How many cores are too many cores?

Dr. Avi Mendelson,
Intel - Mobile Processors Architecture group
avi.mendelson@intel.com

Disclaimer

- No Intel proprietary information is disclosed.
- Every future estimate or projection is only a speculation
- Responsibility for all analysis, opinions and conclusions falls on the author only. ☹
  - It does not means you cannot trust them… ☺

Agenda

- What is the motivation for new generation of parallel processors on a die
- Comparing many cores vs. multi-cores vs. asymmetric core approaches
- My conclusions and future directions

Motivation of the new trend

“The old good days” of computer architectures

- Performance used to came out of process technology and new single threaded architectural improvements.
  - Every 18-24 months new process is announced
  - The target of any new process is to shrink the dimension of the transistors on 0.7 (ideal shrink)
  - As a result, on the same die area, we can have more transistors each of them runs in higher frequency
- Every 5-7 years we had a new architecture generation
  - Used to target better single threaded performance

Simple processes scaling rules.

- If optimal scaling could be achieved (0.7 shrink) we can
  - Double the number of transistors on same area
  - Improve the frequency (performance) in ~50%
  - Consume the same power for the same are
  - Preserve the power density.

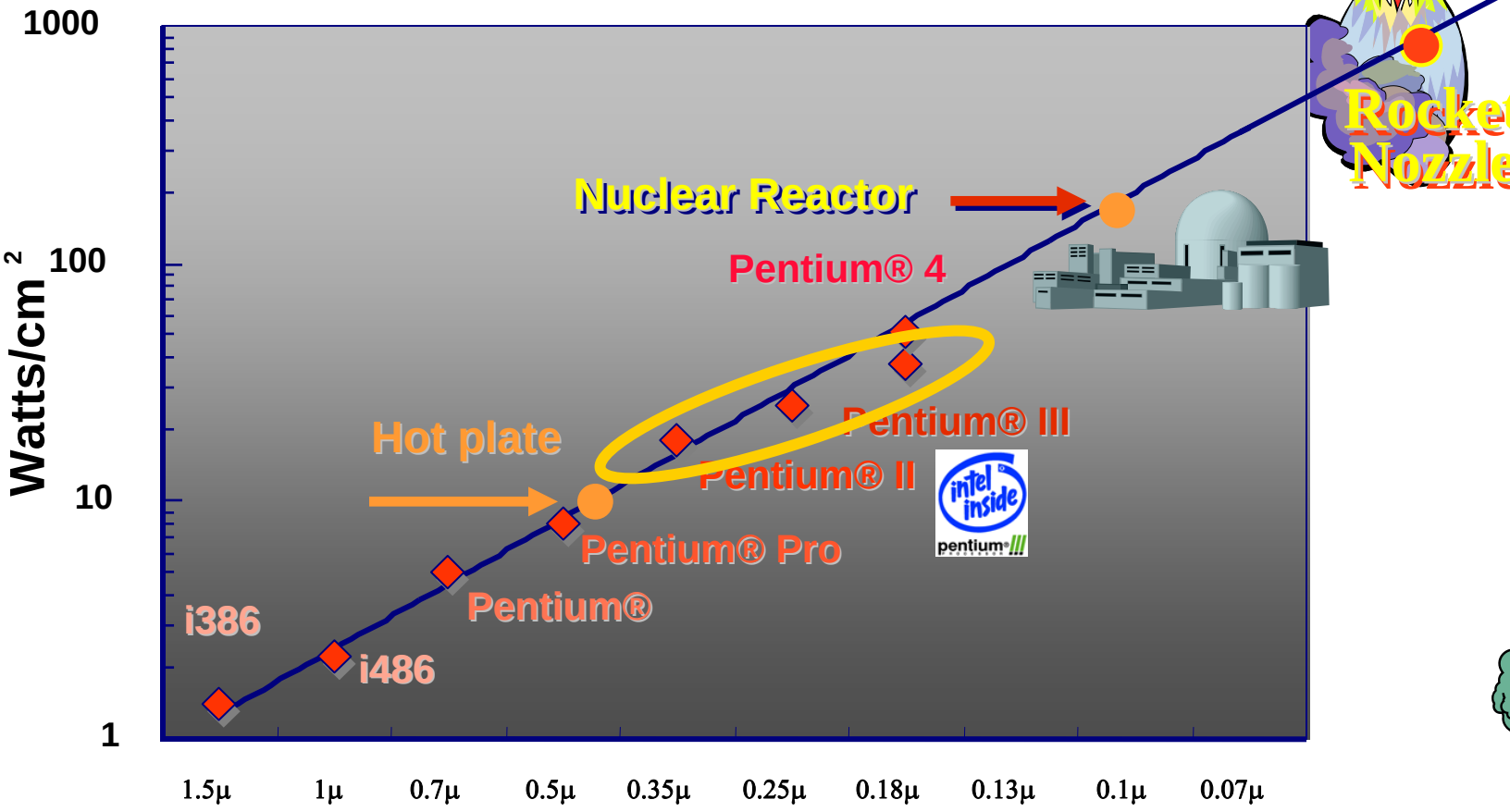
Ideal Scenarios...

