Embedded Computing Systems for Signal Processing Applications

Part 1: Introduction

October 19th 2012

Eric Debes

Introduction

- What is this about?
  - Introduction to power/performance tradeoffs and system architecture
  - Overview of existing processor and system architectures
  - Consumer vs. Industrial/Embedded
- Why do we care?
  - Engineering added value is in complex and critical system architecture
  - Need to know different components available
  - Software/Hardware System Architecture and Modelling
  - Power/Performance/Price Tradeoffs
- What's the plan?

Outline

1. Introduction
2. General-Purpose Processors and Parallelism
3. Application Specific Processors: DSPs, FPGAs, accelerators, SoCs
4. PC Architecture vs. Embedded System Architecture
5. Hard Real-time Systems and RTOS
6. Power Constraints
7. Critical and Complex Systems, MDE, MDA

Embedded Real-time Systems

- Embedded
  - Size and thermal constraints
  - Sometime battery life (energy) constraints
- Real-time
  - Time constraints
  - Can be hard real-time
  - Or soft-real time
- Systems
  - Typically includes multiple components
  - Requires different expertises:
    - Signal Processing, computer vision, machine learning/Cognition and other algorithmic expertise
    - Software Architecture
    - Hardware/Computing Architecture
    - Thermal and mechanical engineering

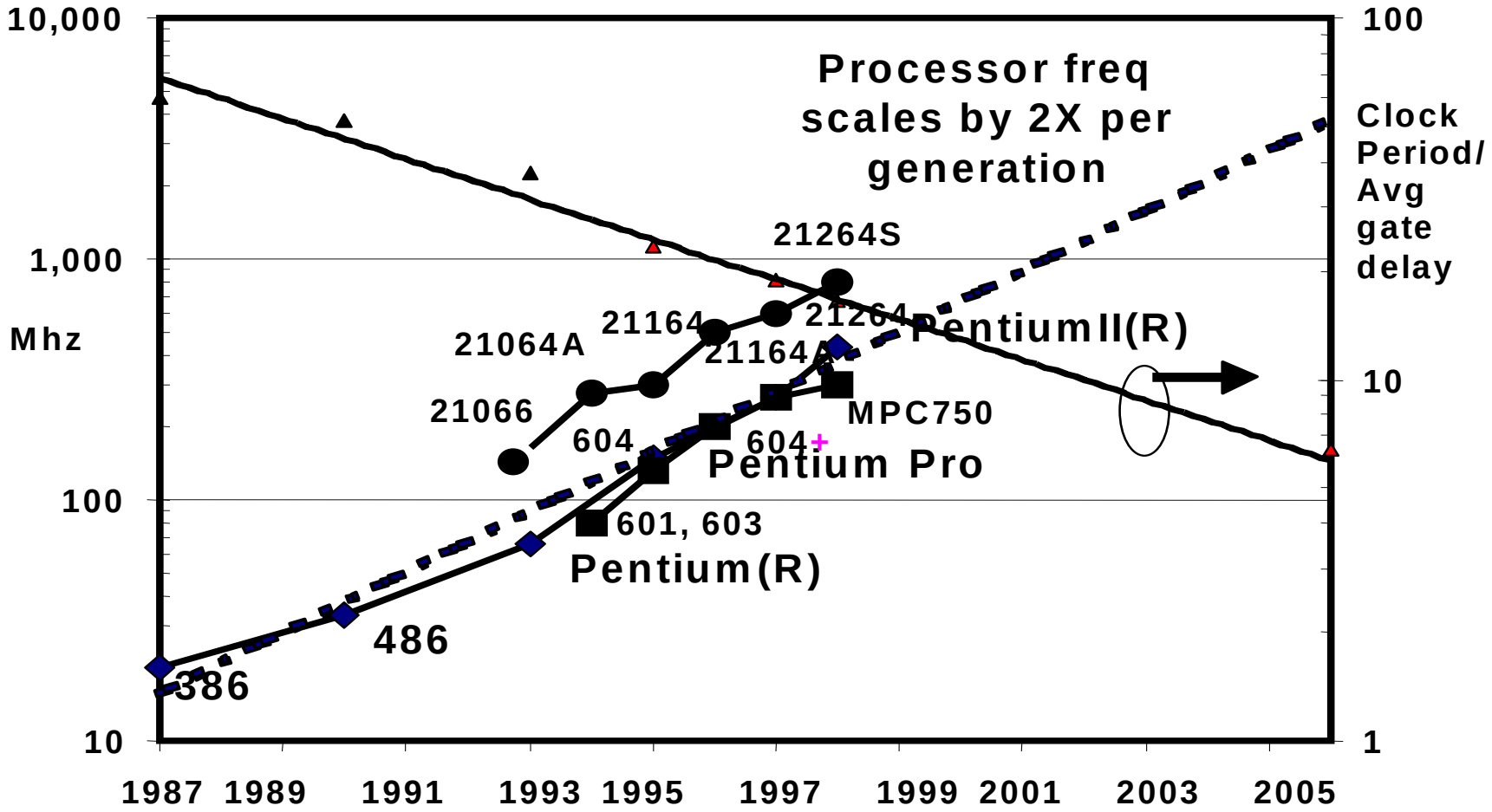
Application Examples

- Consumer : DVD/video players, Set-top-box, Playstation, printers, disk drives, GPS, cameras, mp3 players
- Communications: Cellphone, Mobile Internet Devices, Netbooks, PDAs with WiFi, GSM/3G, WiMax, GPS, cameras, music/video
- Automotive: Driving innovation for many embedded applications, e.g. Sensors, buses, info-tainment
- Industrial Applications: Process control, Instrumentation
- Other niche markets: video surveillance, satellites, airplanes, sonars, radars, military applications

Performance

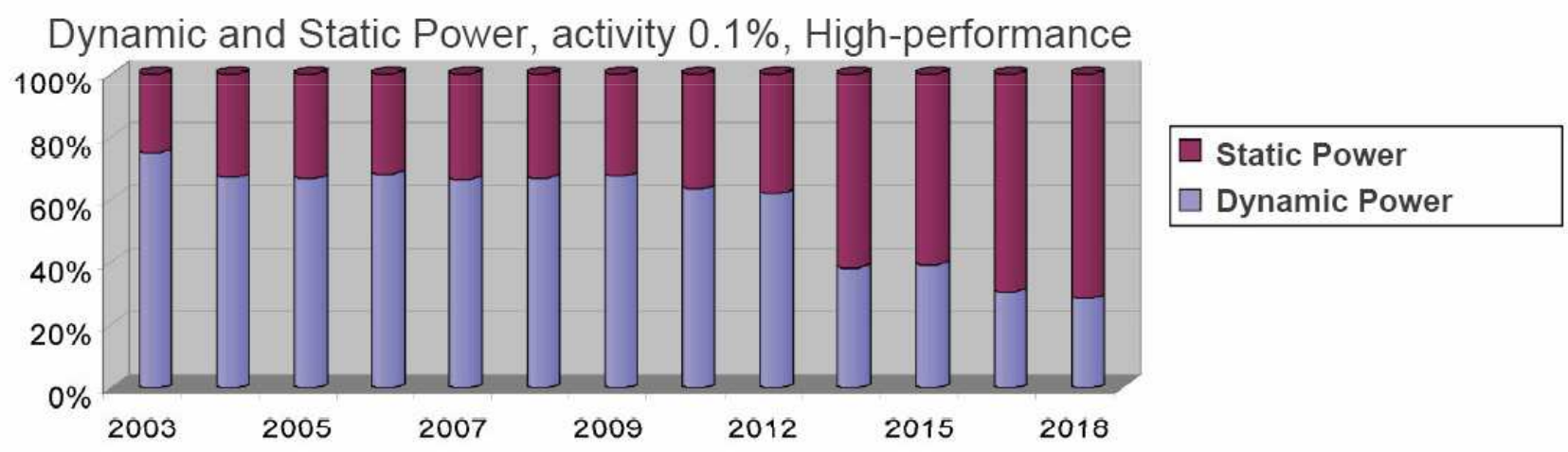
- Texec = NI * CPI * Tc
  - NI = Number of Instructions
  - CPI = Clock per Instruction
  - Tc = Cycle Time
- Texec = NI / (IPC * F)
  - IPC = Instructions Per Cycle
  - F = Frequency
- Performance improves with
  - Silicon manufacturing technology
    - Moore's law contributing to higher frequency and parallelism
  - Microarchitecture improvements
    - Higher frequencies with deeper pipelines
    - Higher IPC through parallelism

Performance

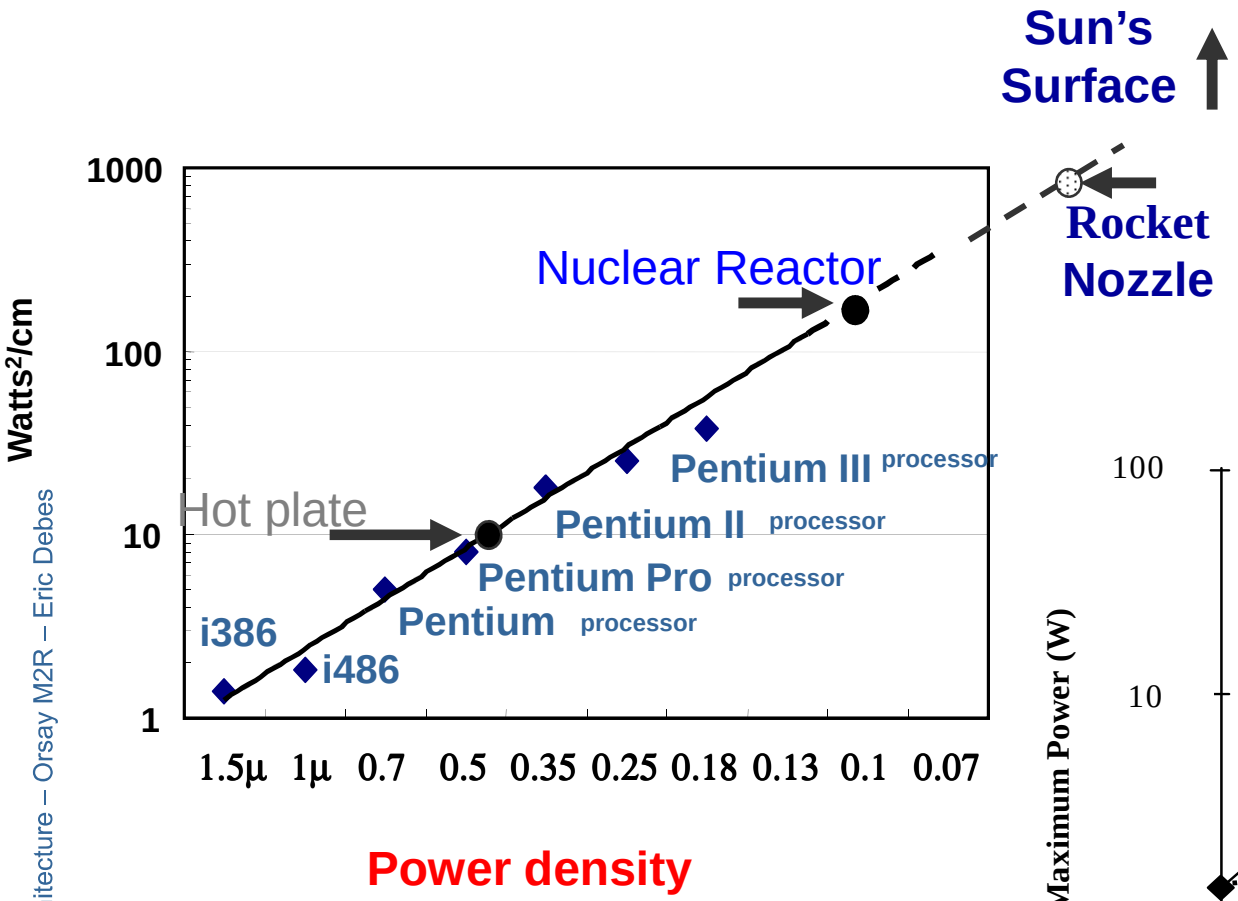


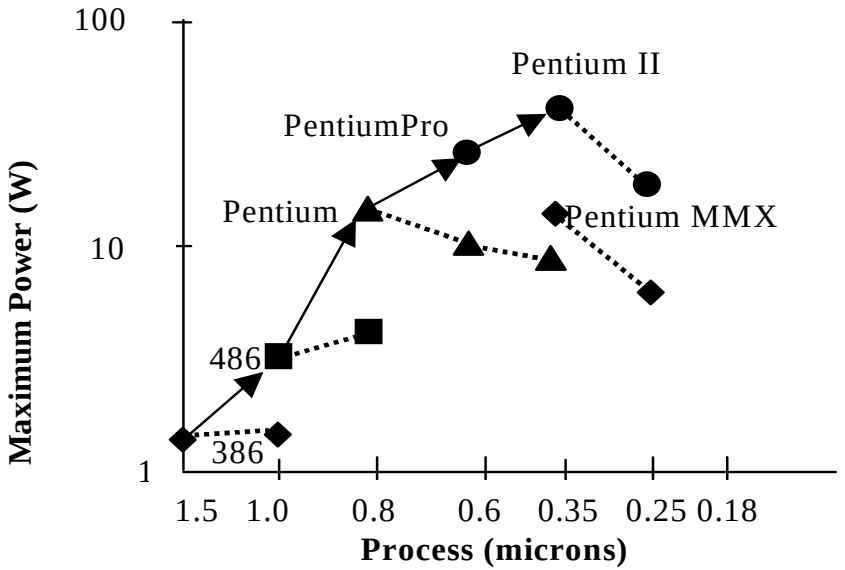
Power

- Dynamic Power = $\alpha C V^2 f$
  - α = activity
  - C = capacitance
  - V = voltage
  - f = frequency
- Power = Pdyn + Pstatic

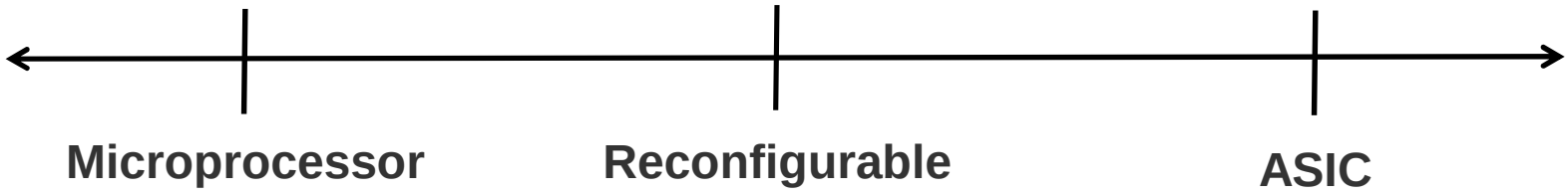


- Power is limited by
  - maximum current (Voltage regulator limitation)
  - Thermal constraints
- Power ≠ Energy


Sun's Surface
Rocket Nozzle
Nuclear Reactor
Hot plate
Pentium III processor
Pentium II processor
Pentium Pro processor
Pentium processor
i386
i486
Watts²/cm
1000
100
10
1
1.5µ 1µ 0.7 0.5 0.35 0.25 0.18 0.13 0.1 0.07
Power density

