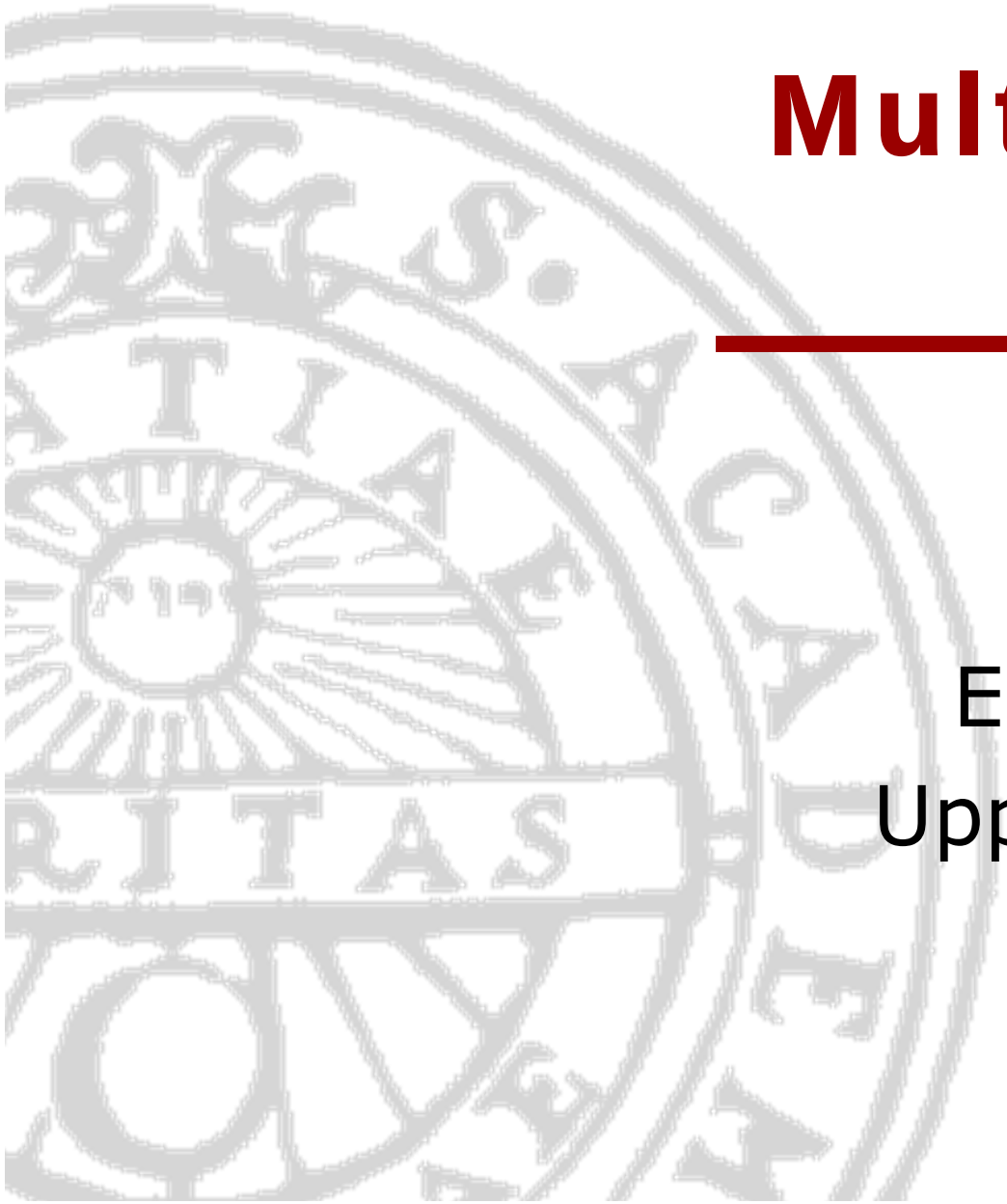
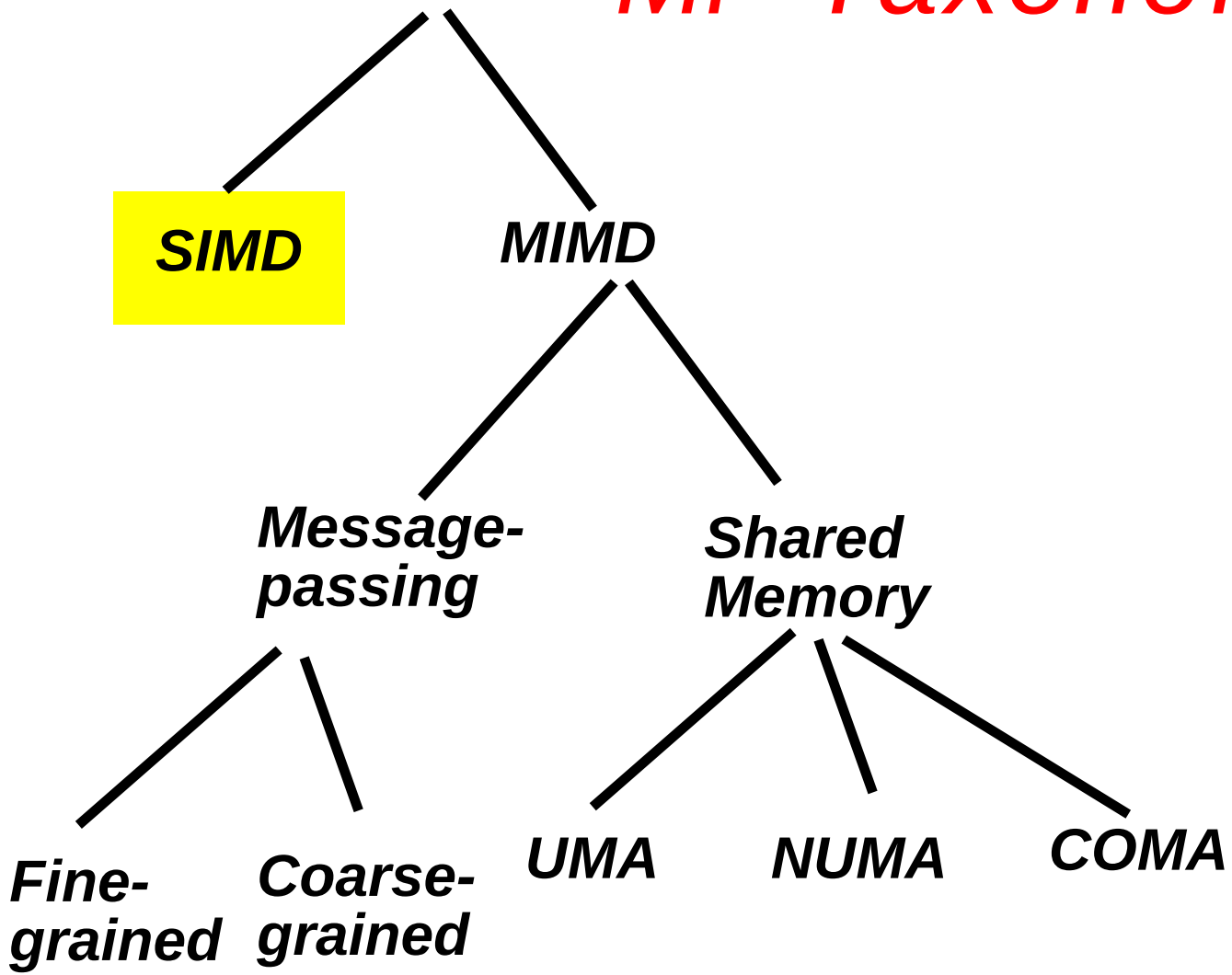