- **Ideal "Shrink"**
  - ***Same μarch***
  - **1X #Xistors**
  - **0.5X size**
  - **1.5X frequency**
  - **0.5X power**
  - **1X IPC (instr./cycle)**
  - **1.5X performance**
  - **1X power density**
- **Ideal New μarch**
  - *Same die size*
  - 2X #Xistors
  - 1X size
  - 1.5X frequency
  - 1X power
  - 2X IPC
  - 3X performance
  - 1X power density

Process Technologies – Reality

- **But in <u>reality:</u>**
  - New process is not ideal anymore
  - New designs squeeze frequency to 2X per process (Moore's law)
  - New designs use more transistors (2X-3X to get 1.5X-1.7X perf)
    - The die size increases
- **So, every new process and architecture generation:**
  - Power goes up about 2X
  - Power density goes up 30%~80%
- **This is bad, and…**
- **Will get worse in future process generations:**
  - Voltage (Vdd) will scale down less
    - No ideal shrinking anymore
  - Leakage is going to the roof and it heavily depends on temperature

Power Density

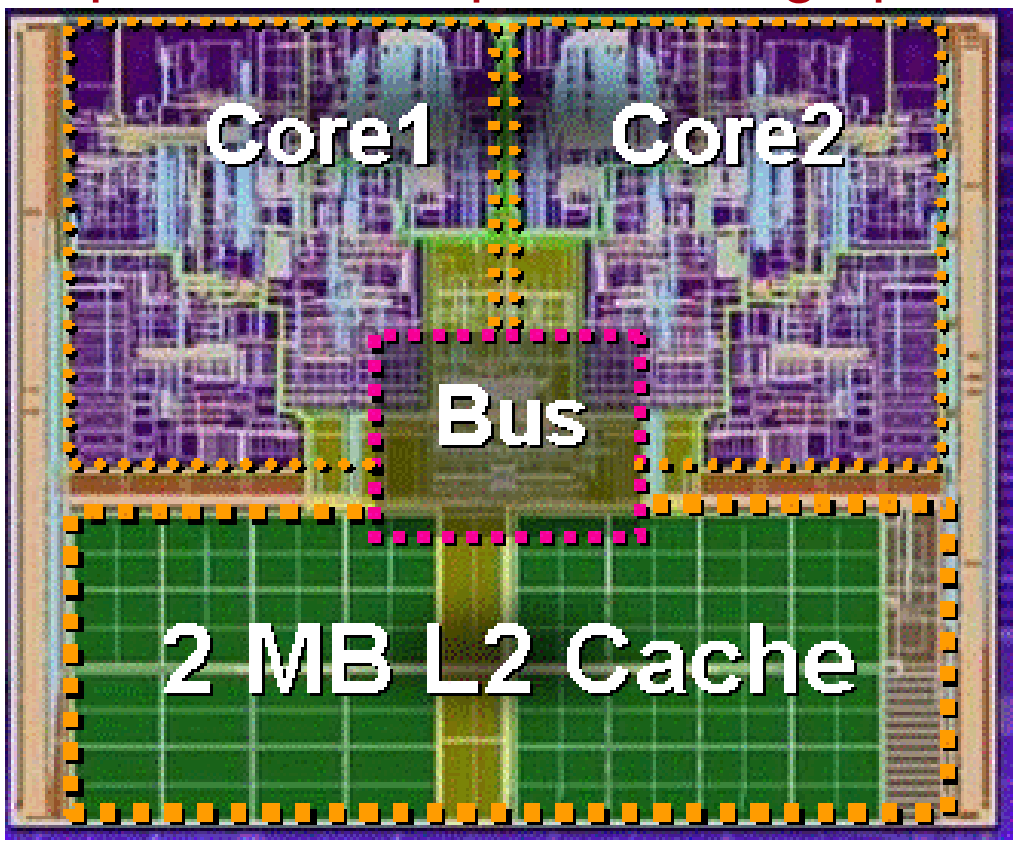


* "New Microarchitecture Challenges in the Coming Generations of CMOS Process Technologies" – Fred Pollack, Intel Corp. Micro32 conference key note - 1999.

Outcome -

- We can't build microprocessors with ever increasing power density and die sizes
- The constraint is power and power density – not manufacturability
- The design of any future micro-processor should take power into consideration. We need to distinguish between different aspects of power:
  - Max power (TJ)
  - Power density - hot spots
  - Energy → static + dynamic
- In order to achieve better single threaded performance improvement at the same power envelop, we need new micro-architecture innovation

Single threaded performance is too difficult to be achieved at the same power envelop, so--let's go parallel



- Intel® core™ Duo (Yonah) was the first CMP processor Intel developed for the mobile market. The following processor core™ Duo-2 (Merom) is used for all the different segments.

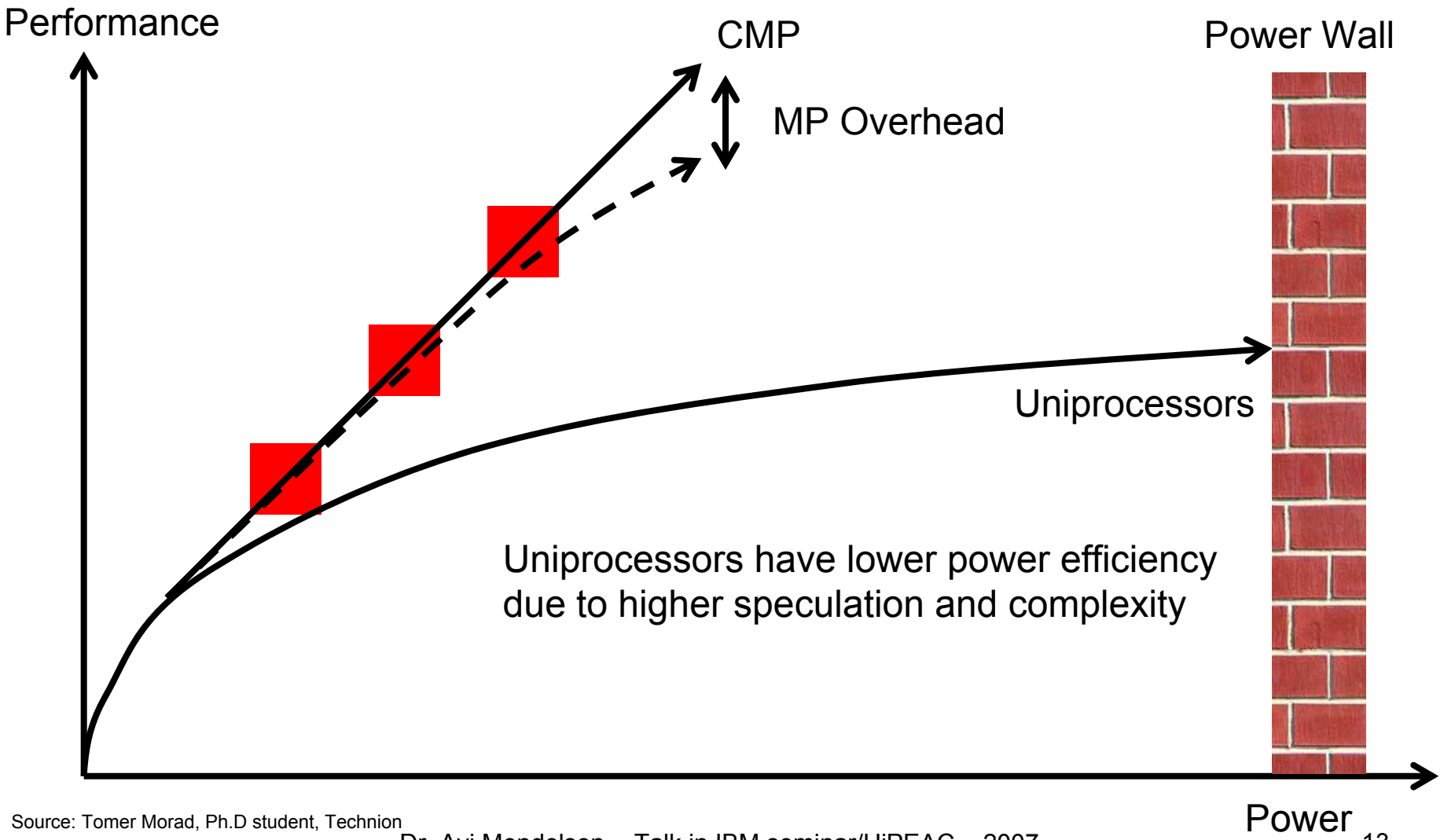
(More information can be found in Intel Journal of Technology, May, 2006)