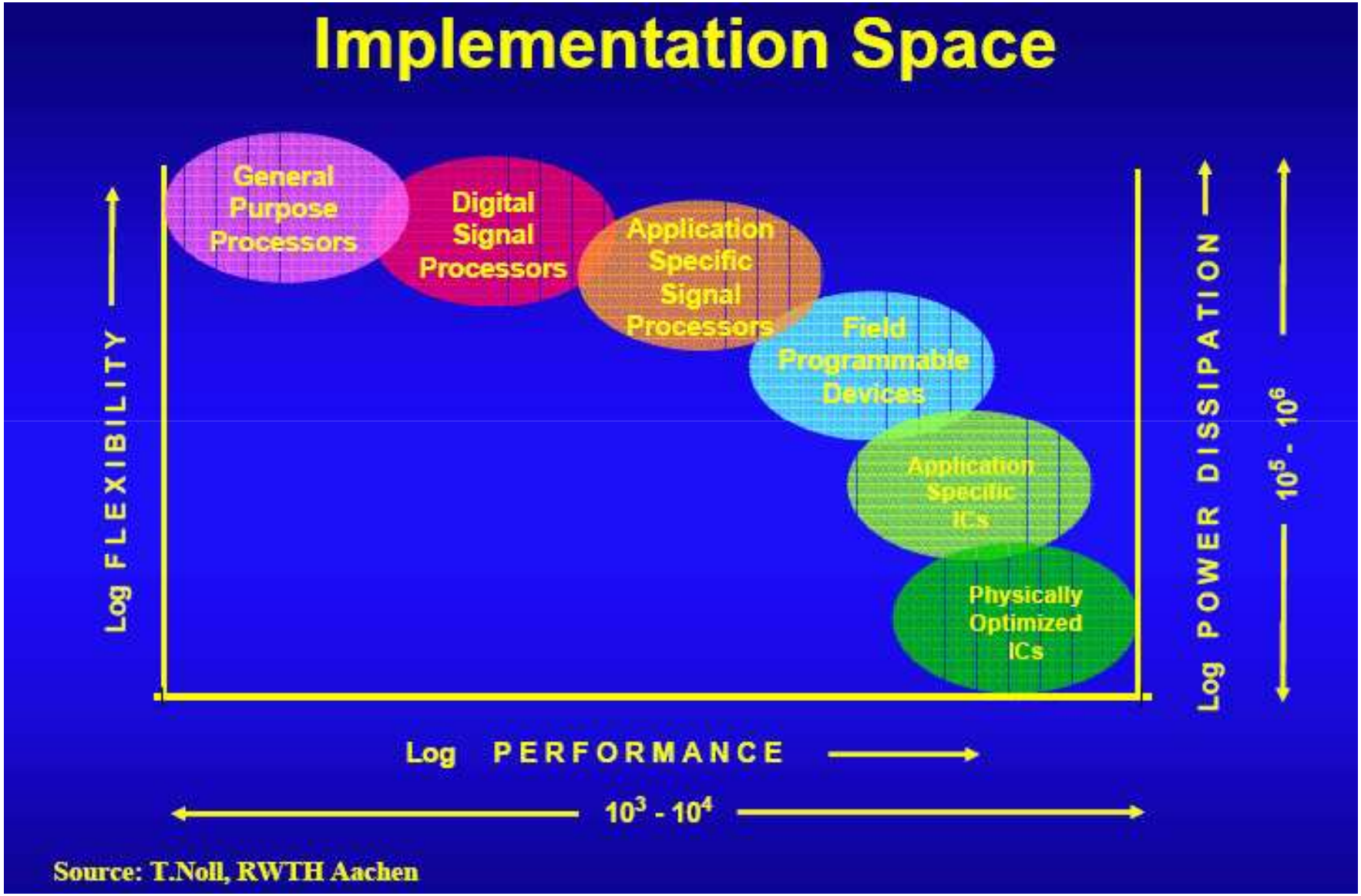

Maximum Power (W)
100
10
1
Pentium II
PentiumPro
Pentium
Pentium MMX
486
386
1.5 1.0 0.8 0.6 0.35 0.25 0.18
Process (microns)

Processor Architecture Spectrum

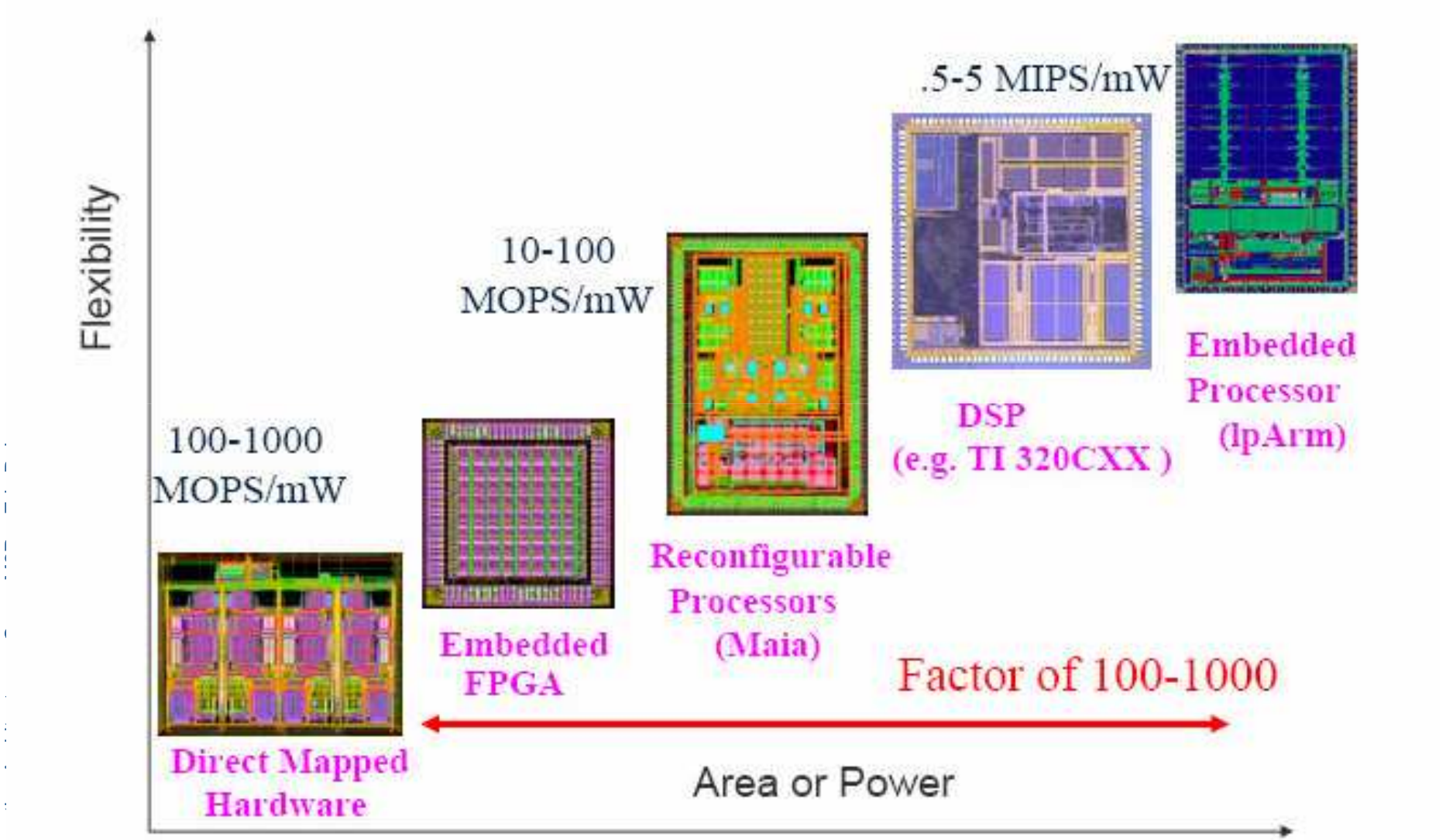


- ASIC
  - High-performance
  - Dedicated to one specific application
  - Not programmable
- Processor
  - Programmable
  - General-purpose
- Reconfigurable Architecture
  - Good compromise between programmability and performance

Processor Architecture Spectrum



Processor Architecture Spectrum



Key Components of a Computing System

What are the key components in a Computing System?

- Processor with
  - Arithmetic and Logic Units
  - Register File
  - Caches or local memory
- Memory
- Buses/Interconnect
- I/O Devices

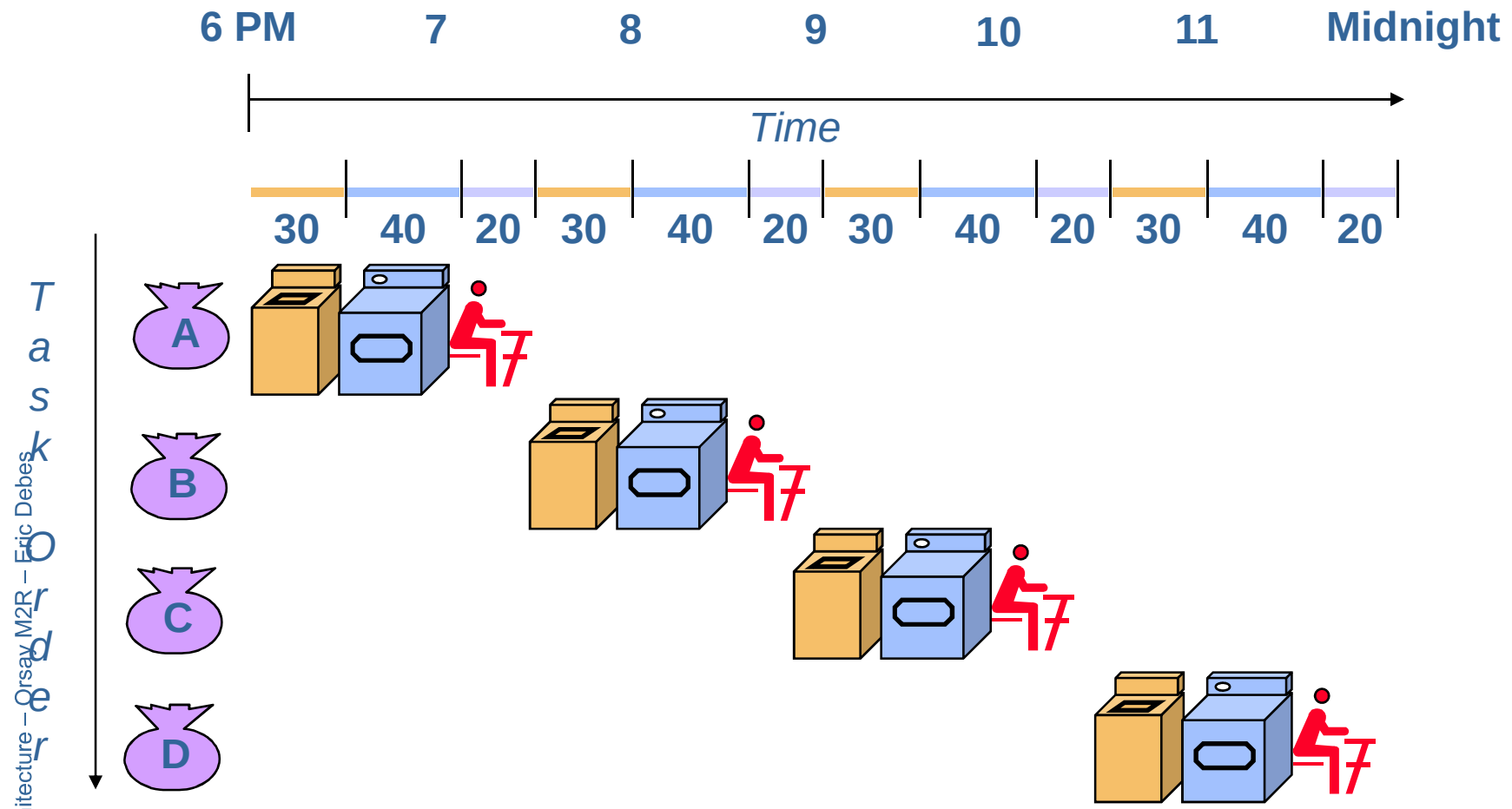
Embedded Computing Systems for Signal Processing Applications

Part 2: General-purpose Processors and Parallelism

October 19th 2012

Eric Debes

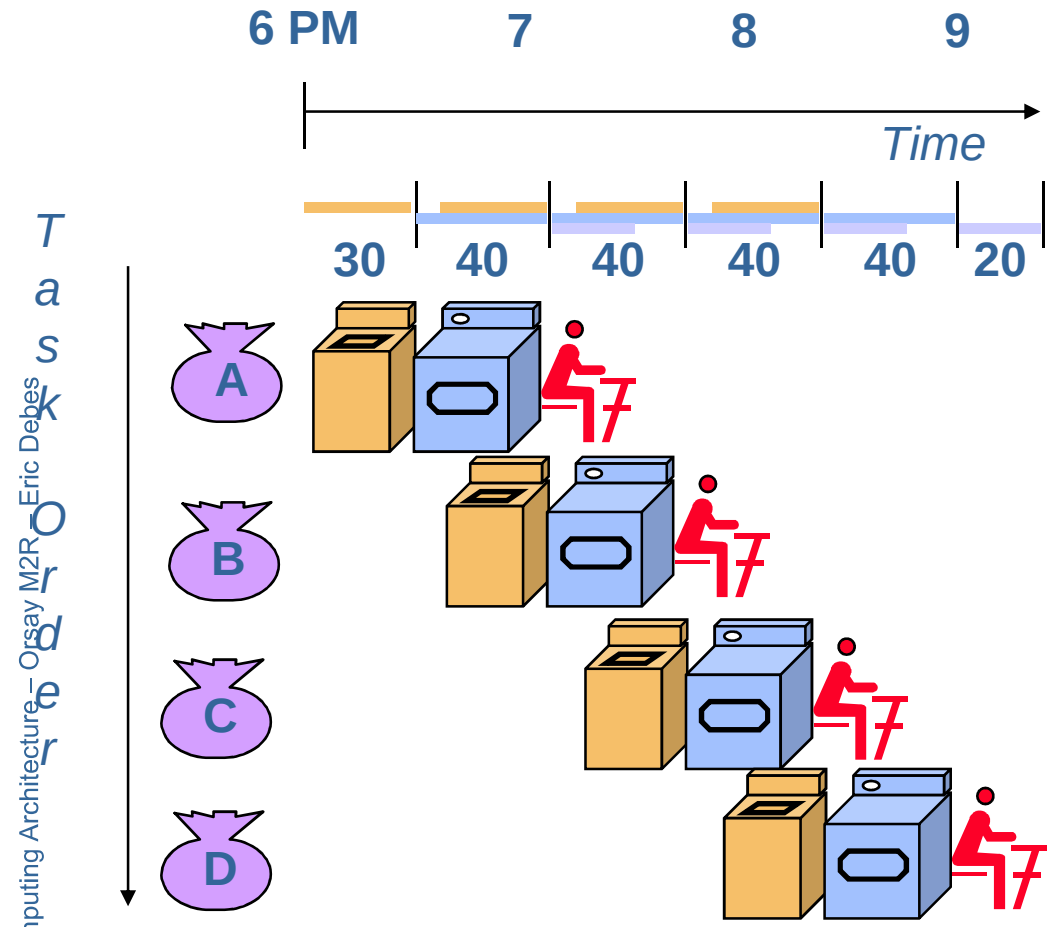
Sequential Laundry



Sequential laundry takes 6 hours for 4 loads

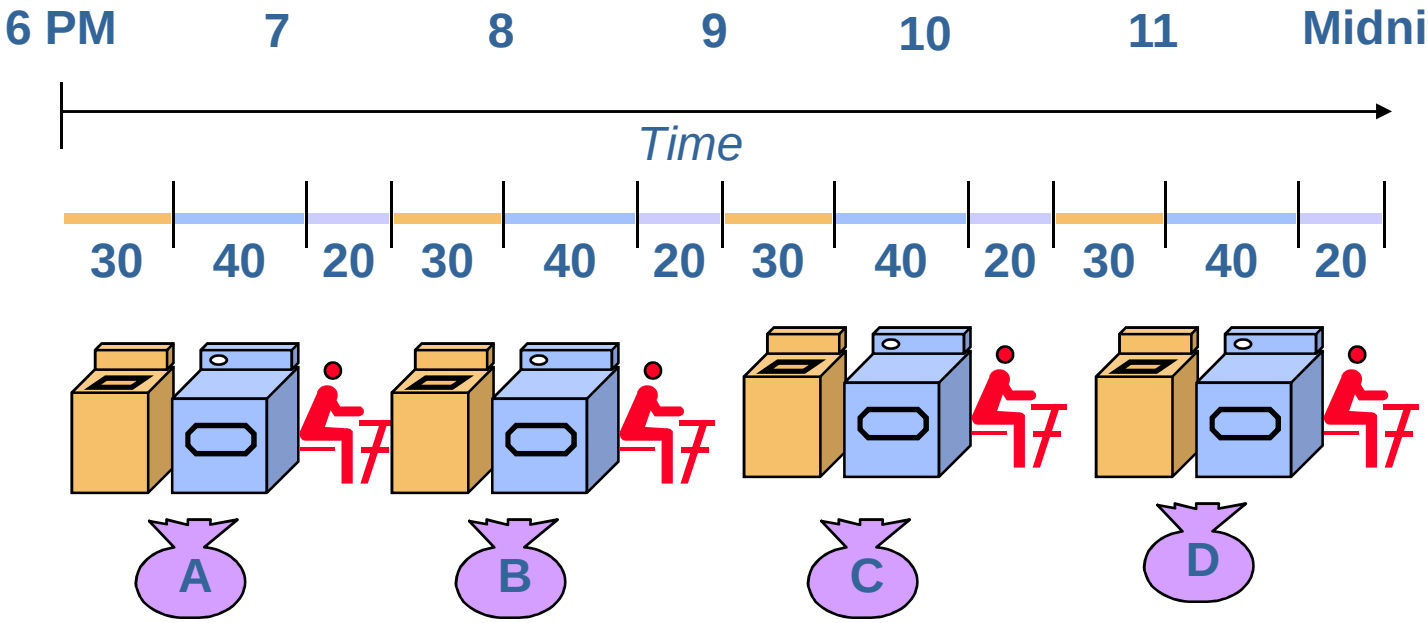
If they learned pipelining, how long would laundry take?