Multiprocessor Options

Erik Hagersten
Uppsala University

MP Taxonomy


SIMD
MIMD
Message-passing
Shared Memory
Fine-grained
Coarse-grained
UMA
NUMA
COMA

UPPSALA UNIVERSITET

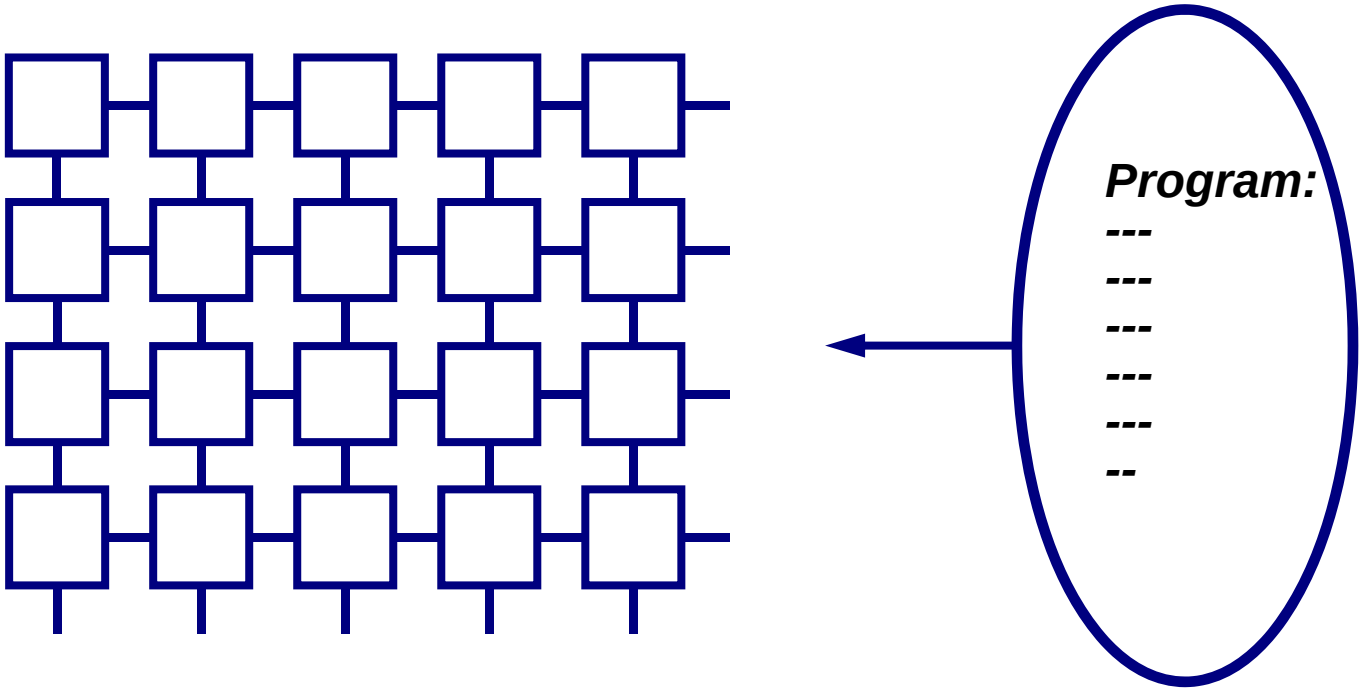
Flynn's Taxonomy

{Single,Multiple}Instruction +
{Single,Multiple}Data

- SISD - Our good old "simple" CPUs
- SIMD – Vectors, "MMX", DSPs, CM-2,…
- MIMD – TLP, cluster, shared-mem MP,…
- MISD – Can't think of any…

UPPSALA UNIVERSITET

SIMD = "Dataparallelism"



UPPSALA UNIVERSITET

SIMD: Thinking Machine

- Connection Machine: CM1, CM2, CM200
  (at KTH ~1990: CM200 "Bellman")
- One-bit ALU and a small local memory
- FP accelerator available
- Programmed in "ASM", *C and *Lisp
- Hard to program (in my opinion…)

