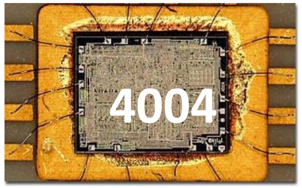
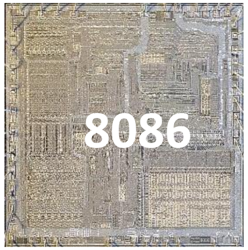
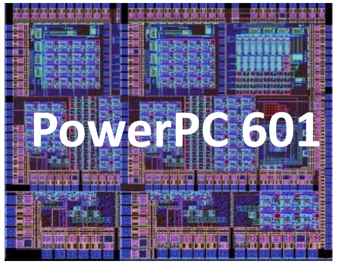


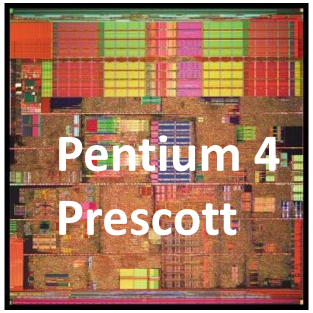
1971



1978



1992

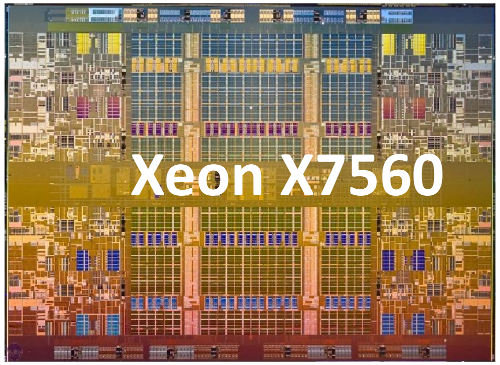


2004

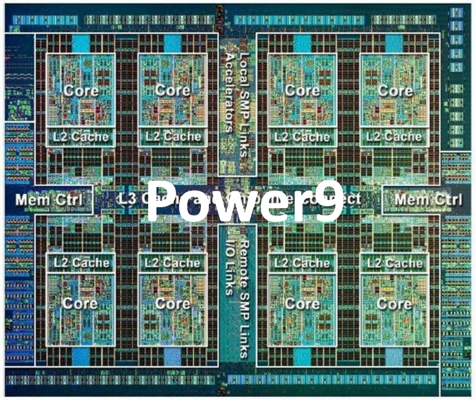
45-year CPU Evolution: 1 Law -2 Equations

Daniel Etiemble

LRI – Université Paris Sud



2010



2017

Nvidia Pascal
SM
Instruction Cache
Instruction Buffer
Warp Scheduler
Dispatch Unit
Register File (32,768 x 32-bit)
Core
DP Unit
LD/ST
SFU
Texture / L1 Cache
Tex
64KB Shared Memory

2016

Exponential Evolution

- *Are there some fundamental rules?*
  - **Moore’s Law**
  - Program execution time

Hennessy-Patterson

$$T_{ex} = IC * (CPIcpu + CPImem) * T_c = \frac{IC}{IPC * F}$$

Instruction count

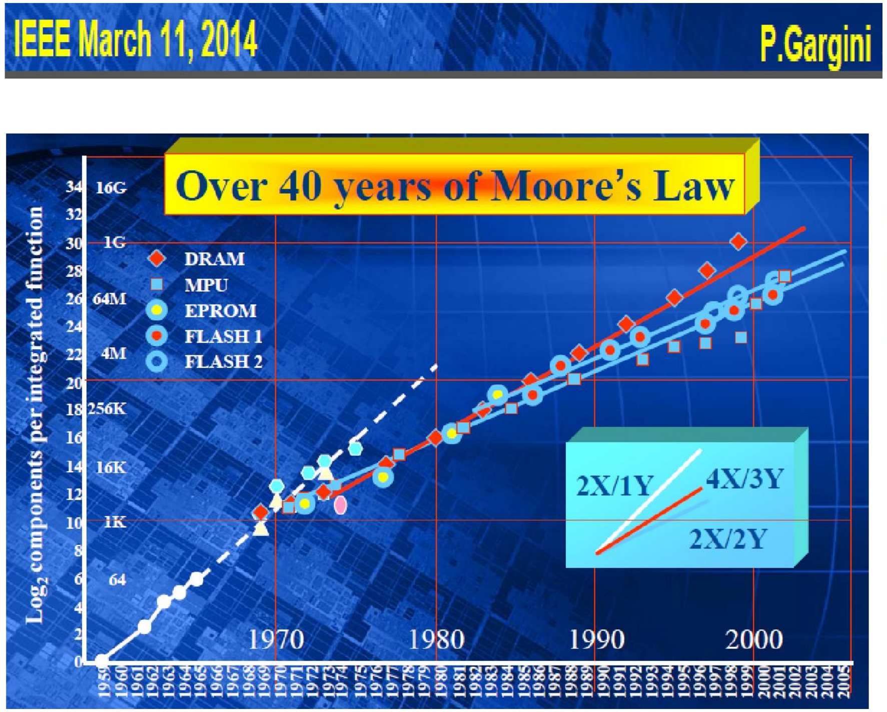
Avg Cycles/Instruction
Computing – Waiting data