Pipelining Lessons



- Pipelining doesn't help latency of single task, it helps throughput of entire workload
- Pipeline rate limited by slowest pipeline stage
- Multiple tasks operating simultaneously
- Potential speedup = Number pipe stages
- Unbalanced lengths of pipe stages reduces speedup
- Time to "fill" pipeline and time to "drain" it reduces speedup

Parallel Laundry



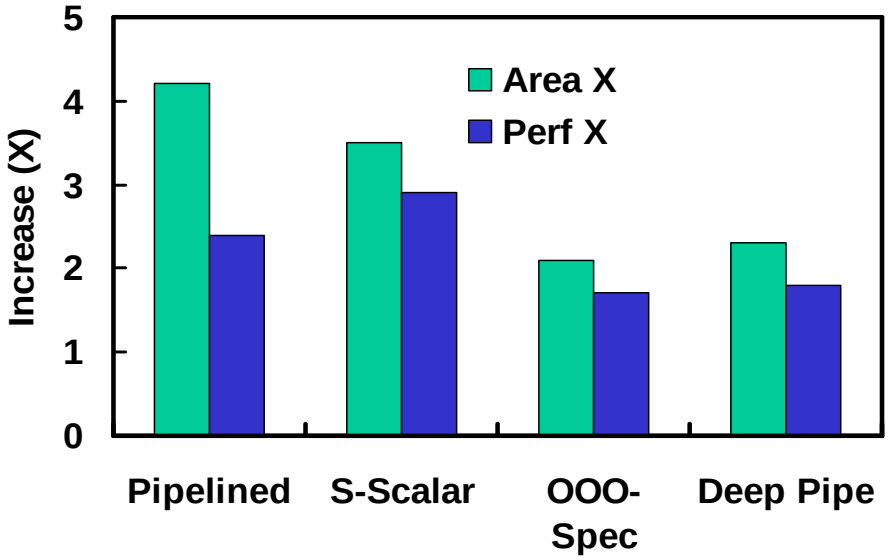
Parallel laundry takes 1.5 hours for 4 loads

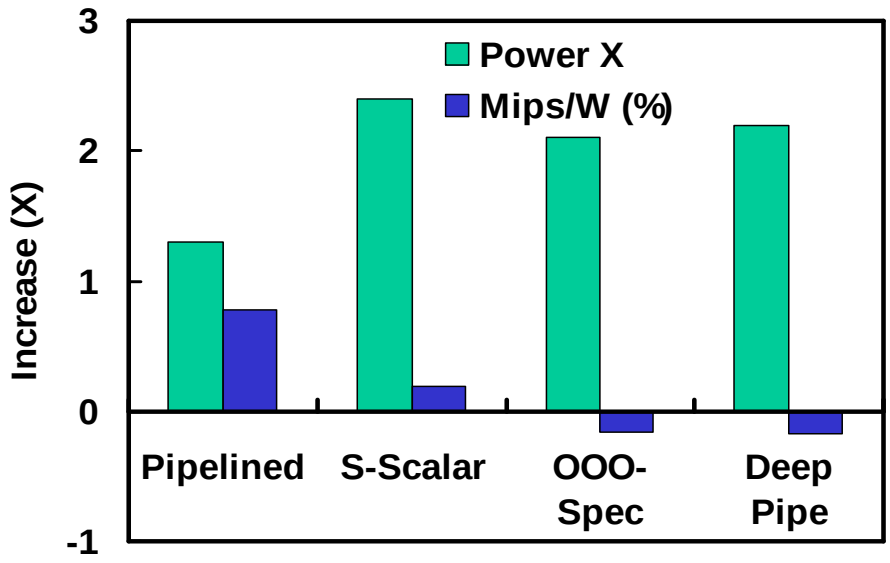
Throughput is the same as in pipeline

Cost more

What is better?

History: How did we increase Perf in the Past?





Moore's Law ⇨ more transistors for advanced architectures

Delivers higher peak perf

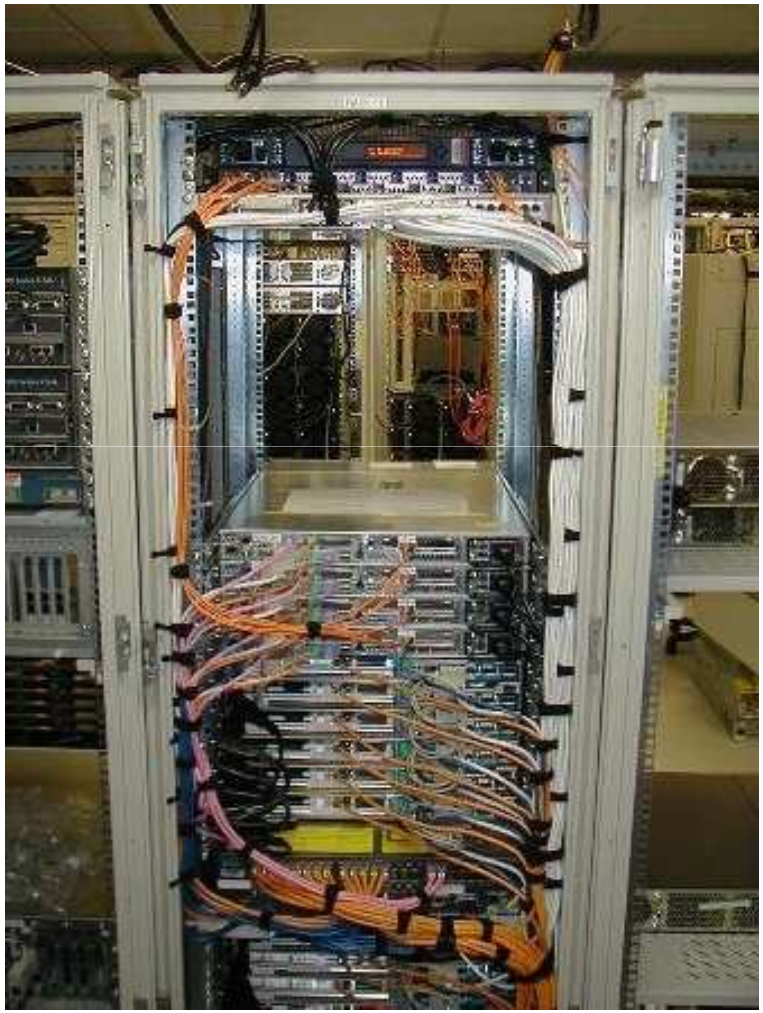
*But l*ower power efficiency

Performance = Frequency x Instruction per Clock Cycle

Power = Switching Activity x Dynamic Capacitance x Voltage x Voltage x Frequency

Why Multi-Cores?

In many systems today power is the limiting factor and will drive most of the architecture decisions



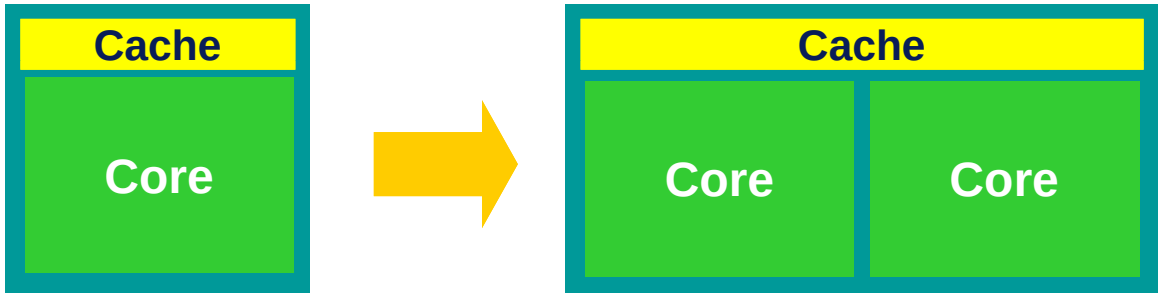
New Goal: optimize performance in a given power envelope

Power = Dynamic Capacitance x Voltage x Voltage x Frequency

Rule of thumb (in the same process technology)

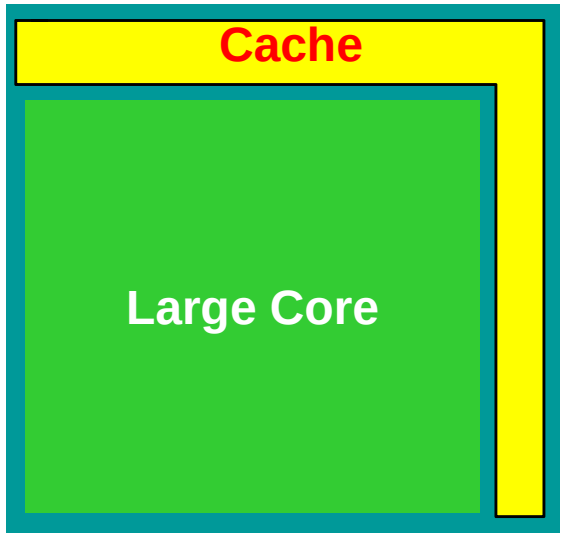
Voltage	Frequency	Power	Performance
1%	1%	3%	0.66%

How to maximize performance in the same power envelope?

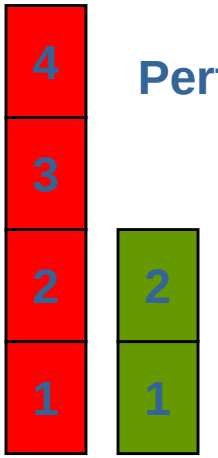


Voltage = 1
Freq = 1
Area = 1
Power = 1
Perf = 1

Voltage = -15%
Freq = -15%
Area = 2
Power = 1
Perf = ~1.8



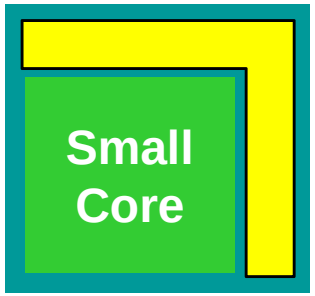
Power

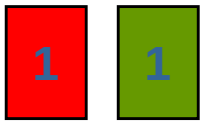


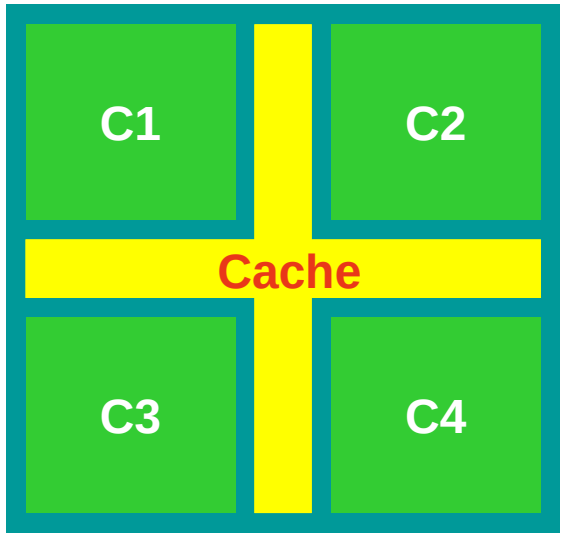
Performance

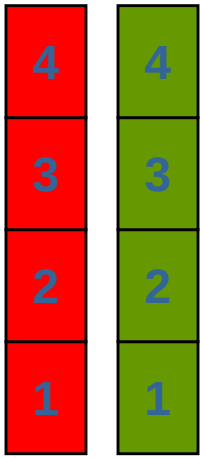
Power = 1/4

Performance = 1/2









Multi-Core:
Power efficient
Better power and thermal management

Summary: Why Multi-Cores?

Thermal is the main limitation factor in future design (not size)

Move away from Frequency alone to deliver performance

Challenges in scaling → need to exploit thread level parallelism to efficiently use the transistors available thanks to Moore's law.

Power/performance tradeoffs dictate architectural choices

Multi-everywhere

- Multi-threading
- Chip level multi-processing

Throughput oriented designs

Application-driven Architecture Design

Processors are designed to address the need of the mass market.

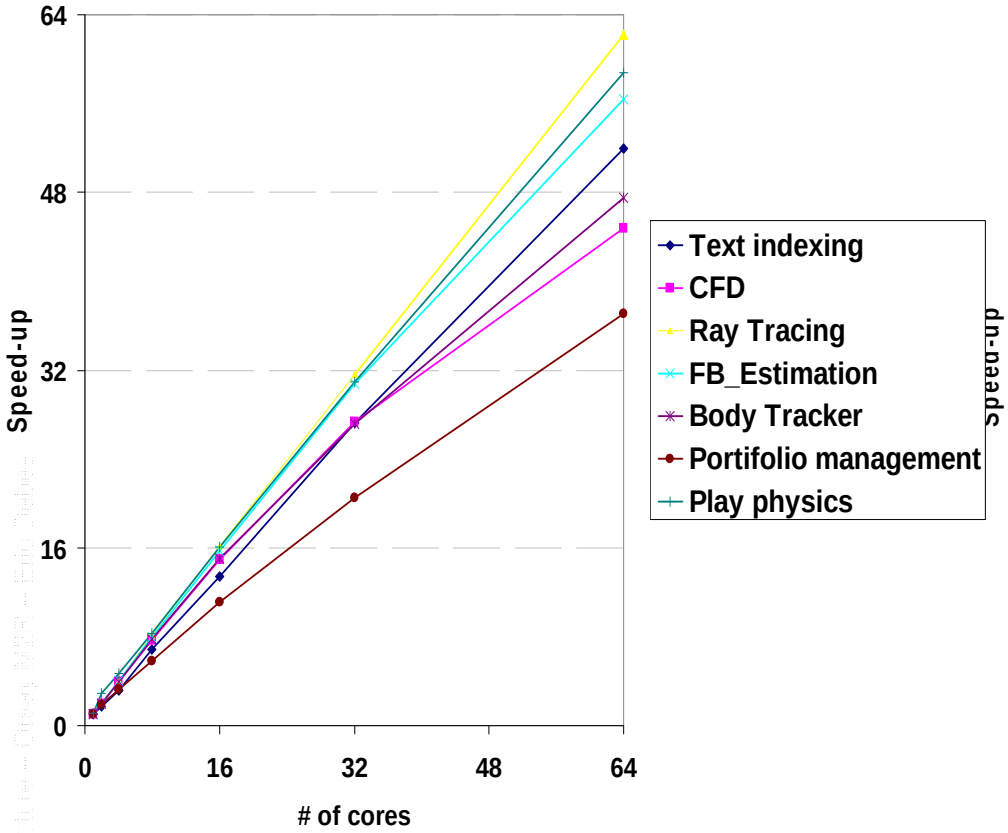
- **Mobile applications** → low power and good power management are top priorities to enable thinner systems and longer battery life

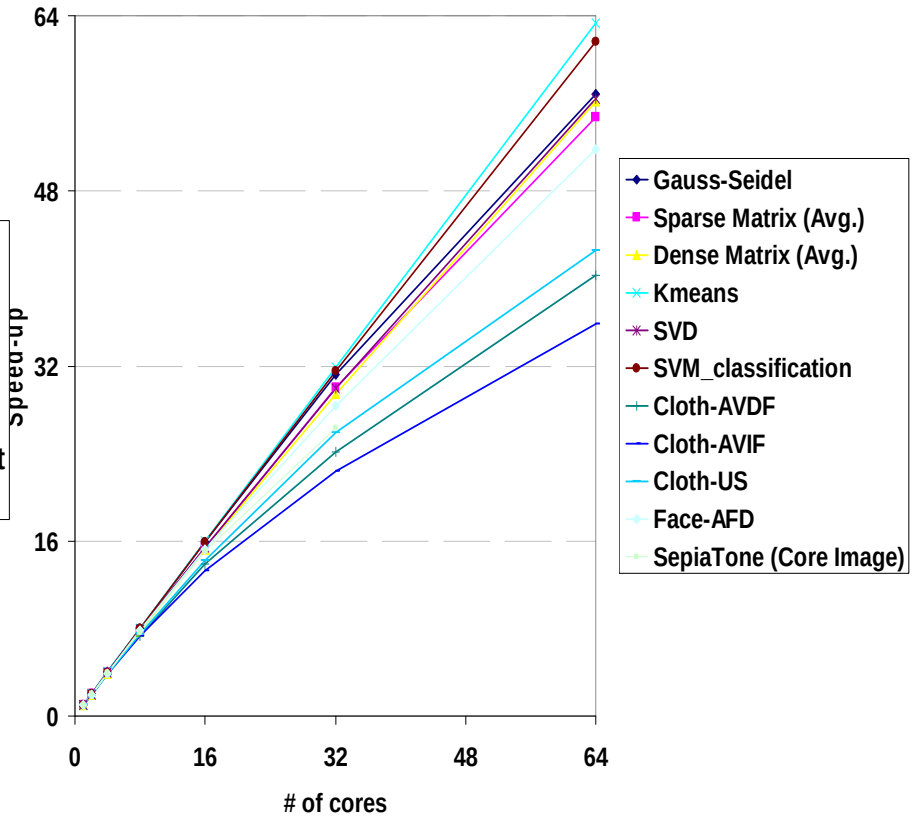
- **Office, image, video** → single threaded perf matters, some level of multithreaded perf → Multi-core

- **RMS (Recognition, Mining, Synthesis) Applications and Model based Computing** → massively parallel apps, good scaling on a large number of cores → Many-core

Because of the large markets in each of the classes above, they are the focus of silicon manufacturers and are driving innovation in the semiconductor market

