컴퓨터및회로설계특론

첨단 **System-on-Chip** 기술 **(2)**

Eui-Young Chung

Outline

- 연구실 소개
- SoC architecture
- System-level low power design

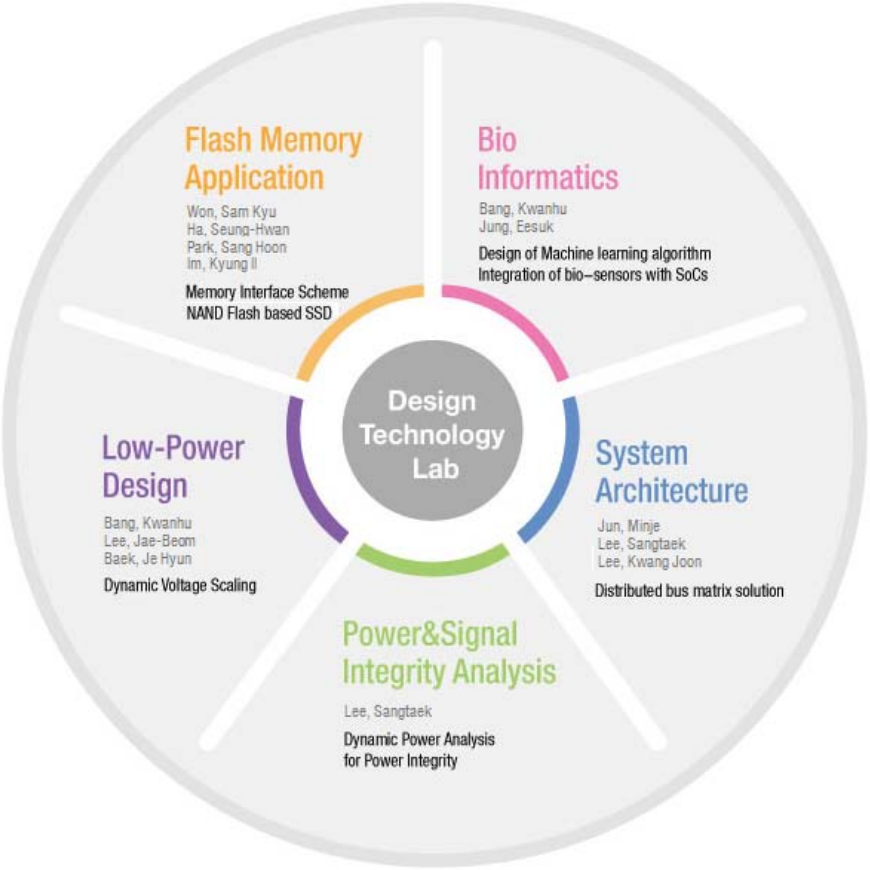
연구실 소개

❑ 구성원

- 4 (박사과정) + 9 (석사과정) (학연 4명: 삼성, LG, 하이닉스)

❑ 연구 분야

- SoC / Computer / Bio
  - HW + SW
- Industry-aligned area
  - SoC / Computer
    - Solid-State Disk (SSD)
    - System architecture
    - Low power design
    - VLSI CAD
- Academia interests
  - Bio-informatics
    - Parallelization
    - Machine learning



연구 실적

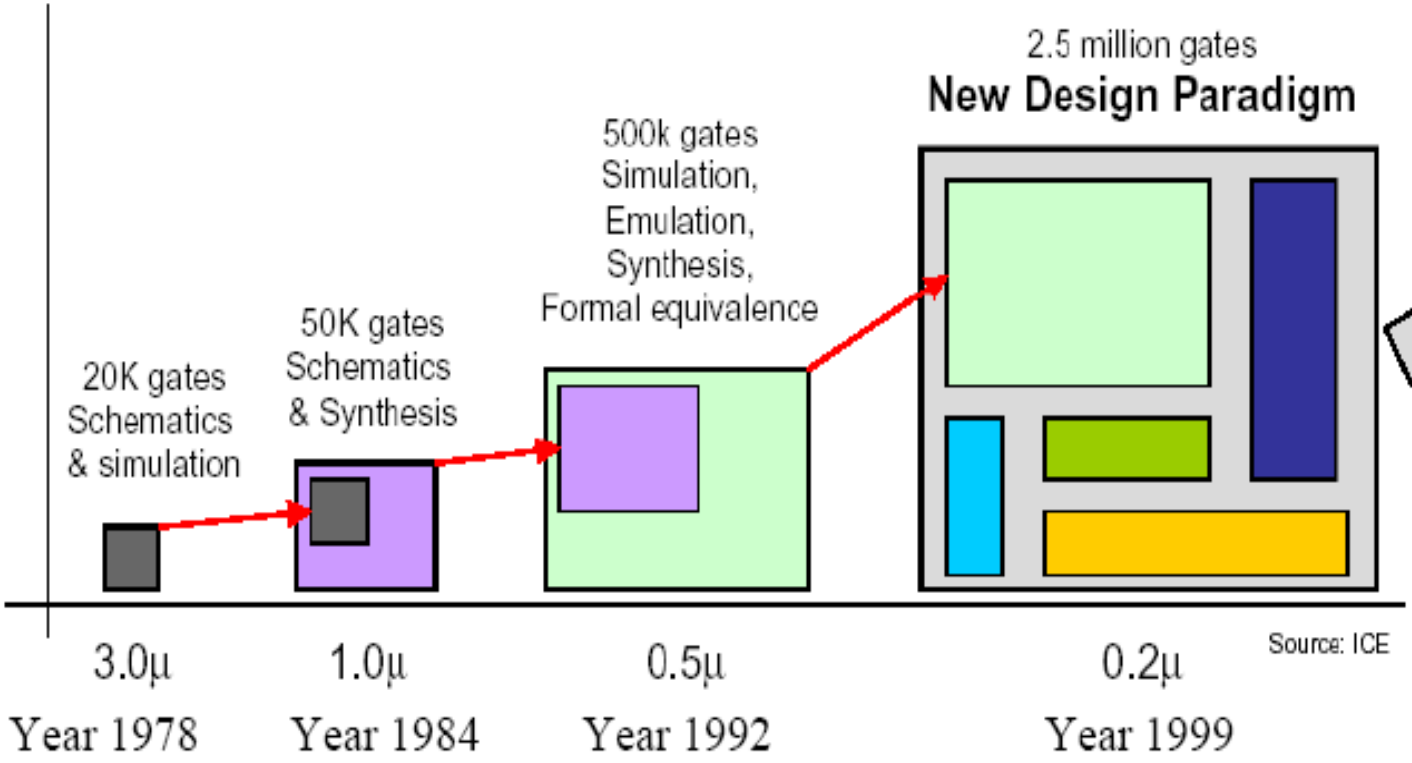
- On-going projects (연 2억내외)
  - SSD 관련 연구: 삼성 / 하이닉스 / 한국연구재단 기초연구
  - Low power design 관련 연구: 삼성 (3월 종료)
  - System Architecture 관련 연구: 산업원천기술 (서울대 와)
  - Bioinformatics 관련: 한국연구재단 신진연구
- 논문
  - SCI(E) 급 논문 21편 (책임 14편) from 2005.09
  - 유관분야 grand slam
    - IEEE Tcomputers, IEEE TCAD, IEEE TVLSI
  - IEEE transaction or IF > 3 이상 논문: 10편 (책임 6편)
- 특허 / 기술이전
  - 보강해야 할 부분임 ☹
  - 산학협력단 나성권 변리사와 SSD에 대해 집중키로 함

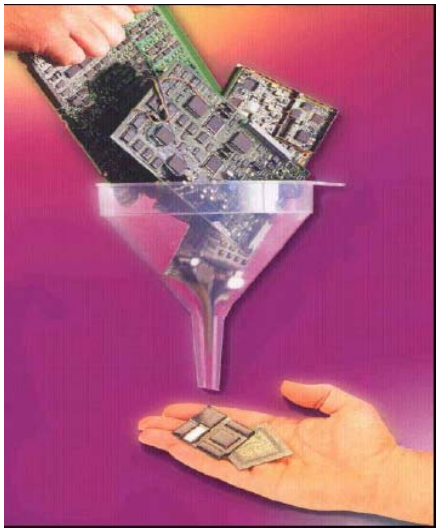
Outline

- 연구실 소개
- SoC architecture
- System-level low power design

Evolution of Microelectronics

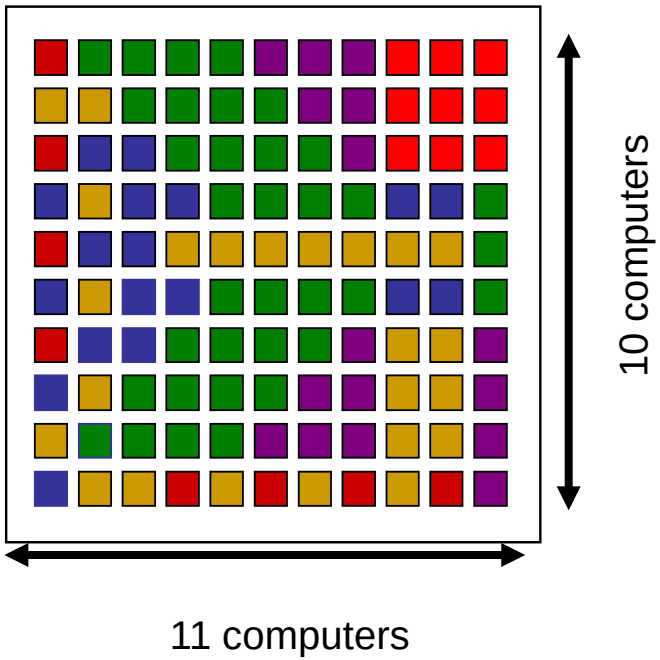
- Yesterday's chip is today's function block!





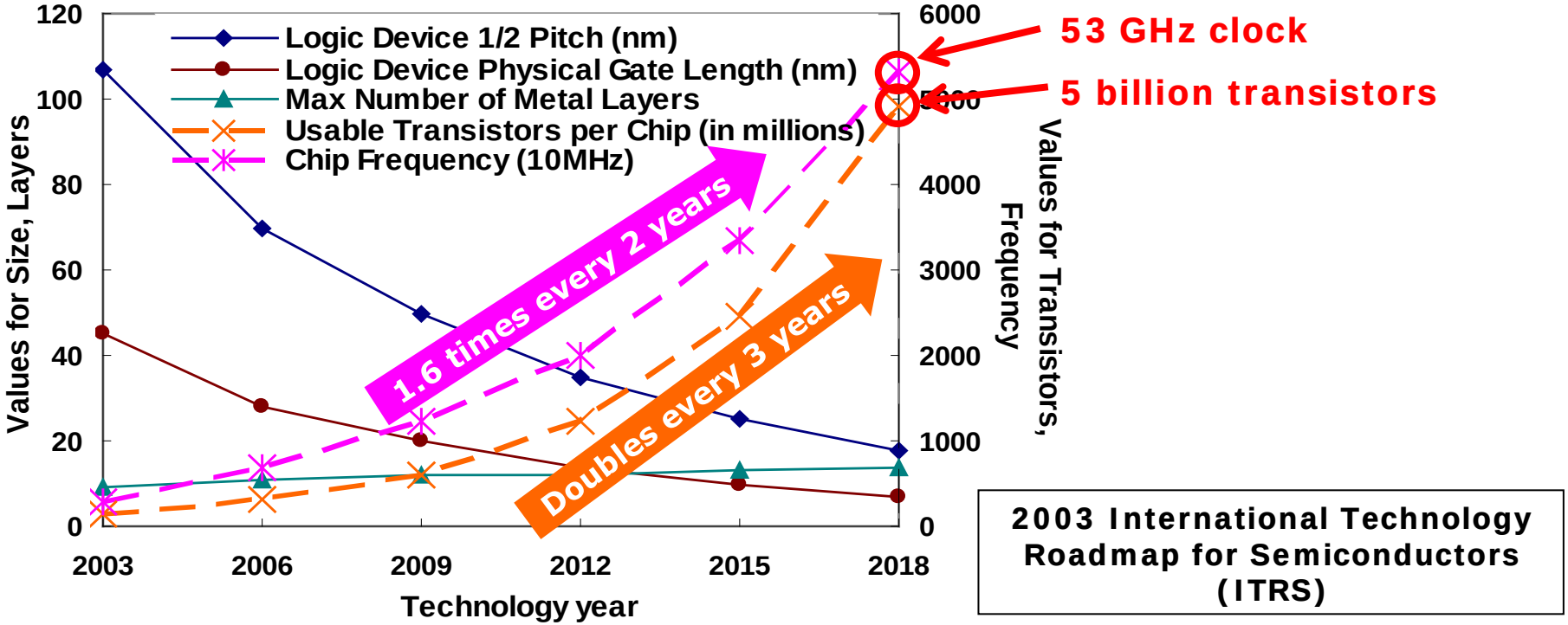
Billion Transistor Era

- 1 billion transistor SoCs are expected to be used in products by 2008.
  - Tens or even hundreds of computer-like resources in a single chip.
- According to ITRS, SoCs at 50nm will have 4 billion transistors and operate at 10Ghz in the next decade.