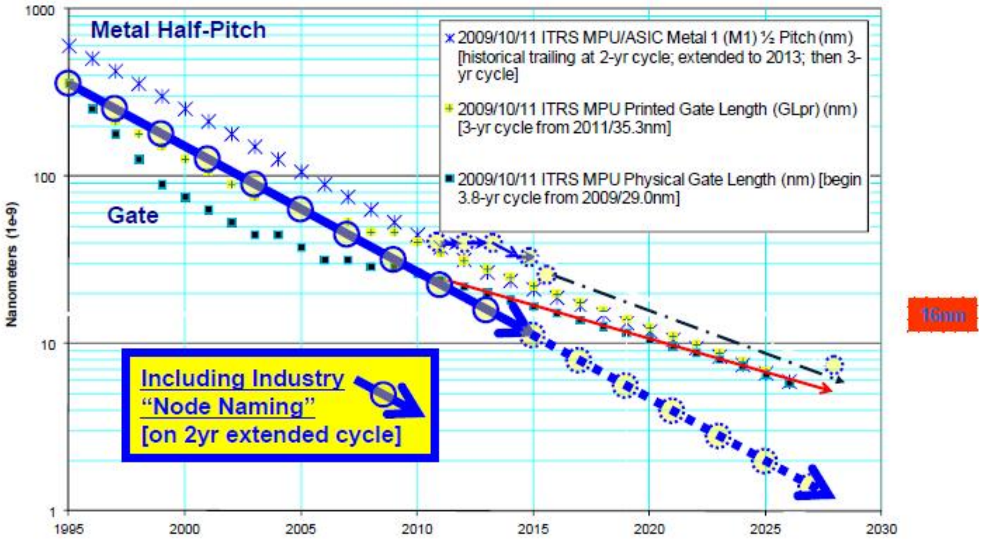
Cycle time

  - CMOS power dissipation

$$\boldsymbol{P_d = V_{dd} * I_{leakage} + \alpha * \sum C_i * V_{dd}^2 * F}$$

Moore’s law



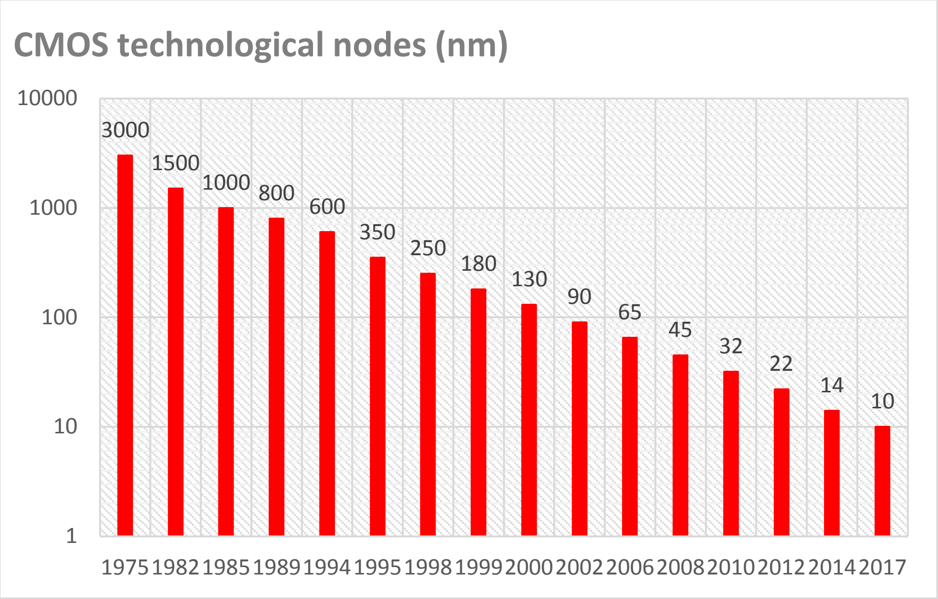


2010 2020 2030

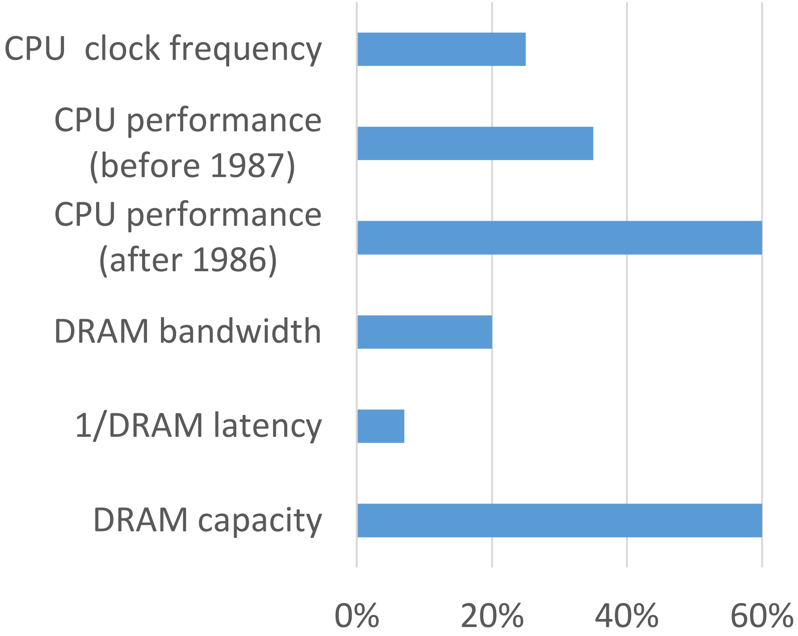
N increases

Nb transistors/chip doubles every N months (12/18/24)

Technological nodes



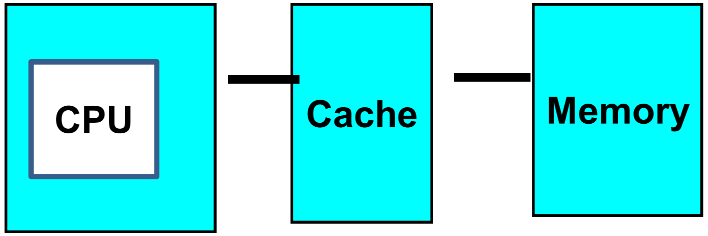
But MISMATCHES



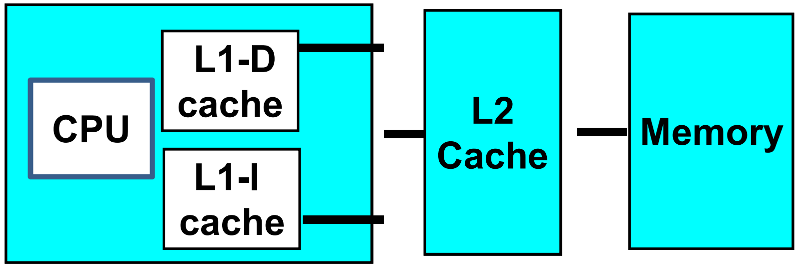
- From one node to the next one
  - **(rough approximation)**
  - Gate delay /1.4 => Higher clock frequencies
  - Increase of transistor count per area unit.
- Consequences
  - From multi-chips to one chip
  - Larger on-chip memories
  - More functionalities per chip