RMS Scaling on a Many-Core Simulator

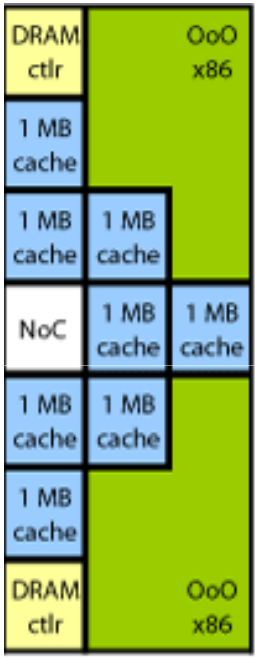


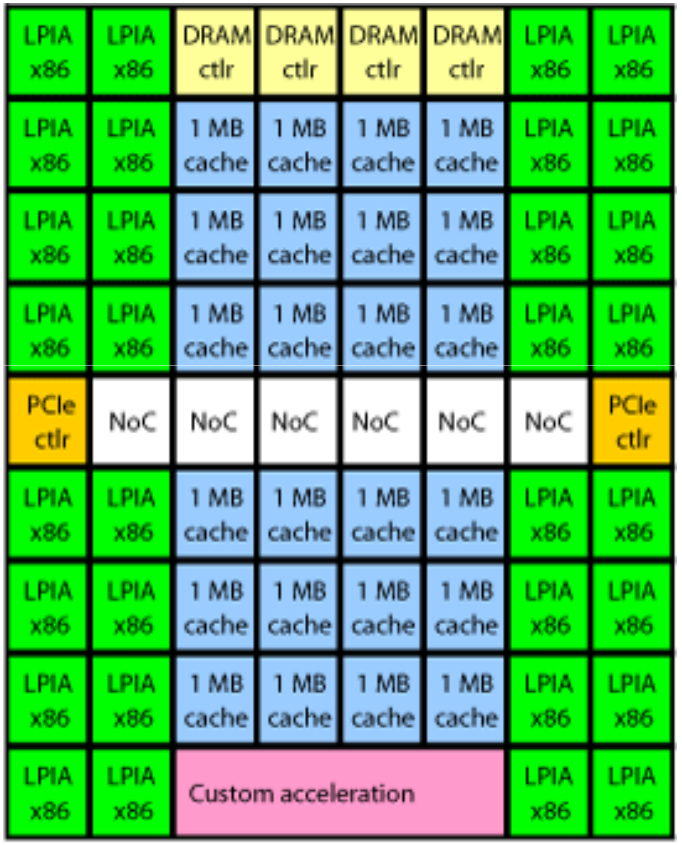


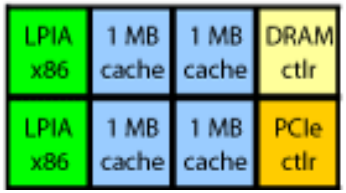
Data from Intel Application Research Lab

3 Classes of Applications→ 3 Types of Processors

- **Low-power architecture and SoCs**
  - ARM based
  - LPIA/Atom based
- **Multi-core**
  - Core microarchitecture
  - PowerPC
- **Many-core**
  - GP GPU
  - Larrabee



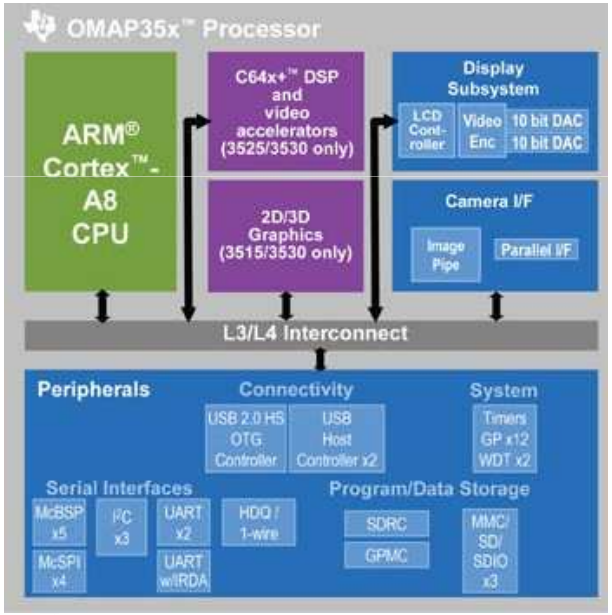


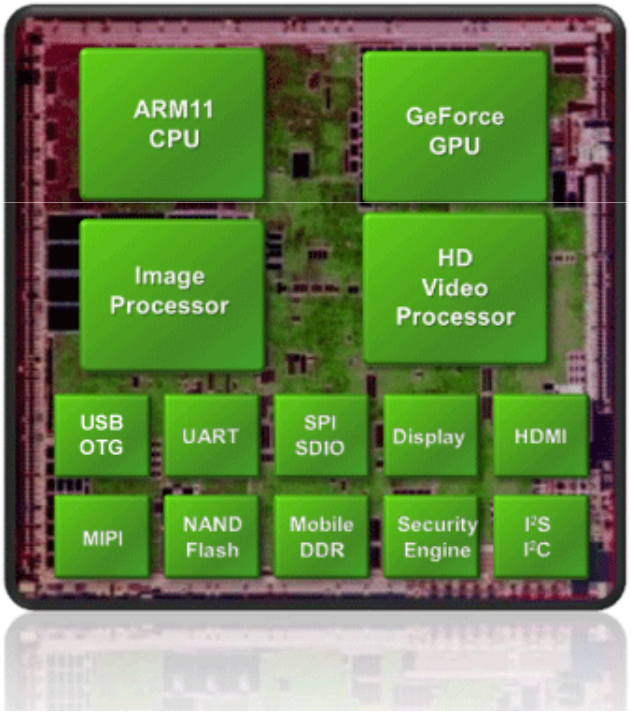


Low-power Architecture and SoCs

Examples of Low power architectures and SoCs

- ARM-based: TI OMAP, Nvidia Tegra, Apple A4/A5
- Atom based: Moorestown, Tunnel Creek(MID), Canmore (CE)

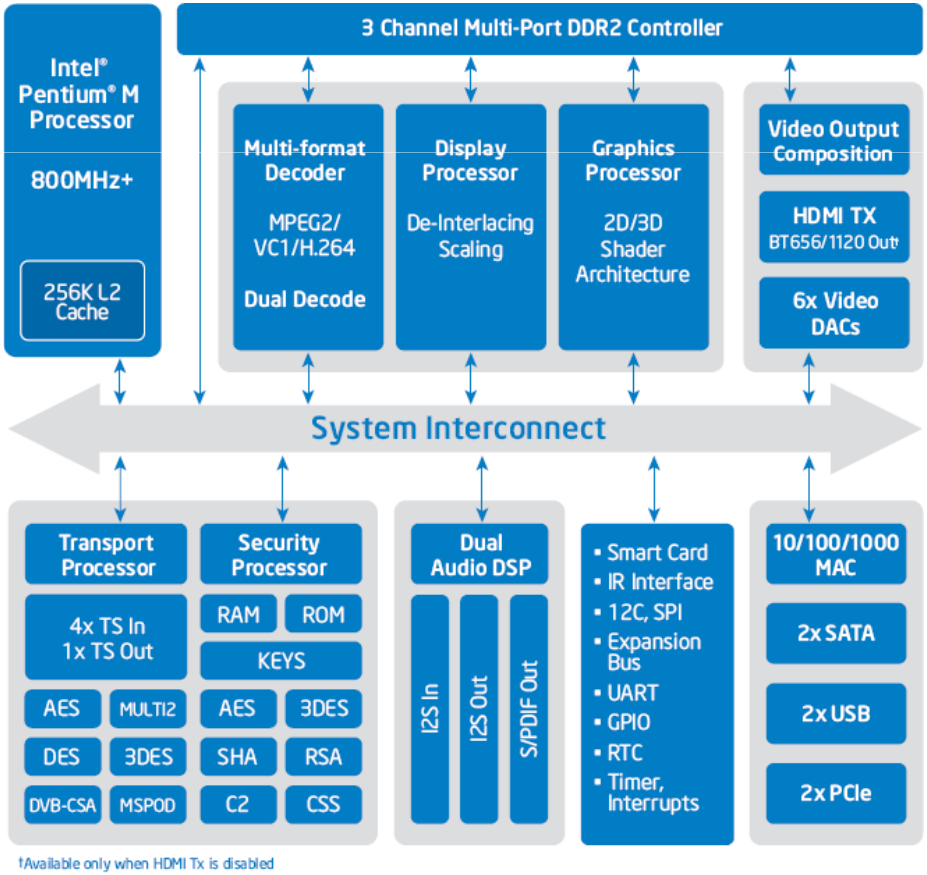


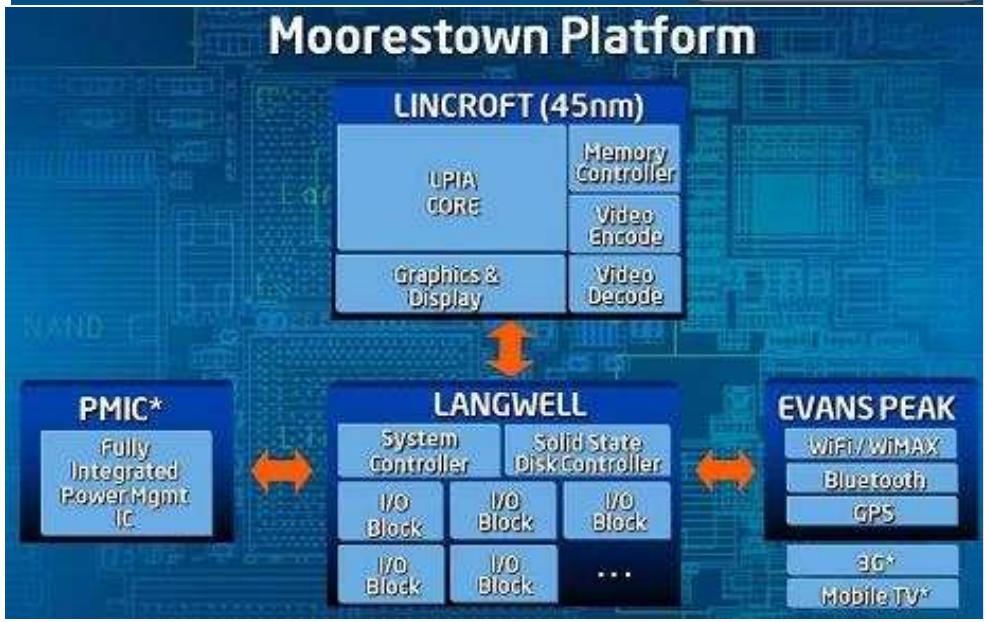


Towards PC on a chip

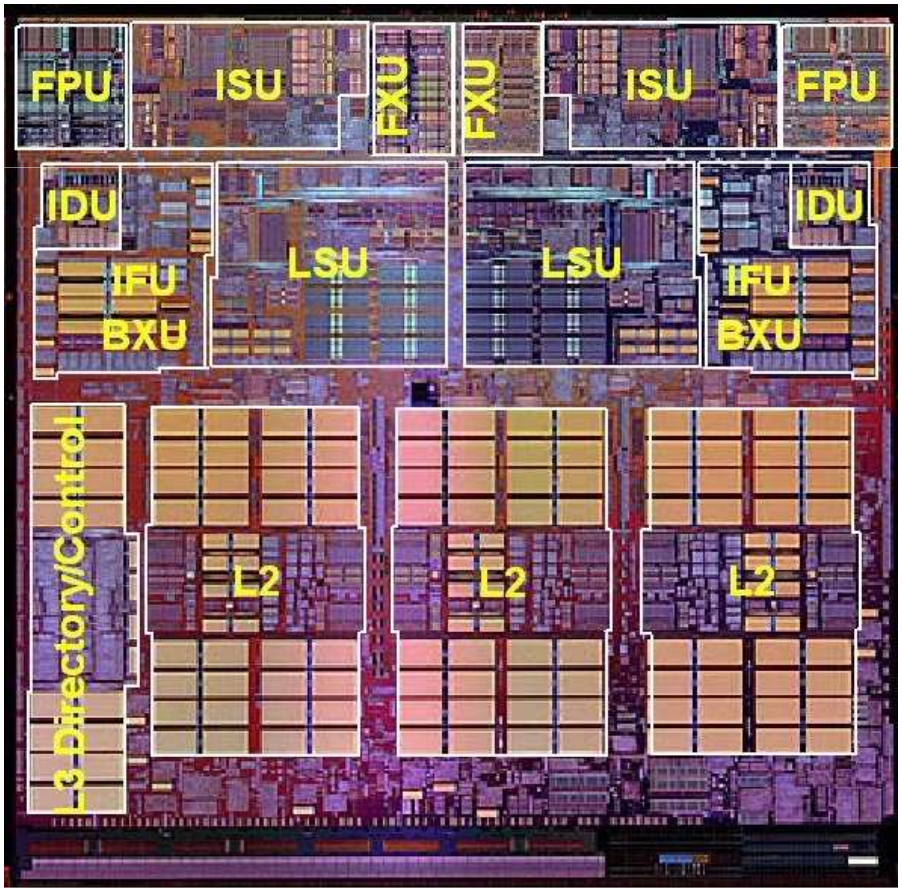
Intel Atom based for:

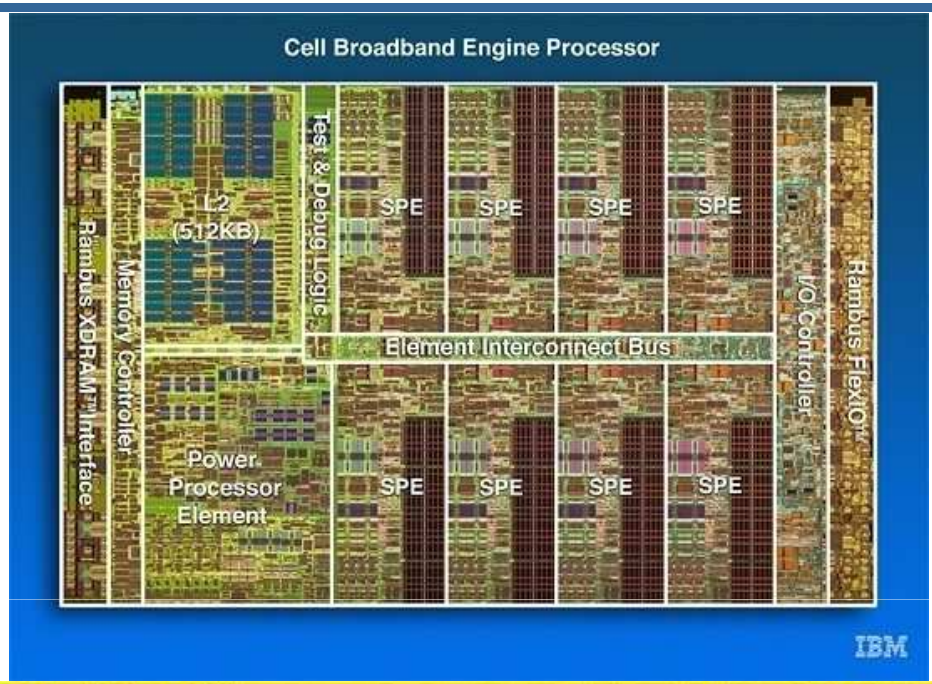
- Mobile Internet Devices
- Consumer Electronic Devices
- Embedded Market