Silicon Technology Advance

- High-volume, high-frequency chips



2003 International Technology Roadmap for Semiconductors (ITRS)

- High integration density
  - Macrosystems ⇨ Microsystems
  - Complex on-chip communication requirements

System-on-Chips (SoCs)

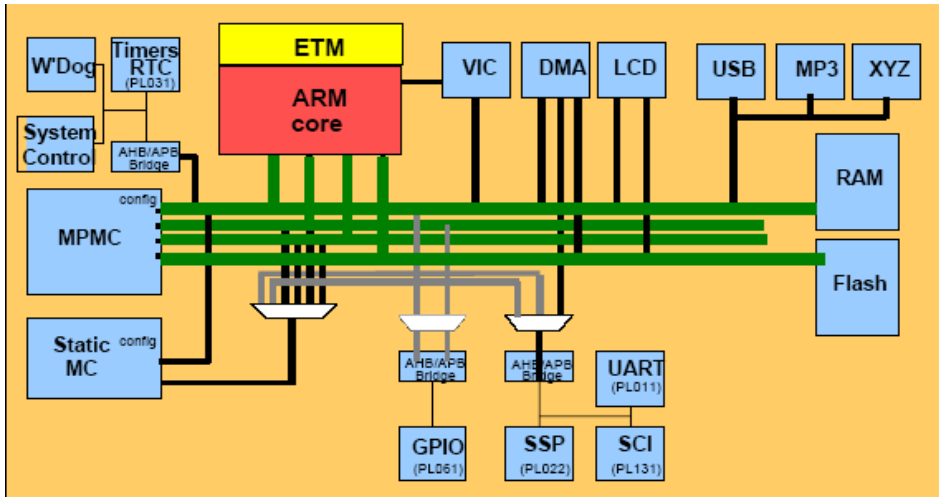
- Solution to cope with increasing circuit complexity.
  - System in terms of subsystems
  - Different Levels of Concepts and Abstraction
- Efficient reuse of designs and design experience
  - Pre-designed Intellectual property (IP) cores
    - Processors, Cache and Memory cores
    - DSP cores
    - Buses (?)
  - Meeting the TTM (time-to-market) constraint
- Facilitated by new design methodologies
  - Interface-based design
  - Platform-based design

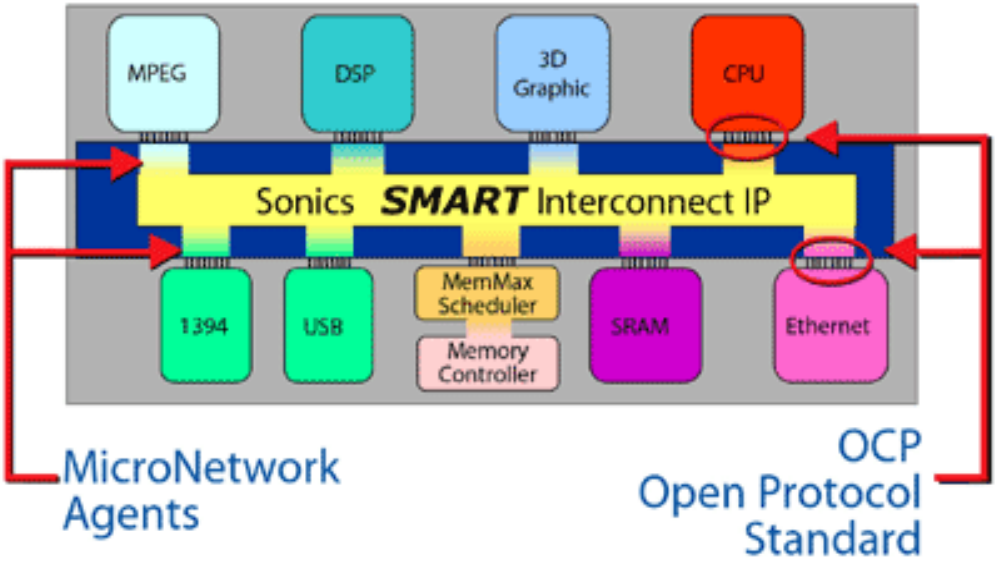
On-Chip Interconnects

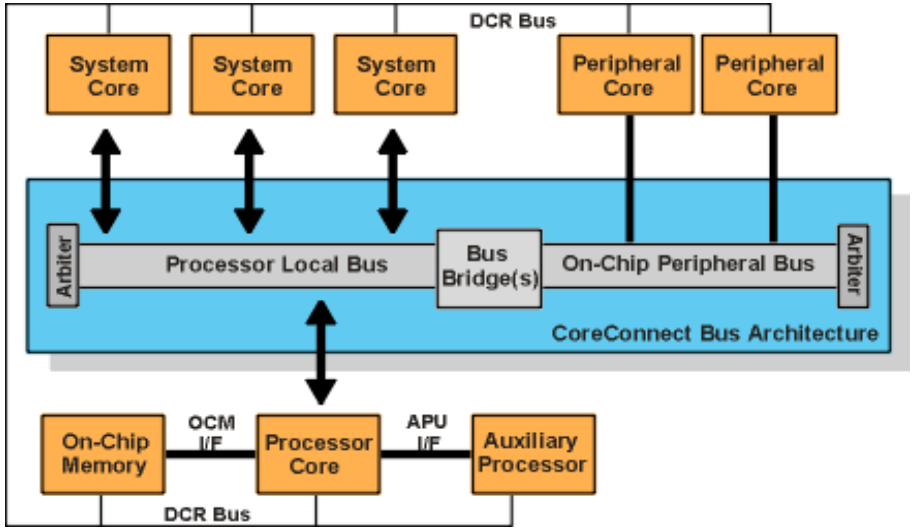
- Communication channels for functional modules (or IP blocks) integrated in a single chip.
  - Shared media like a bus
  - Dedicated point-to-point links
- So far, on-chip interconnects provide limited bandwidth for lower-performance, lower-power cores.
- Standardized bus systems with the incorporation of pre-designed Intellectual Property (IP) cores.
  - AMBA (Advanced Microcontroller Bus Architecture) by ARM
  - SiliconBackplane uNetwork by Sonics
  - CoreConnect by IBM

Popular Industry Solutions

- AMBA (Advanced Microcontroller Bus Architecture)
  - ARM
- SiliconBackplane MicroNetwork
  - Sonics
- CoreConnect
  - IBM

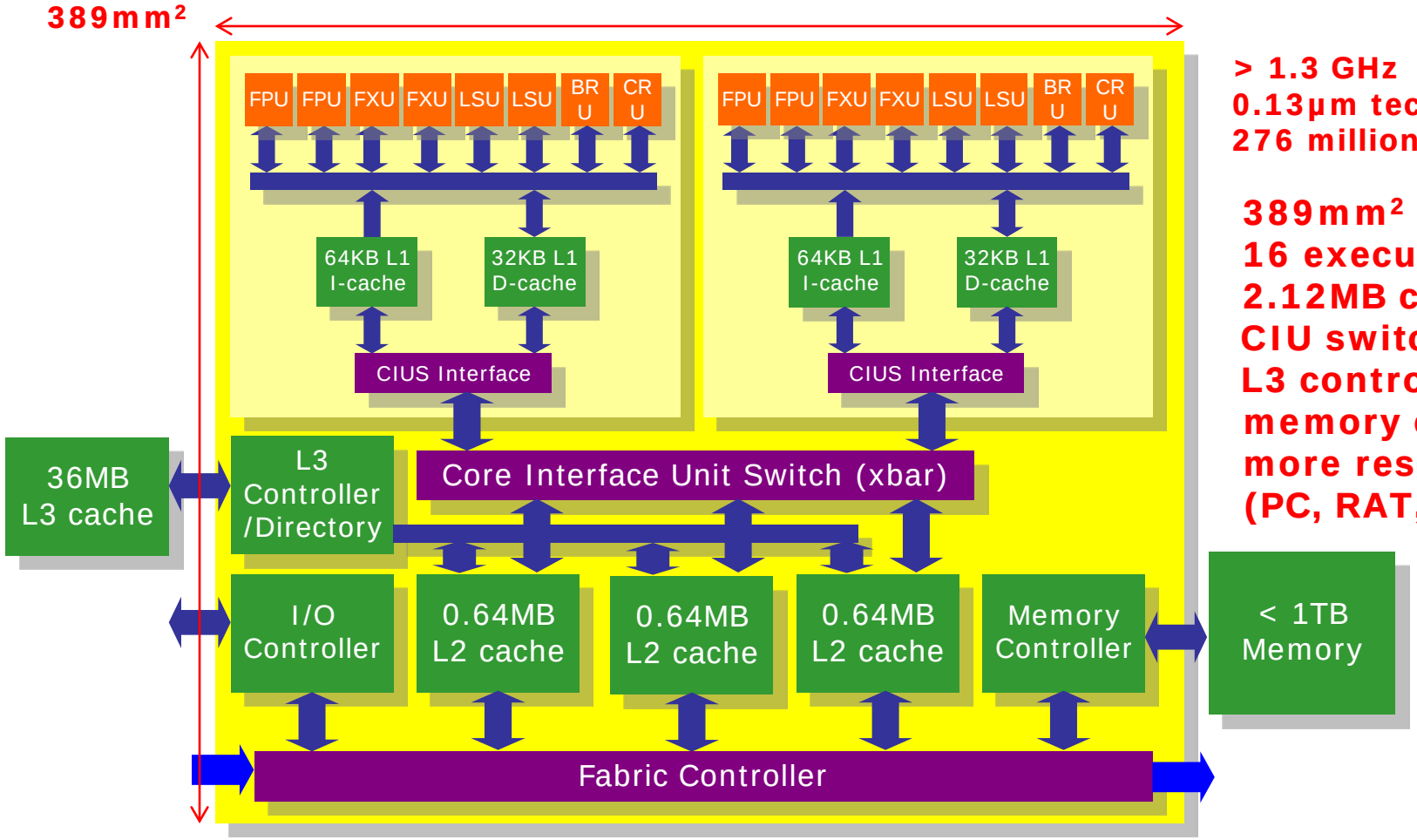






High Density SoC Based on Buses

- IBM POWER5 (Year 2004)



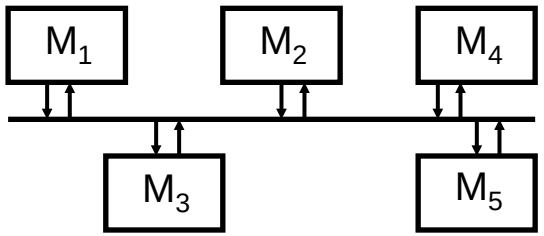
> 1.3 GHz
0.13μm tech
276 million Tr.

389mm² chip has
16 execution units,
2.12MB cache,
CIU switch, I/O controller,
L3 controller/directory,
memory controller, and
more resources for SMT
(PC, RAT, RF, CL, etc.)

*Refer to http://www-03.ibm.com/servers/eserver/pseries/hardware/whitepapers/power4.html

Characteristics of Bus-based Interconnects

- Communication based on shared-medium (e.g. bus)
  - Multiplexer-oriented topologies
- Pros
  - Simple topology, low area cost and extensibility
- Cons
  - Performance bottleneck
  - Scalability problem
  - Power consumption inefficient
  - Unpredictable performance



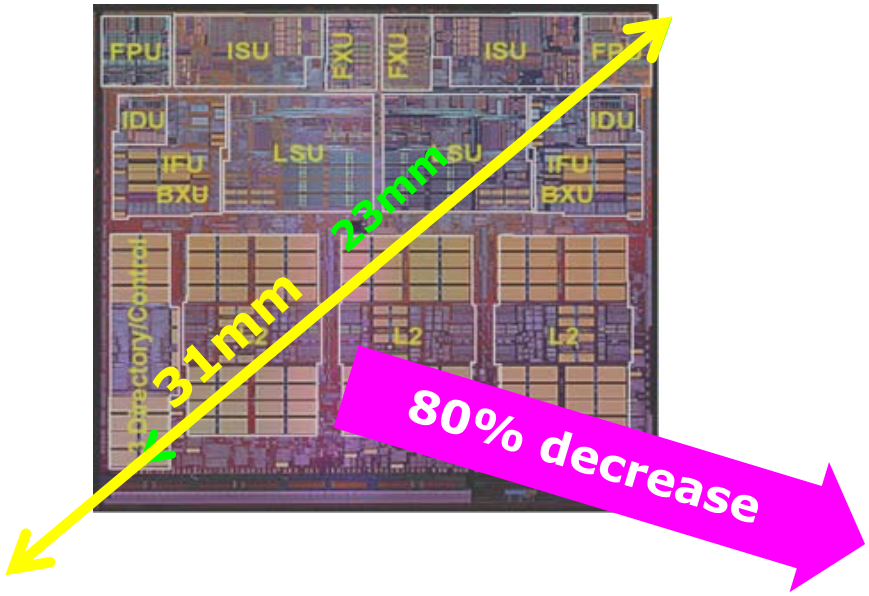
Properties Limiting the Use of Bus

- Wire delay
  - Wires become “longer”, and wire delay becomes a performance bottleneck
  - Partition a long wire in segments with repeaters
  - Synchronization problem
- Power
  - More energy consumption due to longer wires
  - To reduce delay, bigger drivers are used, which increase energy consumption
  - Typical solutions
    - Reduce voltage swing
      - Good for performance and power
      - But reduces noise margins => more errors!
    - Differential signaling
- Signal integrity
  - Growing capacitive and inductive coupling between wires
  - IR drop, Cross-talk, Electro-migration, ...

