

“End of Moore’s law or end of General Purpose Computing?”

This presentation is based on a panel I lead as part of the BMW workshop

Dr. Avi Mendelson – Microsoft R&D Israel

Hertzeliya, 6/1/2011

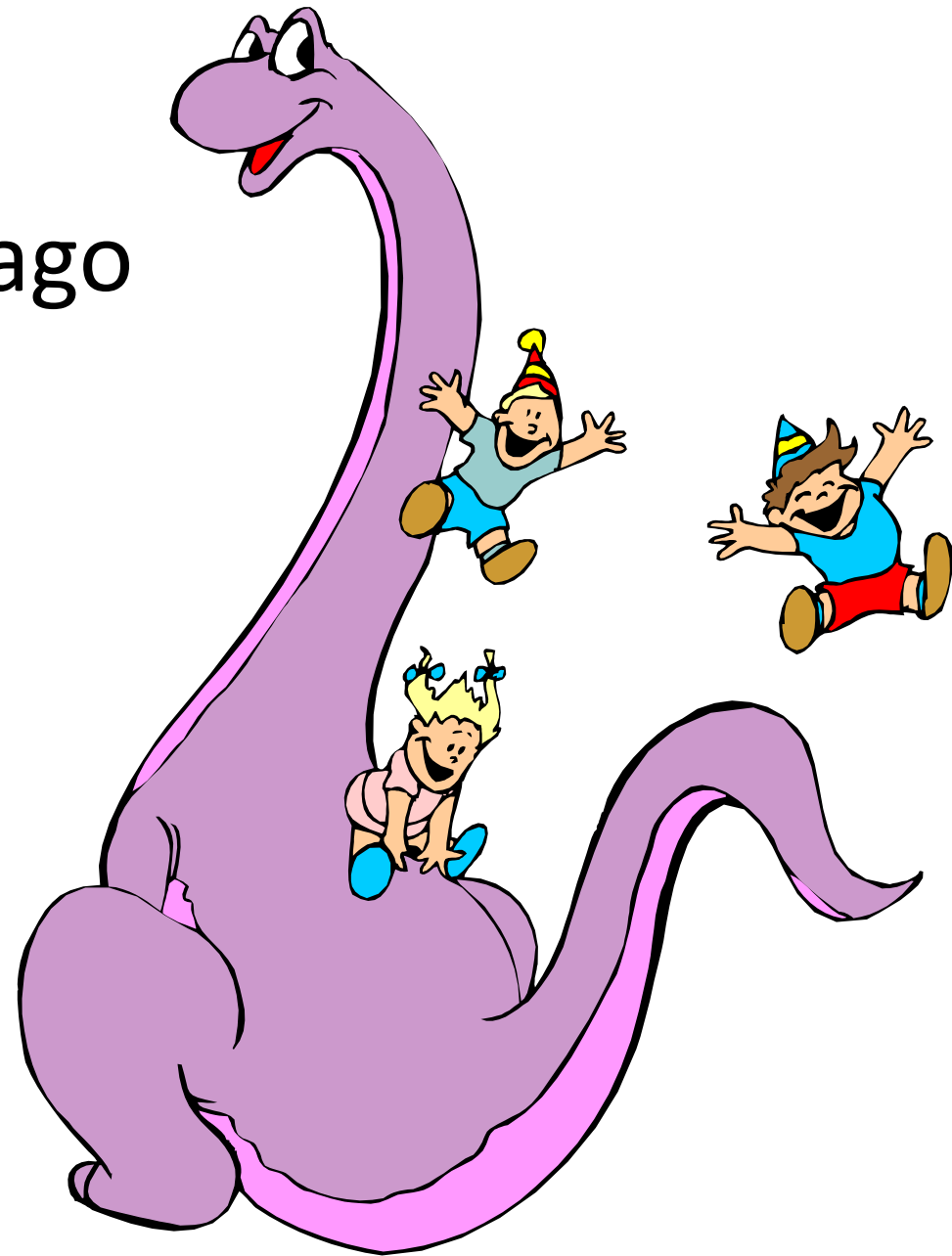
Disclaimer

- All information and ideas presented in this document are mine.
- No information presented in this document represents Microsoft's view and/or any of its future products.