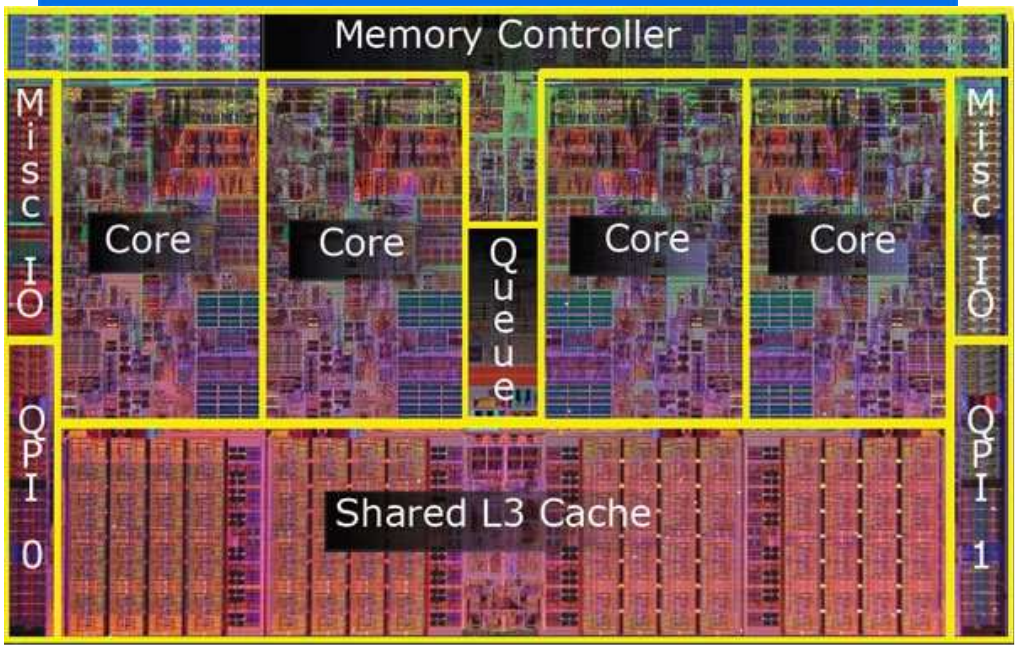




- **Multi-core**
  - IBM Power4
  - IBM Cell
  - Intel Core microarchitecture

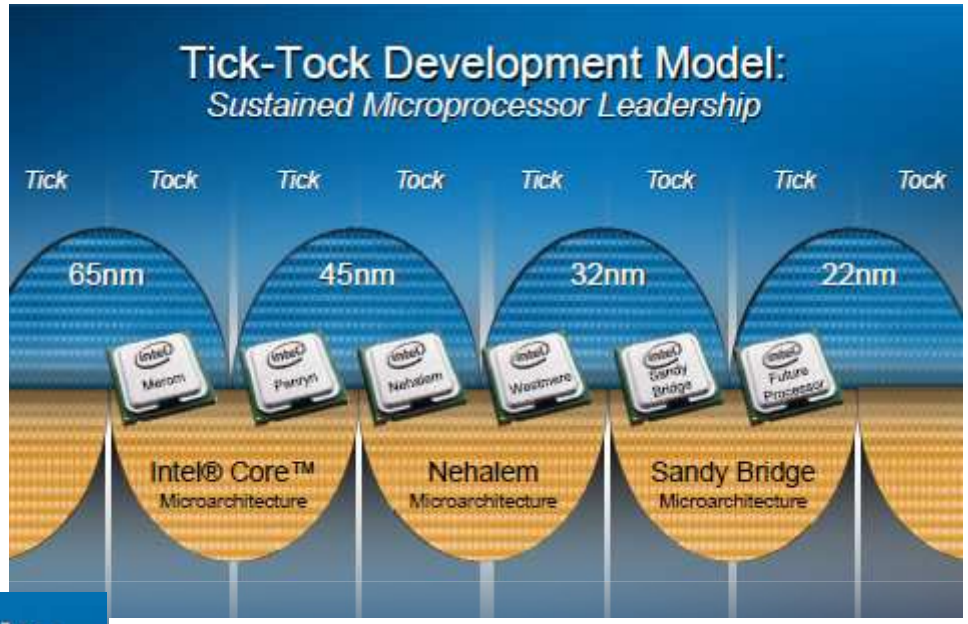


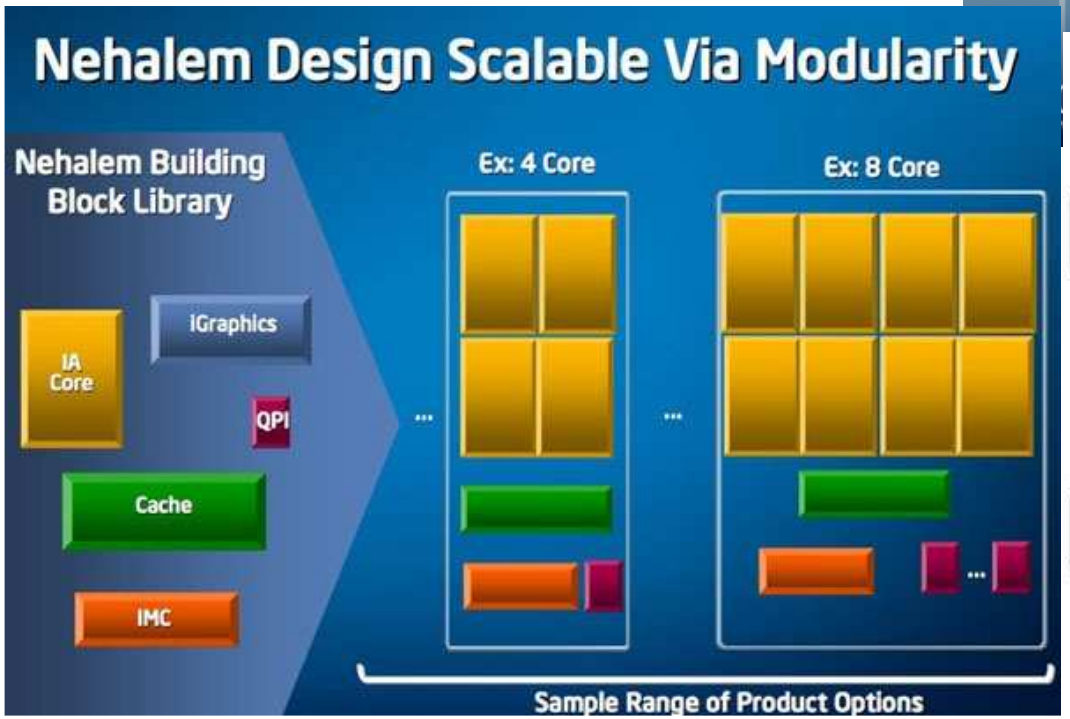


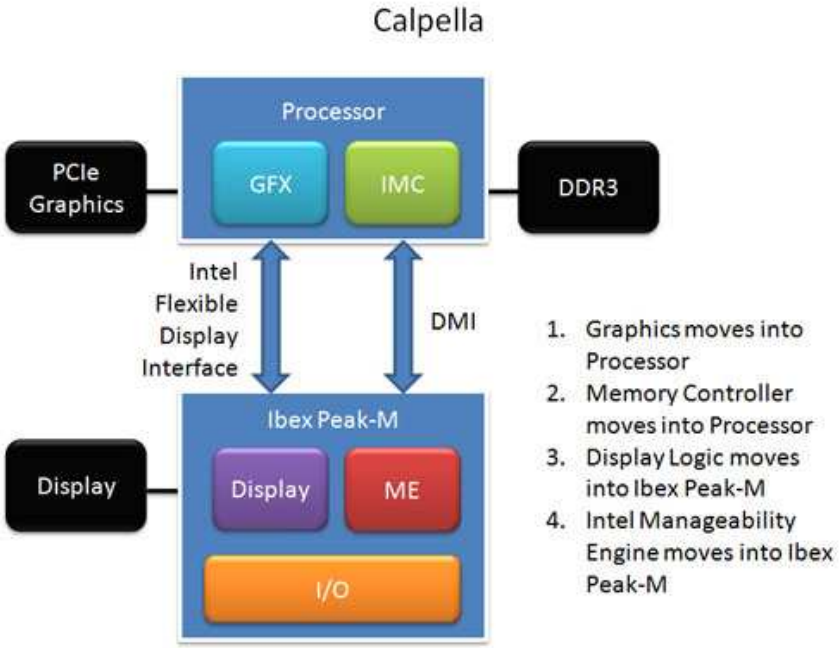


Intel Roadmap for Intel Core Microarchitecture

- **Tick-Tock model**
- **Modular design to decrease cost (design, test, validation)**
- **Integrate graphics on chip**



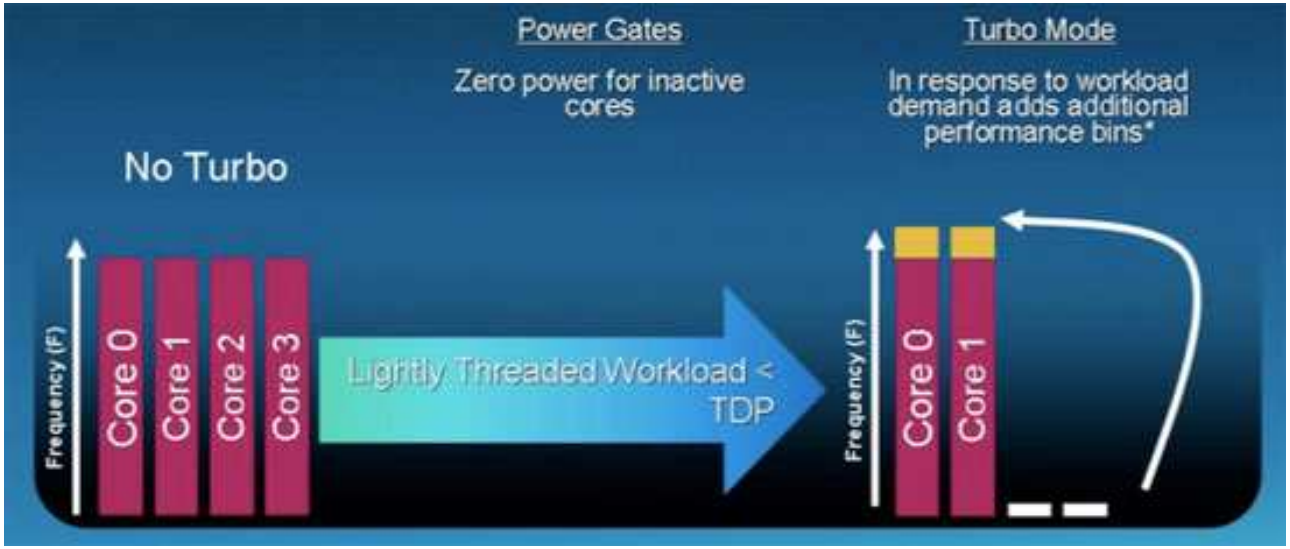




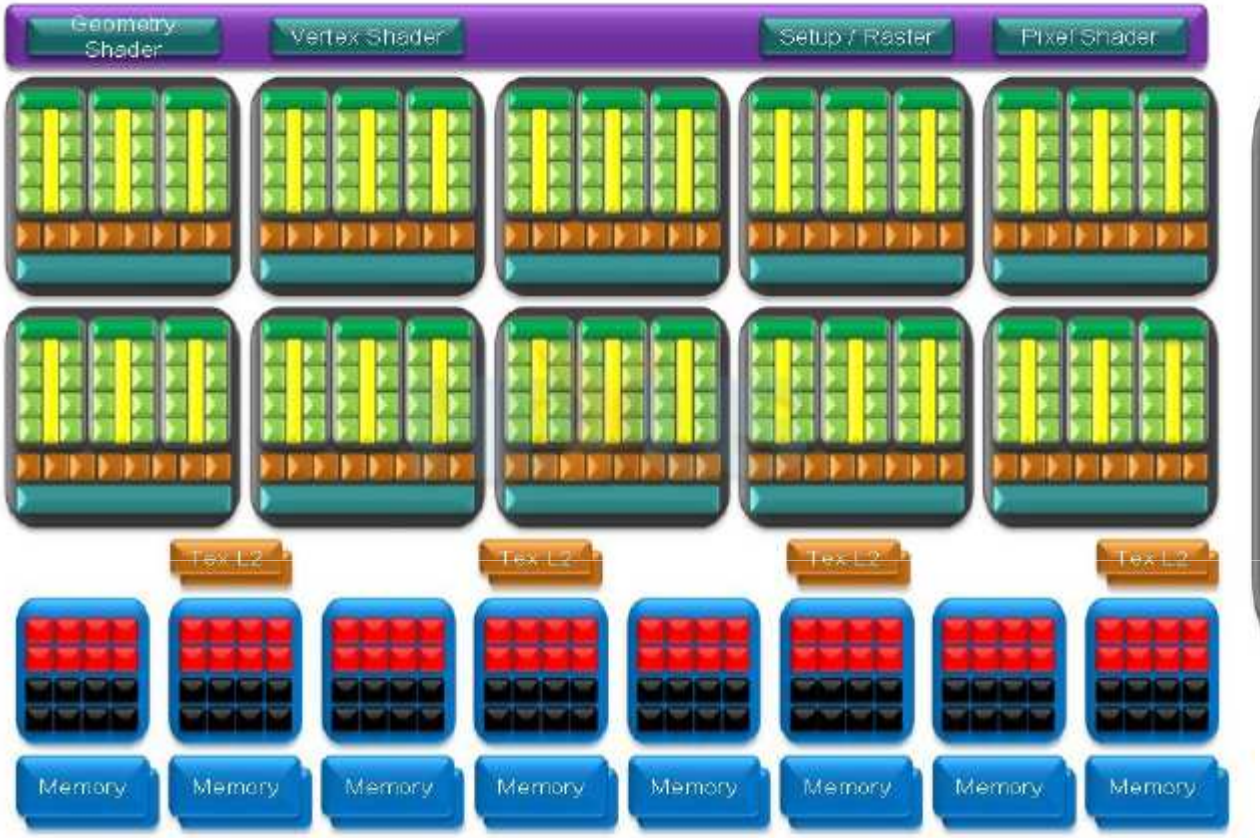
Power/Performance Tradeoffs

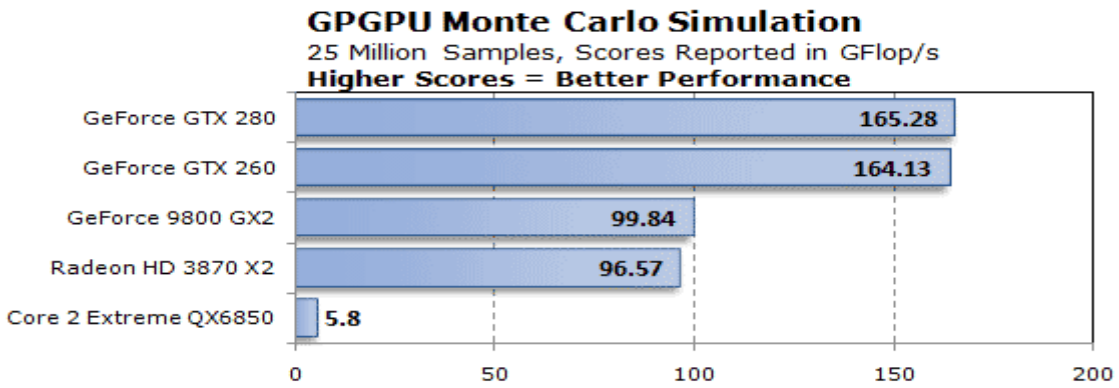
- **Binning for leakage distribution and performance**

  $P = \alpha.C.v^2.f$ + leakage
- **Turbo mode to optimize performance under a given power envelope**
- **Policy to balance thermal budget between general purpose cores, and between GPP cores and graphics**
- **Next: Maximize performance under a given thermal envelope at the platform level**



GP GPU: NVidia GeForce more than 2000 PEs





CPUs vs. GPUs

- No need to put a lot of cache for GPUs because the number of threads are hiding the latency. The chip is designed for DRAM latency through a huge number of threads. Local memory are still present to limit bandwidth to GDDR
- CPU need multi-level large caches because the data need to be close to the execution units
- Fast growing video game industry exerts strong economic pressure that forces constant innovation

Performance/Power for different architectures

For a given application, processor architectures should be chosen depending on the performance/power efficiency

- **MIPS/Watt or Gflops/Watt**
- **Energy efficiency (Energy Delay Product)**

This is highly dependent on the application and targeted power envelope. Examples:

- ARM and Atom are optimized for mainstream office and media apps for a power envelope between 1W and <10W
- Core microarchitecture is optimized for high-end office and media apps for a power envelope between 15W and ~75W
- GPUs are optimized for graphics applications and some selected scientific applications between 10W and more than 400W

Future: PC on a Chip

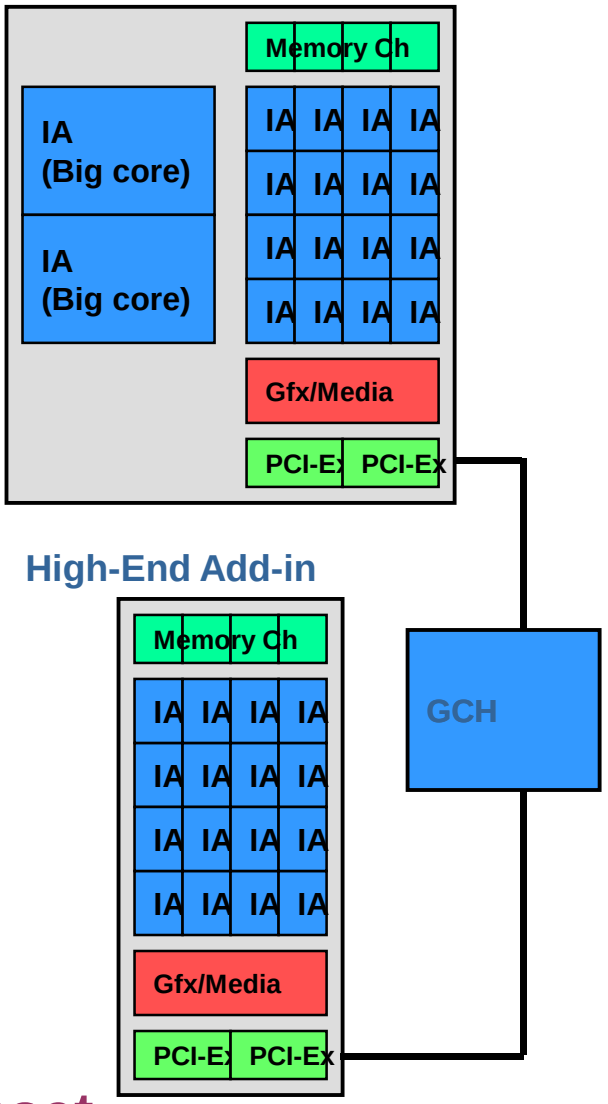
Processor will integrate

- Big core for single thread perf

- Small core for multithreaded perf

- some dedicated hardware units for

- graphics
- media
- encryption
- networking function
- other function specific logic



Systems will be heterogeneous

Processor core will be connected to

- one or multiple many-core cards

- and dedicated function hw in the chipset

+ reconfigurable logic in the system or on chip?