Reachable Physical Distance Per Clock

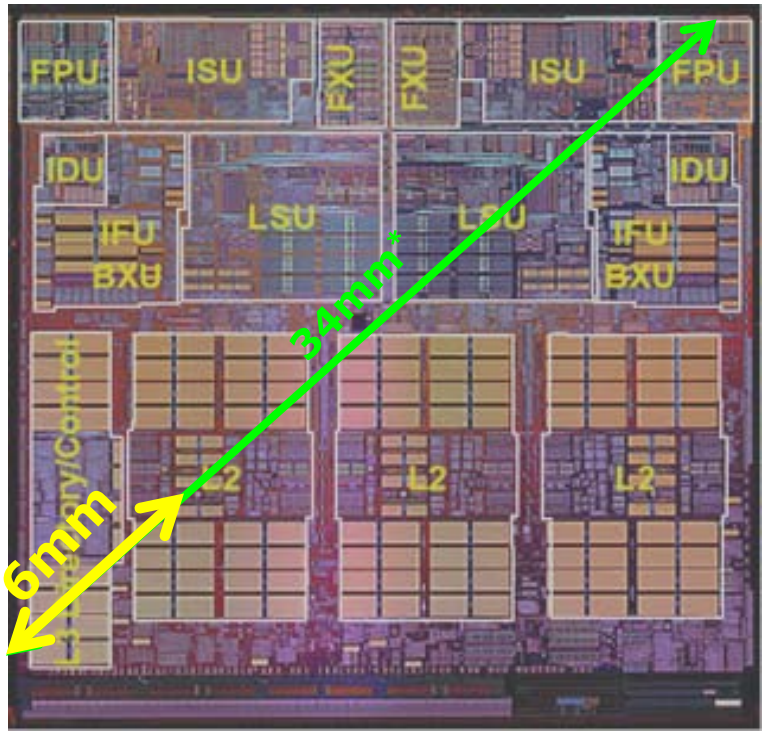
- Reachable physical distance within one clock

1GHz POWER4 (.18µm)



100% reachable in one clock

8GHz* POWER6 (.065µm)



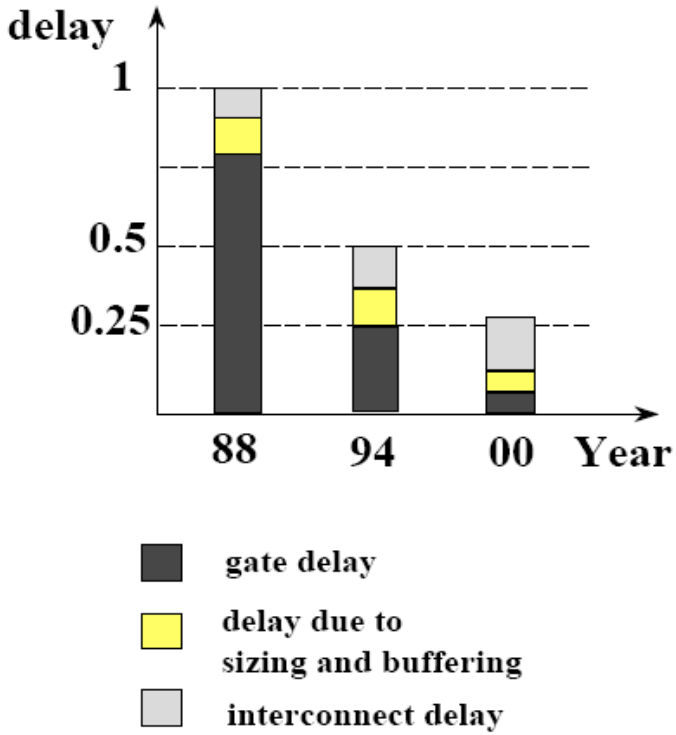
18% reachable in one clock

*** Projected values (Year 2006)**

Performance Impact of On-Chip Interconnect

Operation	Delay (0.13um)	(0.05um)
32b ALU Operation	650ps	250ps
32b Register Read	325ps	125ps
Read 32b from 8KB RAM	780ps	300ps
Transfer 32b across chip (10mm)	1400ps	2300ps
Transfer 32b across chip (20mm)	2800ps	4600ps

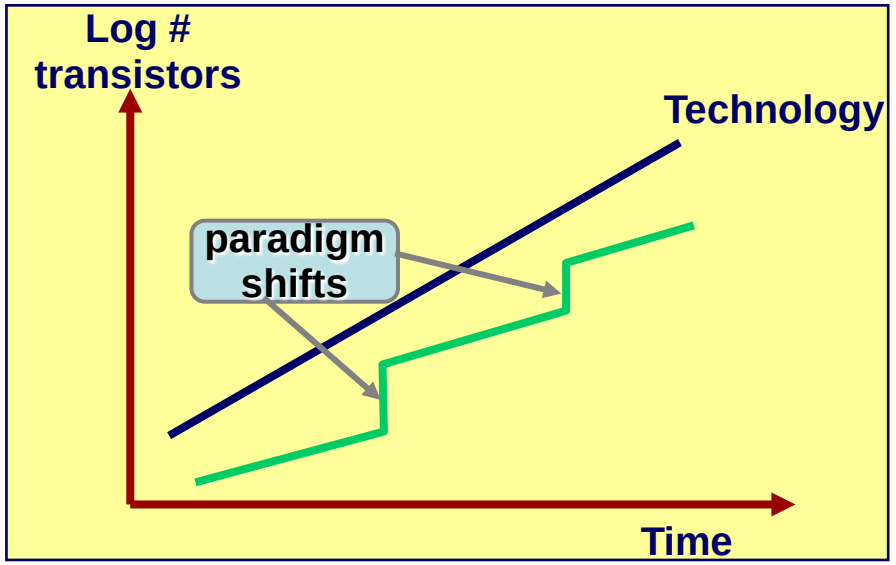
2:1 global on-chip comm to operation delay
9:1 in 2010



Taken from W.J. Dally presentation: Computer architecture is all about interconnect (it is now and it will be more so in 2010) HPCA Panel February 4, 2002

Challenges in SoC Design

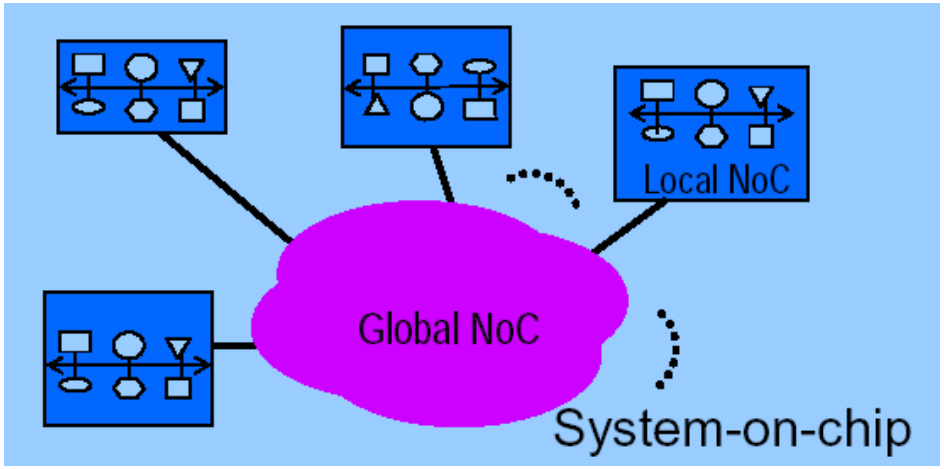
- Time-to-market pressure
  - Design productivity gap
- Complexity
  - Heterogeneous
  - Deep submicron effects
  - Performance/Energy/Cost tradeoff
  - Scalable architecture
- New Design Paradigm
  - IP/Platform-based design
  - Error tolerant design strategy
  - Interconnect oriented design

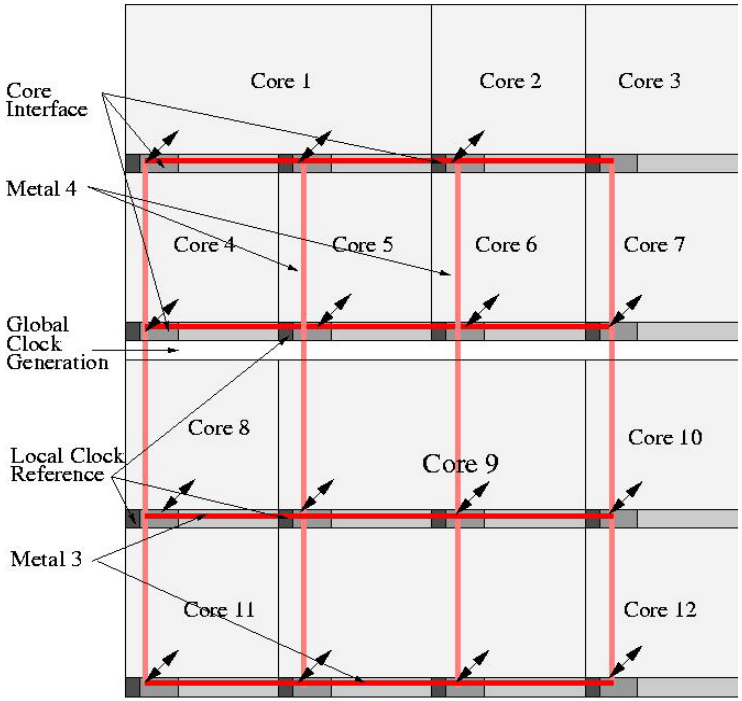


⇨ Paradigms shifts in design methodology is the only escape

Network on Chip

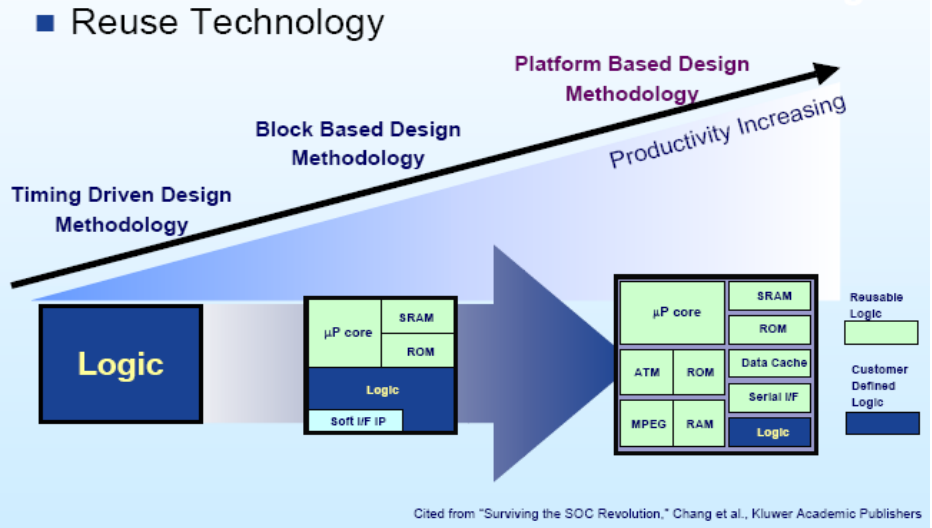
- Communication channels for computer systems or modules integrated within a single chip.
  - Resources are interconnected by a network of switches.
- Large-scale integration of SoCs with the scalable interconnects



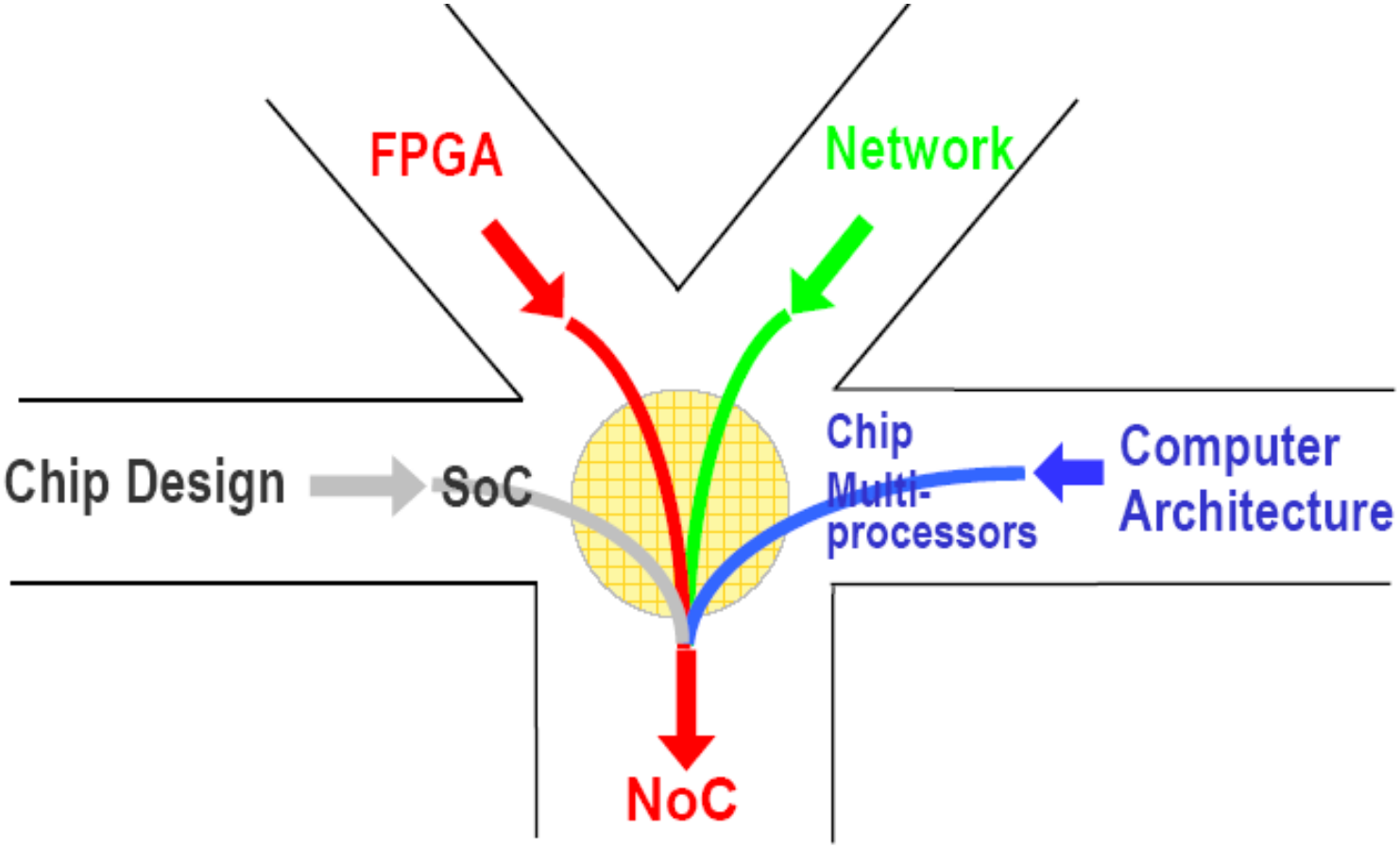


Advantages of NoC

- Reuse
  - Components and resources
  - Communication platform
  - Design and verification time
- Predictability
  - Communication performance
  - Electrical properties
- Scalability
  - Computing, Memory and Interconnection Resources
- Modular, compositional
  - Decoupling computation and communication



