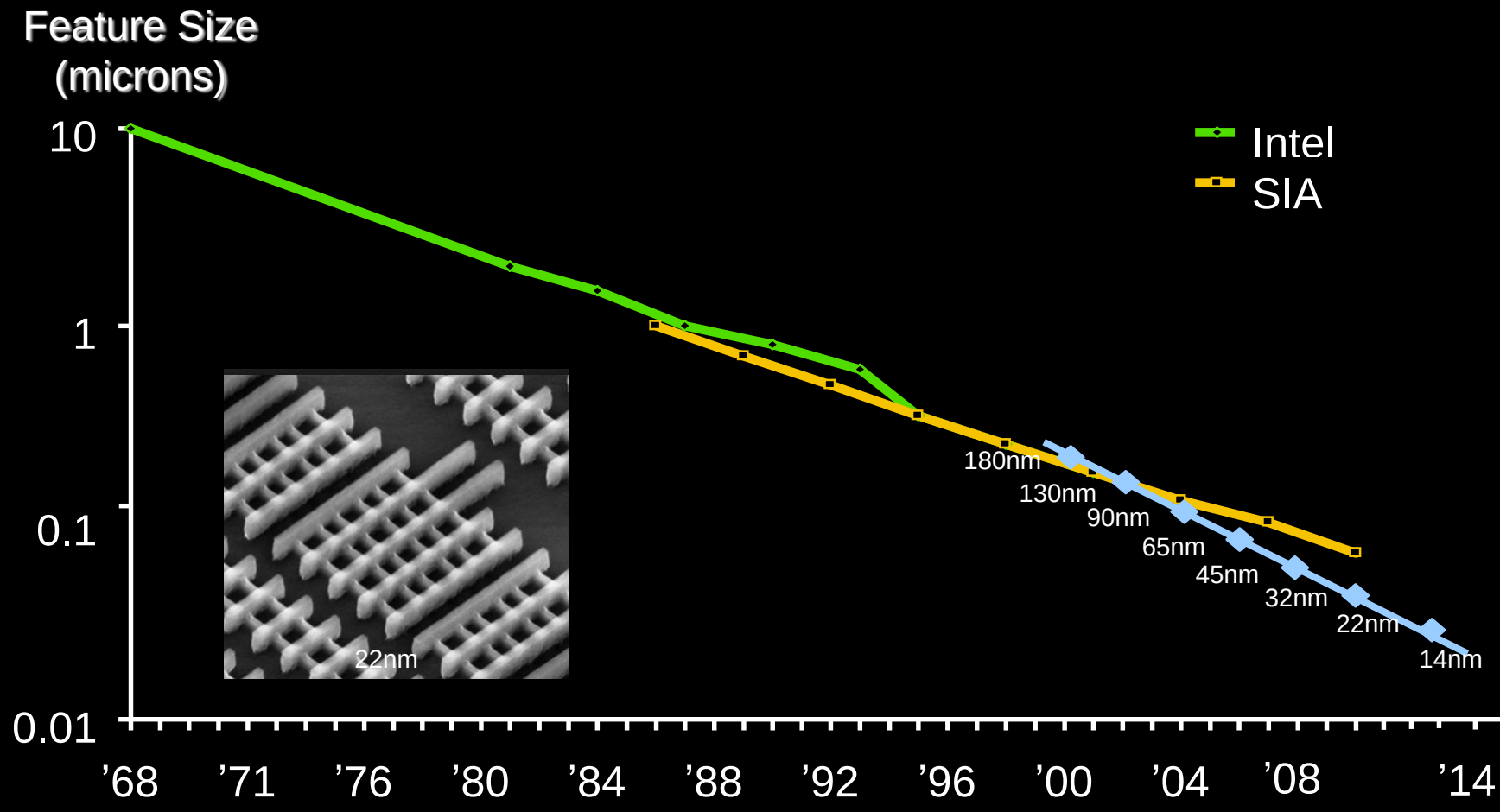
From:

Outline

- The trends
- The implications
- The opportunities
  - Heterogeneous systems – some thoughts
  - Memristor ➔ Memory Intensive Architecture (MIA)
- Energy: Optimal resource allocation in a Heterogeneous system
- How to start to think about Memory Intensive Architecture

The Trends

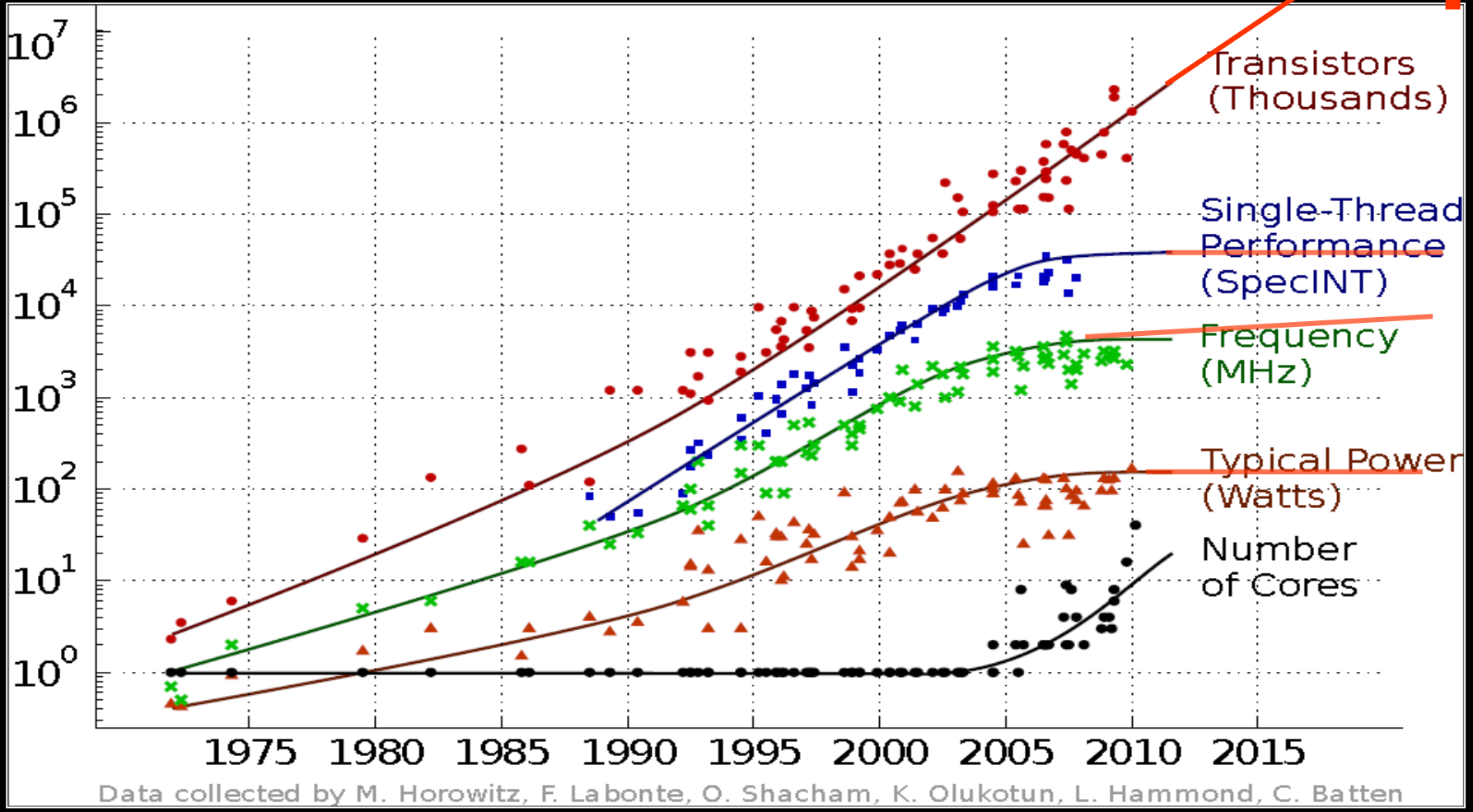
Process Technology: Minimum Feature Size



Source: Intel, SIA Technology Roadmap

SIA: Semiconductor Industry Association

Putting It All Together



The Trend

Where are we going?