Theoretical calculation of Single vs CMP cores power for the same performance

- In theory, power increases in order of the cube of the frequency. (2.5 is more realistic factor)
- If we assume that frequency approximates performance
  - Doubling performance by increasing its frequency growth the power exponentially
  - Doubling performance by adding another core, growth the power linearly.

- Conclusion: as long as enough parallelism exists, it is always more efficient to double the number of cores rather than the frequency in order to achieve the same performance.

CPU Architecture - multicores



Source: Tomer Morad, Ph.D student, Technion

Can we partition the system to have more cores forever?

- There are at least three camps in the computer architects community
  - Multi-cores - System will continue to contain small number of “big cores” – Intel, AMD, IBM, etc
  - Many-cores – System will contain a large number of “small cores” – Sun
  - Asymmetric cores – a combination of small number of small cores together with large number of small cores – IBM *Cell architecture*.

How many core are too many cores? – A simple Analytical model

- Basic assumptions
  - Suppose that performance of a core proportional to square-root of its area (Polack rule)
  - The thermal of each core is proportional to its power and the overall thermal capacitance is
    $$T = \sum(1/T_i)$$
  - Power - *Work* done per time unit (Watts)
    Active power: $P = \alpha C V^2 f$
    (α : activity, C: capacitance, V: voltage, f: frequency)
    Static power is out of the scope of this model.

Small cores – big cores

- Suppose same area is partitioned to m smaller processors
- Each core has the area of A/m, performance of m*SQRT(A/m) and power of $K_1*(C/m * F^{2.5})*m=K_2$
- For the same power we can get much better performance.

- Suppose we like to achieve same performance
- Each core can run at F/m
- $P_{new} = m*K*(C/m)*(F/m)^{2.5}$
  - $P_{new}=K*C(F/m)^{2.5}$
  - $P_{old}/P_{new} = m^{2.5}$

So if m → ∞ P→ 0 Perf. → ∞

But:

- F ≠ performance
- Interconnect do not scale
- Static power mater

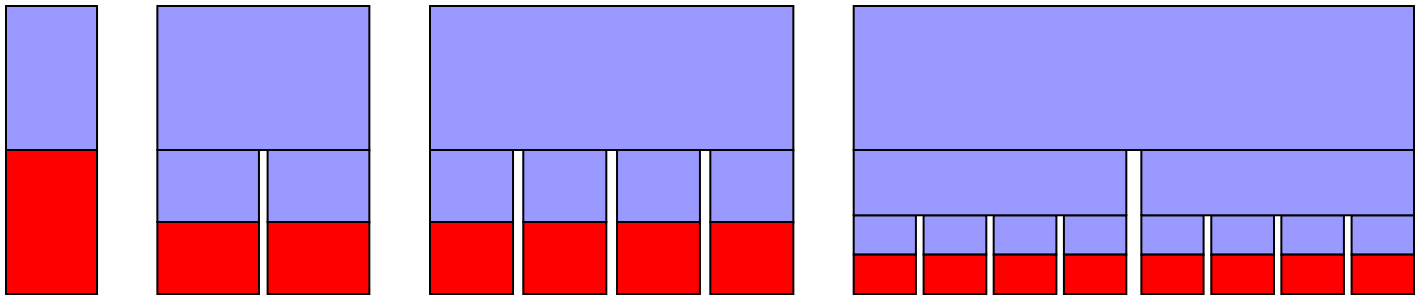
AND

If we try to build it

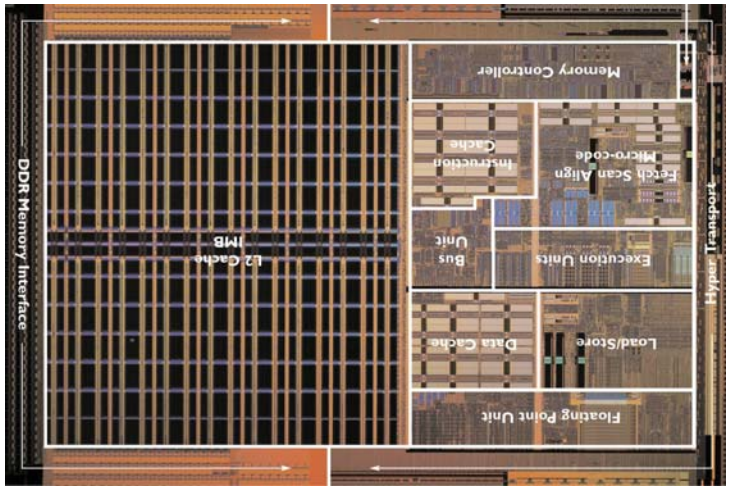
- Small die allows only simple architectures
- Memory BW and latency do not scale