Embedded Computing Systems for Signal Processing Applications

Part 3: App Specific Proc: DSPs, FPGAs, Accelerators, SoCs

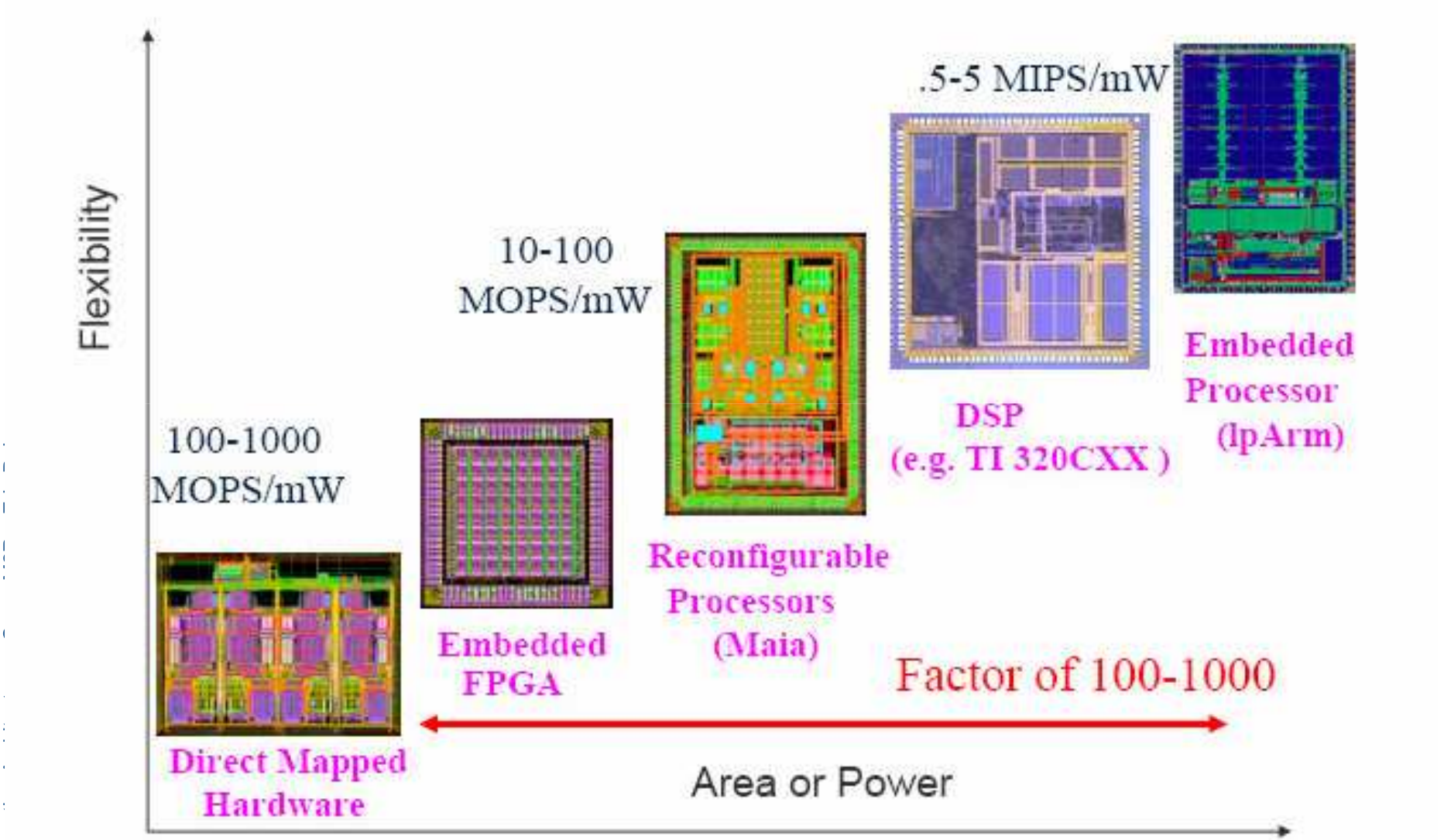
October 19th 2012

Eric Debes

Application Specific Processors

- What are application specific processors?
  - Processors or System-on-chip targeting a specific (class of) application(s)
- Very common for
  - Audio: MP3, AAC coding and decoding in audio players
  - Image: JPEG or JPEG2000 coding and decoding, e.g. Digital cameras
  - Video: MPEG, H264 coding and decoding, e.g. DVD players or set-top-boxes
  - Encryption: RSA, AES
  - Communication: GSM, 3G in cellphones
- Why?
  - Large markets can justify the development of application specific processors
  - Dedicated circuits provide higher performance with lower power dissipation, better battery life and very often lower cost.

Application Specific Signal Processor Spectrum



Different Types of ASPs

- DSPs
- Dedicated ASICs
- FPGAs
- Accelerators as coprocessors
- ISA extensions
- SoCs

Summary of Architectural Features of DSPs

Data path configured for DSP

- Fixed-point arithmetic
- MAC- Multiply-accumulate

Multiple memory banks and buses -

- Harvard Architecture: separate data and instruction memory
- Multiple data memories

Specialized addressing modes

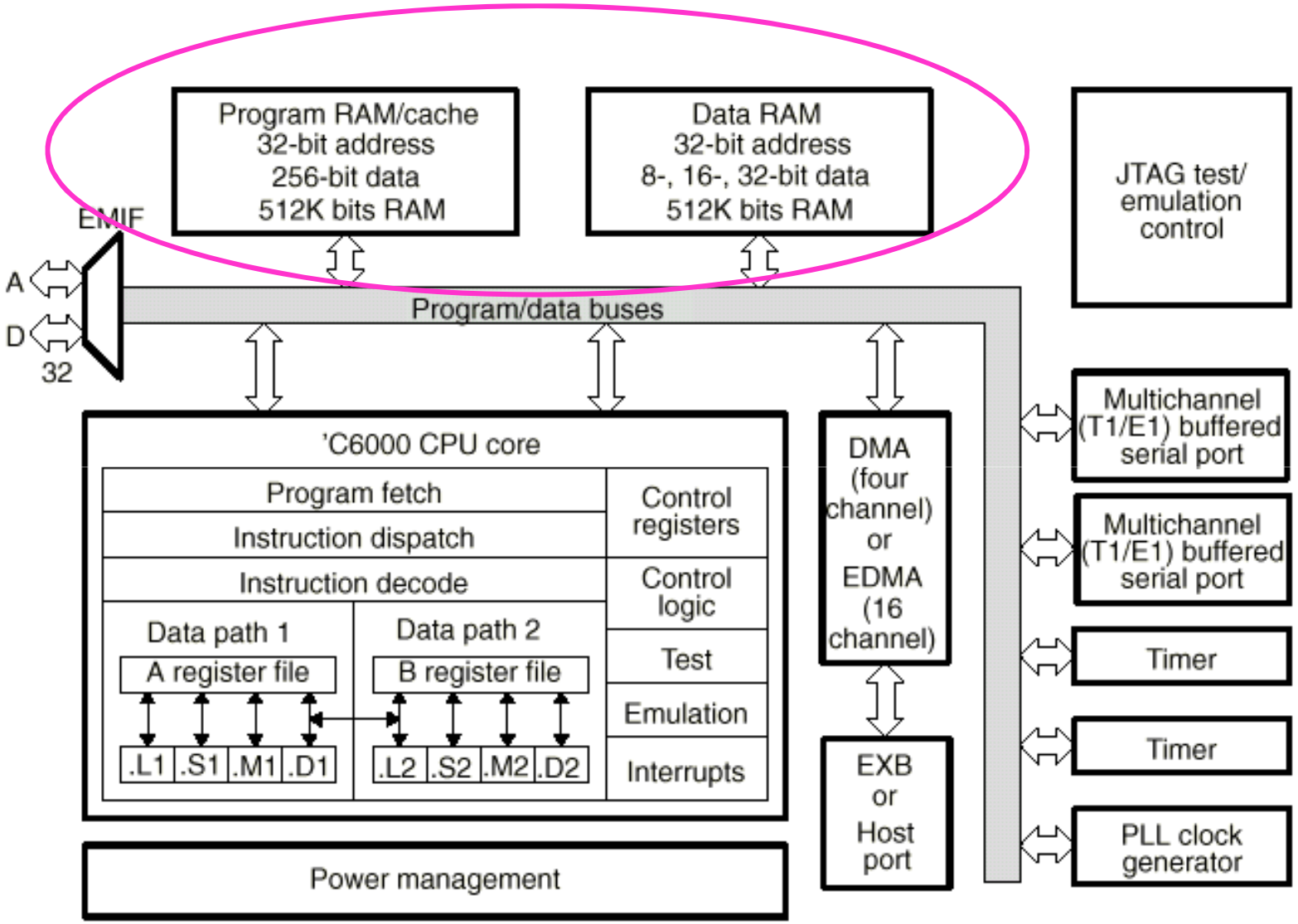
- Bit-reversed addressing
- Circular buffers

Specialized instruction set and execution control

- Zero-overhead loops
- Support for MAC

Specialized peripherals for DSP

DSP Example: 320C62x/67x DSP

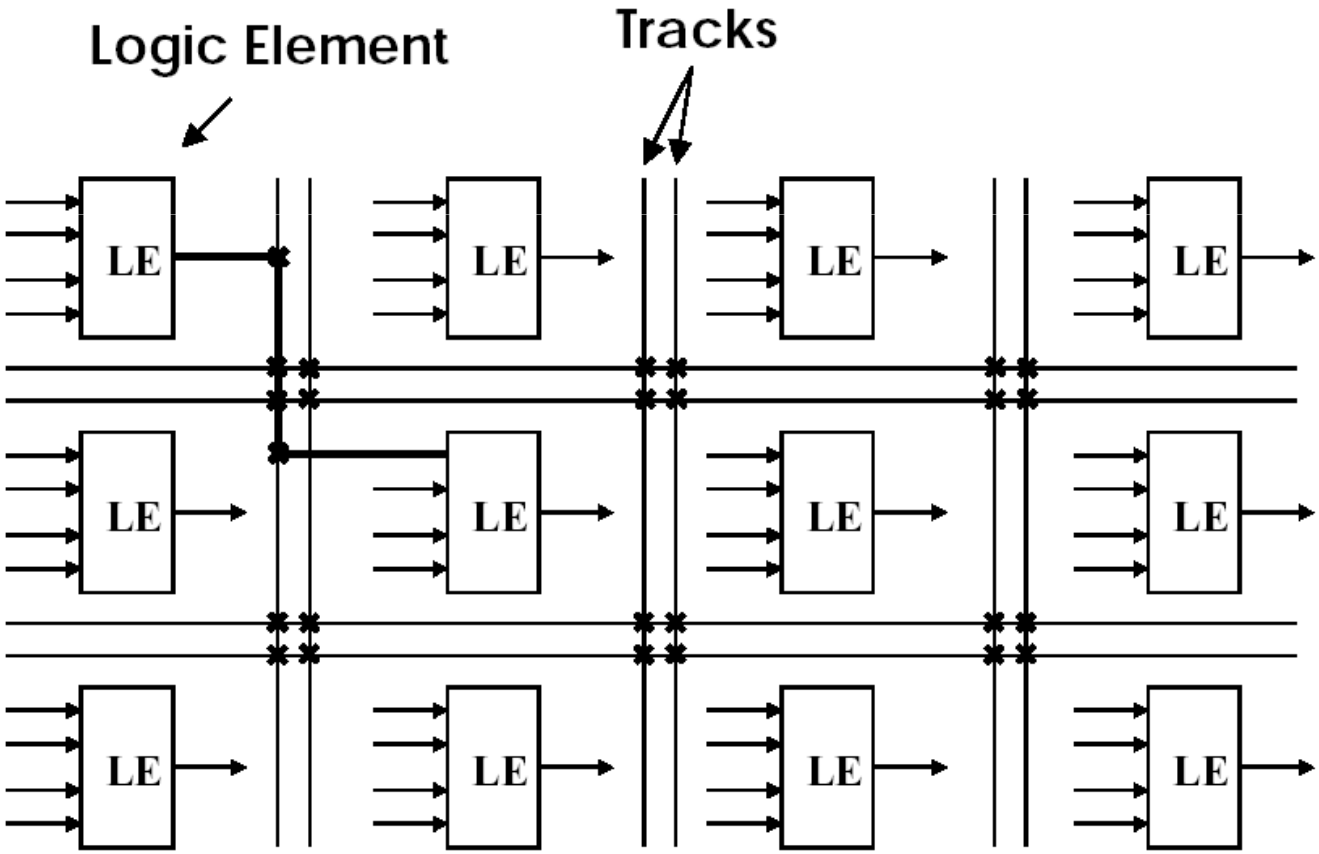


Dedicated ASICs

- Many dedicated ASICs exist on the market, especially for media and communication applications. Example:
  - MP3 player
  - DVD player
  - Video processing engines, e.g. De-interlacing, super-resolution
  - Video Encoder/Decoder
  - GSM/3G
  - TCP/IP Offload engine
- Advantages:
  - Low power, high perf/power efficiency
  - Small area compared to same functionality in DSP or GPP
- Drawbacks
  - Cost of designing ASICs → requires large volume
  - Not flexible: cannot handle different applications, cannot evolve to follow standard evolution

Reconfigurable architectures

- FPGAs contain gates that can be programmed for a specific application
  - Each logic element outputs one data bit
  - Interconnect programmable between elements
- FPGAs can be reconfigured to target a different function by loading another configuration



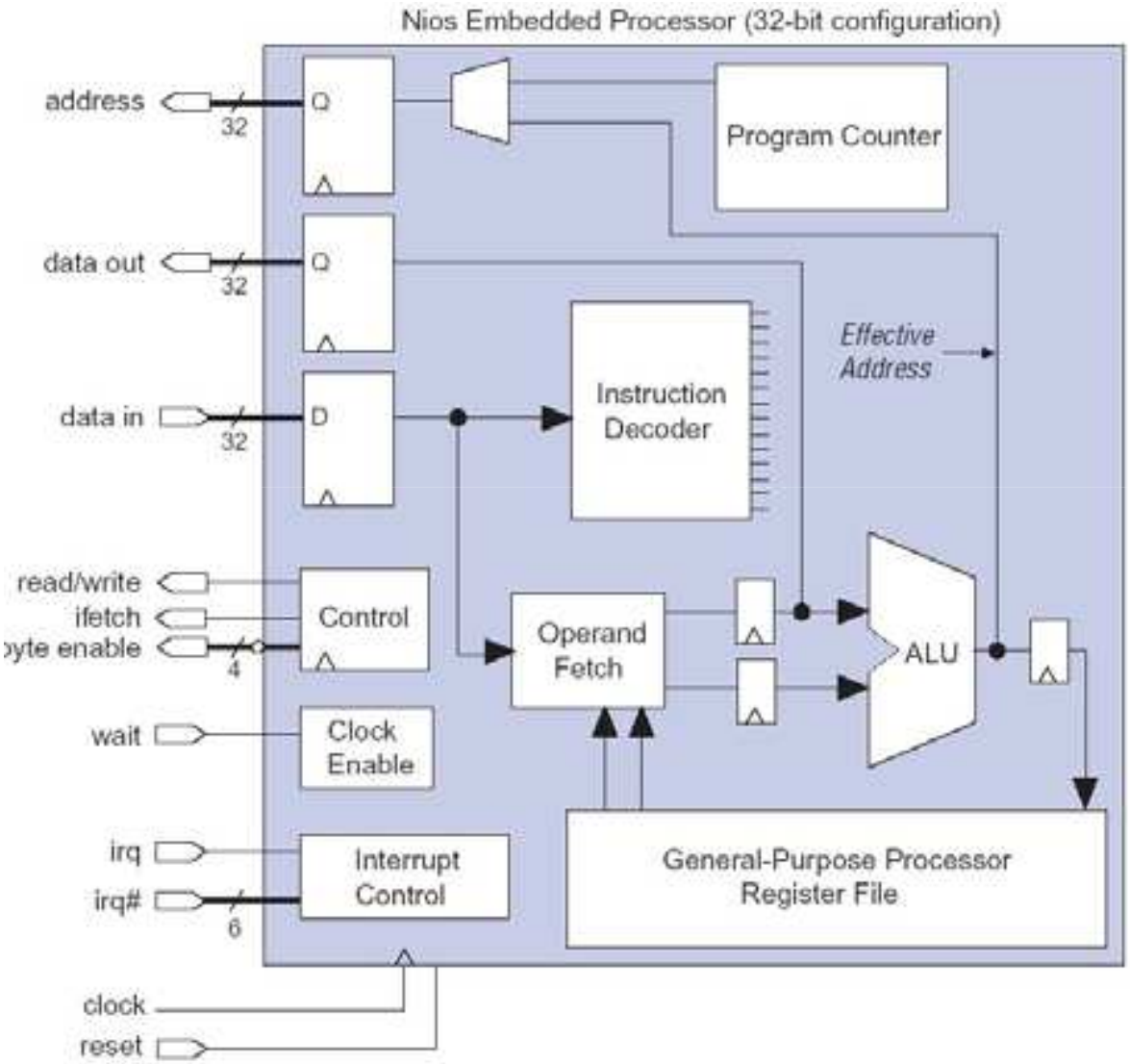
Flot de conception FPGAs

- Spécifications
  - Input: RTL coding → structural or behavioral description
- RTL Simulation
  - Functional simulation → check logic and data flow (no temporal analysis)
- Synthesis
  - Translate into specific hardware primitives
  - Optimisation to meet area and performance constraints
- Place and Route
  - Map hw primitives to specific places on the chip based on area and performance for the given technology
  - Specify routing
- Temporal Analysis
  - Verification that temporal specification are met
- Test and Verification of the component on the FPGA board