The power wall

Microarchitecture

- VLSI Microarchitecture has been influenced by concepts that have been around for a long time
- We hit a power wall
- Solutions
  - Top down – improve performance/power or Throughput/power ➔ Heterogeneous Architecture
  - Bottom up – new devices ? Memory resistive devices?

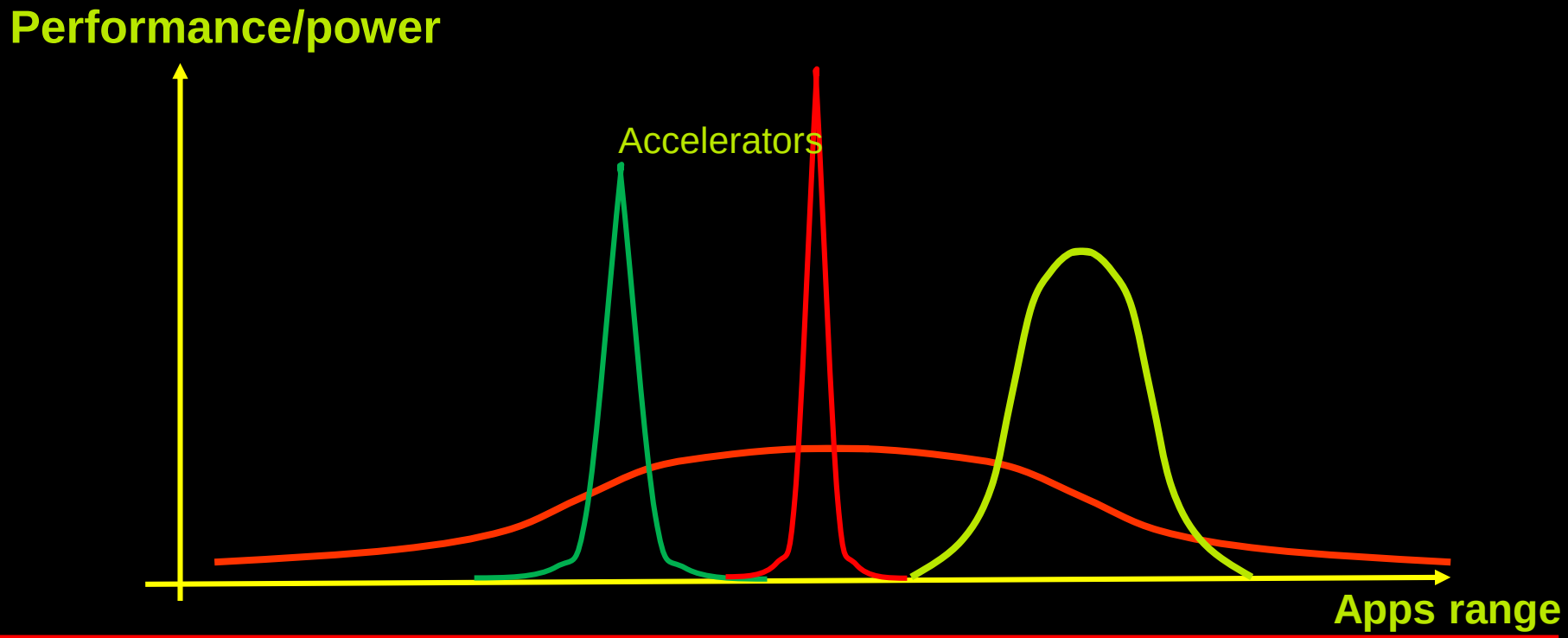
Hetero vs. Memory Intensive

- Heterogeneous Architecture ➔
  - For a while no major breakthrough in CPU technology
  - But the main reason is the POWER wall and energy/task
  - Accelerators to the rescue
- Memory Intensive Architecture ➔
  - Either a huge amount of memory cells close to logic, or
  - Logic cells close to lots of memory
  - Does it imply Symmetric processing?

Heterogeneous Systems

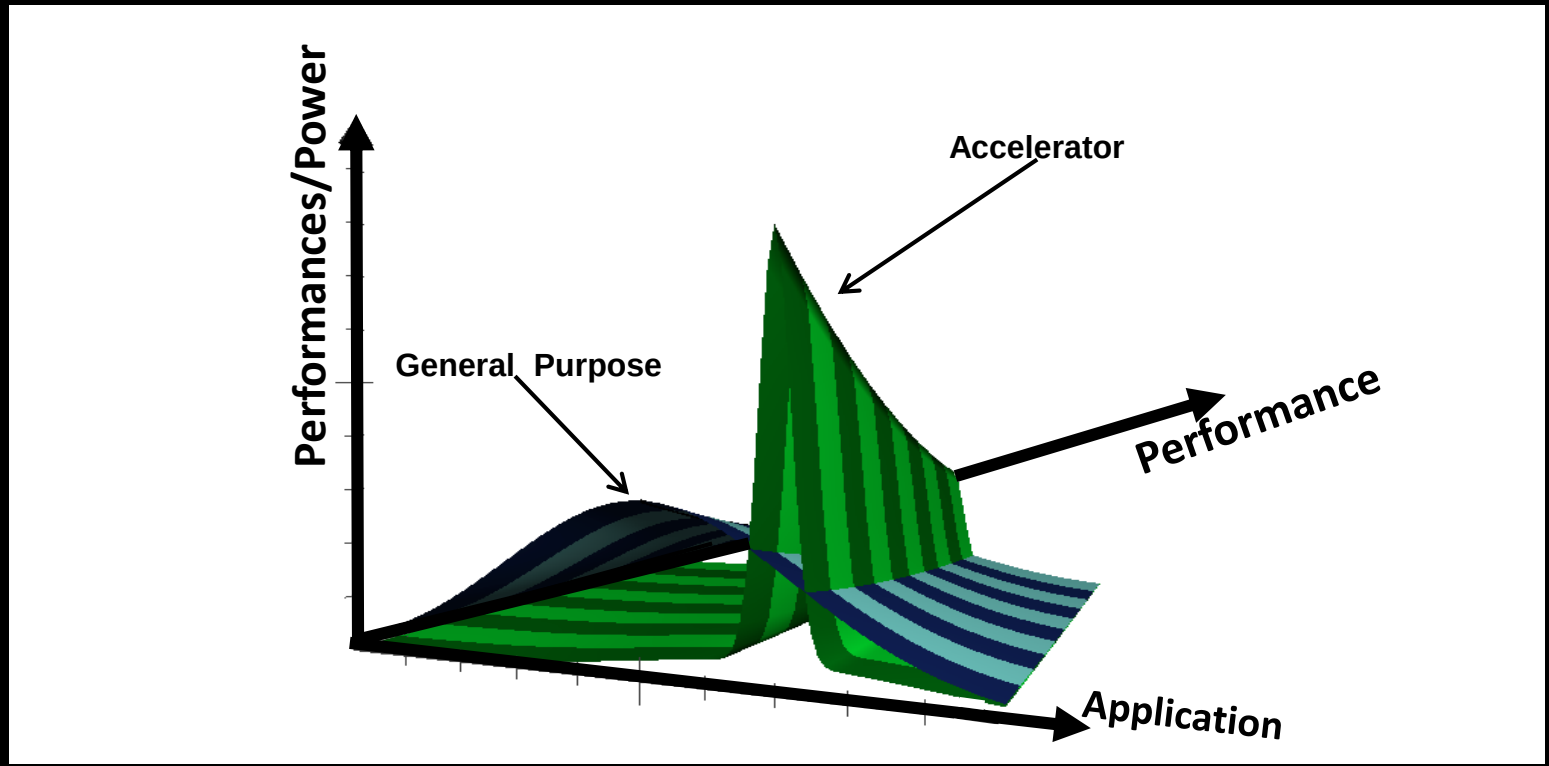
Flying machines - are they all the same?