Example: Caches and I/O traffic

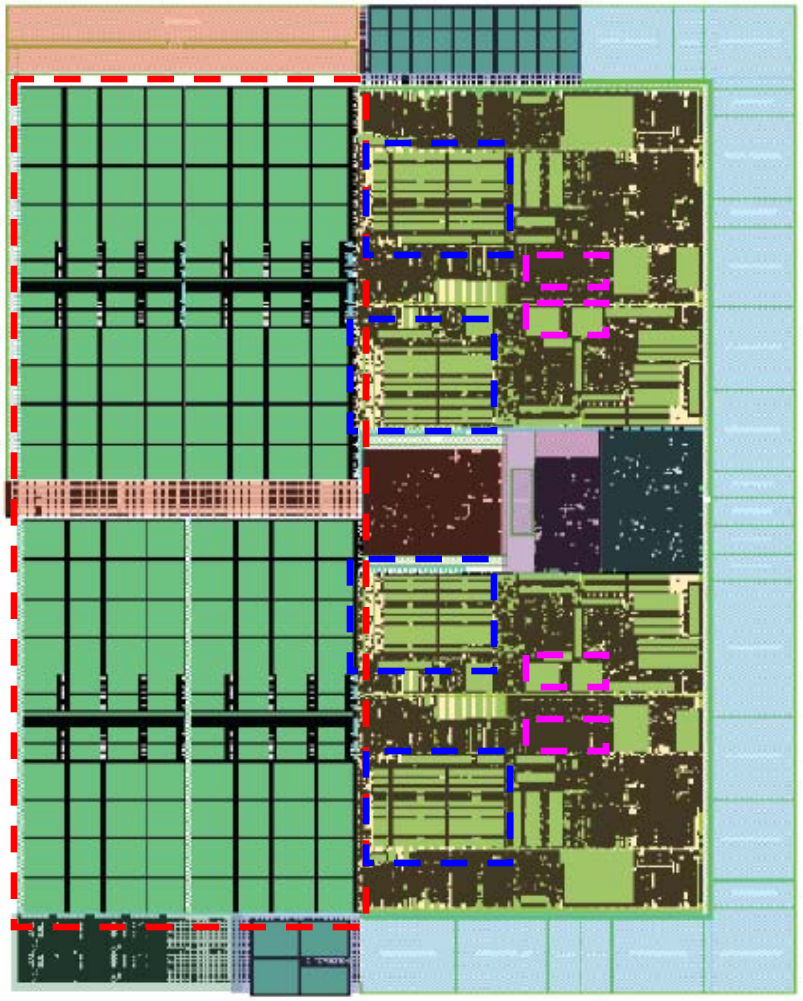
- In order to reduce I/O we need to increase the local memory (caches) respectively.
- Thus the proportion of the memory within the overall logic increases in time.
- Let's assume an hierarchy model where at each node half of its area is devoted to memory and half to logic
- At most 2 cores can share a memory hierarchy



	1	2	4	16	32
Logic	1/2	1/4	1/4	1/8	1/8



Opteron: 2M L2, 2x64K L1



K8L: 512Kx4 L2, 4x2x64K L1
shared 2M L3 extendable.

Implications:

When increasing the number of cores, the "active" area of each core is less than 1/m

It farther reduces the performance of each core

Another example of the slogan - *"The difference between theory and practice is always greater in practice than it is in theory"*

Symmetric Multiprocessor Model

- Parallelism coefficient: $0 \leq \lambda \leq 1$
- Area of each Core: a
- Number of cores: n