Cache hierarchies

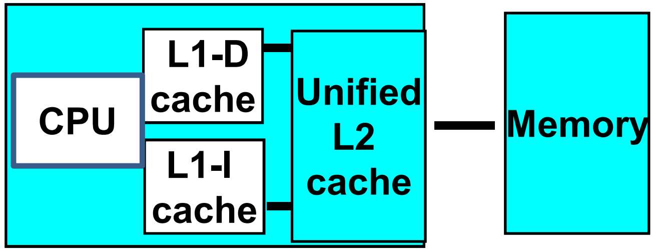
Evolution of cache hierarchies



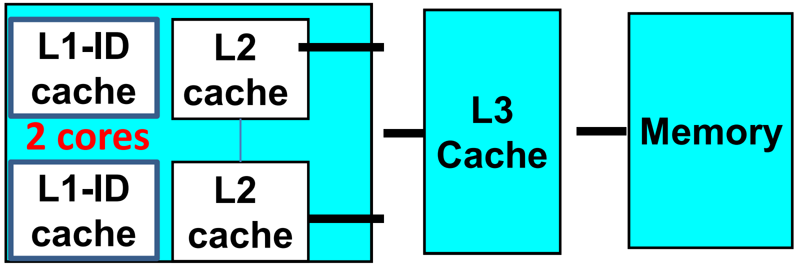
386



Pentium



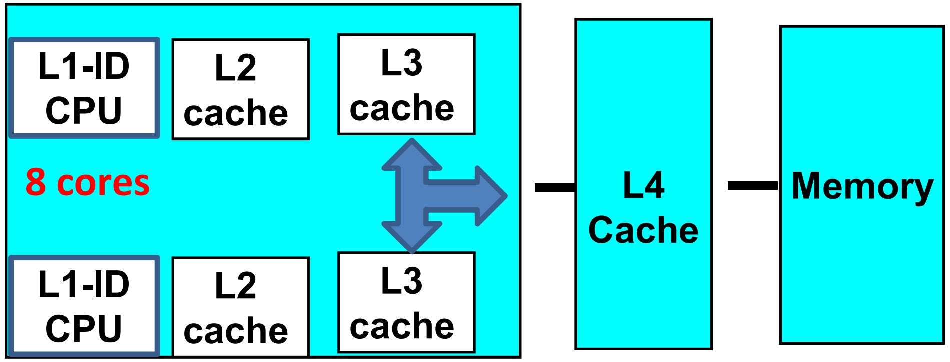
Pentium 4



Power6

Reducing CPImem

Latency & Bandwidth



Power 8

NUCA

Improving Performance $T_{ex}= \frac{IC}{IPC * \boldsymbol{F}}$

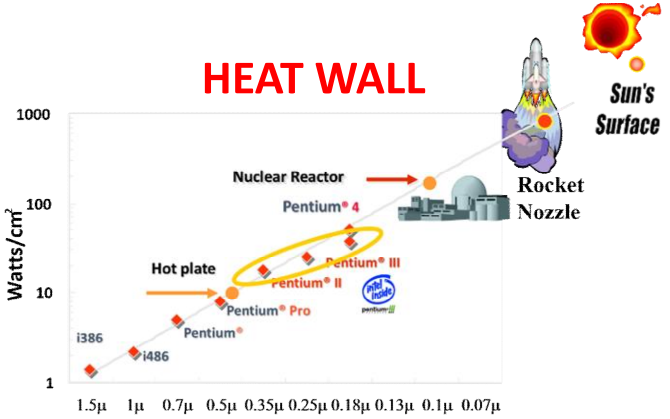
Increase **F** (Technological nodes)

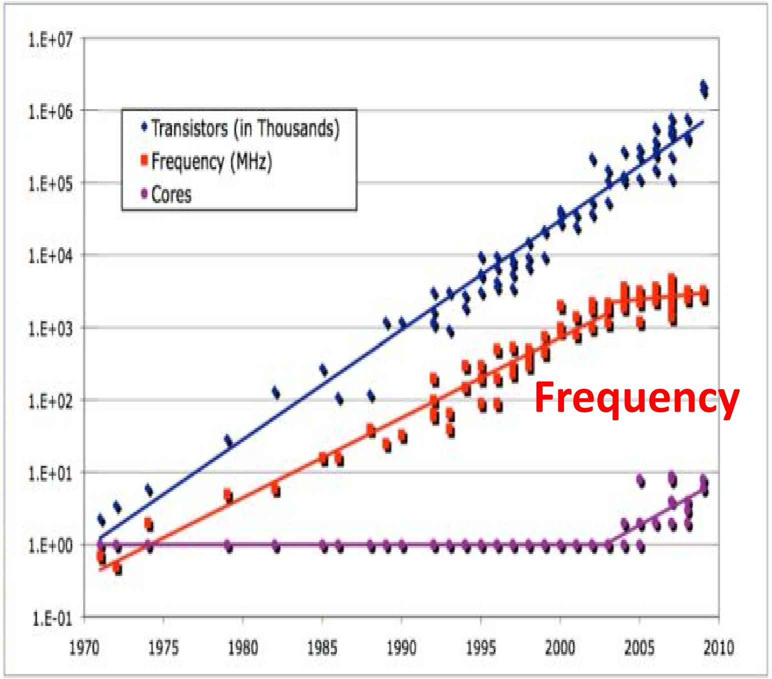
- Gate delay /1.4 => Higher clock frequencies
- Increased Performance