Yet Another Interconnection Network

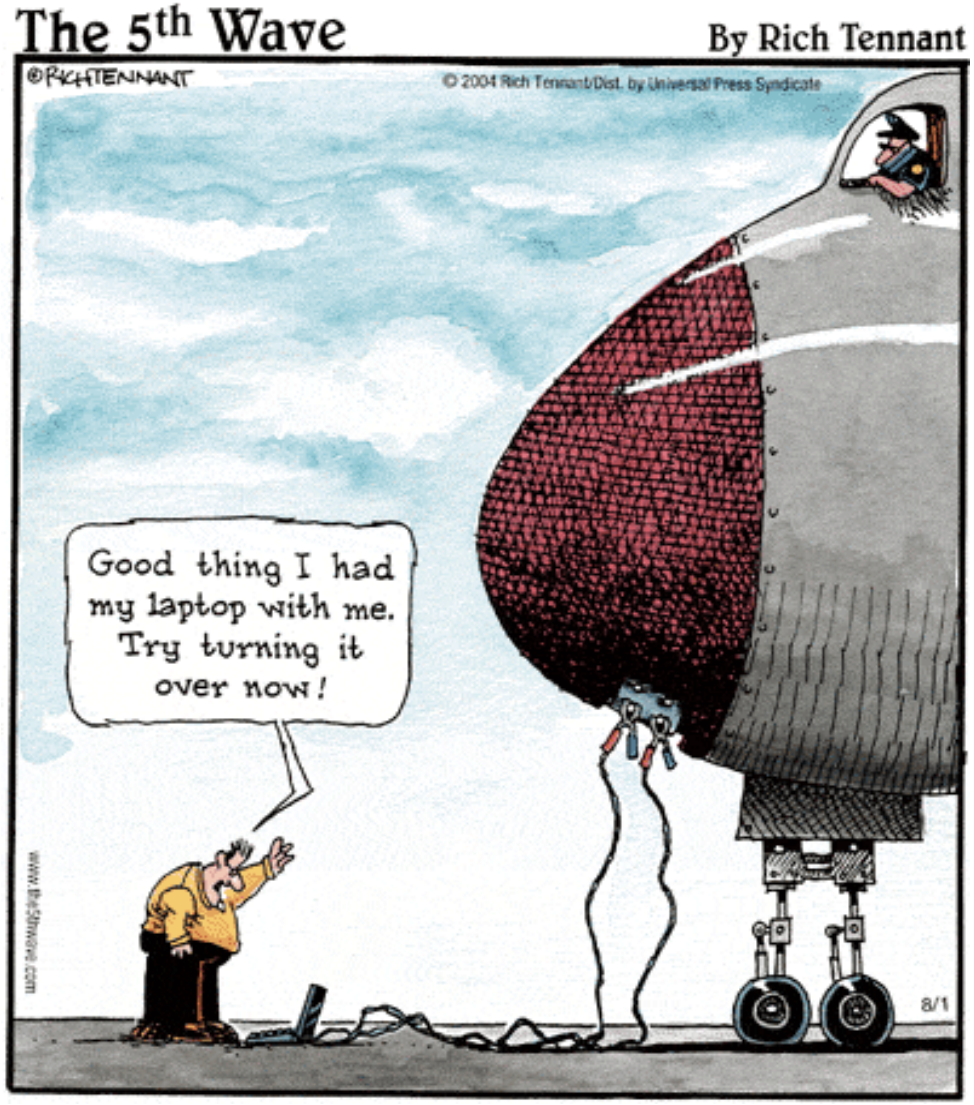


(Networks-on-a-Chip;
On-chip interconnect networks)

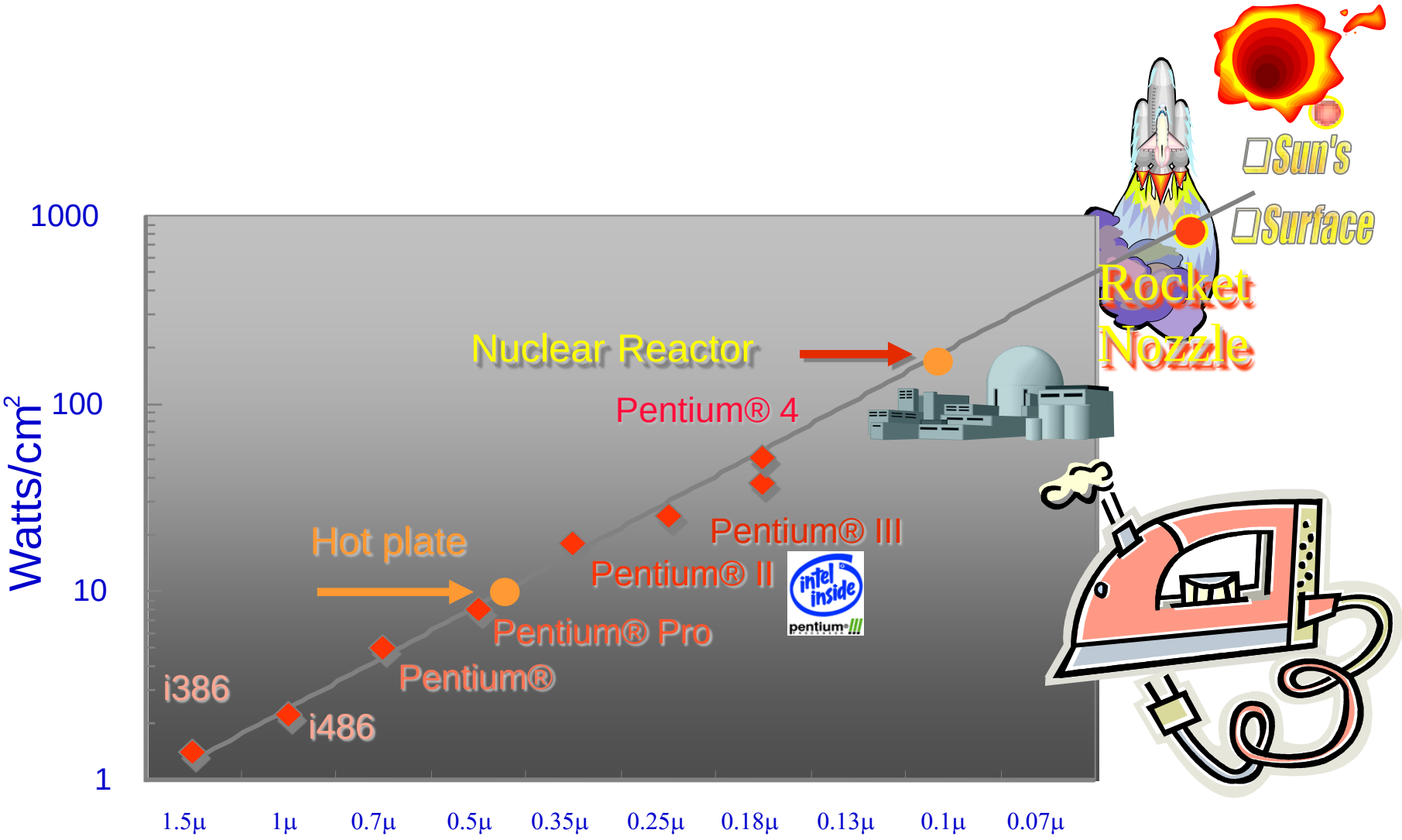
Outline

- 연구실 소개
- SoC architecture
- System-level low power design

High Energy Consumption incurs ...

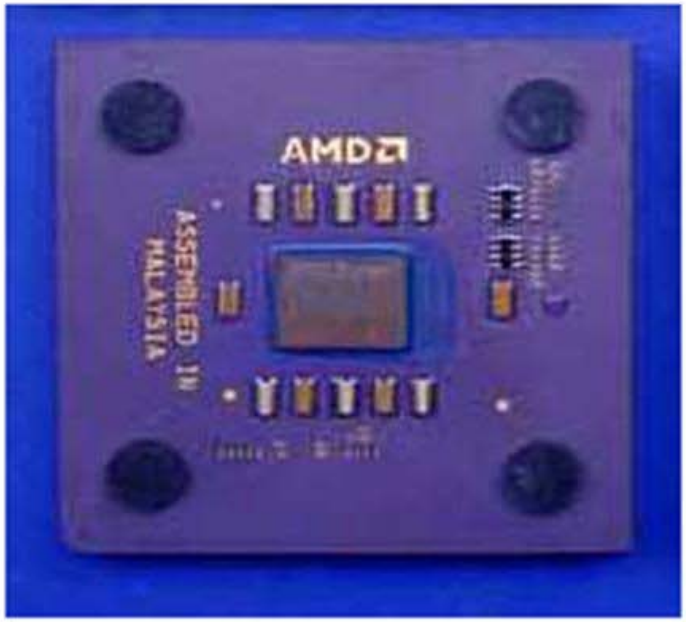


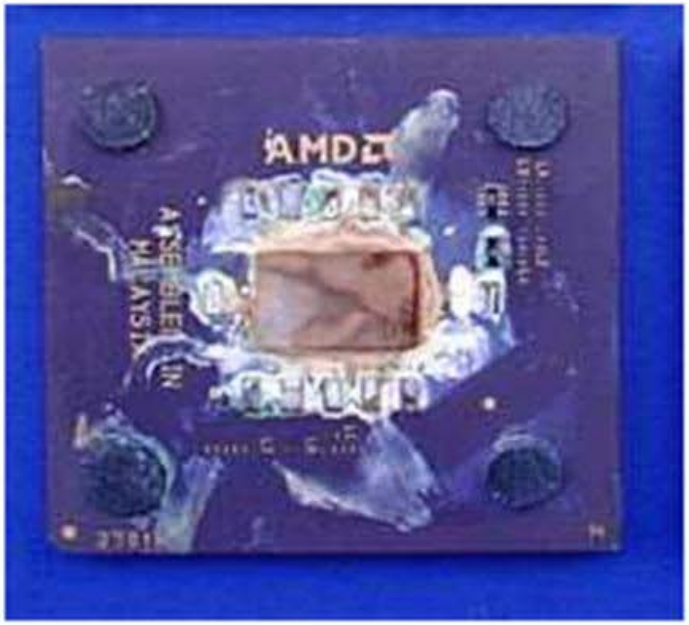
Hot Chips are No Longer Cool!



* "New Microarchitecture Challenges in the Coming Generations of CMOS Process Technologies" – Fred Pollack, Intel Corp. Micro32 conference key note - 1999.

High Temperature incurs ...



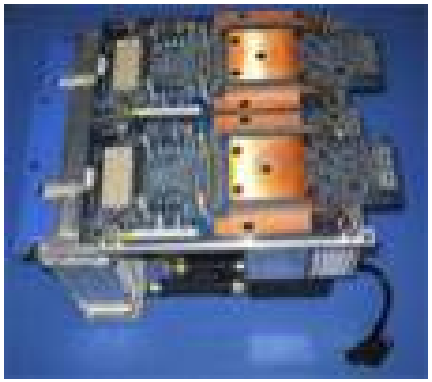


Source: Tom's Hardware Guide
http://www6.tomshardware.com/cpu/01q3/010917/heatvideo-01.html

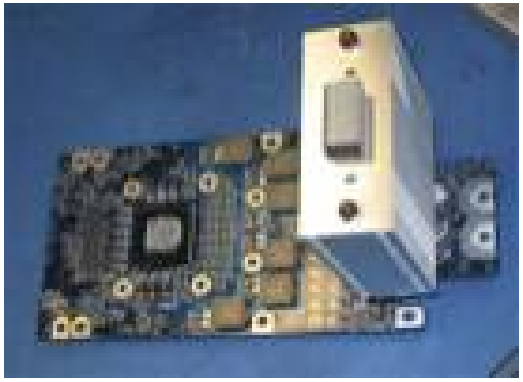
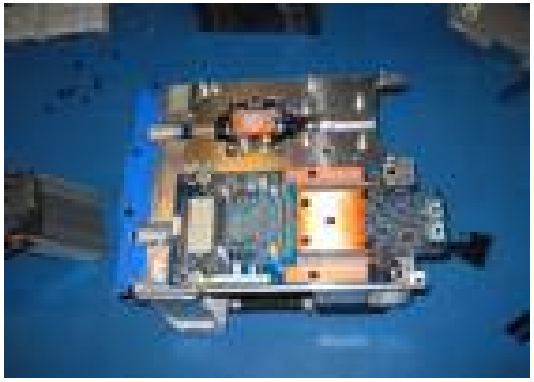
What's the benefit from High Temperature?