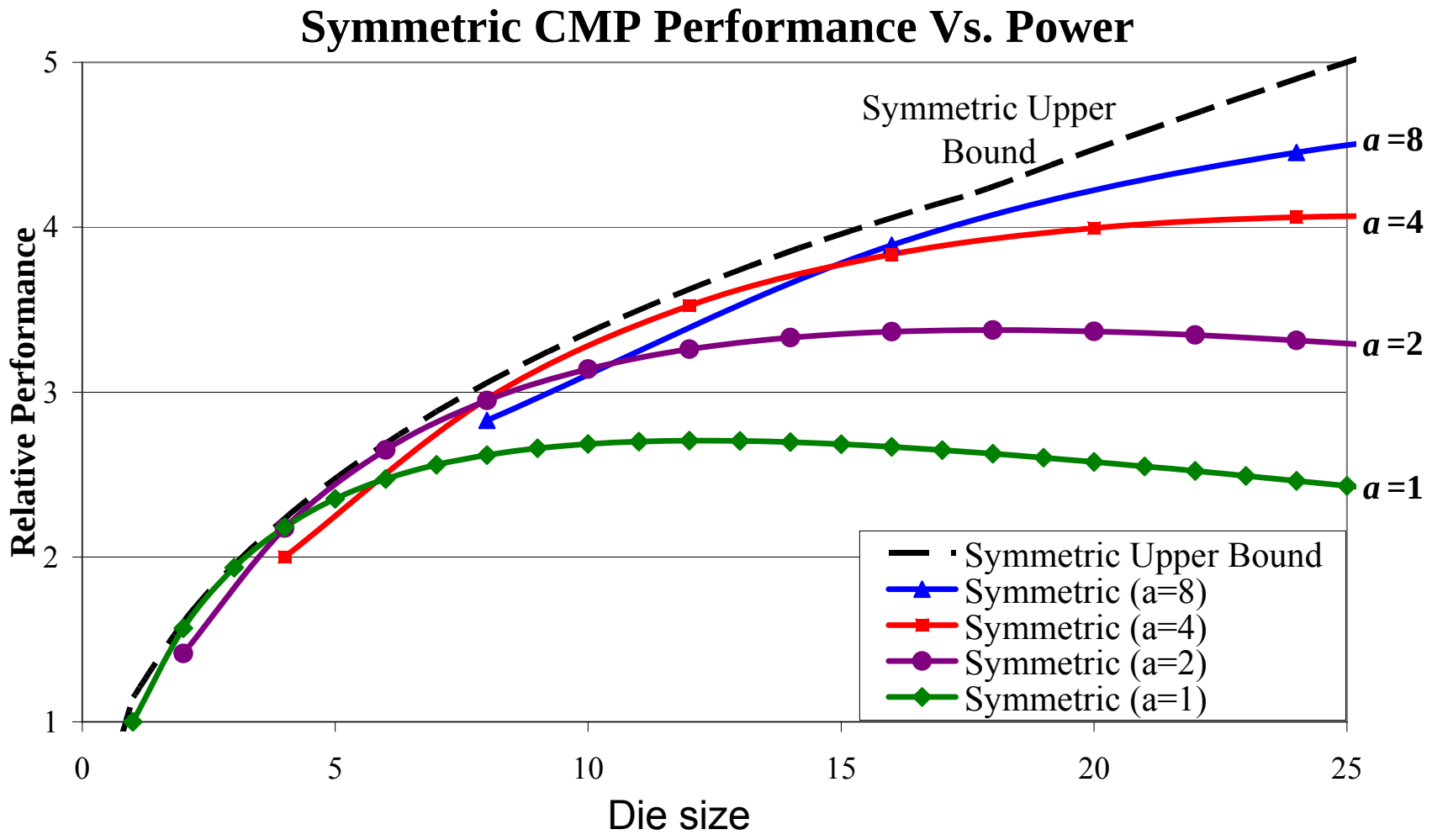
CPU1	CPU2
CPU3	CPU4

λ Denotes the number of dynamic instructions within parallel phases, divided by the total number of dynamic instructions.
Fully serial programs have λ=0
Fully parallel programs have λ=1

- Sou

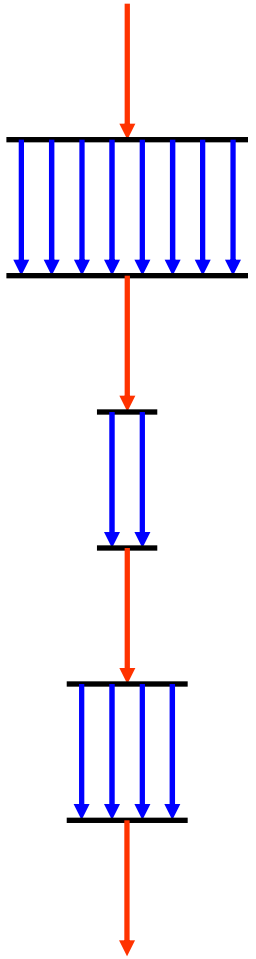
eiser, Avinoam Kolodny, Mateo Valero, and Eduard Ayguadé. "Performance, Power Efficiency, and Scalability of Asymmetric Cluster Chip Multiprocessors." In Computer Architecture Letters, Volume 4, July 2005.

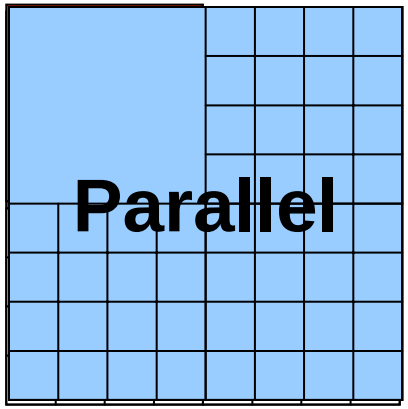
Symmetric CMP Performance (λ=0.75)



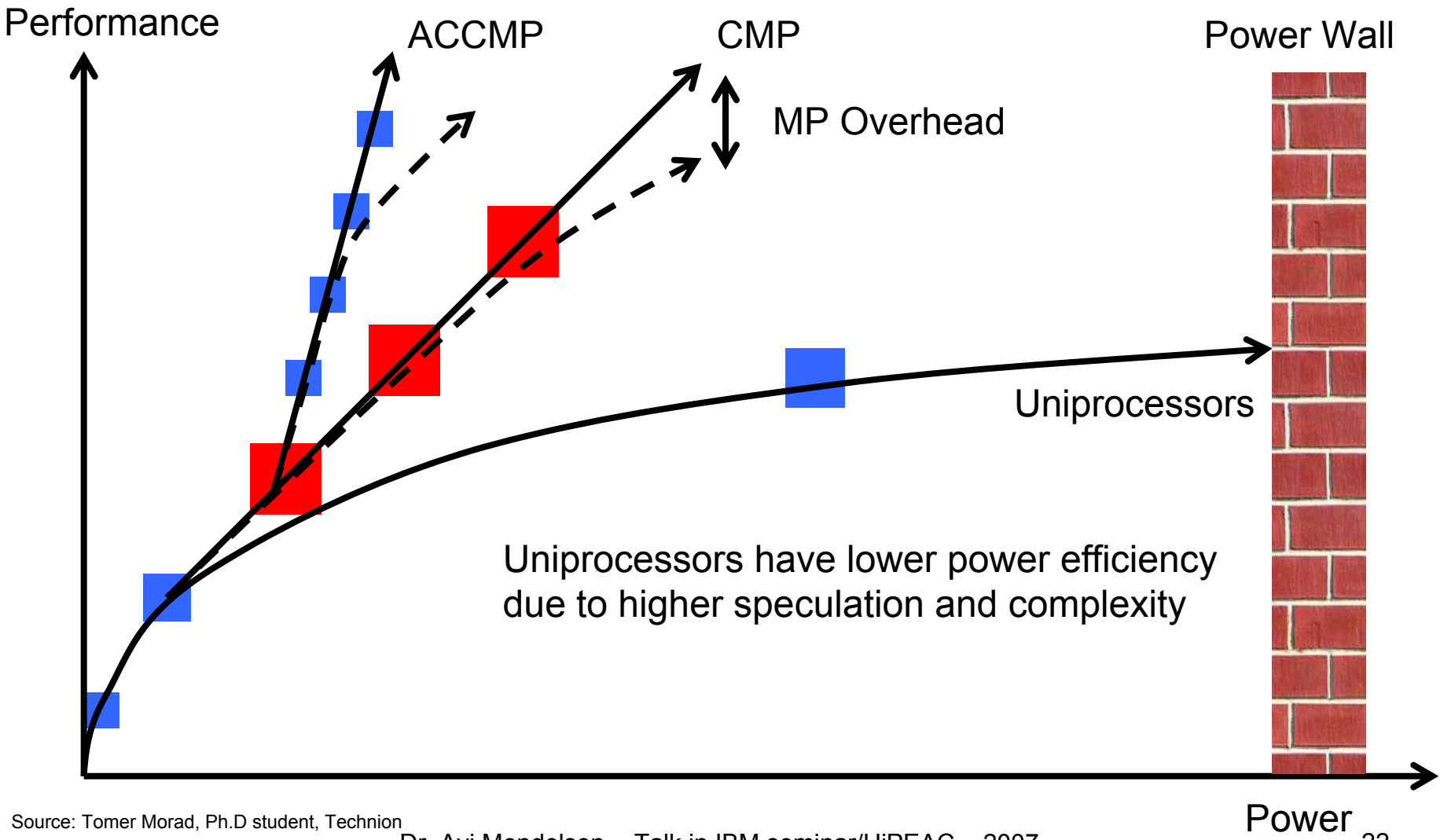
Asymmetric approach – Single Application