Increase F

- Increased Power and Power density

$$\boldsymbol{P_d = V_{dd} * I_{leakage} + \alpha * \sum C_i * V_{dd}^2 * F}$$





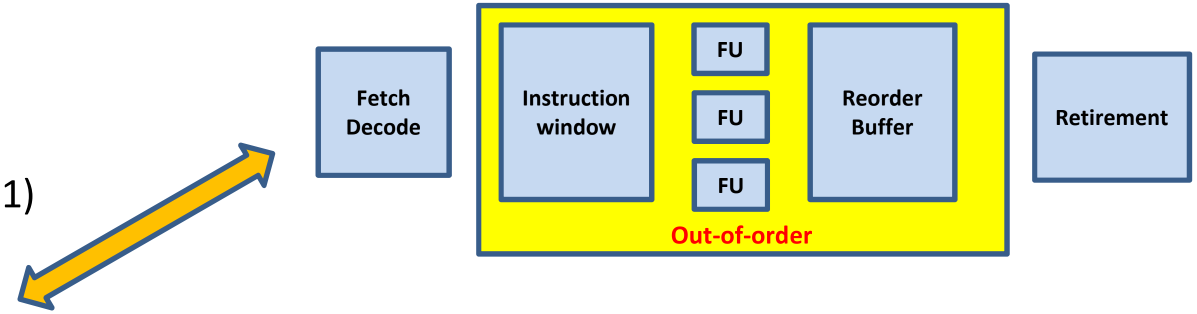
2017 : F in the 3-4 GHz range

Except water cooled IBM z14 CPU (5.2 GHz)

Improving Performance $T_{ex} = \frac{IC}{IPC * F}$

ILP in mono-processor CPU

- Scalar CPU (IPC <1)
  - Pipeline and superpipeline
- Superscalar CPU (IPC >1)
  - In-order CPUs
  - **Out-of-order CPU**
- VLIW CPU (IPC >1)



Fetch Decode

Instruction window

FU

FU

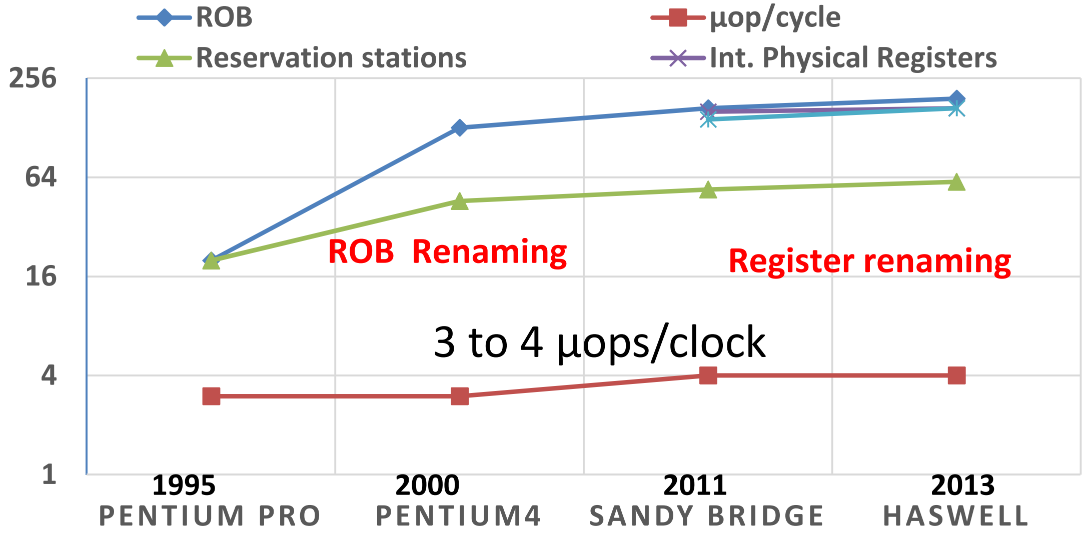
FU

Reorder Buffer

Retirement

Out-of-order

- Intrinsic limits of ILP in a sequential program & “ HW diminishing return”.
  - Larger buffer to extract µops to launch, but launching width remains constant !



Improving Performance $T_{ex} = \frac{IC}{IPC * F}$

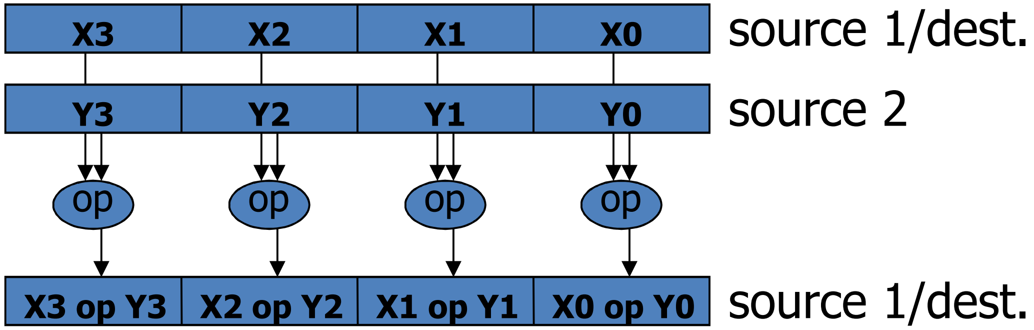
Data Parallelism in mono-processor CPU

CPU

SIMD

1 instruction with several data

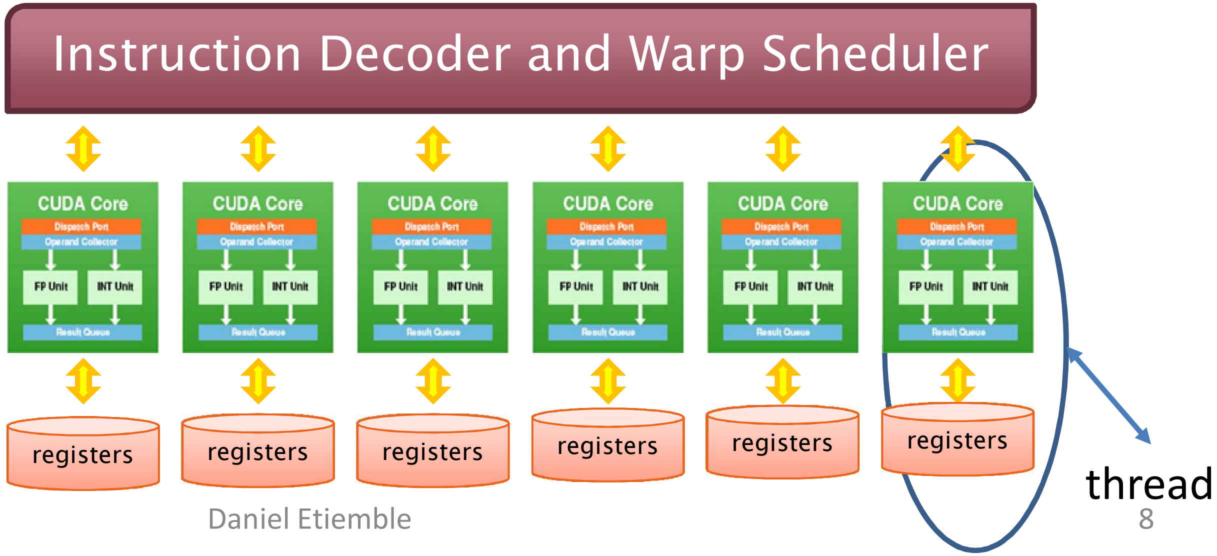
SSE2/3/4 – Neon – Altivec
AVX – AVX2 – AVX 512…