Heterogeneous Computing: Application Specific Accelerators



Continue performance trend using Heterogeneous computing to bypass current technological hurdles

Heterogeneous Computing



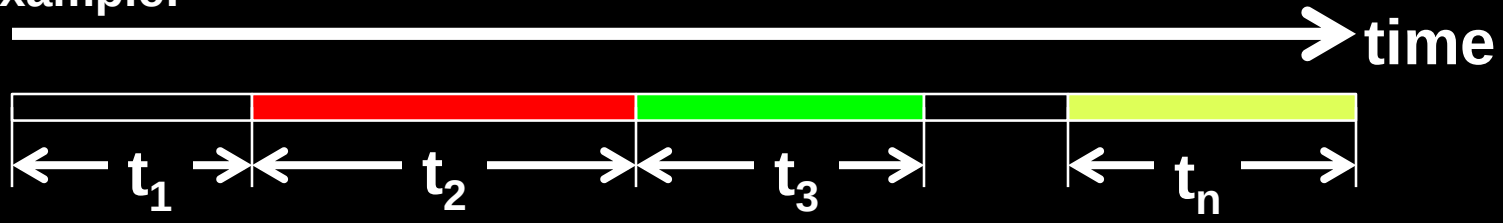
Heterogeneous Systems' Environment

- Environment with limited resources
- Need to optimize system's targets within resource constrains
- Resources may be:
  - Power, energy, area, space, $
- System's targets may be:
  - Performance, power, energy, area, space, $

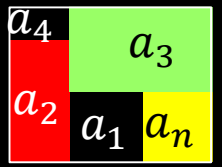
Heterogeneous Computing

- **Heterogeneous system design under resource constraint**
  how to divide resources (e.g. area, power, energy) to achieve maximum system's output (e.g. performance, throughput)

Example:



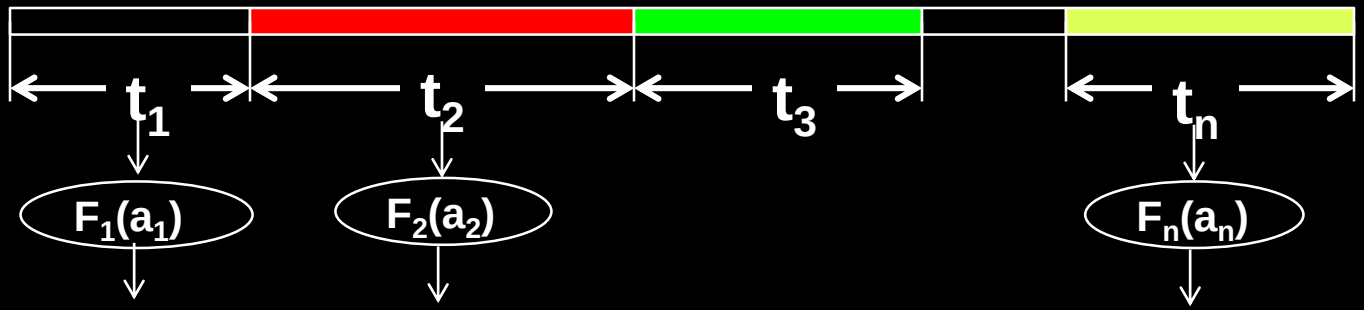
t_i = execution time of an application's section (run on a reference computing system)



$$A = \sum_{i=1}^{i=n} a_i$$

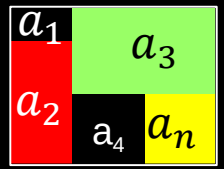
Accelerator target (an example): Minimize execution time under Area constraint

MultiAmdahl:



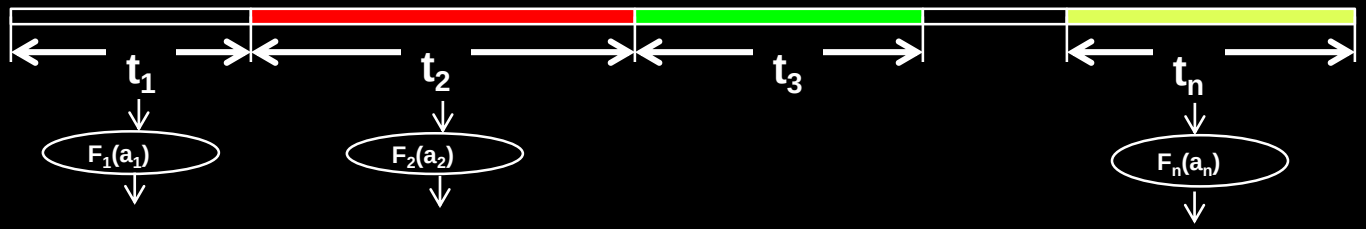
$T = t_{1*} F_1(a_1) + t_{2*} F_2(a_2) + \ldots + t_{n*} F_n(a_n)$