UPPSALA UNIVERSITET

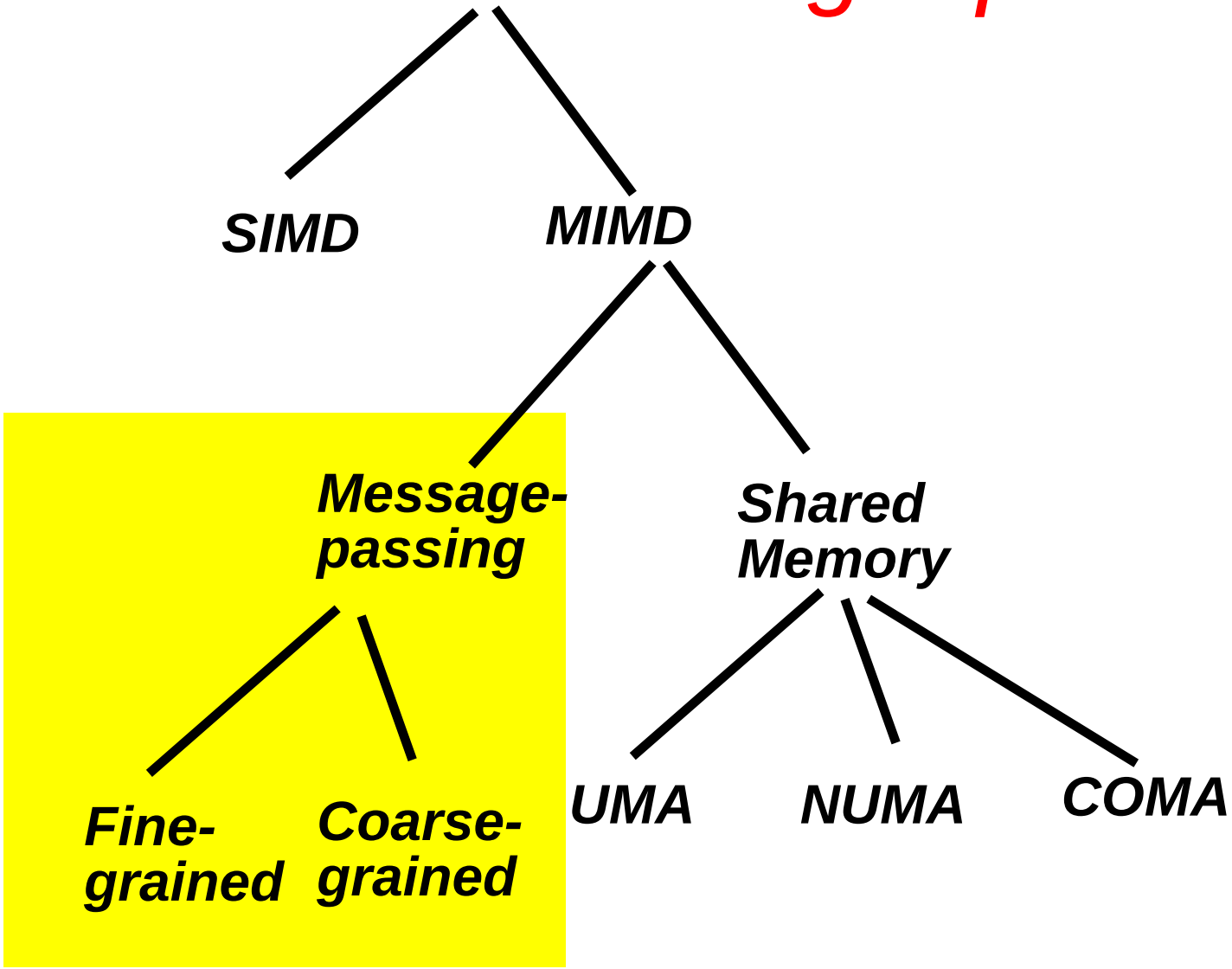
Vector architectures

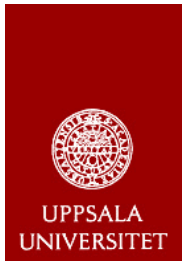
CRAY, NEC, Fujitsu,

- Vectory Processors
  - LD/ST operate on vectors of data
  - ALU Ops operate on vectors of data

- Example:
  - 8 “vector register” contain 64 vector “words” each
  - A single LD/ST instr loads/stores entire vectors
  - A single ALU instr V1 ← V2 op V3
  - Overlaps Mem and ALU ops
  - Special scatter/gather instructions
  - 64 bit mask vectors make execution conditional