GPU

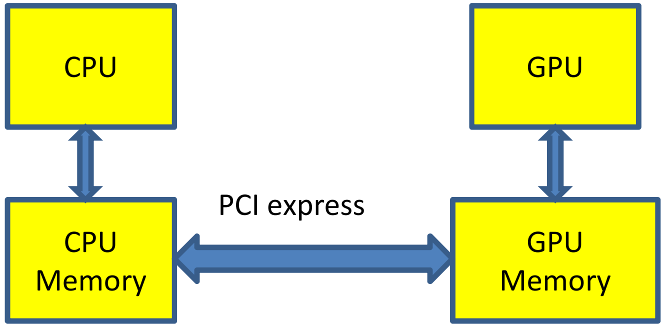
SIMT

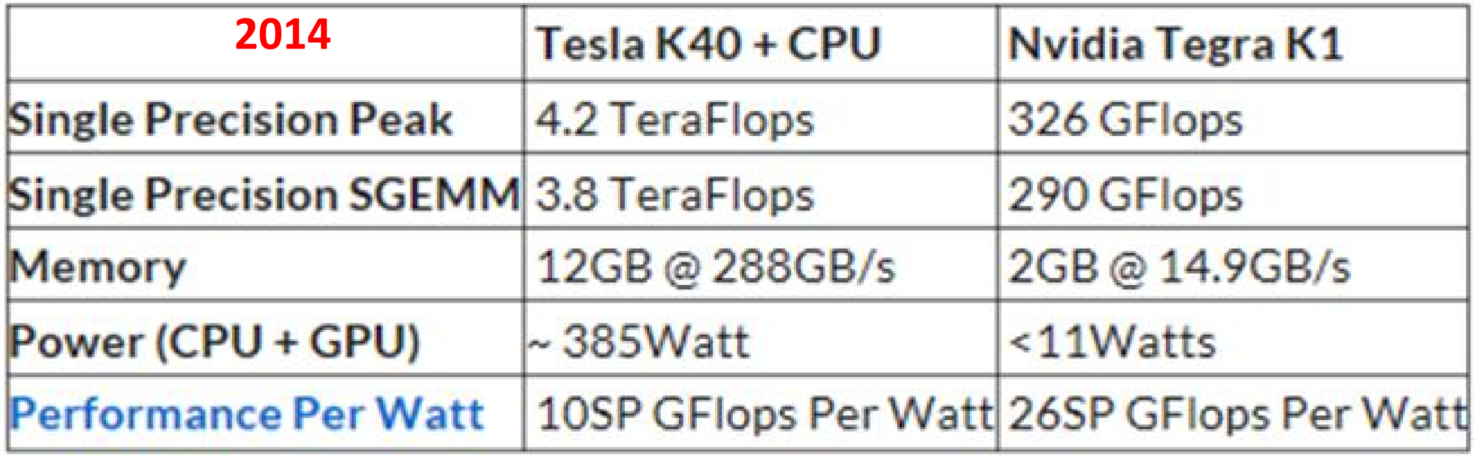
1 instruction for several threads



CPU + GPU

- Two different programming models
- Two chips or one chip?
  - CPU + GPU or APU





2014	Tesla K40 + CPU	Nvidia Tegra K1
Single Precision Peak	4.2 TeraFlops	326 GFlops
Single Precision SGEMM	3.8 TeraFlops	290 GFlops
Memory	12GB @ 288GB/s	2GB @ 14.9GB/s
Power (CPU + GPU)	~ 385Watt	<11Watts
Performance Per Watt	10SP GFlops Per Watt	26SP GFlops Per Watt

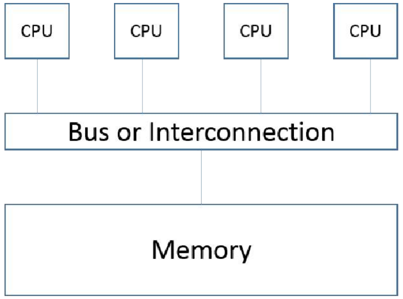
Improving Performance $T_{ex} = \frac{\boldsymbol{IC}}{IPC * F}$

Data or/and Thread Parallelism in Parallel Architectures

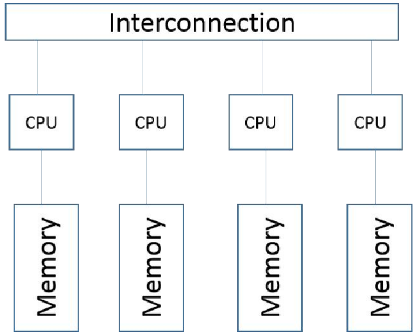
Dispatch IC among several processors (or cores)

- Except for simple cases or embarrasingly parallel applications, the dispatch depends on the architecture, the programming model, Amdahl law, etc.
- In some architectures, communication times must be included.