$A = a_1 + a_2 + a_3 + \ldots + a_n$



Target: Minimize T under a constraint A

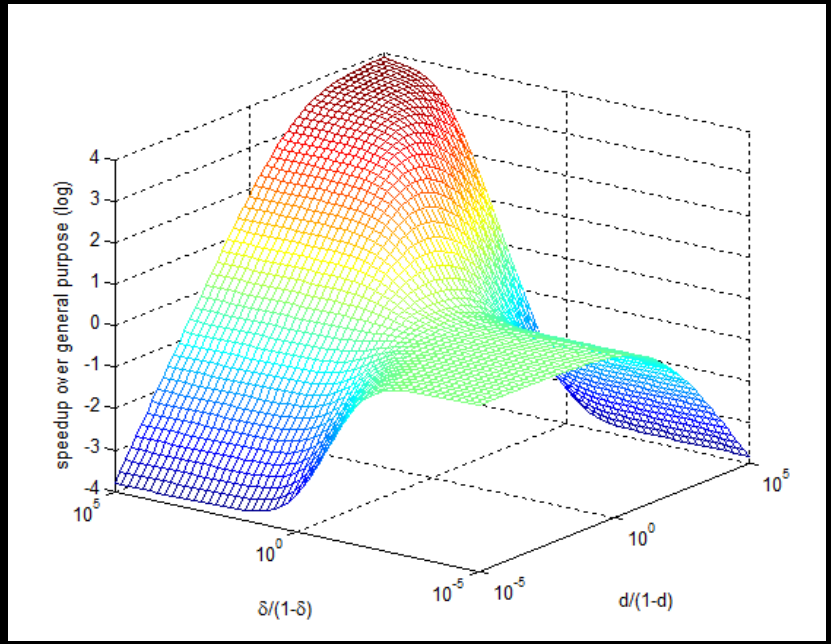
MultiAmdahl:



- **Optimization using Lagrange multipliers**
  Minimize execution time (T)
  under an Area (a) constraint

$$t_j F'_j(a_j) = t_i F'_i(a_i)$$

F'= derivation of the accelerator function
a_i = Area of the i-th accelerator
t_i = Execution time on reference computer



MultiAmdahl Framework

- Applying known techniques* to new environments
- Can be used during system's definition and/or dynamically to tune system

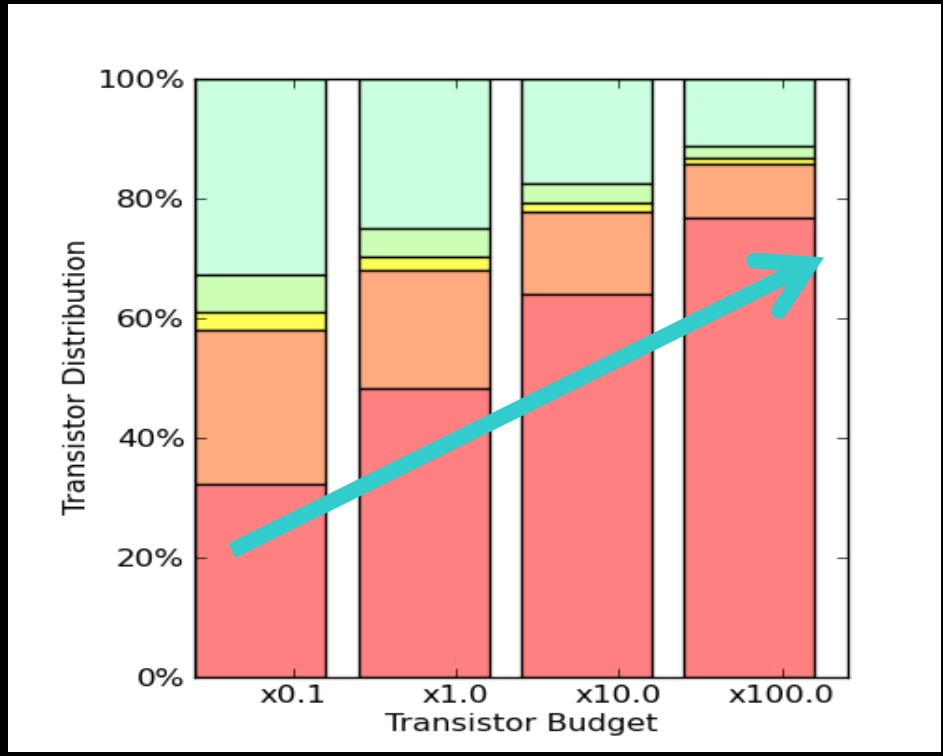
* Gossen's second law (1854), Marginal utility, Marginal rate of substitution (Finance)

Example: CPU vs. Accelerators

Future GP CPU size vs. transistor budget growth

Test case:
4 accelerators and GP (big) CPU

Applications: evenly distributed benchmarks mix w/ 10% sequential code



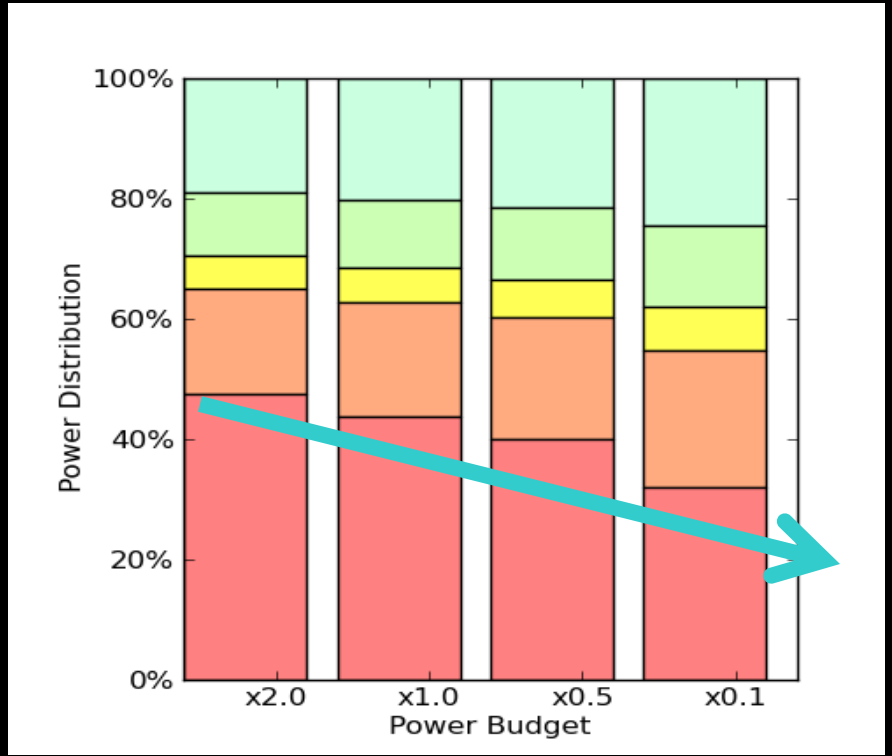
Heterogeneous Insight:
In an increased-transistor-budget-environment, General Purpose (big) CPU importance will grow

Example: CPU vs. Accelerators

GP CPU size vs. power budget

Test case:
4 accelerators and GP (big) CPU

Applications: evenly distributed benchmarks mix w/ 10% sequential code



Heterogeneous Insight:
In a decreased-power-budget-environment, Accelerators importance will grow

Environment Changes

Is it time for a change in implementation?