FPGAs with On-chip GPP

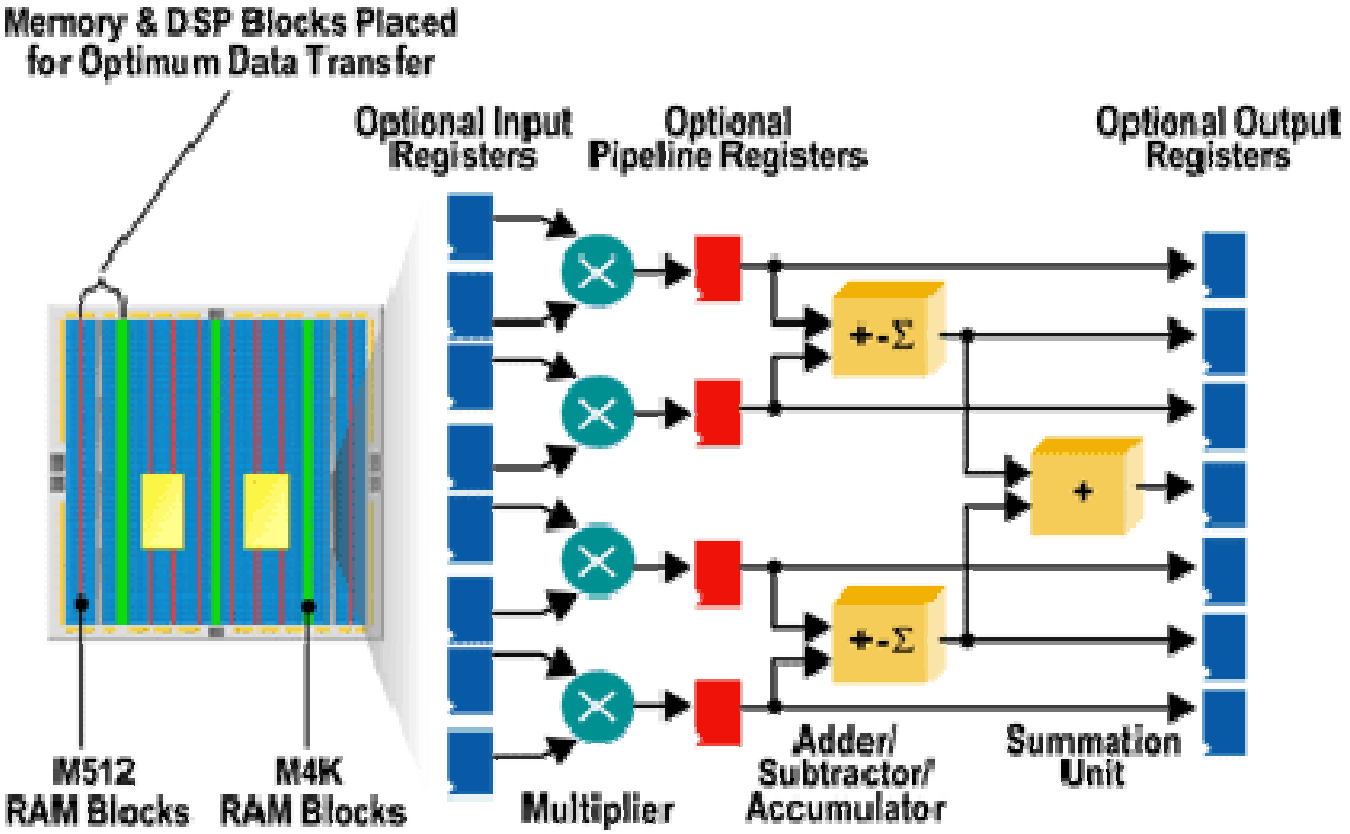
Current generations of FPGAs add a GPP on the chip

- Hardwired PowerPC (Xilinx)
- NIOS Softcore (Altera)
- MicroBlaze Softcore (Xilinx)



DSP blocks in reconfigurable architectures

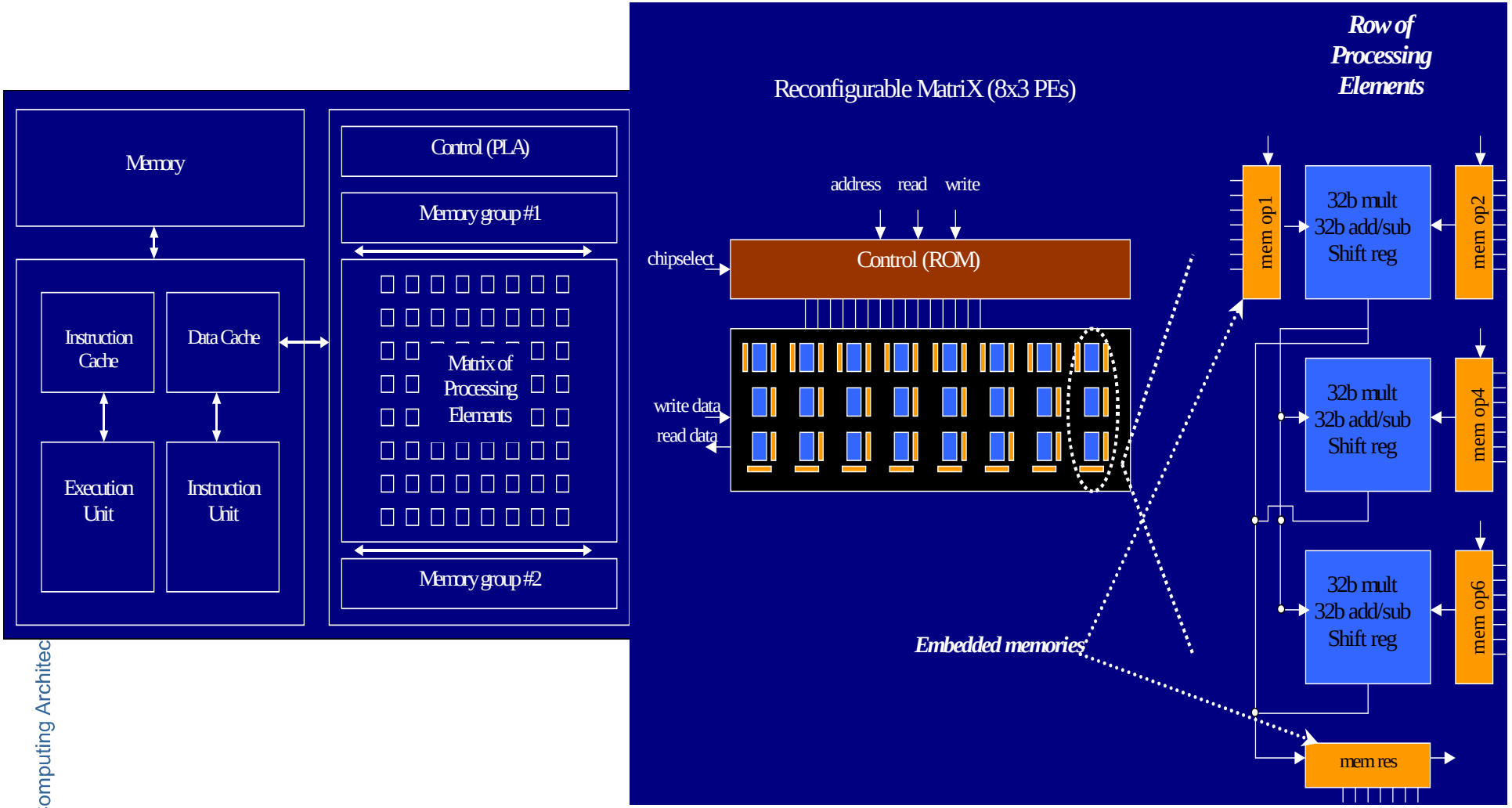
Some FPGAs add DSP blocks to increase performance of DSP algorithms
Example: Stratix DSP blocks



Stratix DSP blocks consist of hardware multipliers, adders, subtractors, accumulators, and pipeline registers

Reconf matrix of DSP blocks as media coproc.

It is possible to build complex system based on recent FPGA architectures
Taking advantage of the regular structure of the DSP blocks in the FPGA matrix



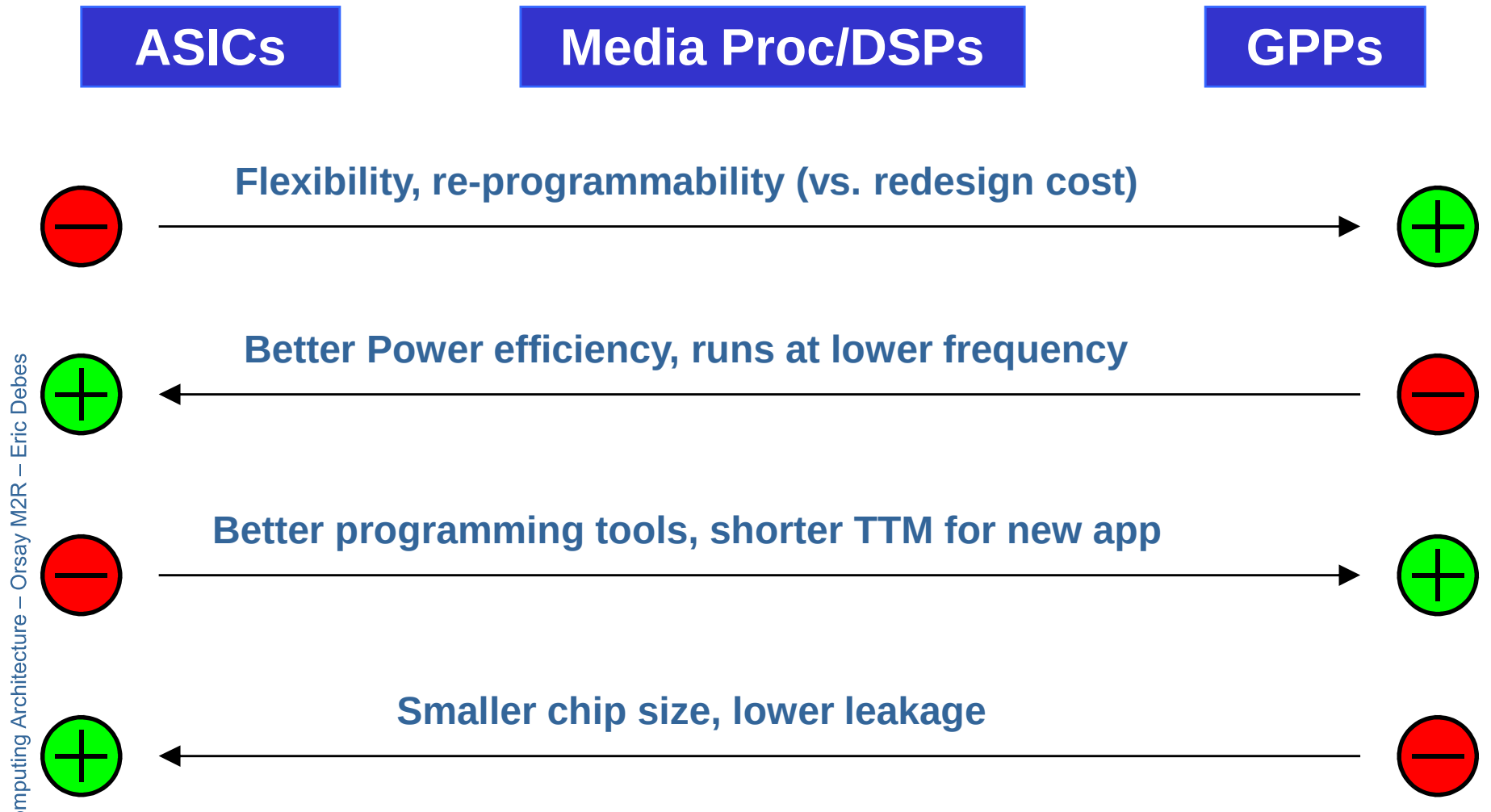
Accelerators as Coprocessors

- Dedicated circuits to accelerate a specific part of the processor
- Typically will be connected to a general-purpose processor or a DSP
- Granularity can vary
  - accelerator for a DCT function
  - Accelerator for a whole JPEG encoder
- Accelerators are very common in system on chip
- Are typically called through an API function call from the main CPU

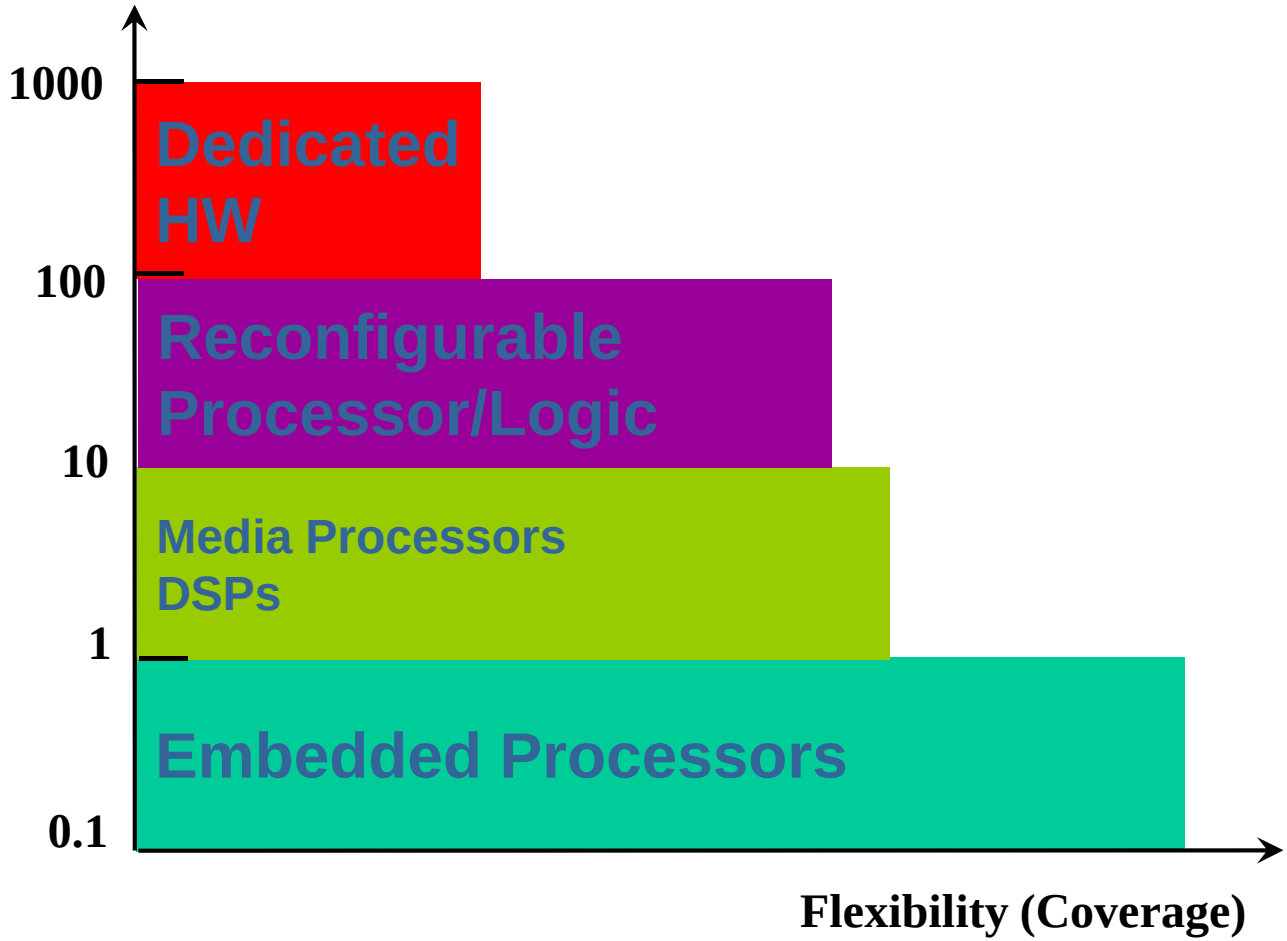
ISA extension in General-Purpose Processors

- Extending the ISA of a general purpose processor with SIMD instructions and specific instructions targeting media and communication applications is very common
- It adds application specific features to a processor and turns a general purpose processor into a signal/image/video processor.
- Example:
  - Intel MMX, SSE
  - PowerPC AltiVec
  - SUN VIS
  - Xscale WMMX
  - ARM Neon, Thumb-2, Trustzone, Jazelle, etc.