OpenMP
Pthreads



Multiprocessors



Multi-computers

MPI

Switching from sequential to parallel programming

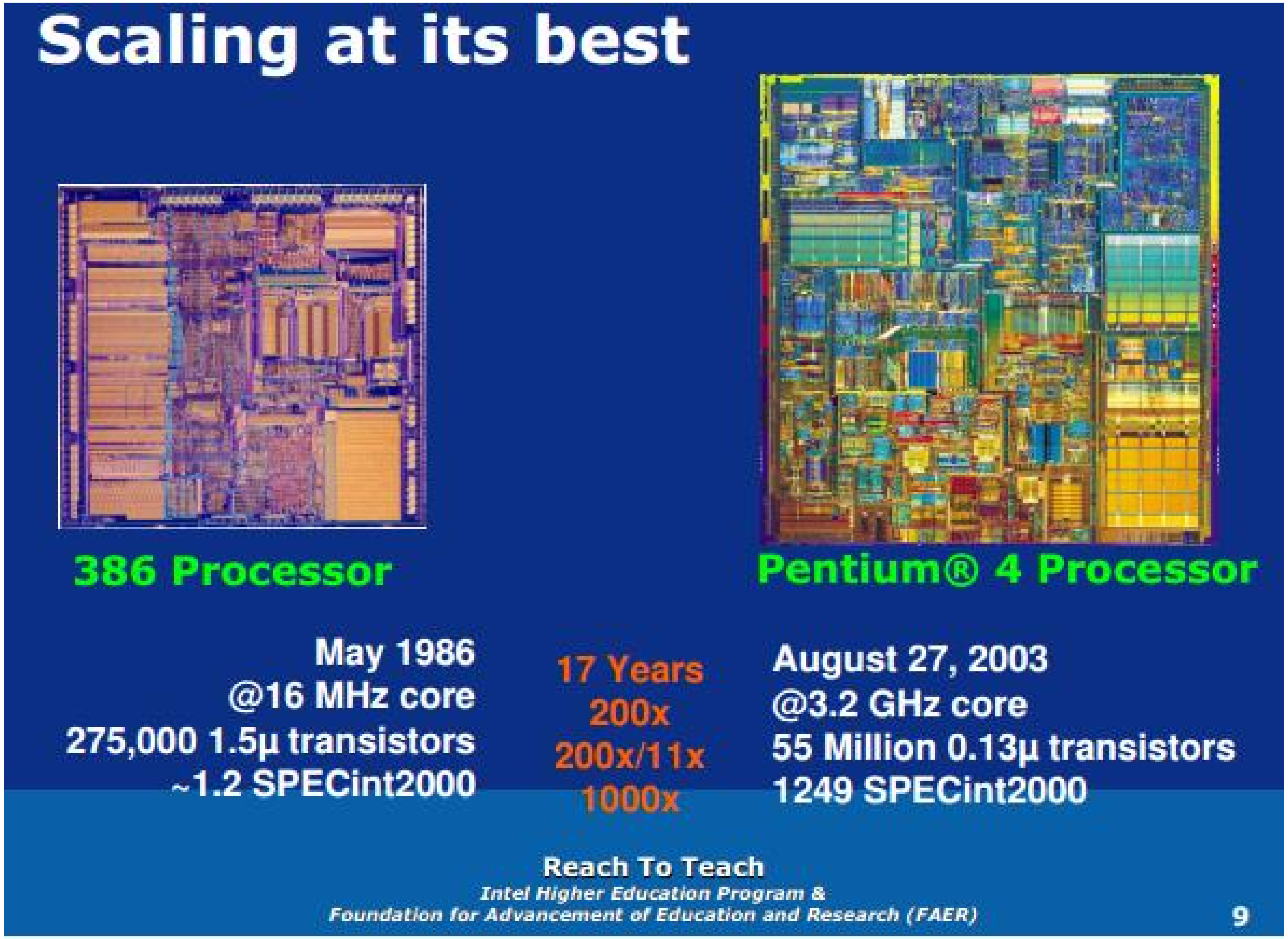
- Limited to servers and super-computers during the « free lunch » period

Free lunch… (by Intel)

F

IPC

IC (SIMD)



Scaling at its best

386 Processor

May 1986
@16 MHz core
275,000 1.5µ transistors
~1.2 SPECint2000

17 Years
200x
200x/11x
1000x

Pentium® 4 Processor

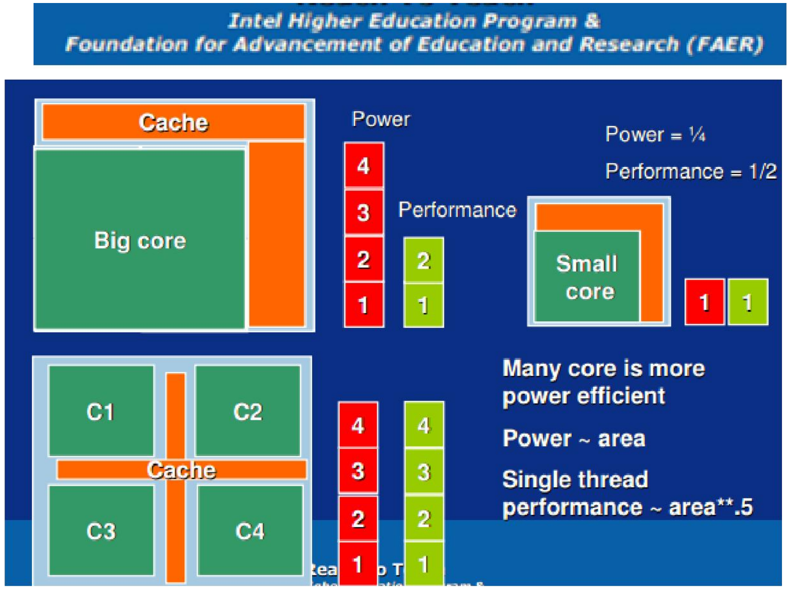
August 27, 2003
@3.2 GHz core
55 Million 0.13µ transistors
1249 SPECint2000