- Throughput became an essential Microprocessor target
- Data footprint became bigger
- Multi-Core systems are everywhere
  ➔ more performance = more memory usage

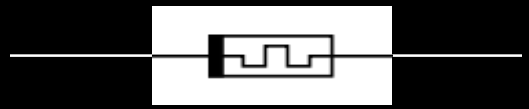
➔ Memory pressure is increasing

- Significant CPU die power (>30%) is consumed by IO (access to out-of-die memory)

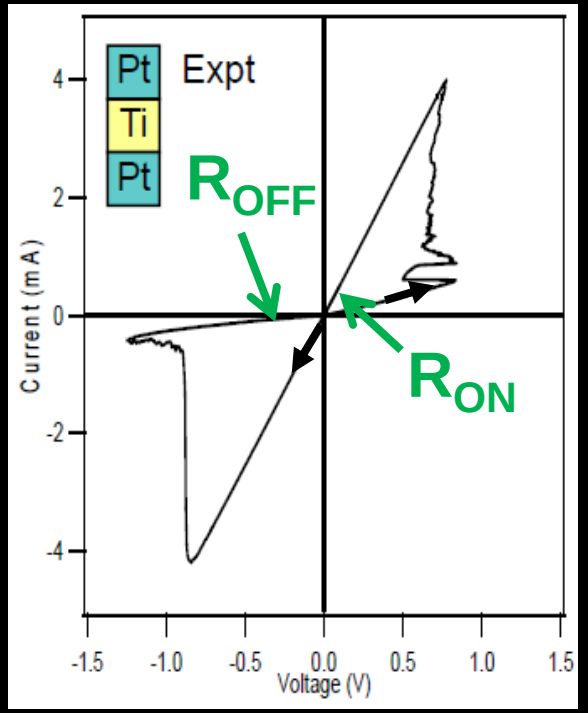
Bottom up approach:
New device - Memristor?

What is a Memristor?

- 2-terminal resistive nonvolatile device

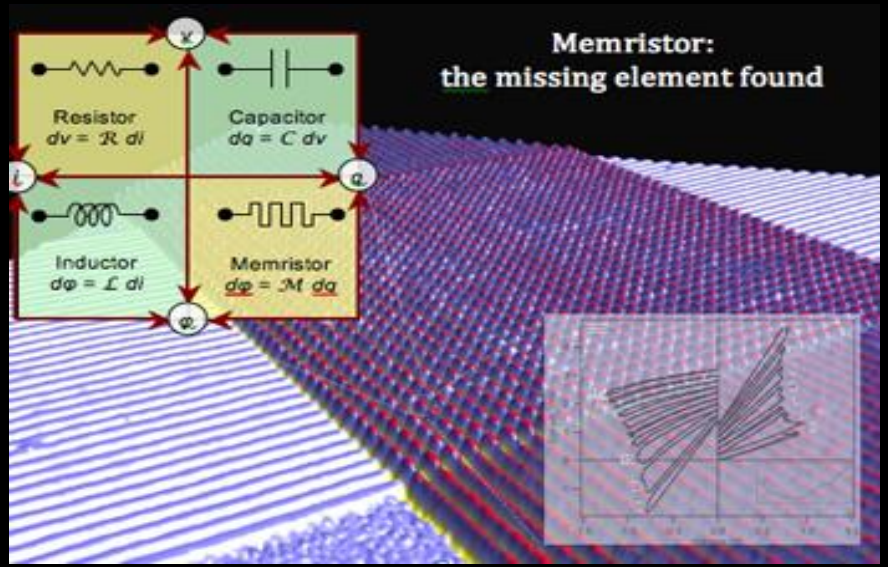


- Device's resistivity depends on past electrical current
- Device is constructed of 2 metal layers with oxide in between (e.g. T_iO_2)
- Can be implemented in Multi (physical) layer memory



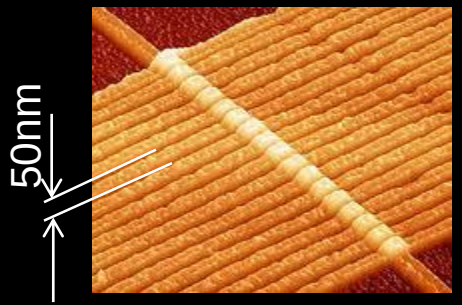
Jul 30, 2013
Panasonic Starts World's First Mass Production of ReRAM Mounted Microcomputers
[1] ReRAM (Resistive Random Access Memory)
A type of non-volatile memory which records "0" and "1" digital information by generating large resistance changes with a pulsed voltage applied to a thin-film metal oxide.
The simple structure of the metal oxide sandwiched by electrodes makes the manufacturing process easier and provides excellent low power-consumption and high-speed rewriting characteristics.

Memristor



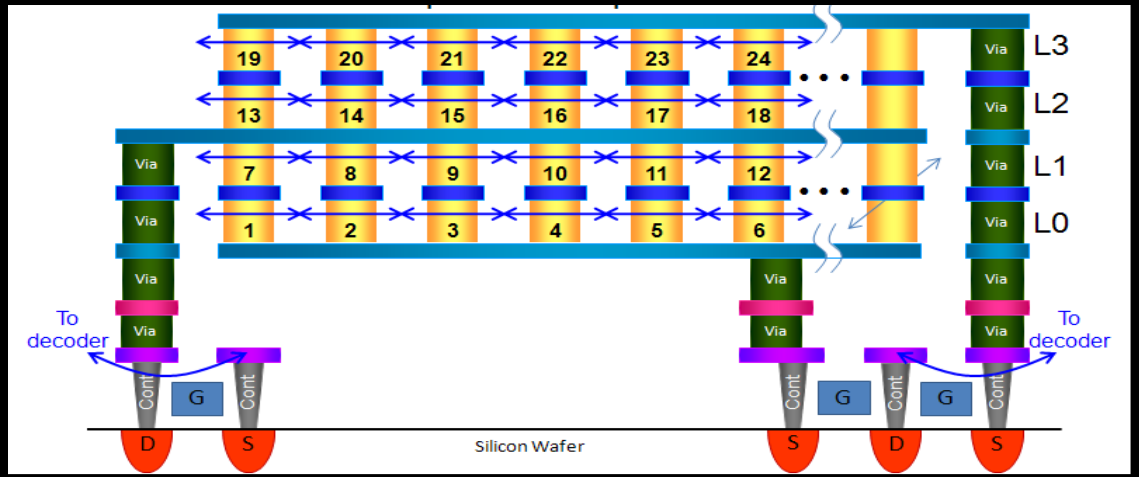
- Theoretical idea by Chua in 1971
- Implemented today by Hewlett Packard
  SK Hynix, HRL Labs
- Memory products by Pannasonic

Array of 17 oxygen-depleted titanium dioxide memristors (HP Labs)



Memristor Microarchitecture “Vision”

- Layers of memory cells above logic



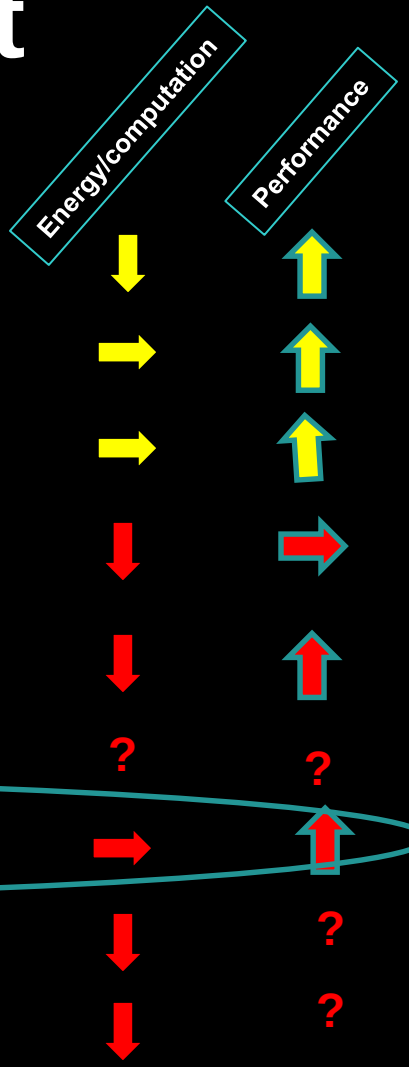
- Does this new structure open the possibility for new Microarchitecture?