- Large core area: βa (β>1)
- Small cores of area: a
- Serial phases execute on large core
- Parallel phases execute on all cores
- This model can be extended to other models such as GPGPU and parallel execution between BIG and small cores





CPU Architecture - multicores



Source: Tomer Morad, Ph.D student, Technion

Conclusions for Symmetric cores

- The sequential portion of the code governs the overall performance
- Adding cores (in a naïve way) may reduce the performance
  - For same area, the sequential part will be executed slower
- λ decrease when number of cores increases (more sequential part as a result of the cost of synchronization primitives
  - We need “almost perfect parallelism” in order to gain many cores architectures.

How to ease the problem

- Reduce the serial part
  - “smart prefetch”; e.g., scout, helper threads
  - Push technology; e.g., Cell – push the data to the right locations in parallel (DMA) and then execute
  - Fast synchronization – can be done in HW or SW
- Change the programming model; e.g., transactional memory
- Increase parallelism within the application

Easier say than to be done

We have been there too many times!!!

So, how many cores are too many cores?

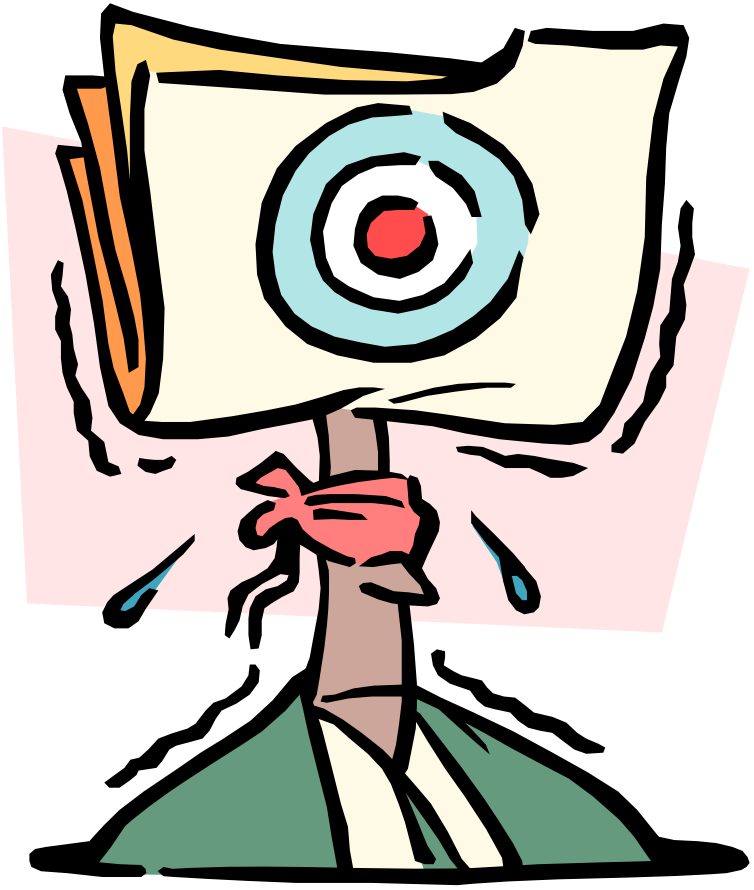
- Many cores will never flourish without a breakthrough in compilers/application/new programming models – but this can take looooong time
- Many cores can be very efficient for special purpose computing and for “special MIPS” since we can “tailor”’ the applications to the HW (and the HW to the software).
- Multicore has its limitation since cannot show future longevity but if we can keep improving the single core performance in parallel to moderate increasing the number of cores, we may gap the time till many cores become reality
- Asymmetric computer is mostly appealing as a combination of general purpose computers and special purpose. It can also become in a form of FPGA and GPGPU (e.g., CUDA, CTM)

Asymmetric computer - challenges

- I assume that asymmetric computers will be used mainly for “special purpose”
  - Graphics, Multi-Media, Fix functions,
- In order to achieve that many technologies are still at a research phase
  - Interconnect – most likely NoC--Network on Chip
  - Memory hierarchy:
    - NUCA, DNUCA, UMA, NUMA?
  - Programming model:
    - OpenMP, MPI, CUDA, CTM, other?
    - Comilers/debugger/ developing environment
  - OS
    - Do we need one scheduler or two (as today) or unified one?
    - Do we need virtualization layer for that?