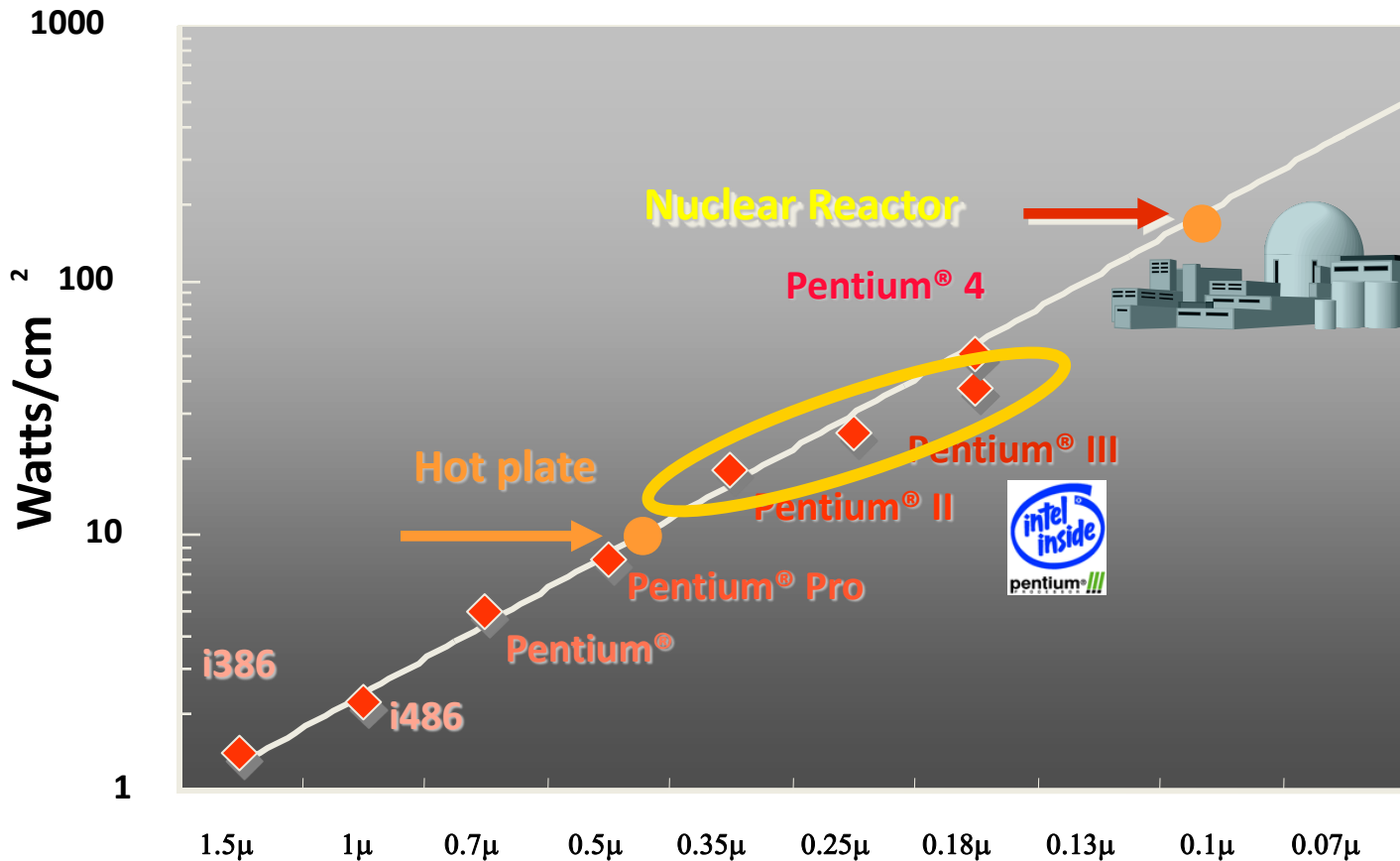
Agenda

- Past
- Present
- Future
- My Conclusions + possible implications

Past – what we
presented 5 years ago



Let's start from basics; we all start with this slide about Power Density that becomes a hard limitation