Solutions from Mechanical Engineering







Design challenges: ITRS predicts ... (I)

Table 9 System Functional Requirements for the PDA SOC-LP Driver

Year of Production	*2003*	*2006*	*2009*	*2012*	*2015*	*2018*
Process Technology (nm)	101	90	65	45	32	22
Supply Voltage (V)	1.2	1	0.8	0.6	0.5	0.4
Clock Frequency (MHz)	300	450	600	900	1200	1500
Application (maximum required performance)	Still Image Processing	Real Time Video Codec (MPEG4/CIF)		Real Time Interpretation		
Application (other)	Web Browser Electric Mailer Scheduler	TV Telephone (1:1) Voice Recognition (Input) Authentication (Crypto Engine)		TV Telephone (>3:1) Voice Recognition (Operation)		
Processing Performance (GOPS)	0.3	2	14	77	461	2458
Required Average Power (W)	0.1	0.1	0.1	0.1	0.1	0.1
Required Standby Power (mW)	2	2	2	2	2	2
Battery Capacity (Wh/Kg)	120	200	200	400	400	400

No increase in power consumption required!

Design Challenges: ITRS predicts ... (II)

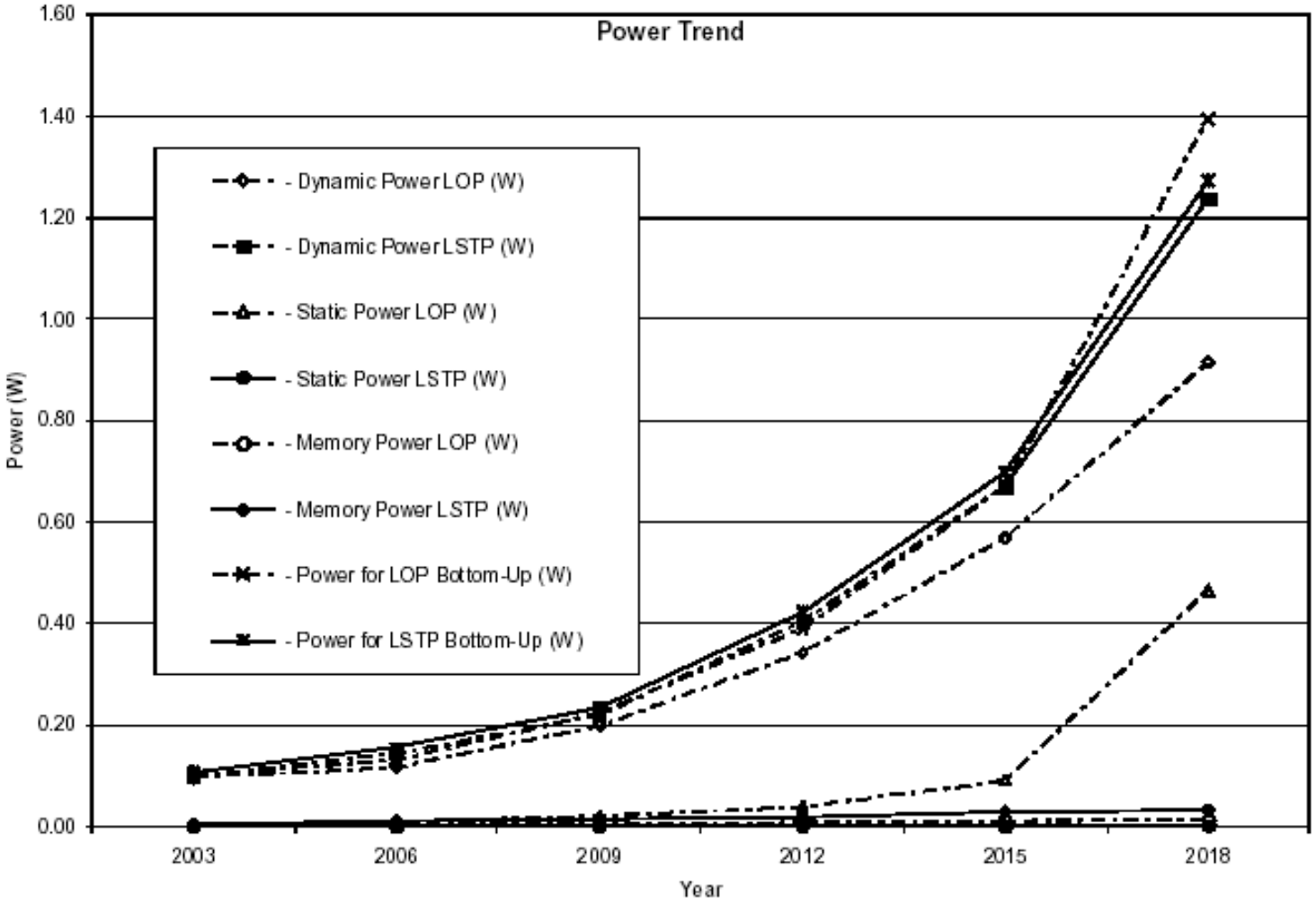


Figure 10 Total Chip Power Trend for SOC-LP PDA Application

Motivation: In 2003, ITRS predicts … (III)

Table 10 Power Management Gap for SOC LP-PDA

	2003	2006	2009	2012	2015	2018
Total LOP Dynamic Power Gap (X)	0.0	0.2	1	2.4	4.7	8.1
Total LSTP Dynamic Power Gap (X)	0.0	0.4	1.2	3.00	5.7	11.4
Total LOP Standby Power Gap (X)	0.37	3.44	8.73	18.79	44.38	231.9
Total LSTP Standby Power Gap (X)	-0.98	-0.96	-0.90	-0.78	-0.53	0.10

- **DP: (Total power – 0.1W) / 0.1W**
- **SP: (Total standby power – 2mW) / 2mW**

- **Need to provide power management scheme**
  - **At the levels of application, OS, architecture, IC**

Motivation of low power design

- Enhance operation time of portable electronic devices
  - Batteries can release limited *energy*
- Reduce heat generation in high-performance processors
  - High *power* consumption requires adequate dissipation
- Reduce energy cost
- Determine *power/performance* spectrum

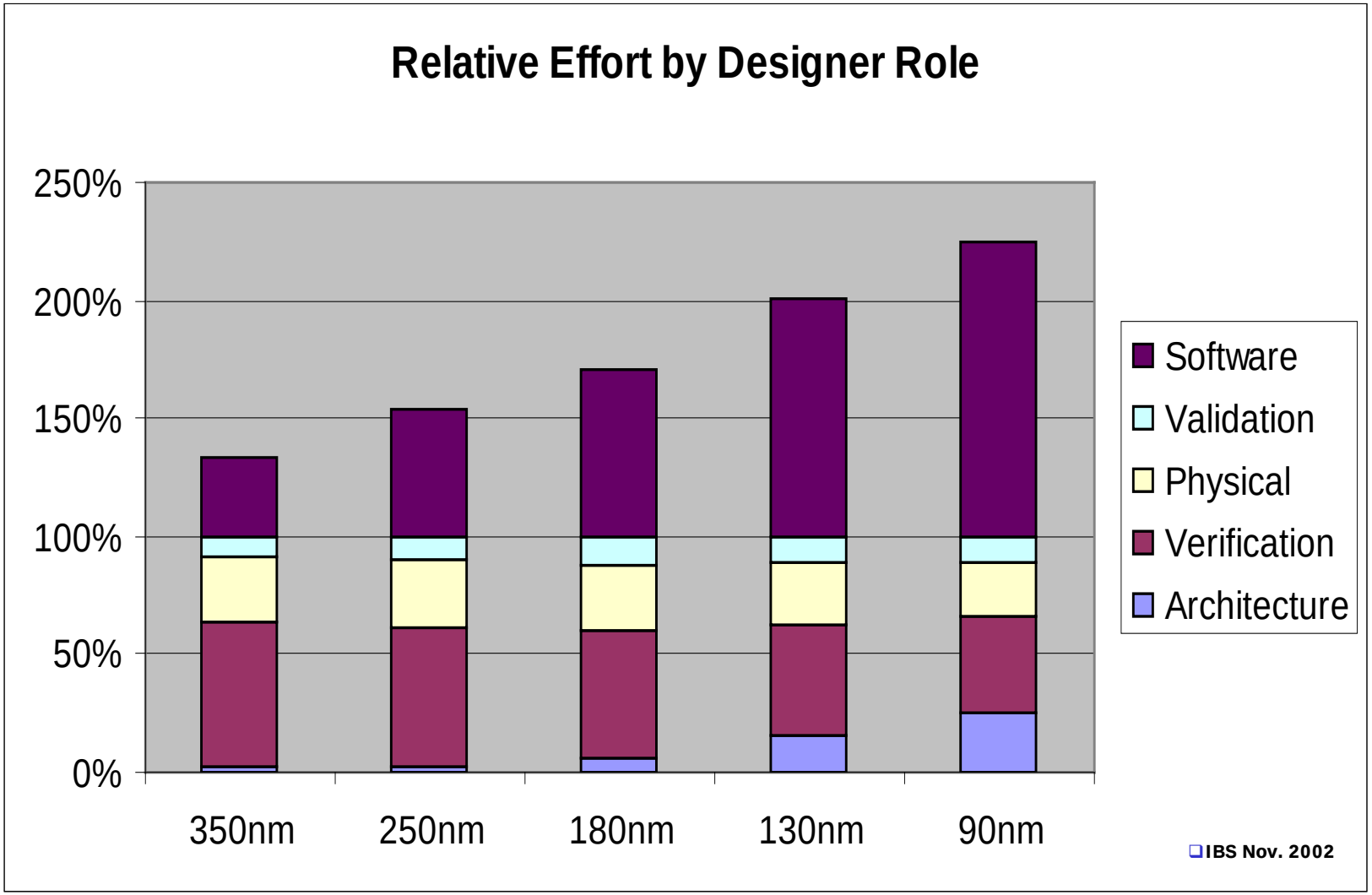
Target system and power reduction techniques

- **Target system: processor based systems**

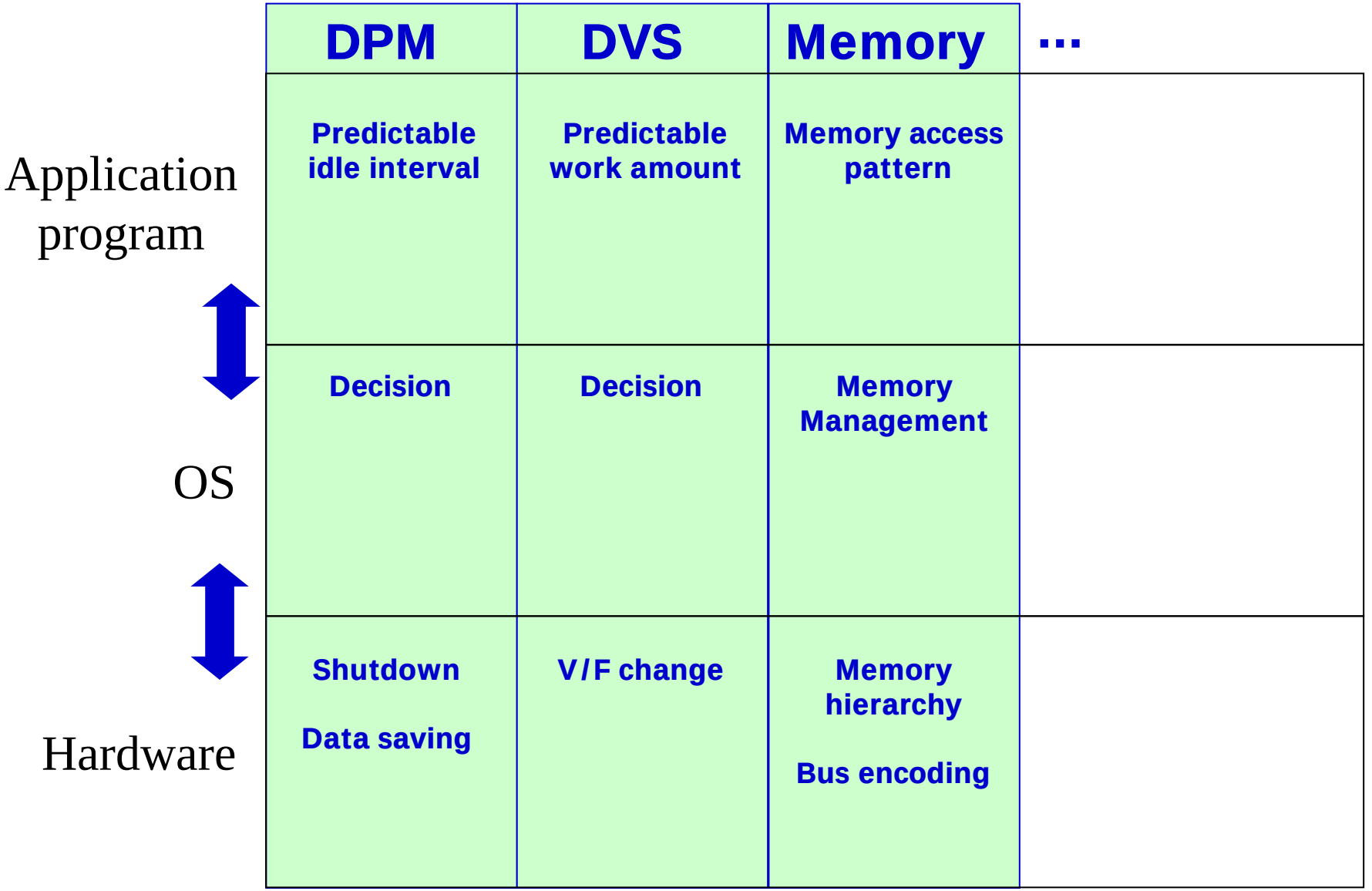
Application program	➡ Algorithm implementation
Operating system	➡ Uniform interface / Resource manager
Hardware	➡ Actual energy consumer