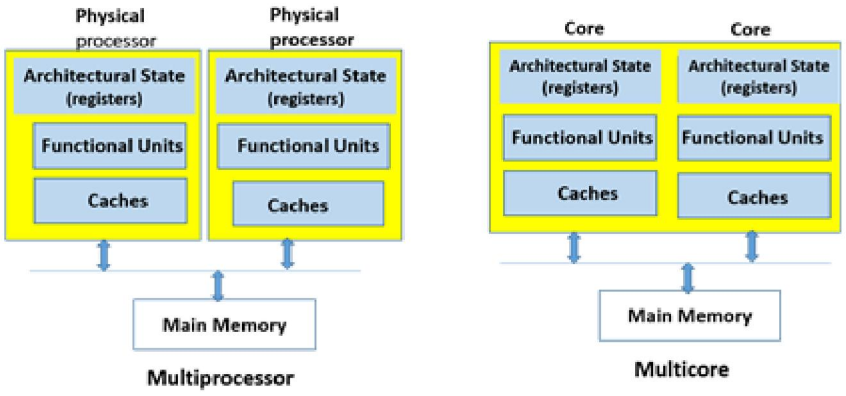
Reach To Teach
Intel Higher Education Program &
Foundation for Advancement of Education and Research (FAER)

9

From mono-processors to multi-cores

- Performance – Power trade-off
- Intrinsic limitations of mono-processors, even multithreaded
- Multi-processors to multi-cores
- Clusters of multi-cores.

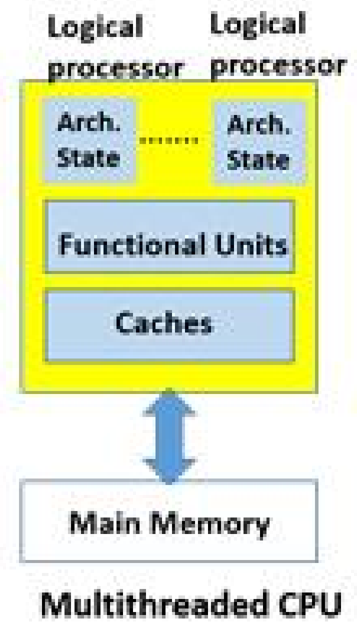

Intel Higher Education Program &
Foundation for Advancement of Education and Research (FAER)
Cache
Big core
Power
4
3
2
1
Performance
2
1
Power = ¼
Performance = 1/2
Small core
1
1
C1
C2
Cache
C3
C4
4
3
2
1
4
3
2
1
Many core is more power efficient
Power ~ area
Single thread performance ~ area**.5



Multithreaded CPUs

$$T_{ex} = \frac{IC}{\boldsymbol{IPC} * F}$$

- Sequential programs
- Several programs (multiprogramming): IC = $\sum IC_i$
- Several threads (TLP) : IC = $\sum IC_i$



Fine grain multithreading

- Switching in one clock from one thread to next one on pipeline hazards
- Sun Niagara, Oracle servers

Simultaneous multithreading

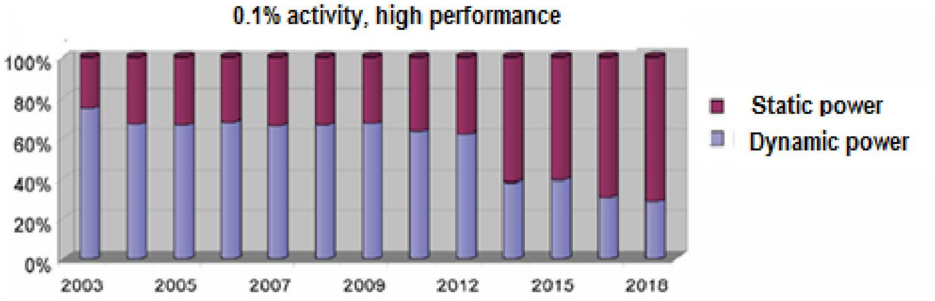
- Issuing instructions from different threads at each clock
- Intel Hyperthreading (2), IBM Power (2 to 8)

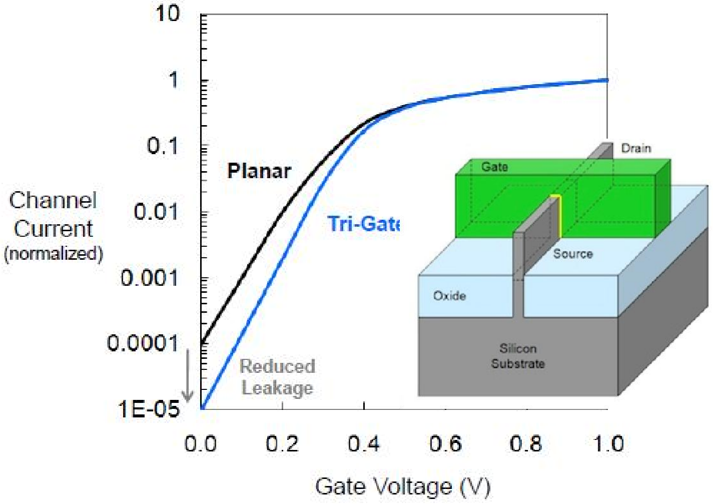
Multithreading reduces $\mathbf{CPI_{Mem}}$