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

Why the market “Go parallel”

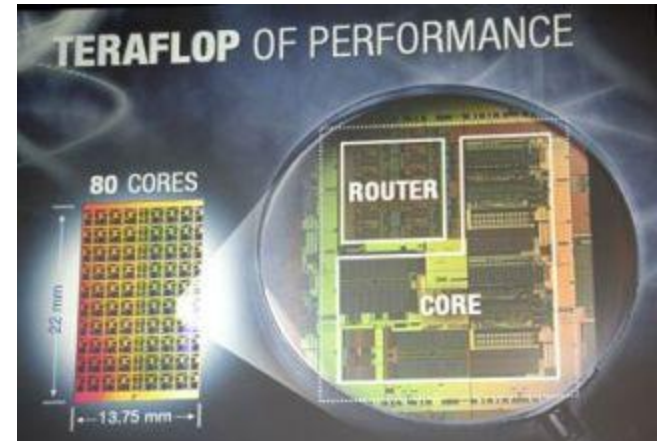
- A simple power calculation of active power is:
 - Active power: **Power = αCV^2f**
(α : activity, C: capacitance, V: voltage, f: frequency)
Static power is out of the scope of this model.
- Since voltage and frequency depend on each other (between V_{\max} and V_{\min}), approximate power change depends on freq. change as follows:
 $\Delta\text{Power} \sim (\Delta f)^{2.5}$
(in theory it should be a factor of 3, in reality the factor is closer to 2.5)
- A naïve tradeoff analysis (assuming frequency maps to performance)
 - Doubling performance by increasing frequency grows power exponentially
 - Doubling performance by adding a core, grows power linearly
- **Conclusions:**
 - (1) As long as enough parallelism exists, it is always more power efficient to double the number of cores rather than the frequency in order to achieve the same performance.
 - (2) In thermally limited environment **POWER == PERFORMANCE**

Why parallel (cont.)

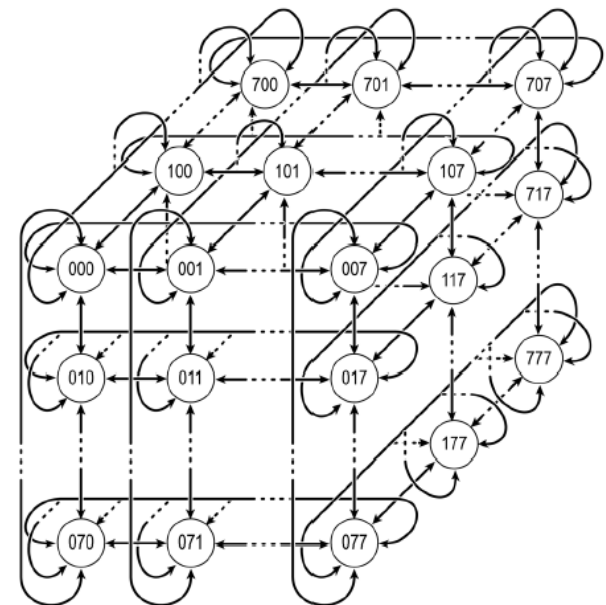
- We can't build microprocessors with ever increasing power, power density and die sizes
- The variability in the process is a major problem
 - If design for worse case, could not keep high performance and high manufacture yield.
 - Most likely we will have to learn how to live with errors at execution time.
- Transistor budget will not be the main problem
 - Devoting significant transistor budget to guarantee reliability/real-time/dependability may be a reasonable tradeoff in future commercial processors
- Cost
 - General purpose cores will look to add more value to the end user
 - Embedded systems may enjoy very low cost and/or one chip solution (including memory)