Software (Application program and OS) control the behavior of actual energy consumer

Design effort breakdown

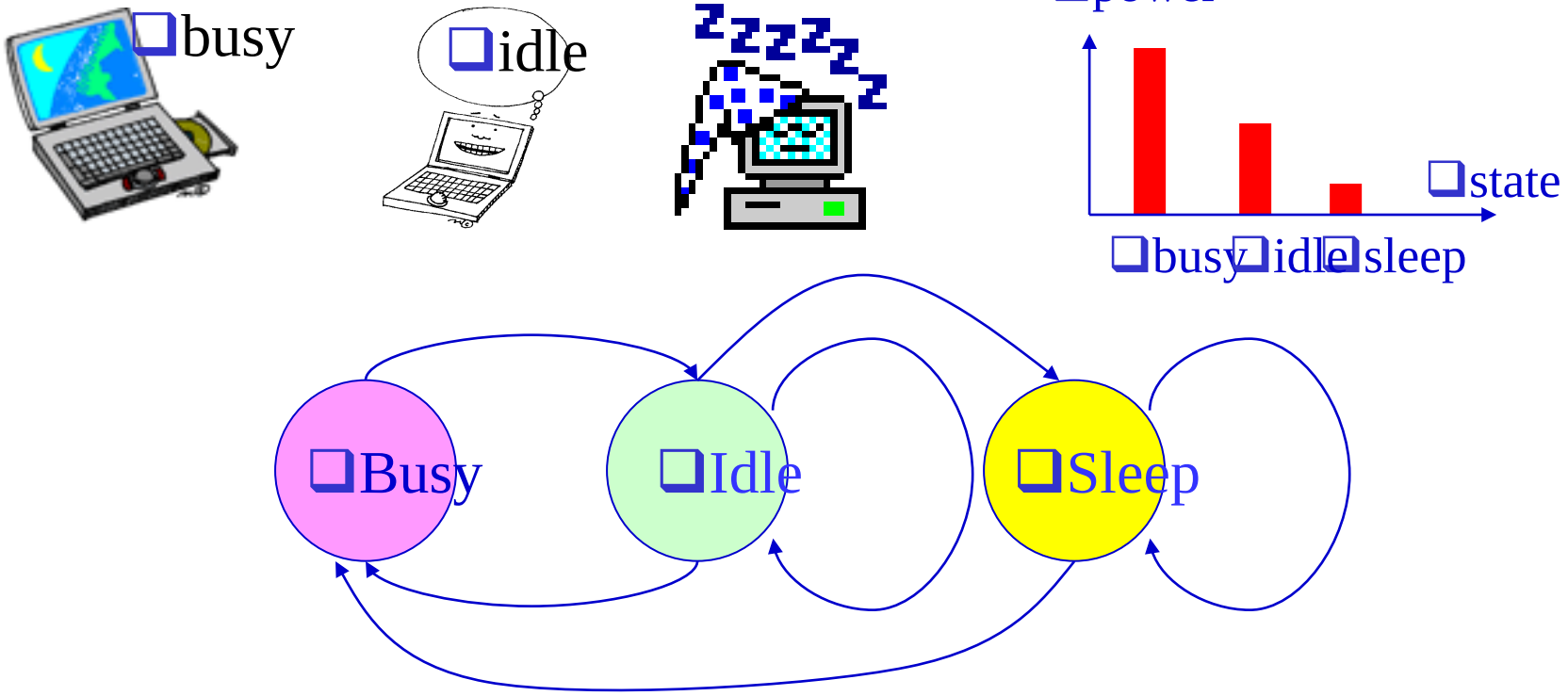


Various system-level low power techniques



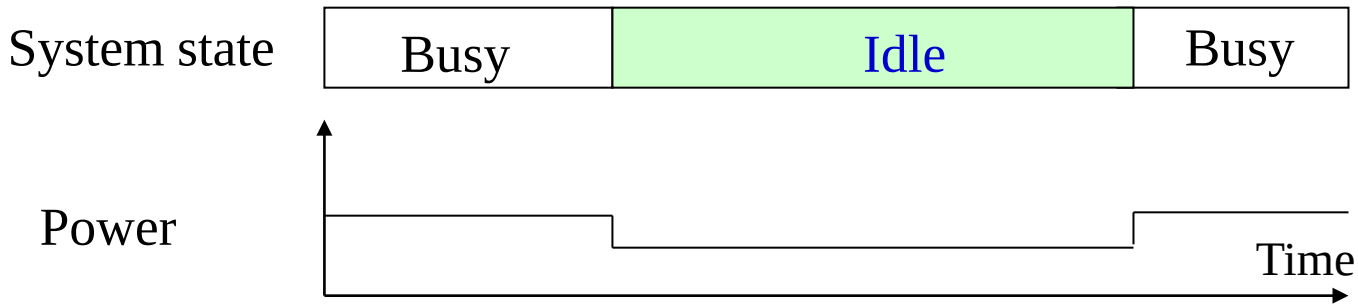
Dynamic Power Management

- **Shut down the system while the system is in idle state**

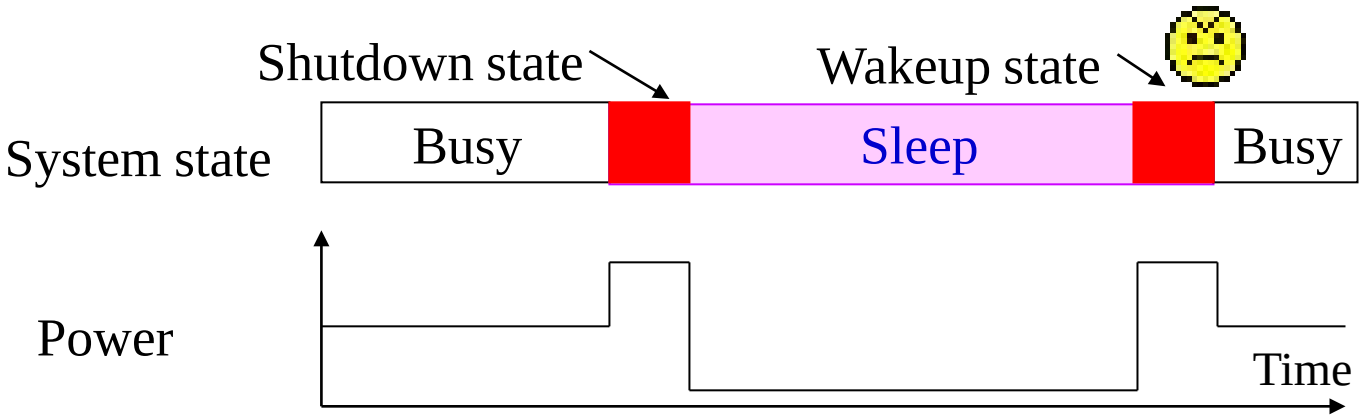


DPM overhead

Without DPM



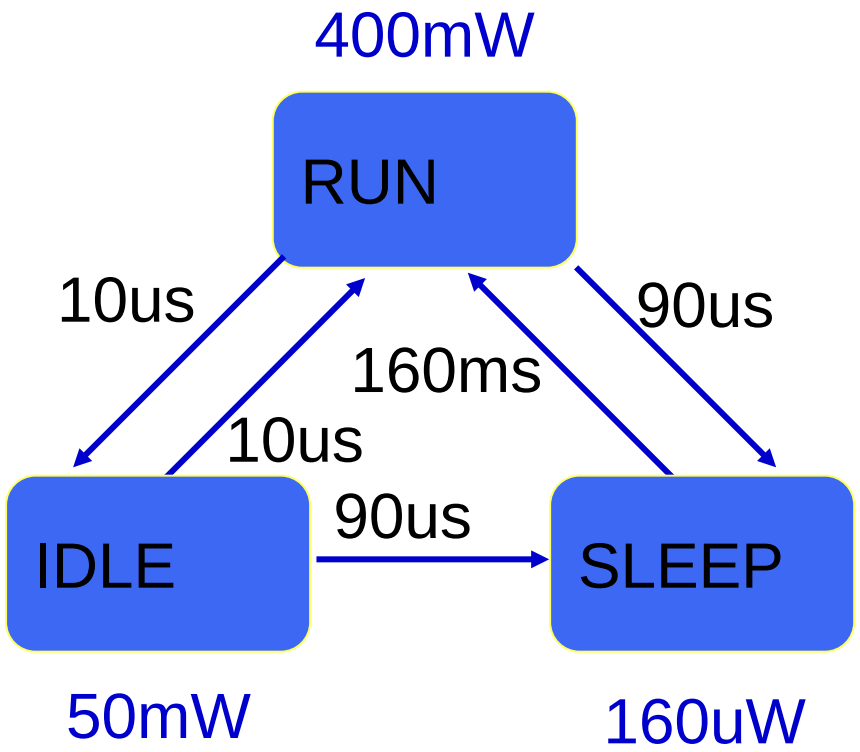
With DPM



Performance and energy overhead due to shutdown and wakeup

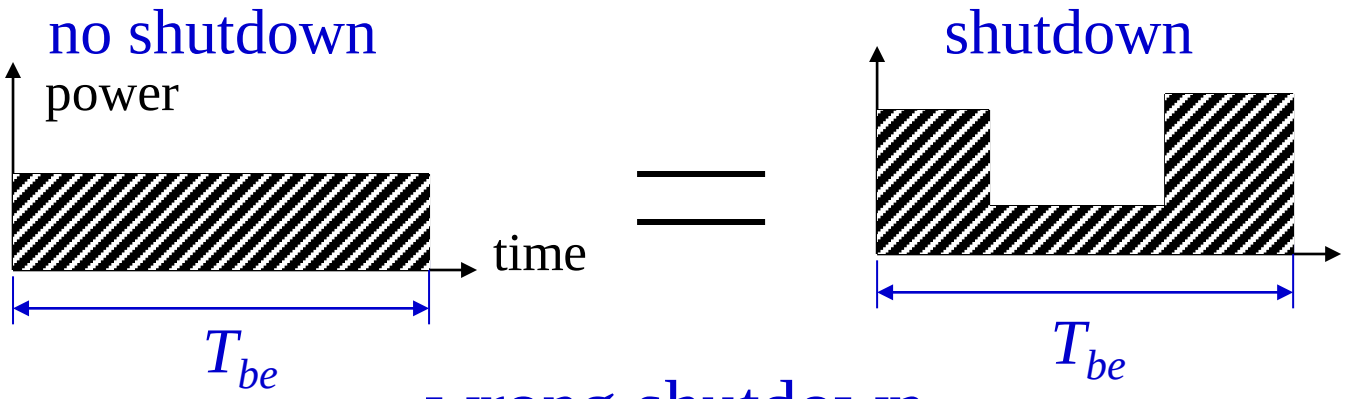
An example: STRONGARM SA1100

- RUN: operational
- IDLE: a sw routine may stop the CPU when not in use, while monitoring interrupts
- SLEEP: Shutdown of on-chip activity