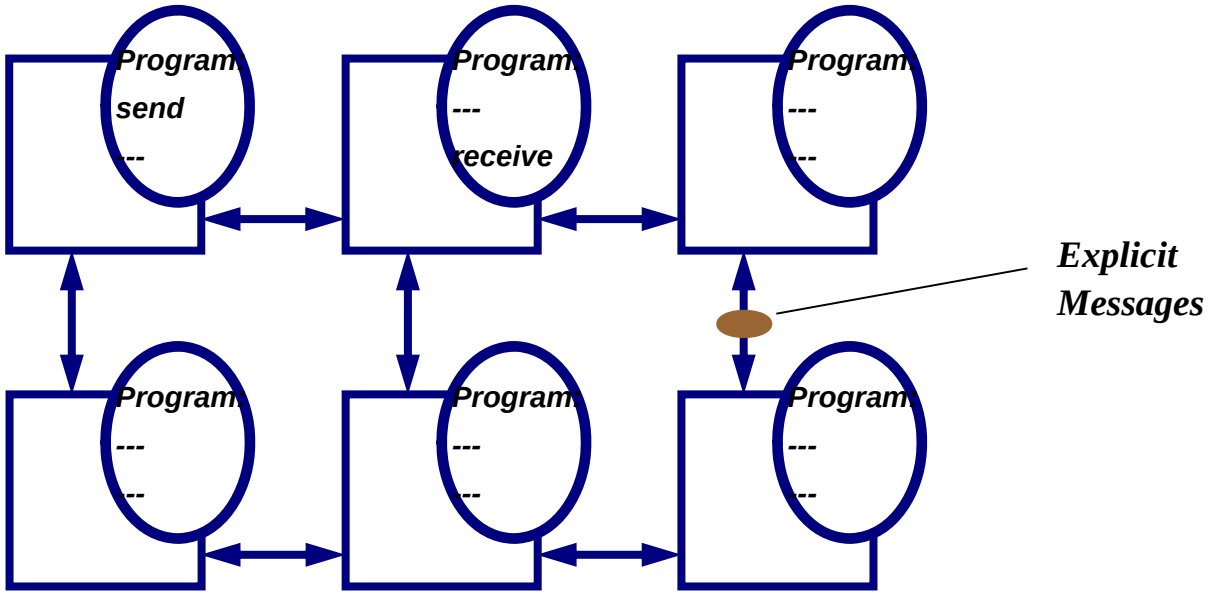
MIMD:Message-passing



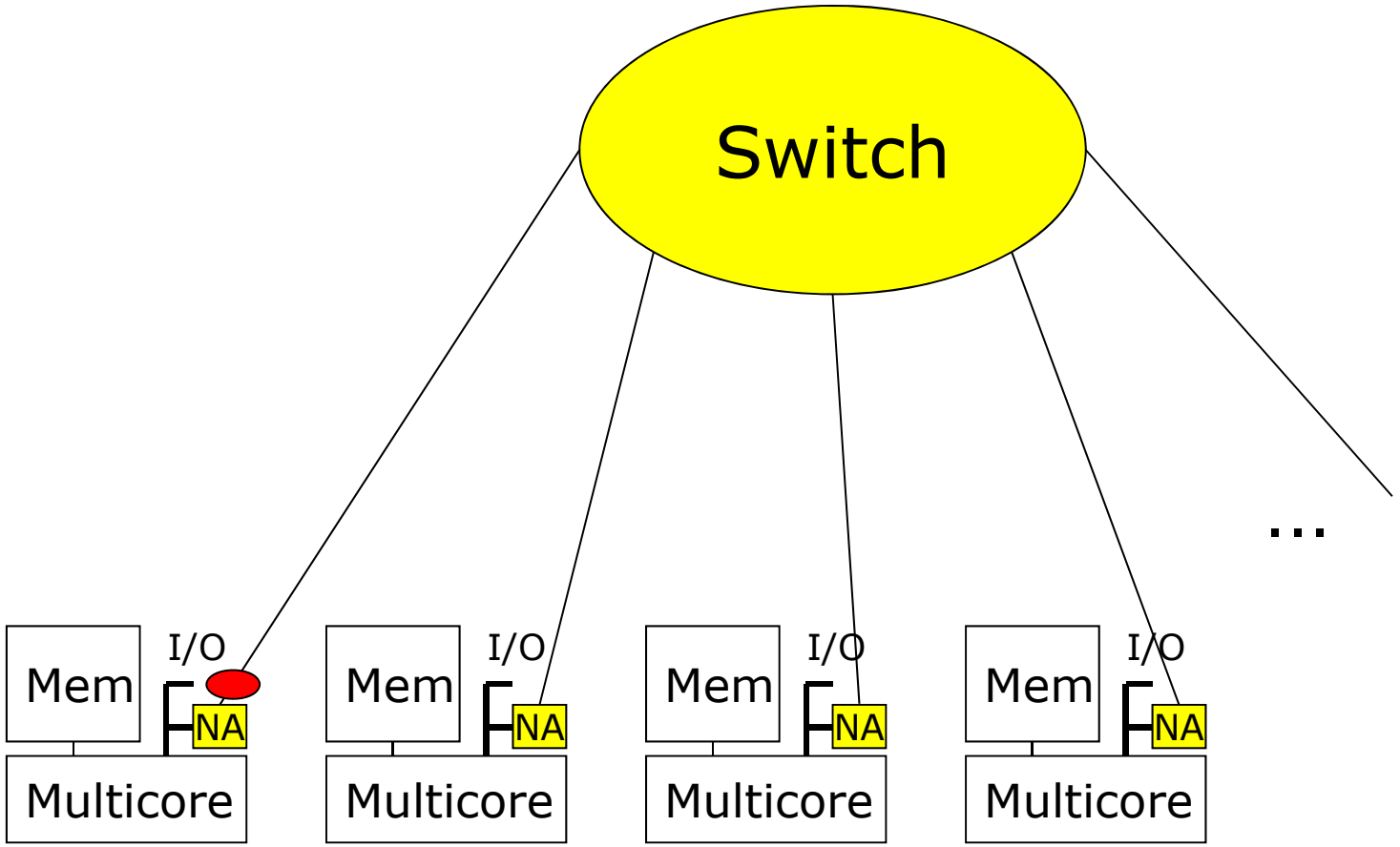


UPPSALA
UNIVERSITET

Message-passing Arch (MIMD)



A modern ”supercomputer”




```
X = vec[i];
MPI_send(X, to_dest);
...
```

```
...
MPI_receive(Y, from_source;
print (Y);
```

UPPSALA UNIVERSITET

Message-Passing HW

- Programmed in MPI or PVM (or HPFortran…)
  - Thinking Machines: CM5
  - Intel: Paragon
  - IBM: SP2
  - Meiko (Bristol, UK!!): CS2
  - Today: Clusters with high-speed interconnect (Important today, but not covered in this course)

- Clusters can be used as message-passing HW. Often used as capacity computing (i.e., throughput computing)