Memristors to the Rescue?

- Huge amount of memory cells
- Very close to logic
- Non volatile
  - No need for power to keep alive
- ~ transistor size
- Fast
- No leakage

Sea of Memory Cells Impact

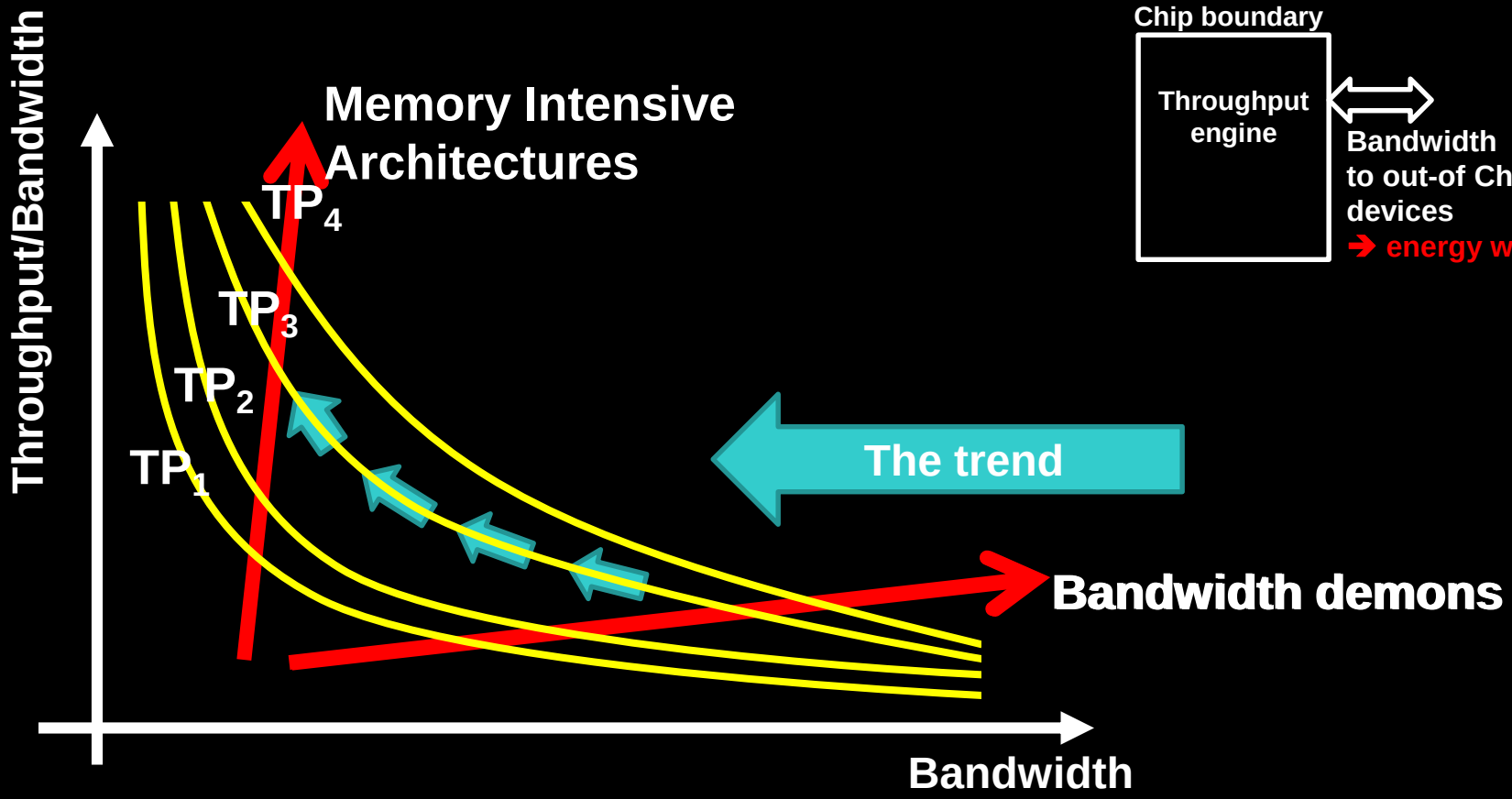
- Conventional vs. Out of the box



- Enhance Multithreading architecture (Graphics like)
- Increase on-die prediction structures
- Instruction queues
- Back to LUT (look-up-tables) implementations
- New caches (e.g. NAHALAL, MC vs. MT, Cache specific content)
- Non-Register Architecture (memory-to-memory operations)
- Continues Flow Multithreading (improved SoE MT)
- Instruction reuse (memoization)
- Computation at the memory level*