Conflicting requirements



The Energy-Flexibility Gap



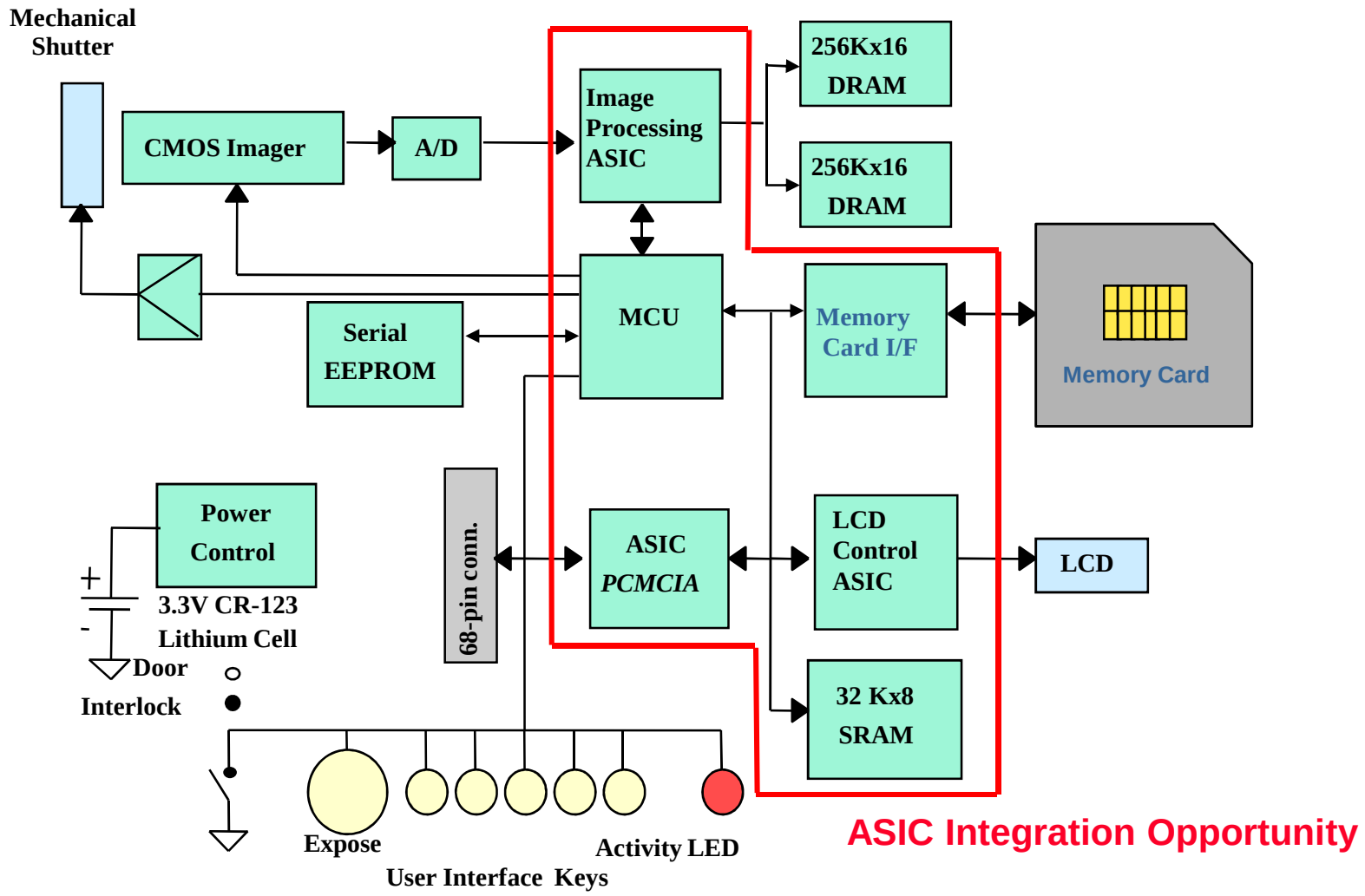
System-on-Chip

- SoCs integrate the optimal mix of processors and dedicated hardware units for the different applications targeted by the system.

- Typically integrate a general purpose processor, e.g. ARM
- Can integrate a DSP
- Accelerators for specific functions
- Dedicated memories

- Integration boosts performance, cuts cost, reduces power consumption compared to a similar mix of processors on a card

Digital Camera hardware diagram



MPSoC: A Platform Story

What's a platform?

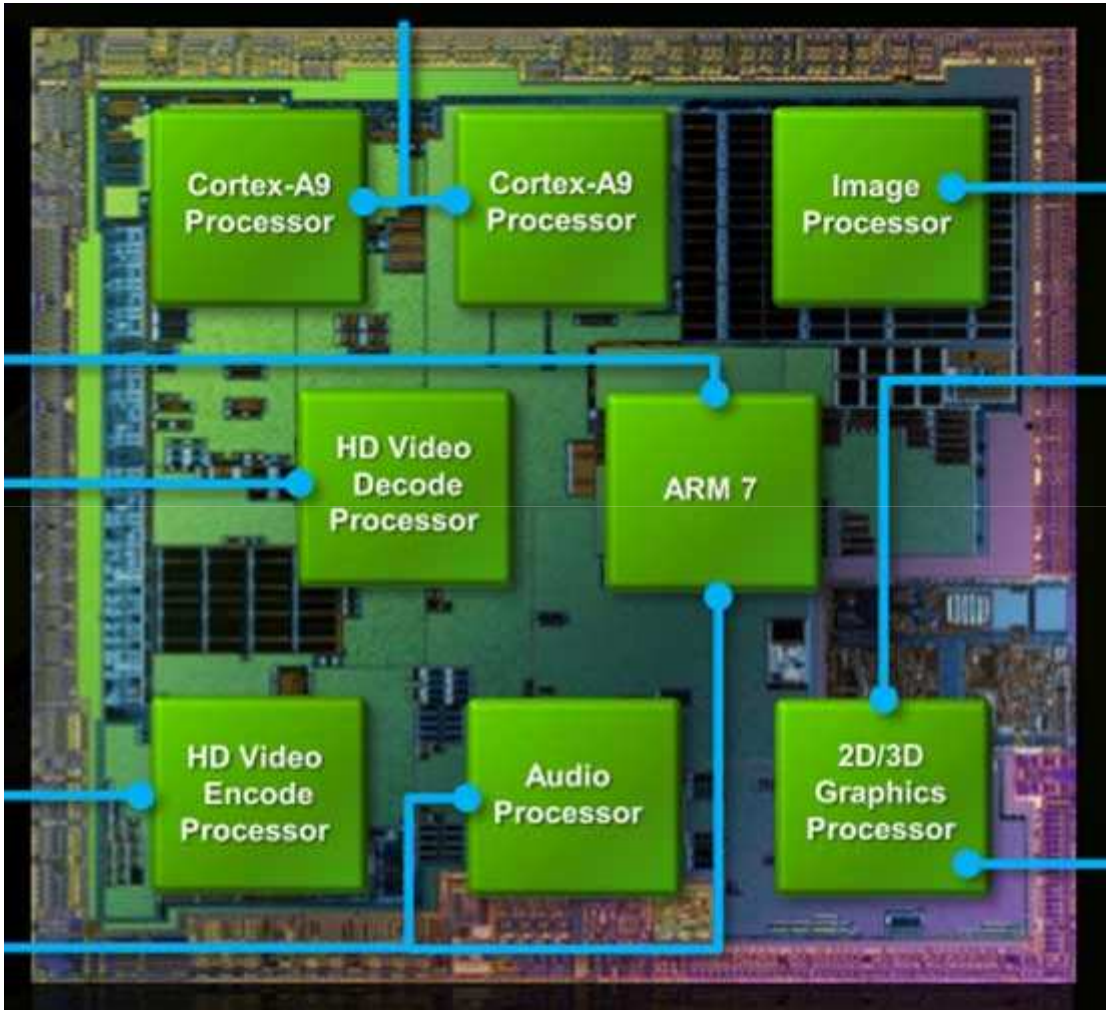
"A coordinated family of architectures that satisfy a set of architectural constraints imposed to support reuse of hardware and software components"

Best of all worlds:

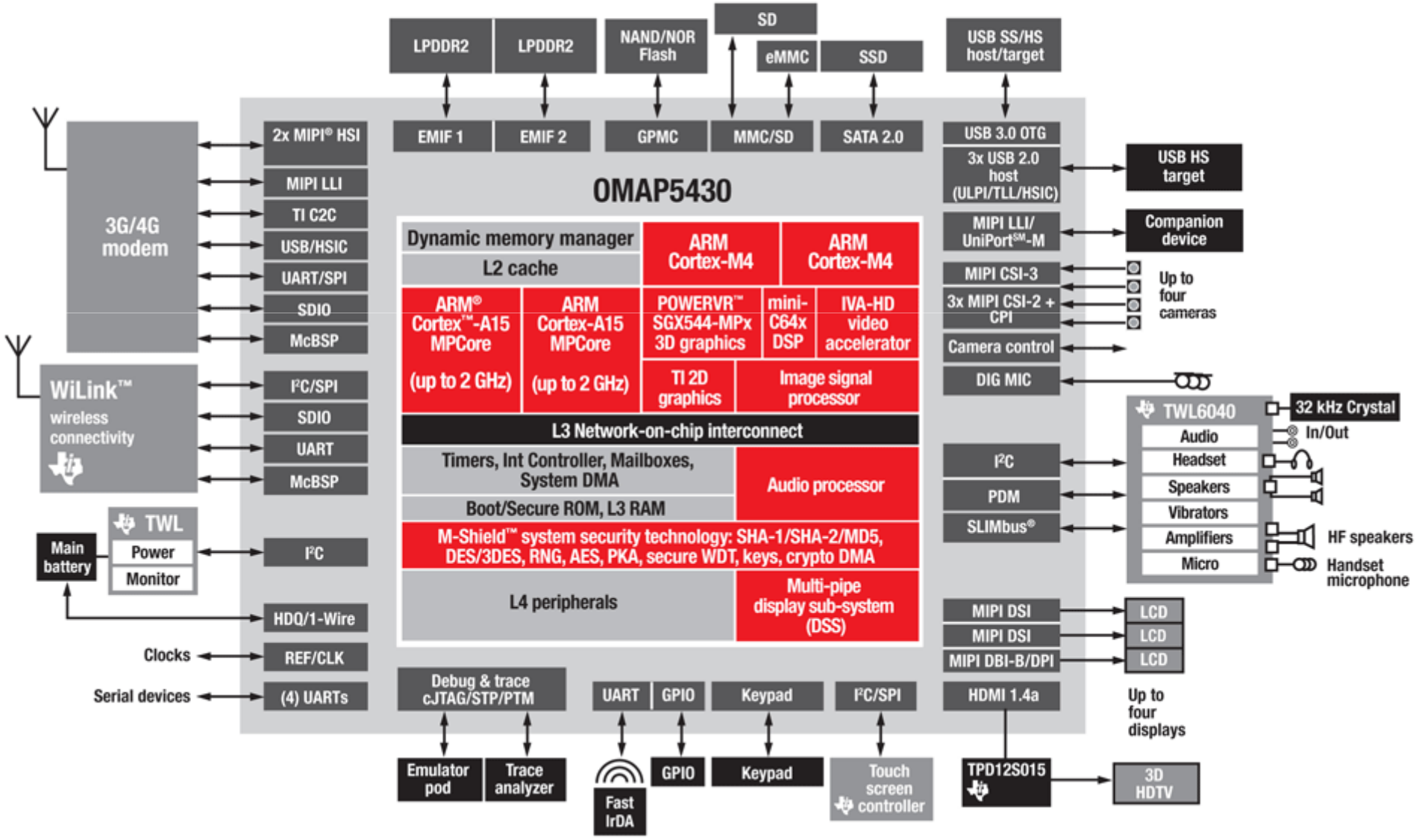
- Provides some level of flexibility
- While being power efficient
- And enabling some level of reusability
- Can last multiple product generations
- Requires forward-looking platform based design to integrate potential future application requirements in today's platform

Programming model and design efficiency are key!

Nvidia Tegra



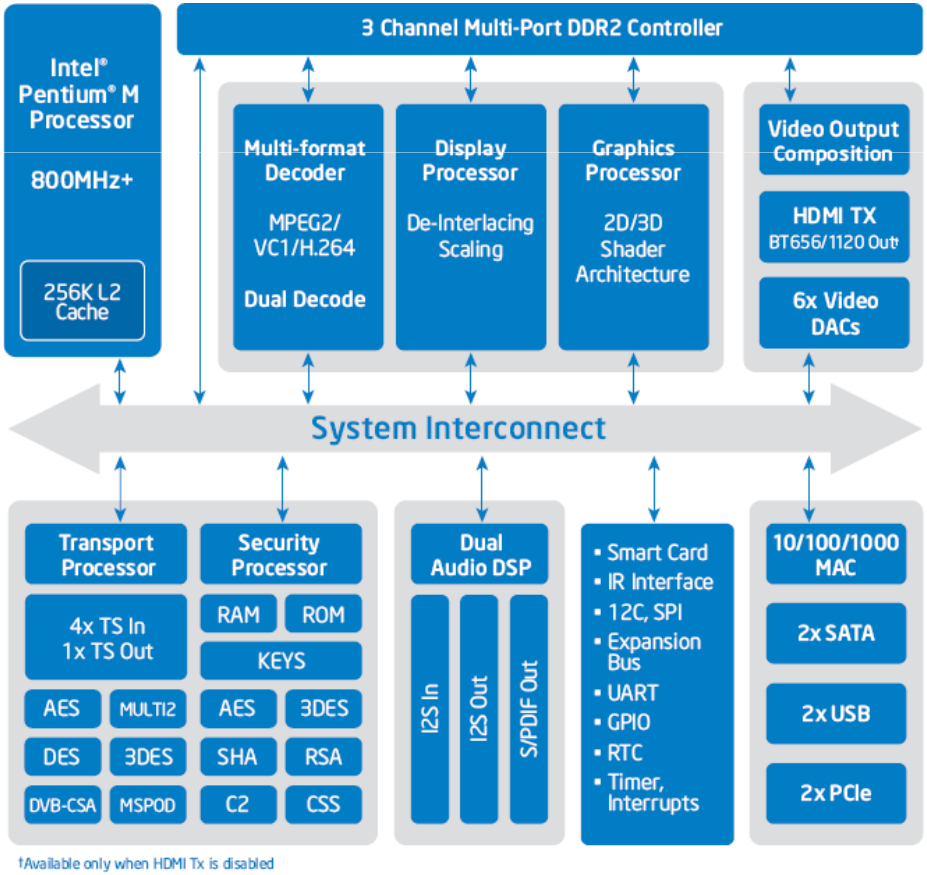
TI OMAP5430 SoC

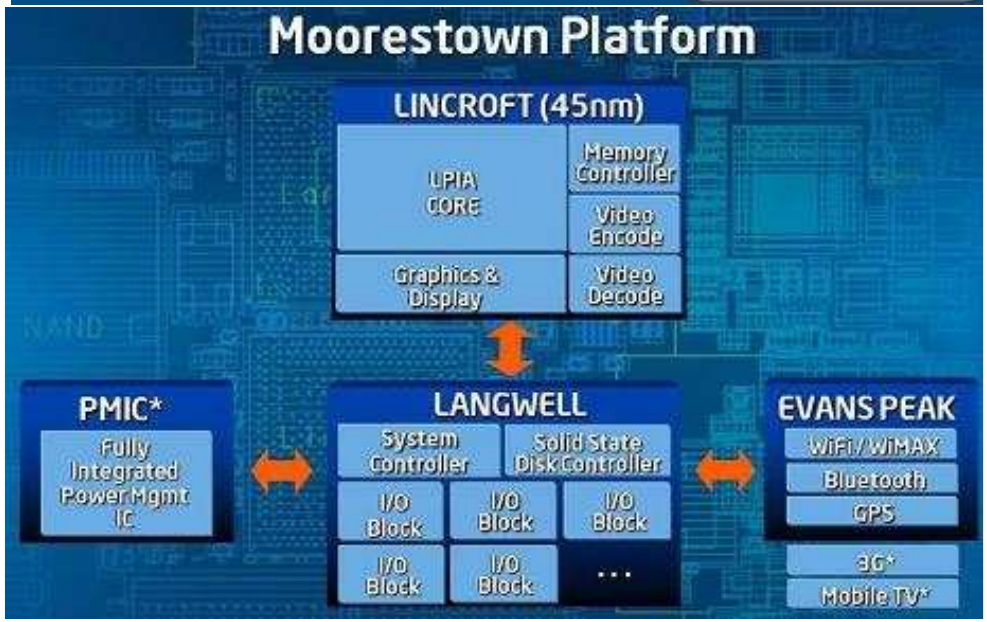


Consumer Electronics Platform Examples

Intel Atom based for:

- Mobile Internet Devices
- Consumer Electronic Devices
- Embedded Market





Summary

- Embedded Signal Processing Architectures have multiple opposite constraints
  - Performance
  - Power
  - Size/Price
- Power/performance tradeoffs are crucial for an efficiently design system
- A wide spectrum of processors to handle such applications
  - From simple in-order pipelined general purpose processors
  - Out-of order processors
  - Symmetric multicore architectures for better power efficiency
  - Heterogeneous System on Chip
  - Many-core/GPGPUs