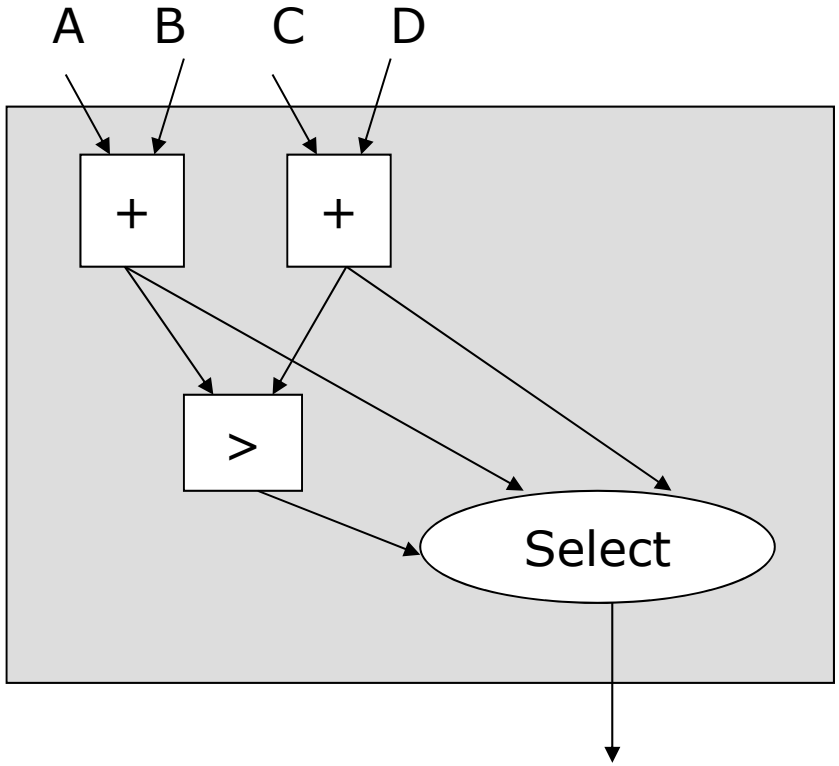
UPPSALA UNIVERSITET

Dataflow

- Often programmed in functional languages (e.g., ID)
- Compile program to Dataflow graph
- Operands + graph = executable
- Operation ready when the source operands are available

Dataflow Example:

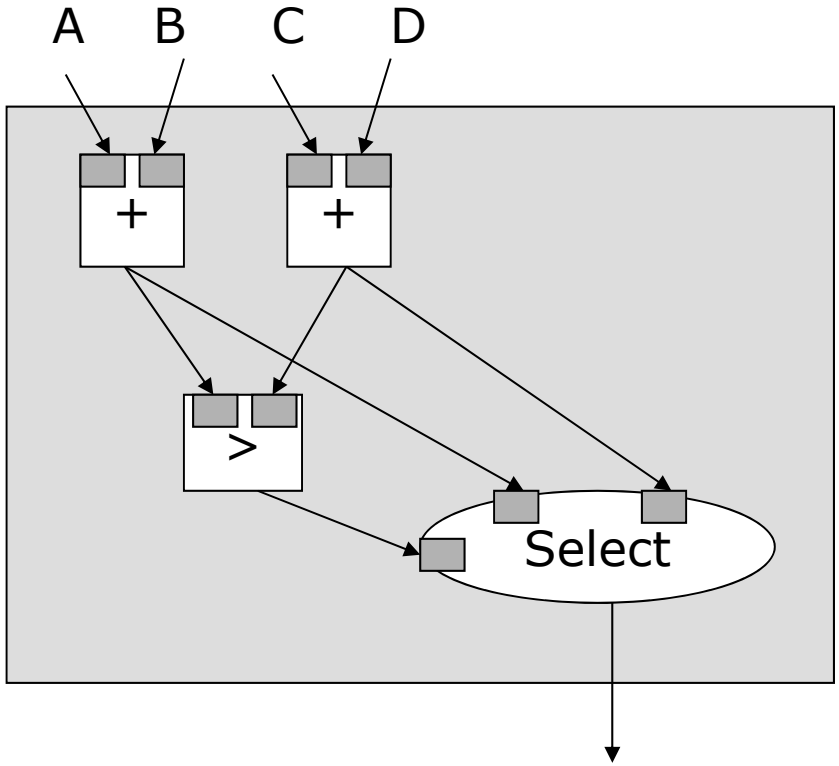
```
X := A + B
Y := C + D
If (X > Y)
  output X
else
  output Y
```



UPPSALA UNIVERSITET

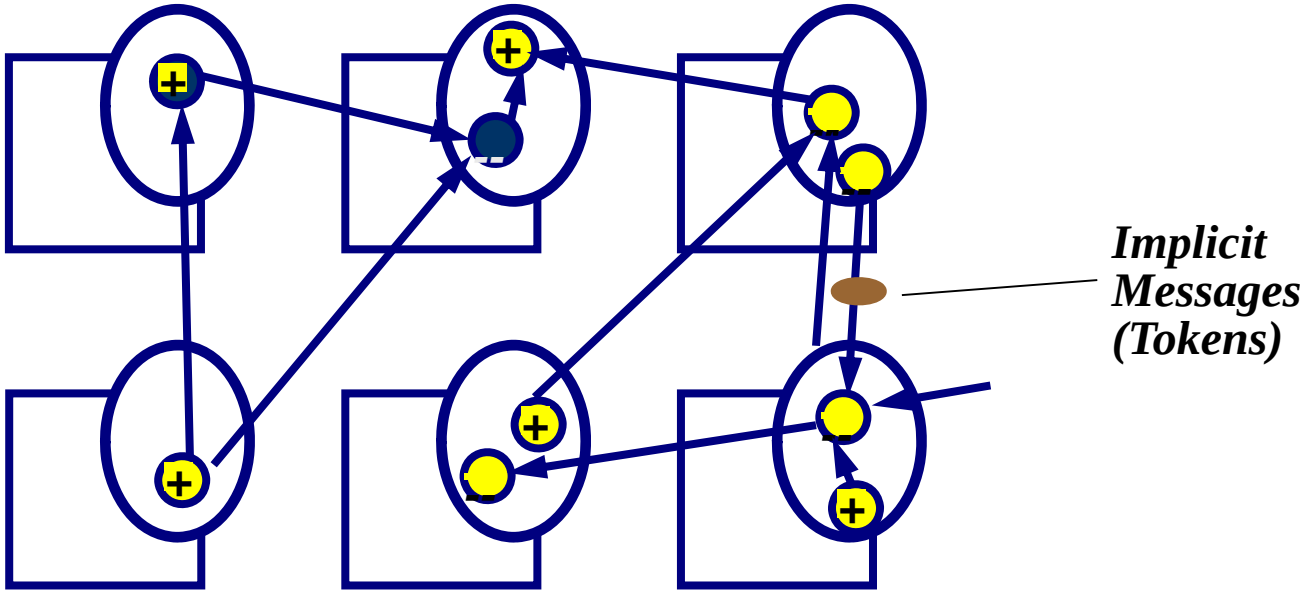
Static Dataflow (Dennis)

```
X := A + B
Y := C + D
If (X > Y)
  output X
else
  output Y
```