I believe that asymmetric computer architectures will generate many new research opportunities for at least the next decay.

Question?



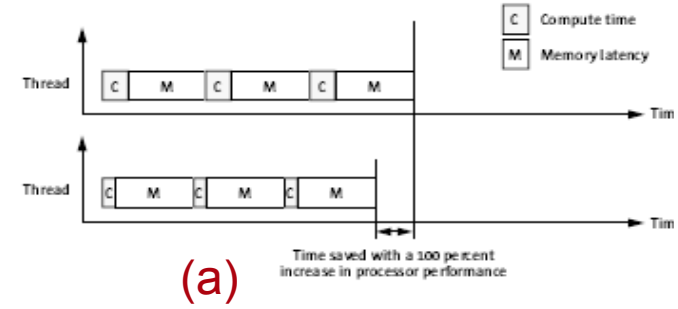
Multi-core – Intel AMD and IBM

- Intel and AMD companies have dual and quad core architectures
  - For Dual core - Intel uses shared cache architecture and AMD introduces split cache architecture for the first generation and shared L3 for the new generation
- For servers, **analysts claims** that Intel is building a 16 way Itanium based processor for 2009 time frame.
- Power4 has 2 cores and power5 has 2 cores+SMT. Analysts claim that IBM consider to move in the near future to 2 cores+SMT for each of them.
- Xbox has 3 Power4 cores.

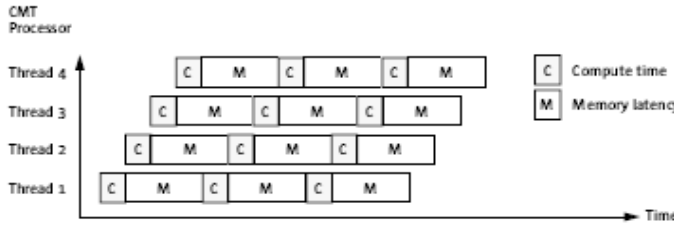
- All the three companies promise to increase the number of cores in a pace that fits the market's needs.

back

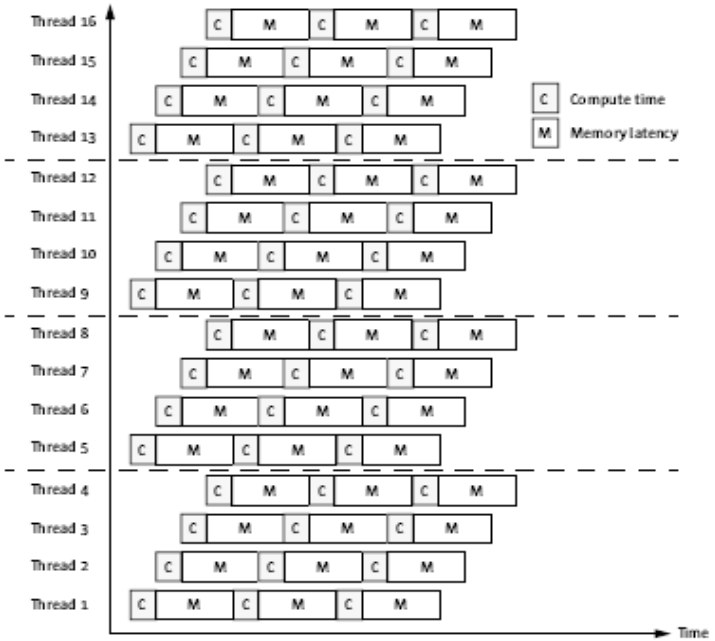
Many-cores –Sun – Sparc-T1:Niagara



(a)



(b)