Source: Avi Mendelson - ICS Keynote speech 2007 ©

Many and Multi Cores Architectures



- Multi-cores – Add more “complex core”
 - Two years ago we introduces dual core, today many systems include four cores
 - Intel announced that next generation of processors will have up to 8 cores (16 threads) on die
- Many-cores: Put many simple cores on Die
 - Sun Negara (T1) contains 32 and 64 thread system
 - Intel Polaris research project targets 80 cores on a die to achieve Teraflop peak-performance
 - Blue Gene has hundreds of embedded cores
 - IBM Cyclops-64: supports 160 HW threads
 - Anton – 512way machine for molecular dynamic simulator



Source: Avi Mendelson - ICS Keynote speech 2007 ©

Conclusions – pointed in the past

- The future of future parallel systems is mainly depended on SW and not on HW.
 - We can build rich enough HW solutions
 - We have no idea how to develop an efficient parallel code (write, debug, runtime support, etc)
 - Heterogeneous systems, or SoC solutions seem to be the right solution, but SW limits their usage

Present



Current Architectures

HW

- Graphics claims to have hundreds of cores
- GP cores still contains handful cores
- Special purpose processors use thousands of cores.
- Heterogeneous (or integration) is here

SW

- Still struggle with many cores for GP
- CUDA is widely used for specific set of applications
- Other solutions such as parallel libraries are still struggling.

Heterogeneous seems to be the right solution

- Multicores are still needed to efficiently handle GP applications, but they are not power efficient.
- Many cores (e.g., graphics) shows a great power efficiency for domain specific applications, but fail to handle GP applications
- The “natural solution” – let’s do heterogeneous computer that will take the best of all worlds.

Different types of Heterogeneity

- Same ISA
 - Like Intel + LRB (almost the same ISA)
- Different ISA
 - Sandybridge (Intel), Barcelona (AMD) has different ISA for the graphics and the main processor.
- SoC
 - Smart phones are built out of different subsystems who are integrated together.
 - TTM (Time to market) for HW, SW and OS are major issues to be solved

Can SW Cope with the fast development in parallel HW?

- General Purpose:
 - TM is still struggling, and even if succeed, will fit only medium number of cores.
 - OpenMP and MPI are being used by experts only.
 - CUDA was proven to be efficient only for small number of applications.
- Special purpose:
 - Many library based solutions
 - Vast development of Domain Specific Libraries/environments.

Panel's agreement

- All panelists agree that the development of programming languages and systems to support parallel, GP HW, is very slow.
- HW companies will not be able to wait forever for the parallel SW to flourish
- If this is true, the SW and the HW markets will change in the near future.

Future



10 years from now

- General
 - Memory management is a big challenge. (total coherency? Island of coherency? (same problem to all other models as well))
 - How to manage I/O?
- Same ISA
 - We can put 100's or 1000's of processors on die, does it make sense to separate “big cores” from “small cores”?
- Different ISA
 - In the form of fixed logic
- SoC
 - This model will be needed, but we do not have the SW layer to support that.

Memory model, SW model and OS are the biggest challenges.

My conclusions for the future

- I believe that SW will **not** cope with the pace of the multicore HW.
- It will not be profitable for HW companies to continue increasing the number of GP cores beyond 8.
- What we will do with the rest of the transistor's budget?
 - SoC
 - Accelerators
 - Reliability
 - **Domain specific systems**

Implications (MY personal view)

- GP Hardware will become commodity → HW based companies such as AMD/Intel will have to change/disappear/find new business model
- Much of the SW for GP systems will be come “open source” and will be developed via marketplaces → traditional SW companies such as Microsoft/CA will have to change/disappear/find new business model
- Highly professional people will move the domain specific areas where small companies/startups will work in synergy with large companies
 - Large companies will build the infrastructure and tools, and the experts will develop specific solution for specific domain of problems.