Each operand executed exactly once per program
Location assigned for each input data

Fine-grained Message-passing Dataflow ==> Multithreading



UPPSALA UNIVERSITET

Dynamic Dataflow (Arvind)

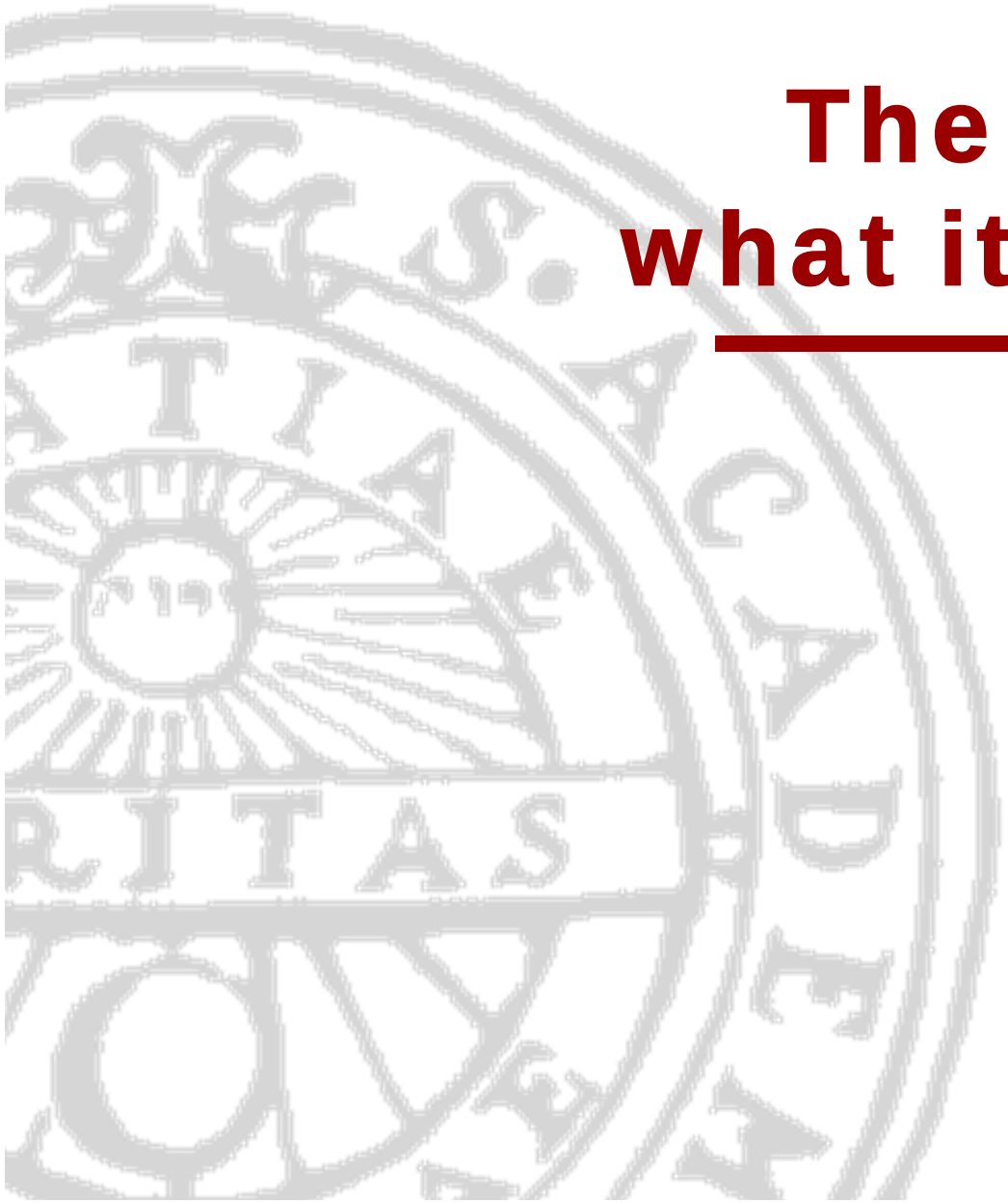
- Allows for recursion and loops
- Each invocation is assigned a "color"
- Pairs of operands are matched dynamically
  - Based on {Color, Operation}
  - In the Waiting-Matching Section (I.e., a cache)
- One problem: too much parallelism in the wrong place

UPPSALA UNIVERSITET

Carlstedts Elektornik

Gunnar Carlstedt, Staffan Truve' et al

- Processor "8601"
  - Gothenburg 1990-1997
  - Functional language "H"
  - Execution performed by a reduction a CAM memory
  - ALU rarely used
  - Many parallel processors on a wafer
    (Wafer-scale integration)
  - ➔ CRT (Carlstedt Research Technology)