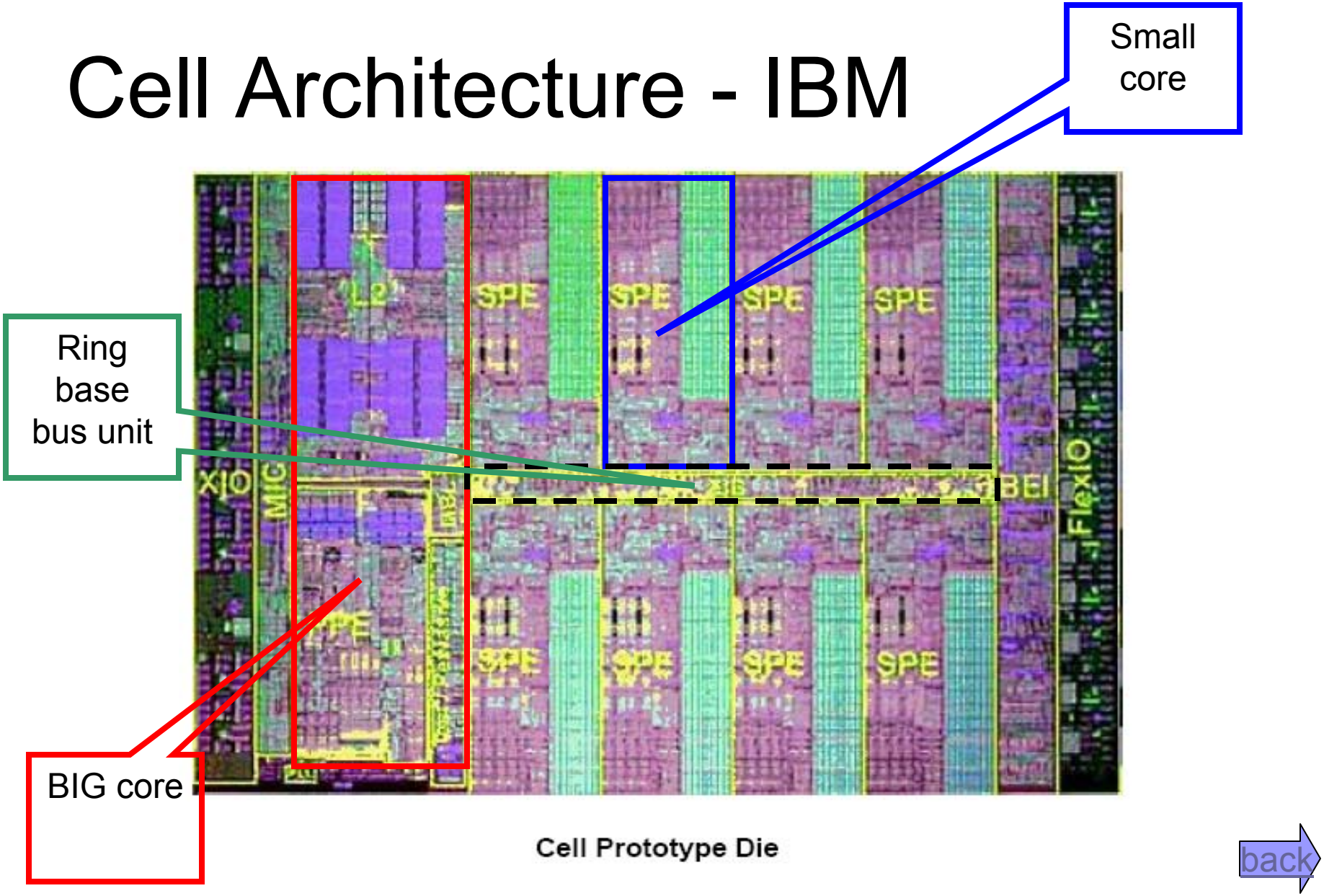


(c)

- Looking at in-order machine, each thread has computational time followed by LONG memory access time (a)
- If you put 4 of them on die, you can overlap between I/O, memory and computation (b)
- You can use this approach to extend your system (c)
- Alewife project did it in the 80's

back

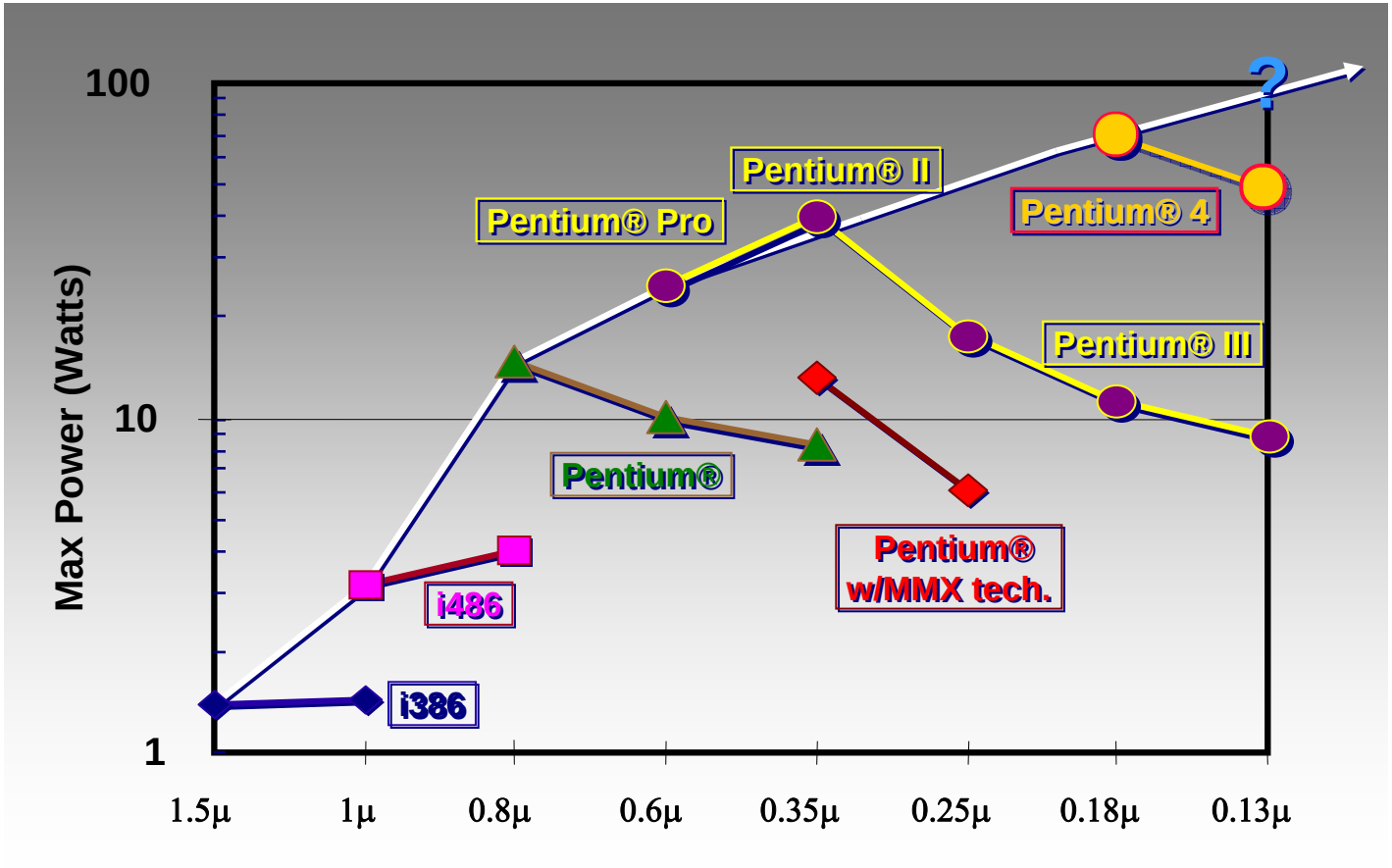
Cell Architecture - IBM



Cell Prototype Die

back

Processor Power Evolution



- *Traditionally:*
  - *new processor generation always increases power*
  - *Using the same micro-architecture with new process decreases its power*