→ **Multi-cores with multithreaded cores**

Reducing static power dissipation

$$P_{d\,static} = V_{dd} * I_{leakage}$$





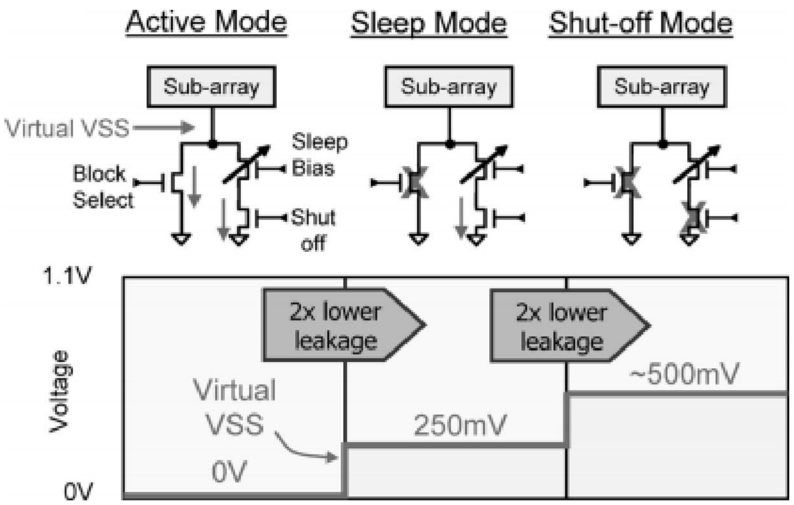
TECHNOLOGY

- Ex: Intel Tri-gate

CIRCUITERY

- Ex: Virtual Vss

65-nm Xeon CPU L3 cache

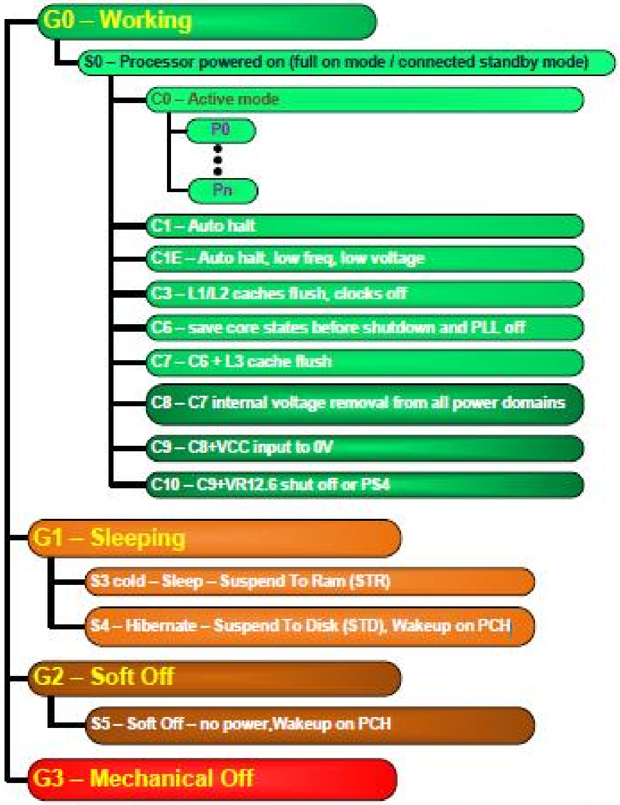


Reducing dynamic power dissipation

$$P_{d\,dynamic} = \alpha * \sum C_i * V_{dd}^2 * F$$

α

Several Operating Modes per blocks: Ex: 5th generation Intel Cores



G0 – Working

S0 – Processor powered on (full on mode / connected standby mode)

C0 – Active mode

P0

Pn

C1 – Auto halt

C1E – Auto halt, low freq, low voltage

C3 – L1/L2 caches flush, clocks off

C6 – save core states before shutdown and PLL off

C7 – C6 + L3 cache flush

C8 – C7 internal voltage removal from all power domains

C9 – C8+VCC input to 0V

C10 – C9+VR12.6 shut off or PS4

G1 – Sleeping

S3 cold – Sleep – Suspend To Ram (STR)

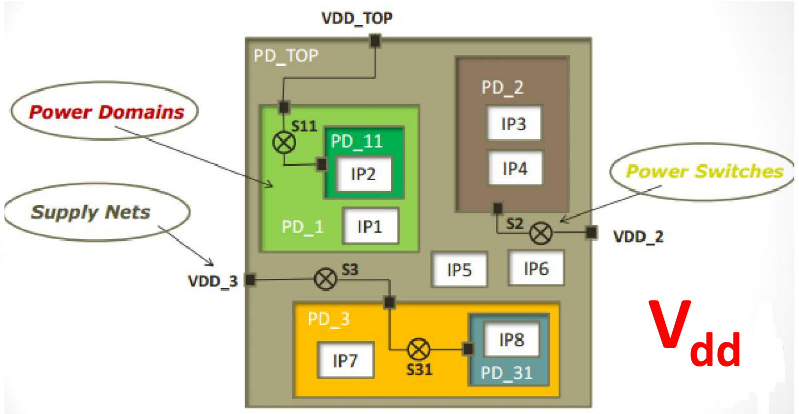
S4 – Hibernate – Suspend To Disk (STD), Wakeup on PCH

G2 – Soft Off

S5 – Soft Off – no power,Wakeup on PCH

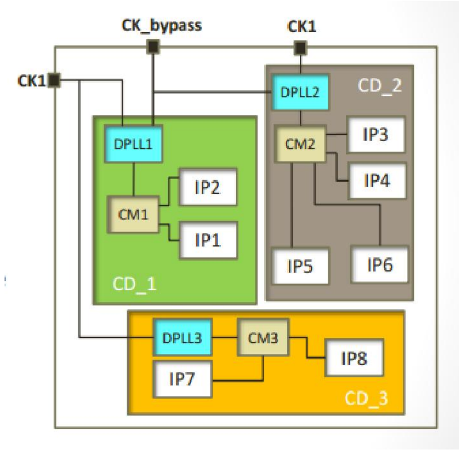
G3 – Mechanical Off

* Note: Power states availability may vary between the different SKUs



V_{dd}

Power domains



F

Clock domains

What about future?

- Moore’s law: towards fundamental limits
- Execution time:
  - F: significant changes are doubtful
  - IPC:
    - Limits of ILP in cores
    - New PIM architectures ? (Data access issues: “Memory Wall”)
  - IC continues to decrease
    - more job per instruction
      - SIMD width (256-512-1024…) and data size (16-bit FP)
      - New 2D instructions: Tensor cores (Nvidia), Matrix Multiplication Unit (Google TPU)…
    - More cores
      - From multi-cores to many-cores. Exponential increase of core number???
- Power dissipation
  - As long as CMOS will be used…

Concluding remarks

- **Main trends (**not details) of CPU evolution can be explained by
  - Moore's law
  - $T_{ex} = IC * (CPICPU + CPIMem) * T_c = \frac{IC}{IPC * F}$
  - $P_d = V_{dd} * I_{leakage} + \alpha * \sum C_i * V_{dd}^2 * F$
- Valid for software programmable processors as long as CMOS technology will be used.
- Mixed HW-SW architectures (FPGA) are more complex to modelize.
- Evolution of CPU architectures is driven by new specific applications (AI, IoT, …).
- Business… (as usual). Ex: "proprietary" versus "open-source" ISAs.