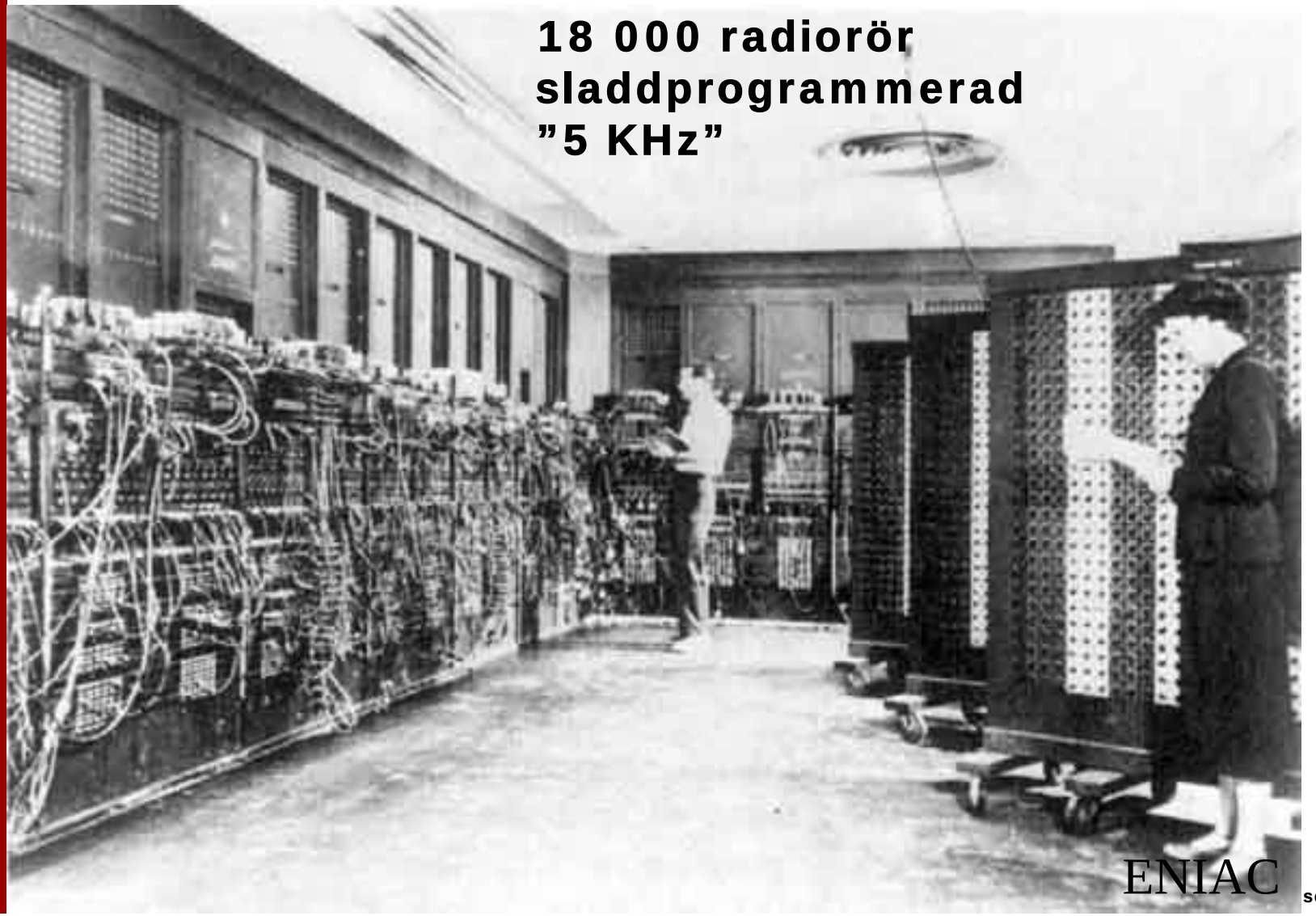


The Future is not what it used to be...

Erik Hagersten

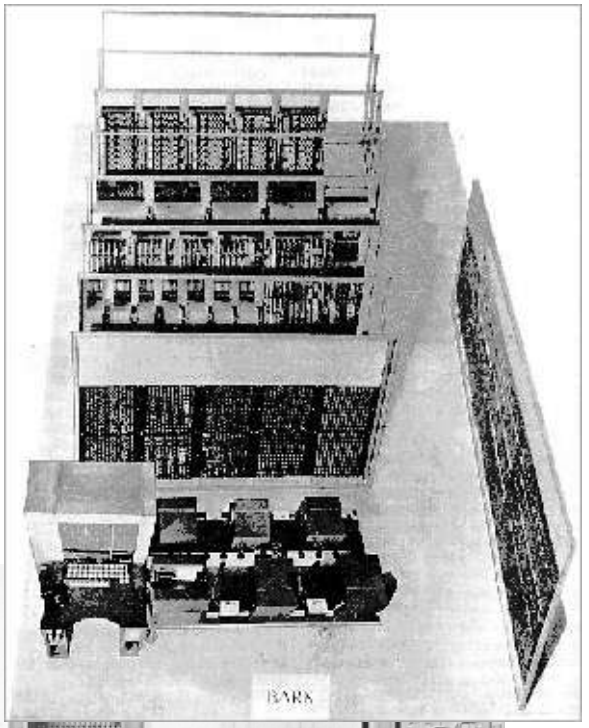
UPPSALA UNIVERSITET

Then…
ENIAC 1946 (”5kHz”)



ENIAC

Then (in Sweden)