Shutdown criteria

- Break even time : T_{be}
  - Shortest idle period for energy saving

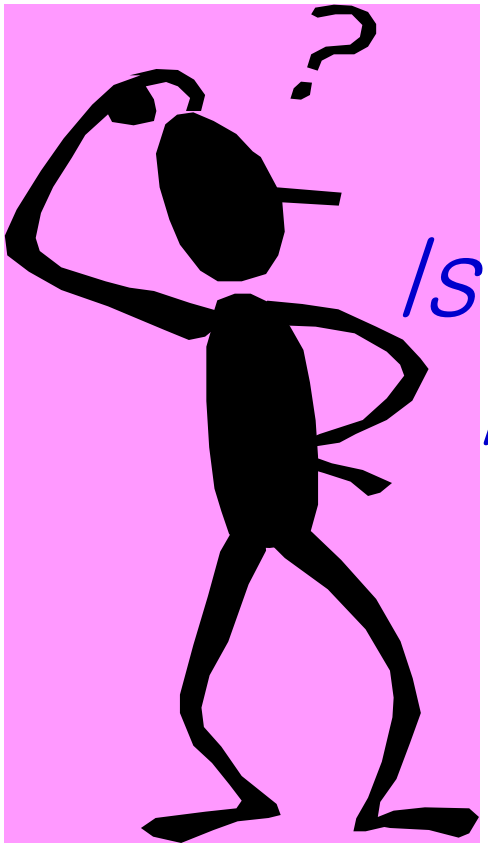


wrong shutdown

<

Idle period shorter than T_{be} is useless for energy saving

The challenge

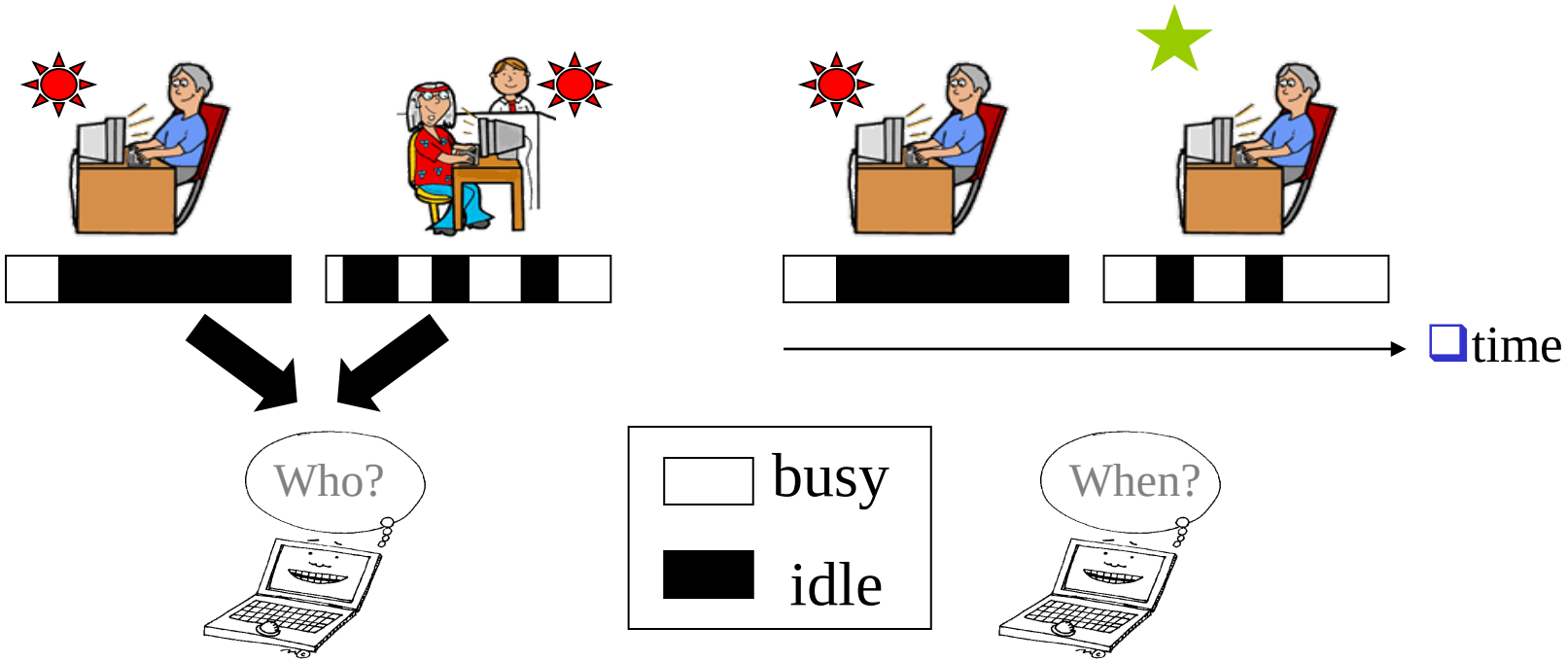


Is an idle period long enough for shutdown (T_{be})?

Predicting the future!

Uncertainty of idle period length

- Idle period is determined by user behavior

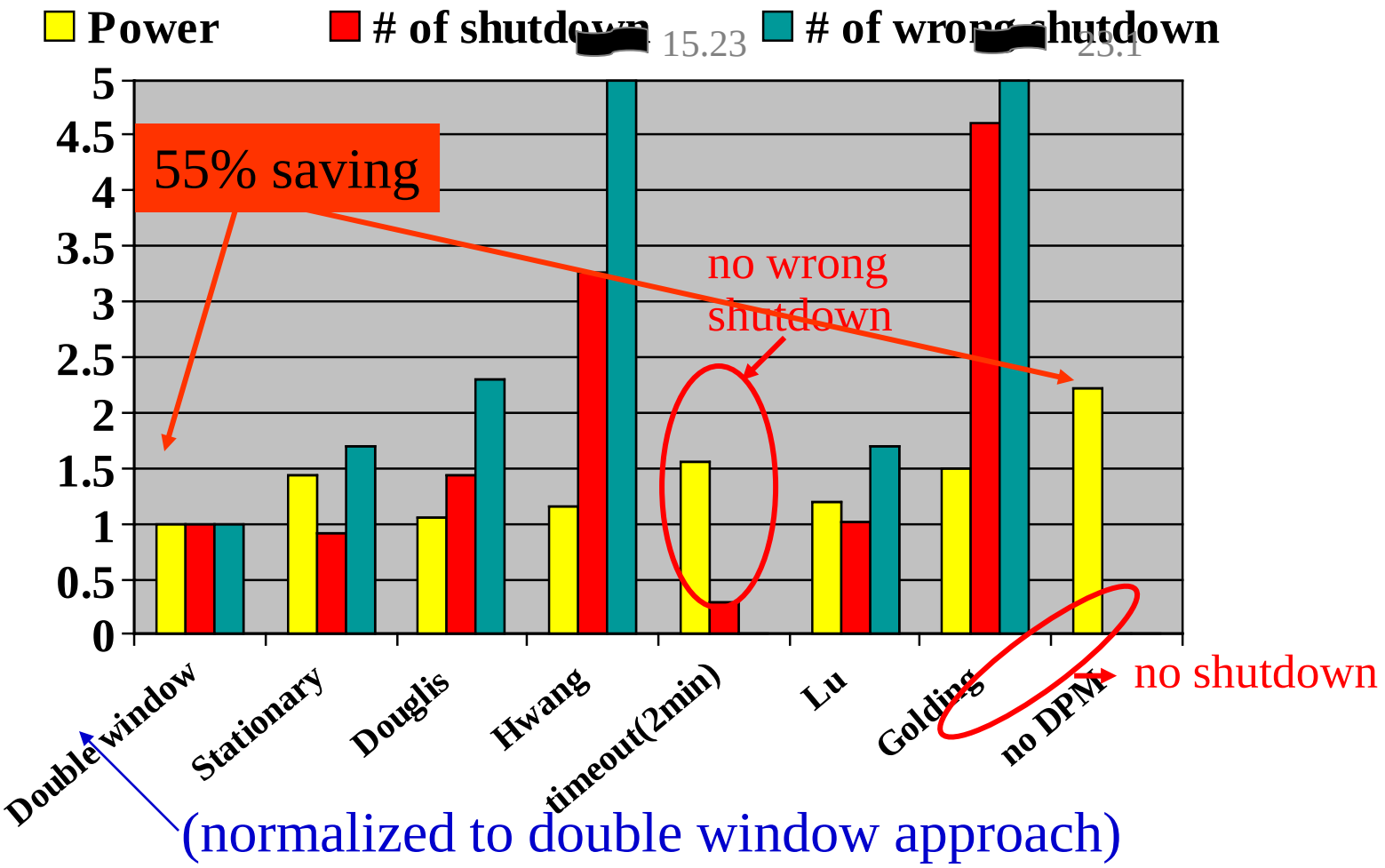


Need a technique to predict idle length!

Categories of DPM techniques

- **Timeout : [Karlin94, Douglis95, Li94, Krishnan99]**
  - Shutdown the system when timeout expires
  - Fixed vs. adaptive
- **Predictive : [Chung99, Golding95, Hwang00, Srivastava96]**
  - Shutdown the system if prediction is longer than T_{be}
- **Stochastic : [Chung99, Benini99, Qiu99, Simunic01]**
  - Model the system stochastically (Markov chain)
  - Policy optimization with constraints
    - Trade off between energy saving and performance
  - Non-deterministic decision
  - Discrete time model / Continuous time model
  - Superior to predictive and timeout

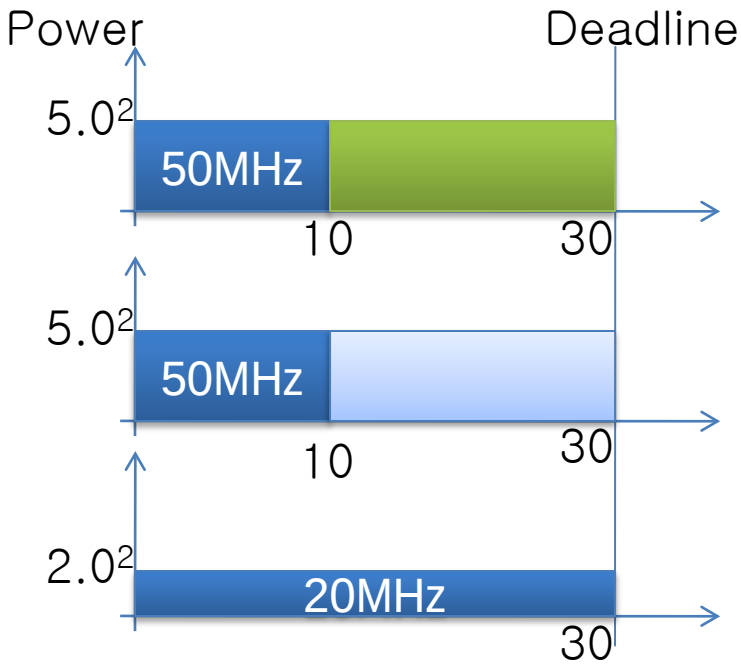
Measurement



Dynamic Voltage Scaling

Basic Idea of DVS

$$E \propto N_{cycle} \cdot V_{DD}^2 \qquad P = C \bullet V \bullet f^2$$



(a) No Power-down

- 15 × 10^8 cycle
- 5.0V
- 37.5J

(b) Power-down (DTM)

- 5 × 10^8 cycle
- 5.0V
- 12.5J

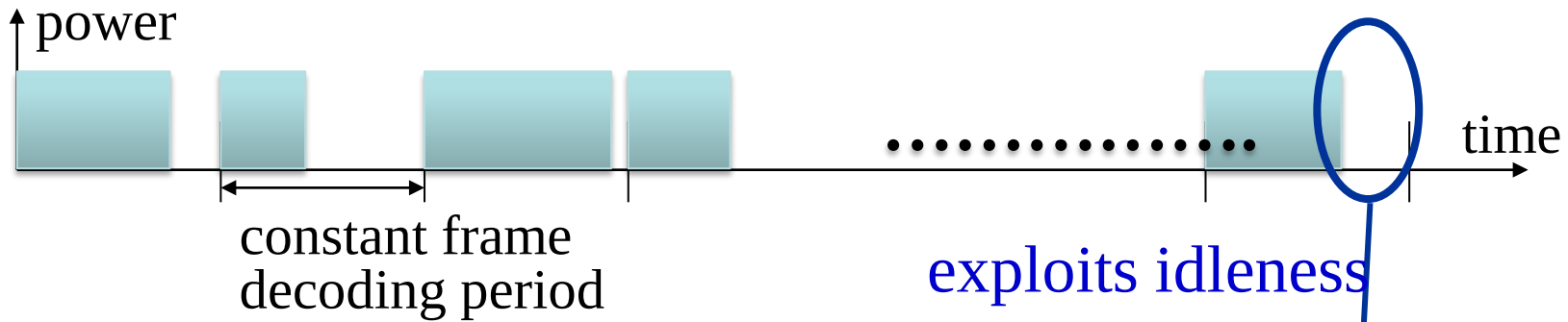
(c) DVS

- 6 × 10^8 cycle
- 2.0V
- **2.4J**

» Slow and Steady wins the race!

Key Issues for successful DVS

- **Efficient detection of slack/idle intervals**
- **Efficient voltage scaling policy for slack intervals**



- **Decoding power variation with constant time**

Dynamic Voltage Scaling

- **Key Question**
  - **How can we predict the future workload?**
  - **Workload Estimation Technique is needed**
    - An amount of power consumption in a certain applications is extremely different according to types of workloads.
    - A large variation of workloads is a challenging problem to achieve low-power consumption in portable devices.