* Ref Dr. Avidan Akerib General manager NeoMagic

Throughput and Bandwidth*

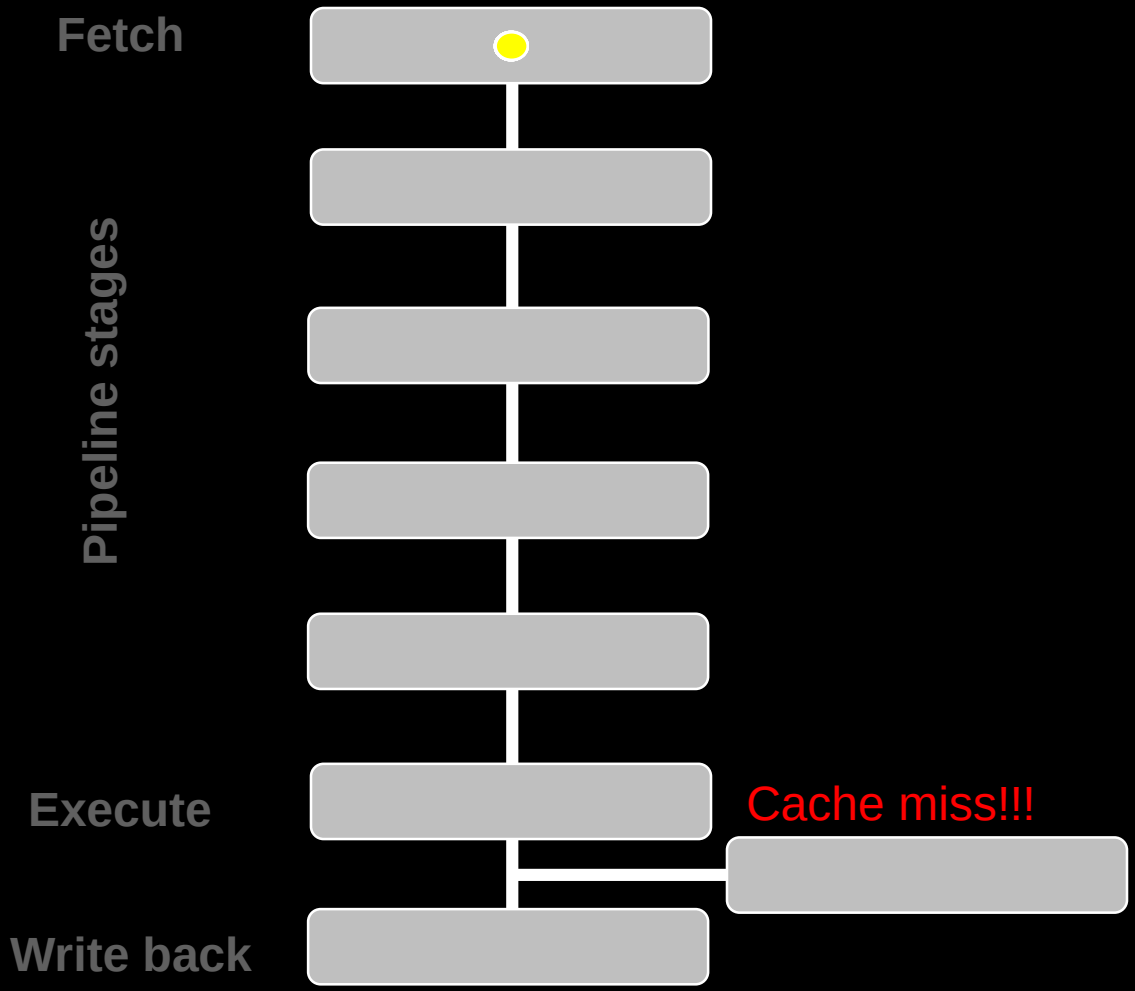


- Traversing on a constant-throughput-line ➔?
- ? ➔ increase on-die-memory (e.g. cache, new ideas)

*Influenced by ISCA 1995 paper: Performance Evaluation of the PowerPC 620 Microarchitecture; (graph: frequency vs. performance/frequency)

Switch on Event Multithreading

Example- processor pipeline



Thread A

Thread B

Thread C

Continuous Flow MT (CFMT)

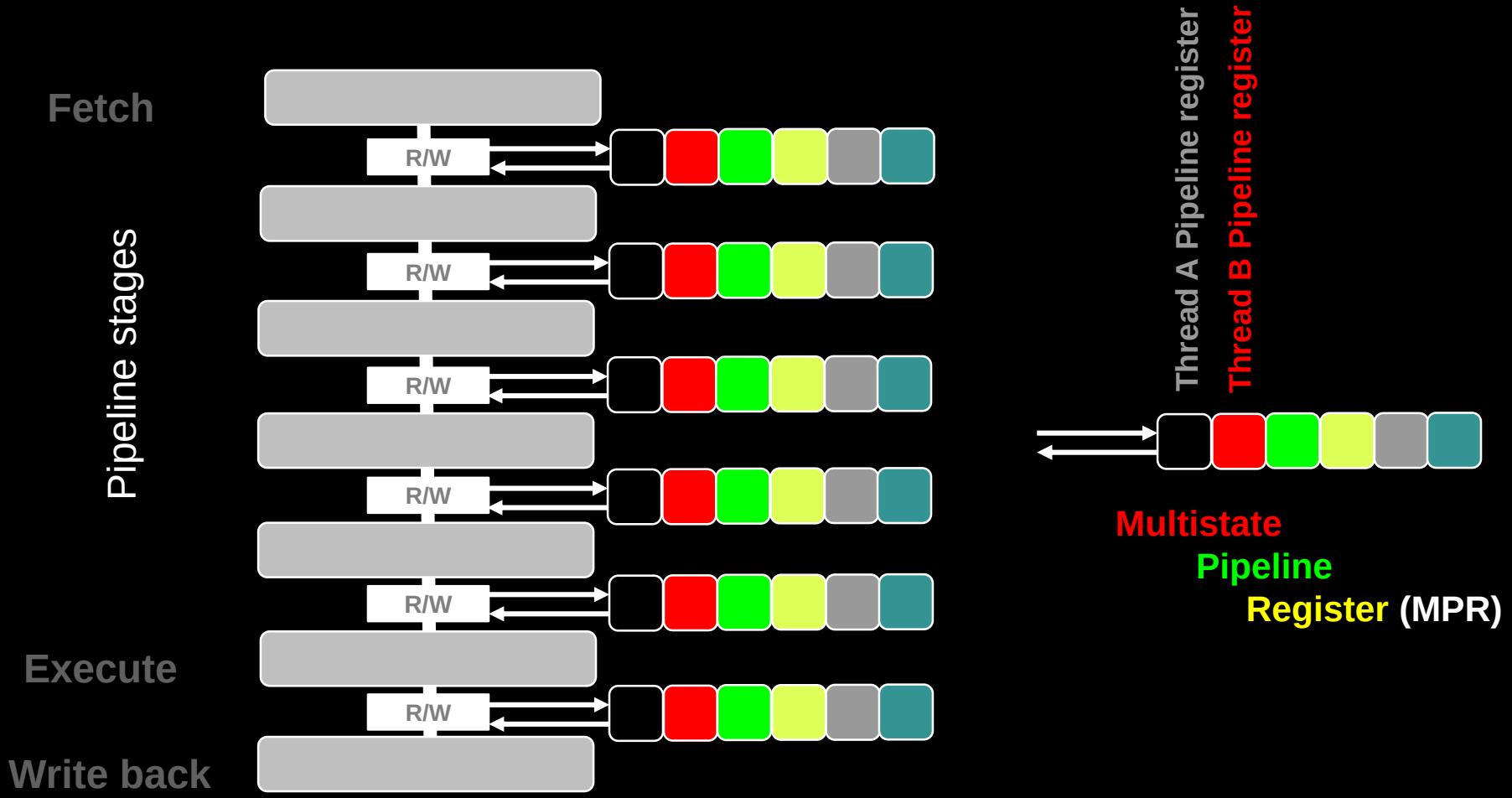
Example – processor's pipeline

- **SOE deficiencies**
  - **Instructions flush beyond the "event instruction"**
    - **➔ waste of energy**
    - **➔ performance degradation**
- **Can we use Memristor to reduce thread switch penalty (bubbles)?**
- **➔ Yes**

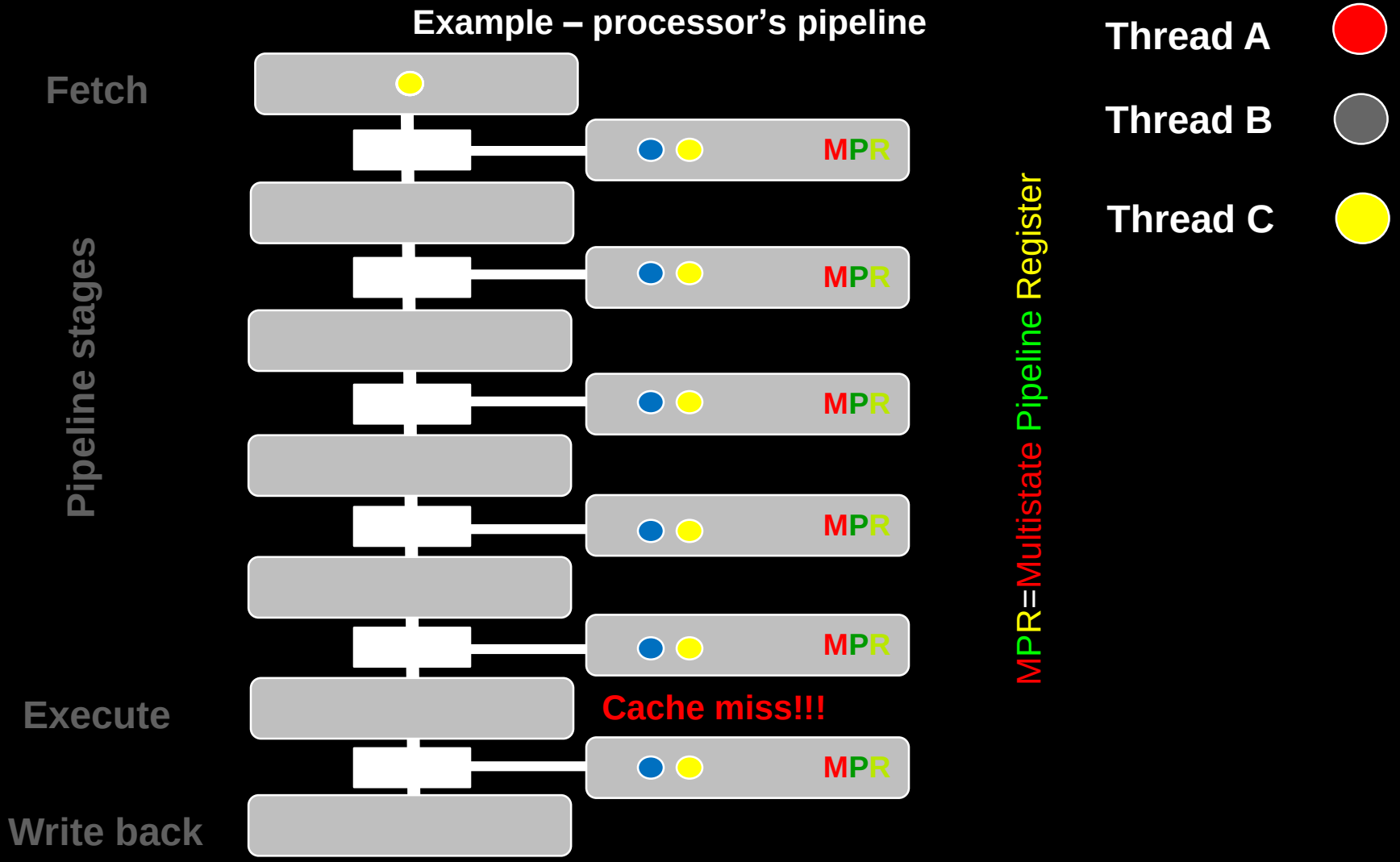
  do not flush, store the thread-pipe-state in memristors (Multistage Pipeline Register)

Continues Flow MT (CFMT)

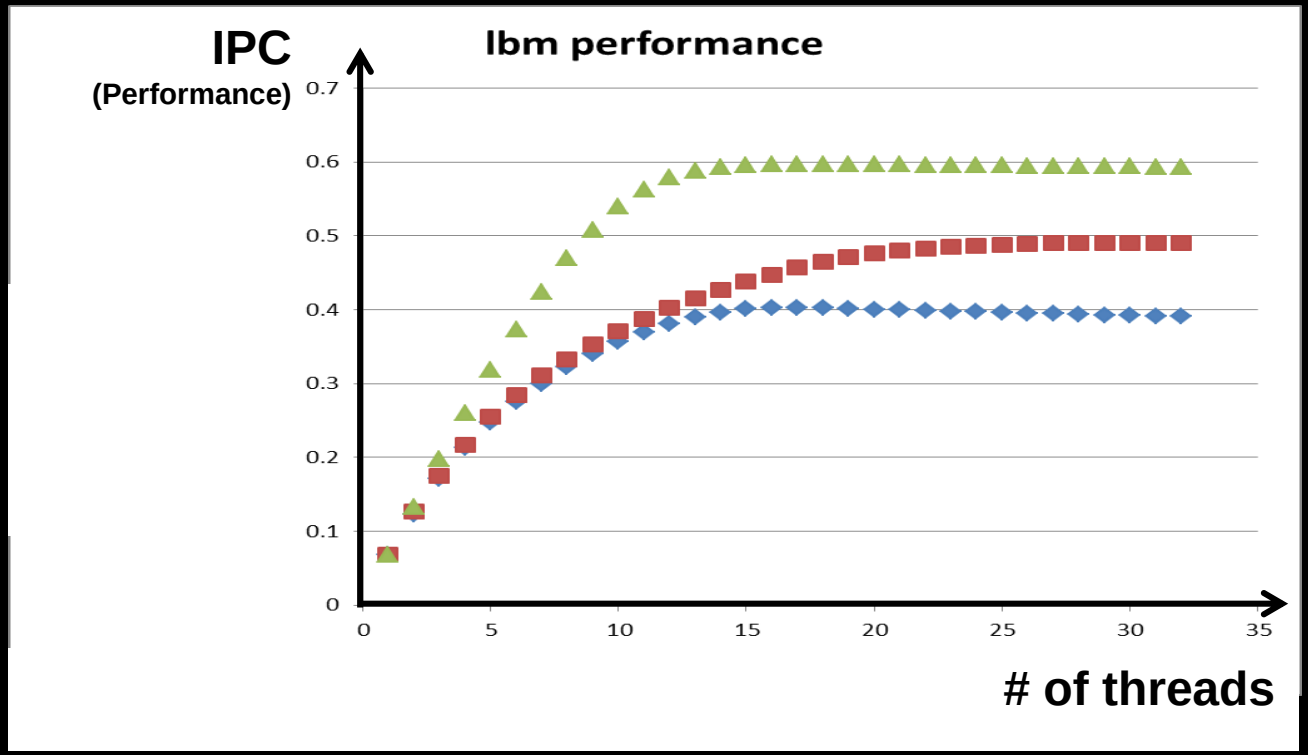
Example – processor's pipeline



Continuous Flow MT (CFMT)



CFMT Initial Simulation (preliminary) (ARM like Microarch V7, lbm from Spec CPU 20



CFMT (Mem and MCE)

CFMT (mem only)

SoE; no CFMT

CFMT for multiple cycle events? not sure yet…

Memory Intensive Architecture

Looking Forward

- **Large on die memory may save energy and change the way we architect our computational machines**
  - **Reduction in Data-Transfer**
  - **Opportunity for dramatic improvement in Performance/Power or Throughput/Power**
  - **Performance improvement (@same power) => energy reduction**
  - **Reduction of static/leakage power**
  - **Energy saving in reactive systems (0 memory energy when no operation)**
  - **NEW!!!**

Summary

- Saving energy via optimal Heterogeneous system
- The introduction of on-die huge memory should alter the way we design computational machines for low energy consumption

Thank You