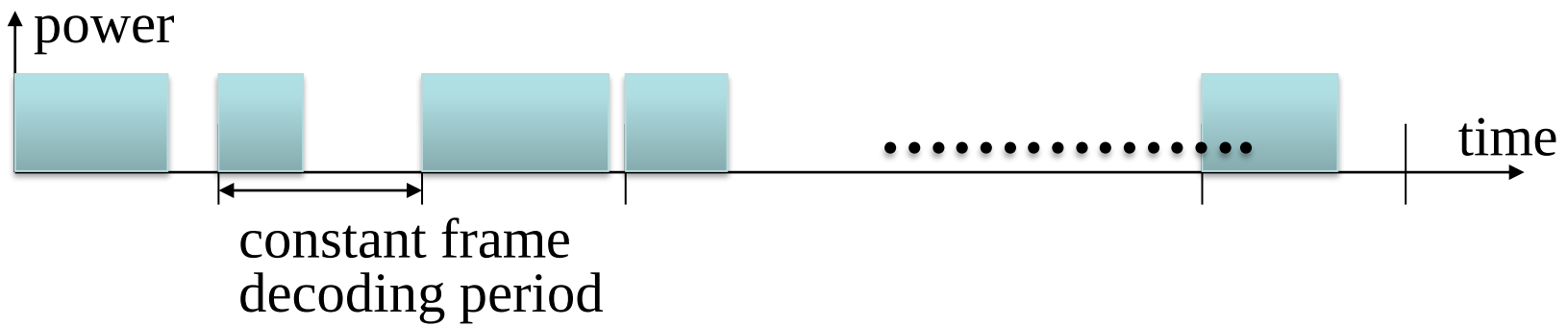
Two type of DVS algorithms

- **Inter-task DVS**
  - Scaling occurs at the start of a task
    - It is unchanged until the task is completed
  - Use worst-case slack time (= $\text{Deadline}_{task} - \text{WCET}_{task}$)
  - Usually used in multi-task scheduling scenario at OS level
- **Intra-task DVS**
  - Scaling occurs at the sub-task level
    - Different frequency is set for each sub-task
  - Use workload-variation slack time

	Voltage scaling method	Scaling decision
Inter-DVS	Linear method	Off-line
	Filter-based method	On-line
Intra-DVS	Path-based method	Off-line
	Stochastic method	

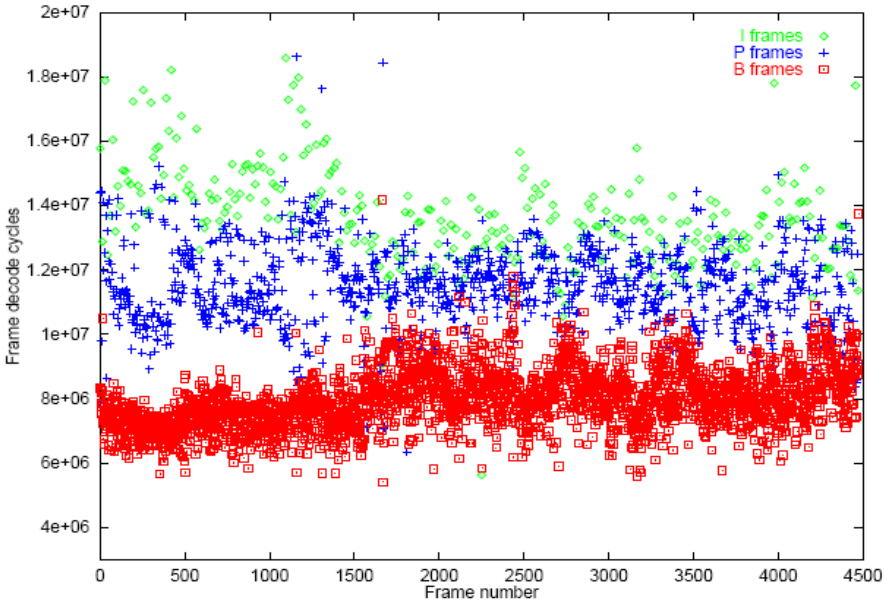
DVS for Multimedia

- Multimedia applications are required to process each unit of data, typically called a **frame**, within a time limit called a **deadline**.
- **The processor may complete a frame's computation before the deadline.**
- The existence of this idle time, called *slack*, implies that the processor can be slowed to save energy.

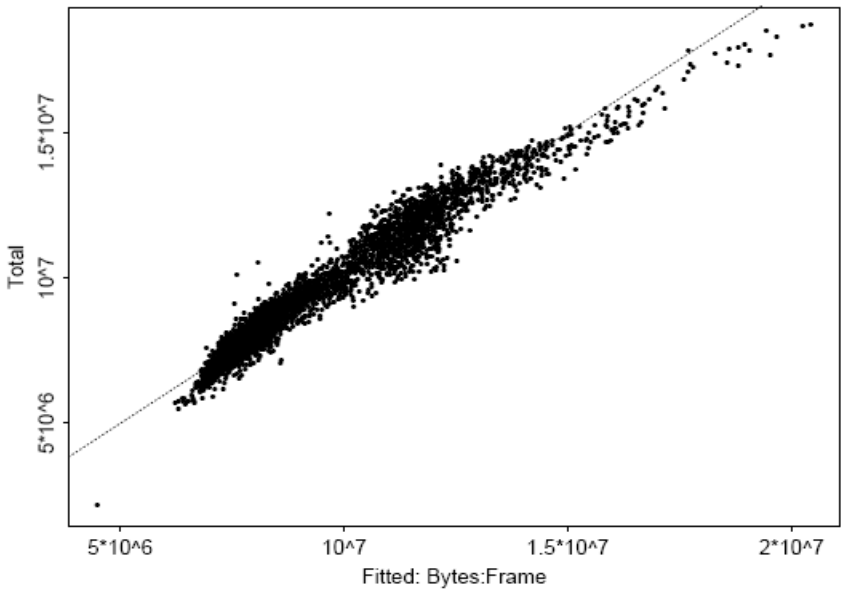


Linear Model

- **Linear model between workload size and timing information [6]**
  - By considering both macroblock types and frame length, it is possible to define tight fitting linear models of MPEG decoding, with R^2 values of 0.97 and above.



❑Frame Decode Times Separated by Frame Type



❑Decode Time As a Function of Frame Length and Type

Inter-task – Filter-based Technique

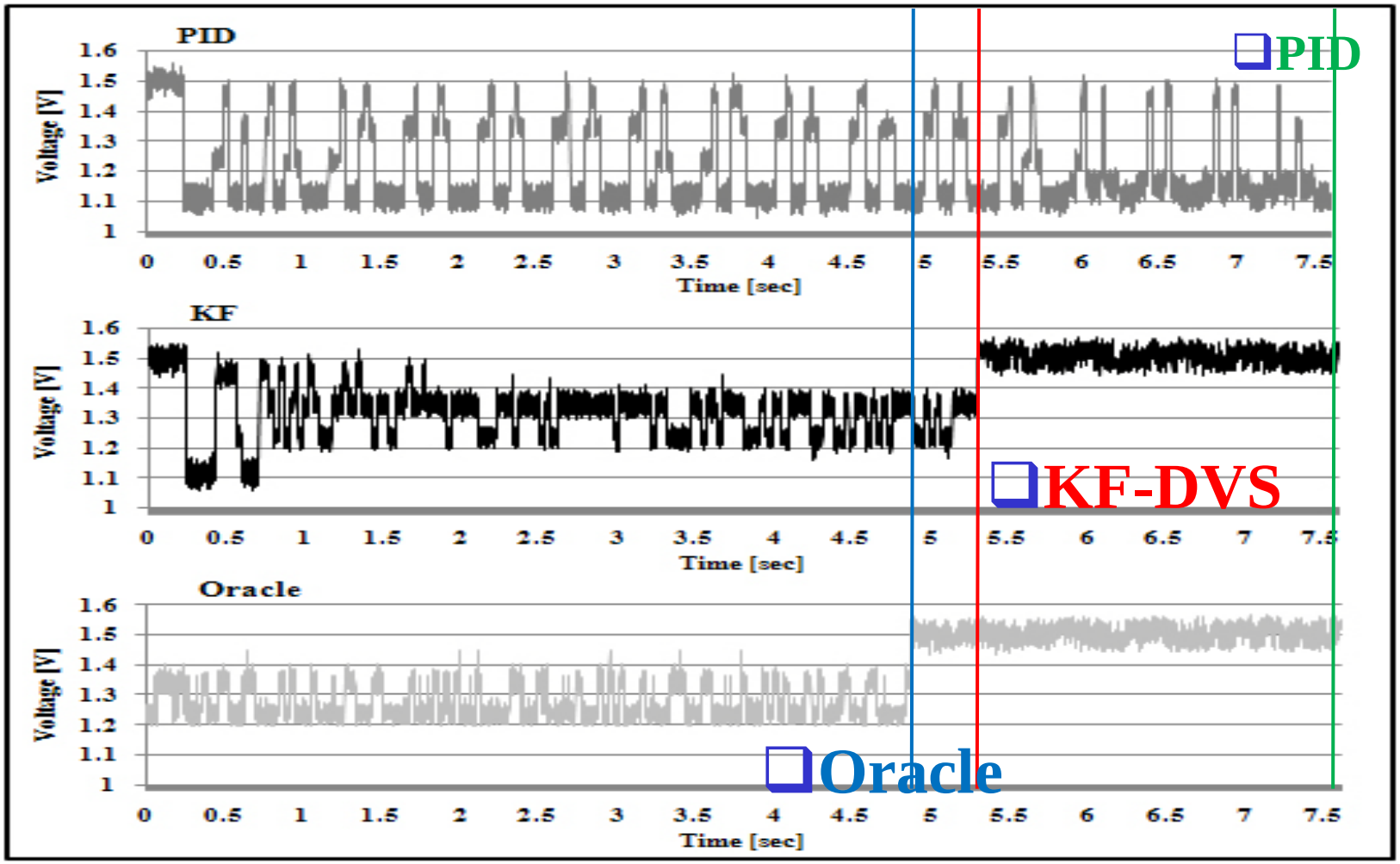
- **Moving Average (MA)**
  - The simplest filter is a time-invariant moving average filter.
- **Exponential Weighted Average (EWA)**
  - This filter is based on the idea that effect of workload k-slots before the current slot lessens as k increases lesser weight to the one before and so on.
- **Least Mean Square (LMS)**
  - It makes more sense to have an adaptive filter whose coefficients are modified based on the prediction error.
- **Proportional – Integral – Derivative controller (PID)**
  - This is generally used for error correction in estimation.
  - Previous estimation error : $\varepsilon_i = w_i - \hat{w}_i$ (w_i: i-th actual workload, $\hat{w}_i$: i-th estimation)
  - A PID controller calculates correction value Δw_i as

$$\Delta w_i = k_P \varepsilon_i + \frac{1}{k_I} \sum_{i}^{W_I} \varepsilon_i + k_D \left(\frac{\varepsilon_i - \varepsilon_{i-W_D}}{W_D} \right)$$

  - Estimation for next workload : $\hat{w}_{i+1} = \hat{w}_i + \Delta w_i$

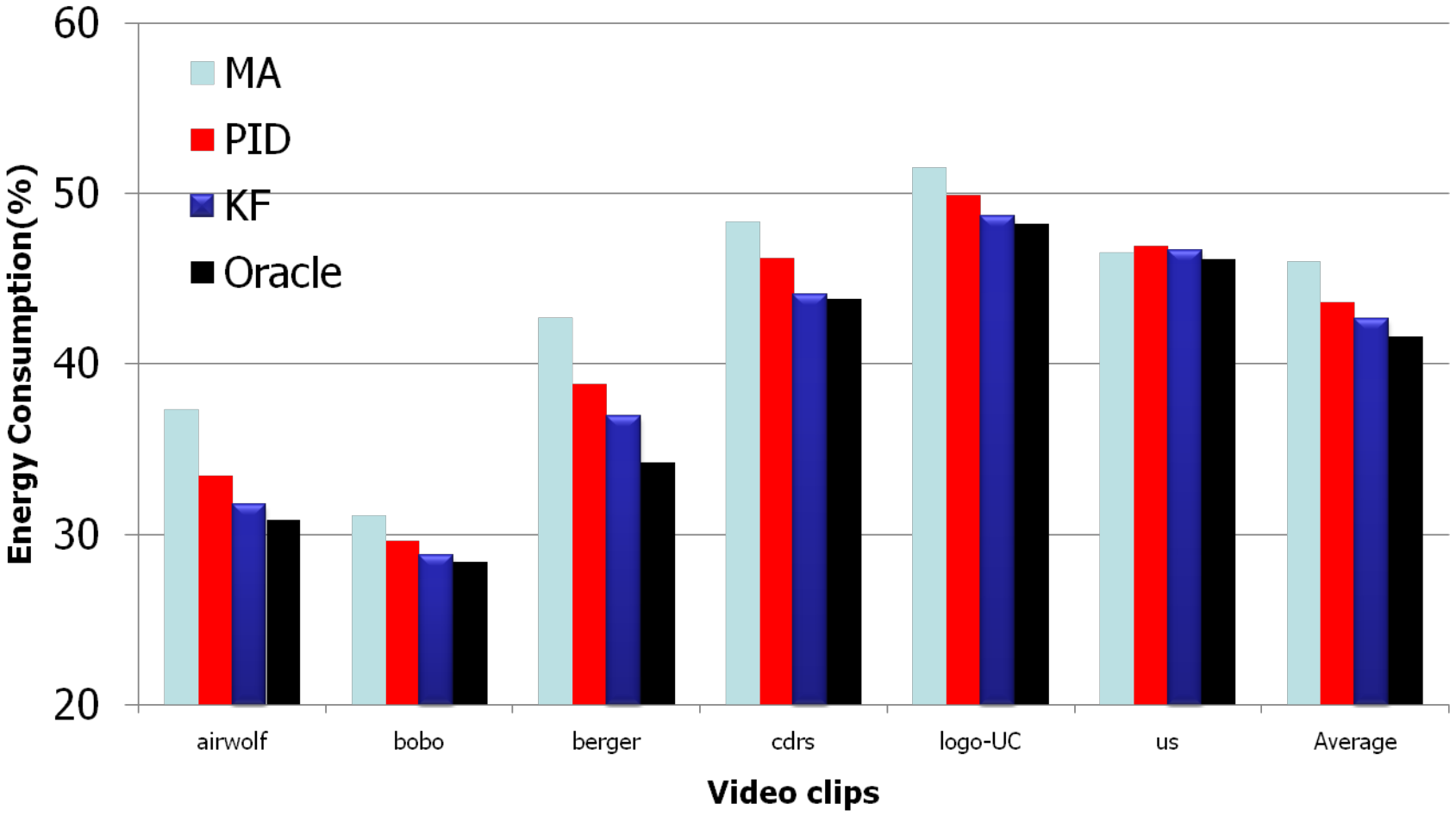
Inter-task DVS with Kalman Filter

- **Voltage selection and decoding time comparison of PID, KF, and oracle method**



Inter-task DVS with Kalman Filter

- **Experimental Results**



Summary

- Diverse research area in SoC Design
- Architecture design becomes more critical
  - On-Chip communication architecture for multi- and many cores
- Low power is must
  - A single technique cannot satisfy the requirement
  - System-level techniques show large effect