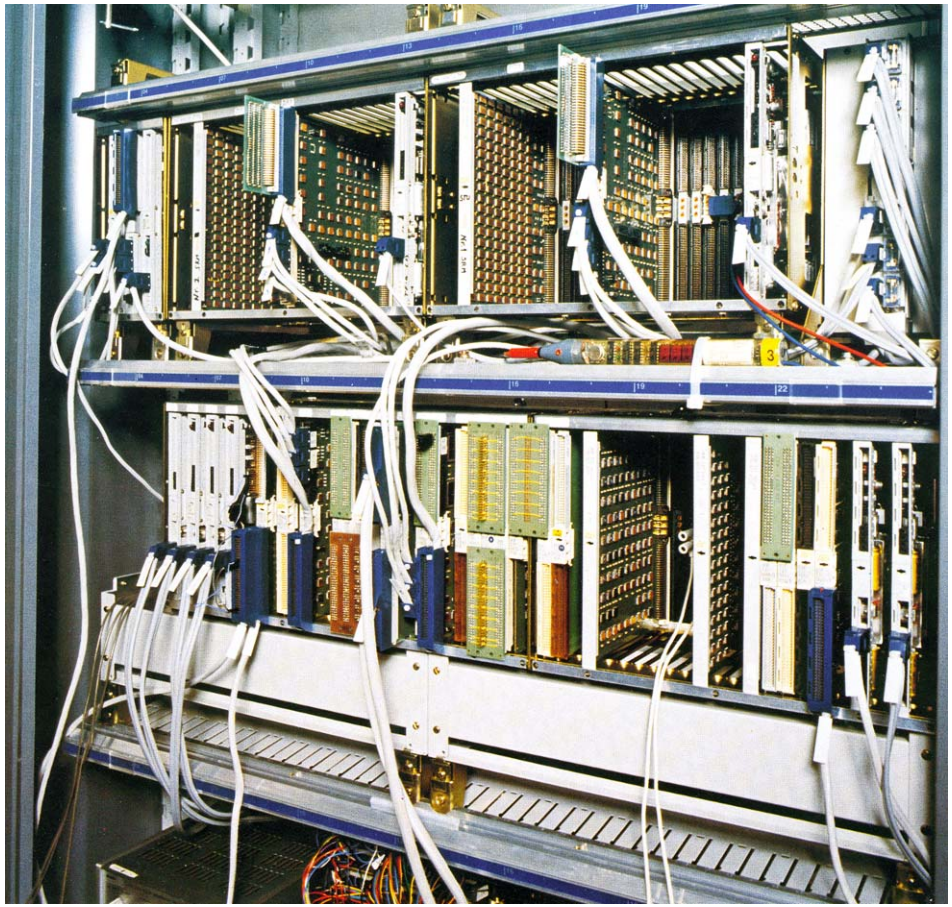


- BARK (~1950)
  - 8 000 relays,
  - 80 km cables
- BESK (~1953)
  - 2 400 vac. tubes
  - ”20 kHz” (world record)

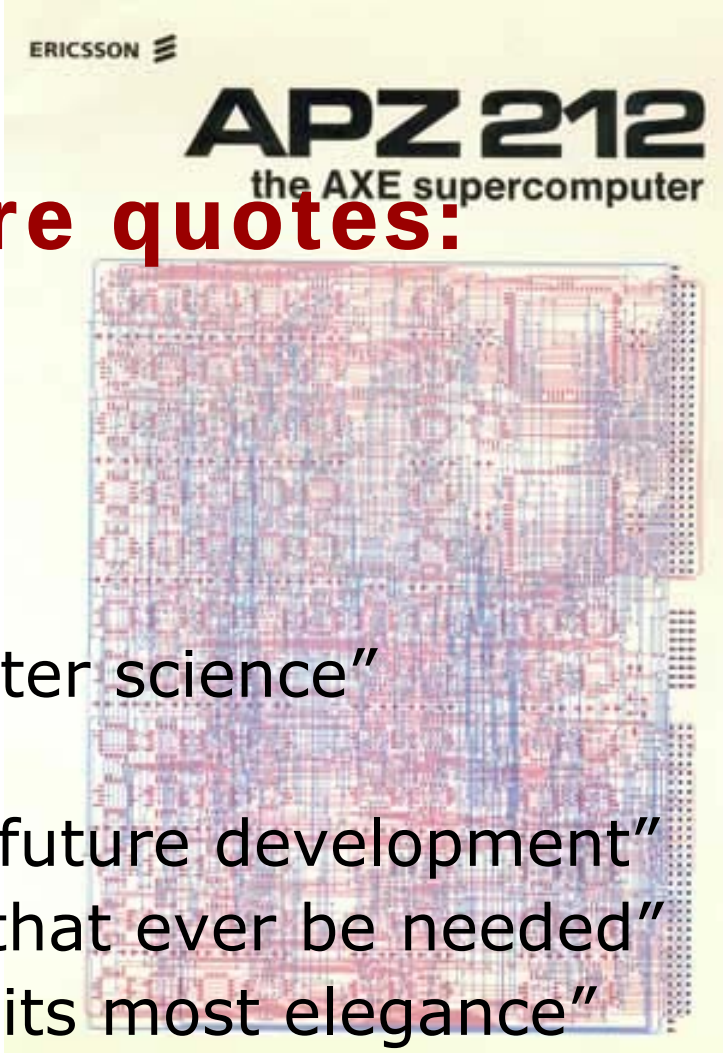
UPPSALA
UNIVERSITET

“Recently” APZ 212, 1983

Ericsson’s Supercomputer (“5 MHz”)



APZ 212 marketing brochure quotes:



- ”Very compact”
  - 6 times the performance
  - 1/6:th the size
  - 1/5 the power consumption
- ”A breakthrough in computer science”
- ”Why more CPU power?”
- ”All the power needed for future development”
- ”…800,000 BHCA, should that ever be needed”
- ”SPC computer science at its most elegance”
- ”Using 64 kbit memory chips”
- ”1500W power consumption

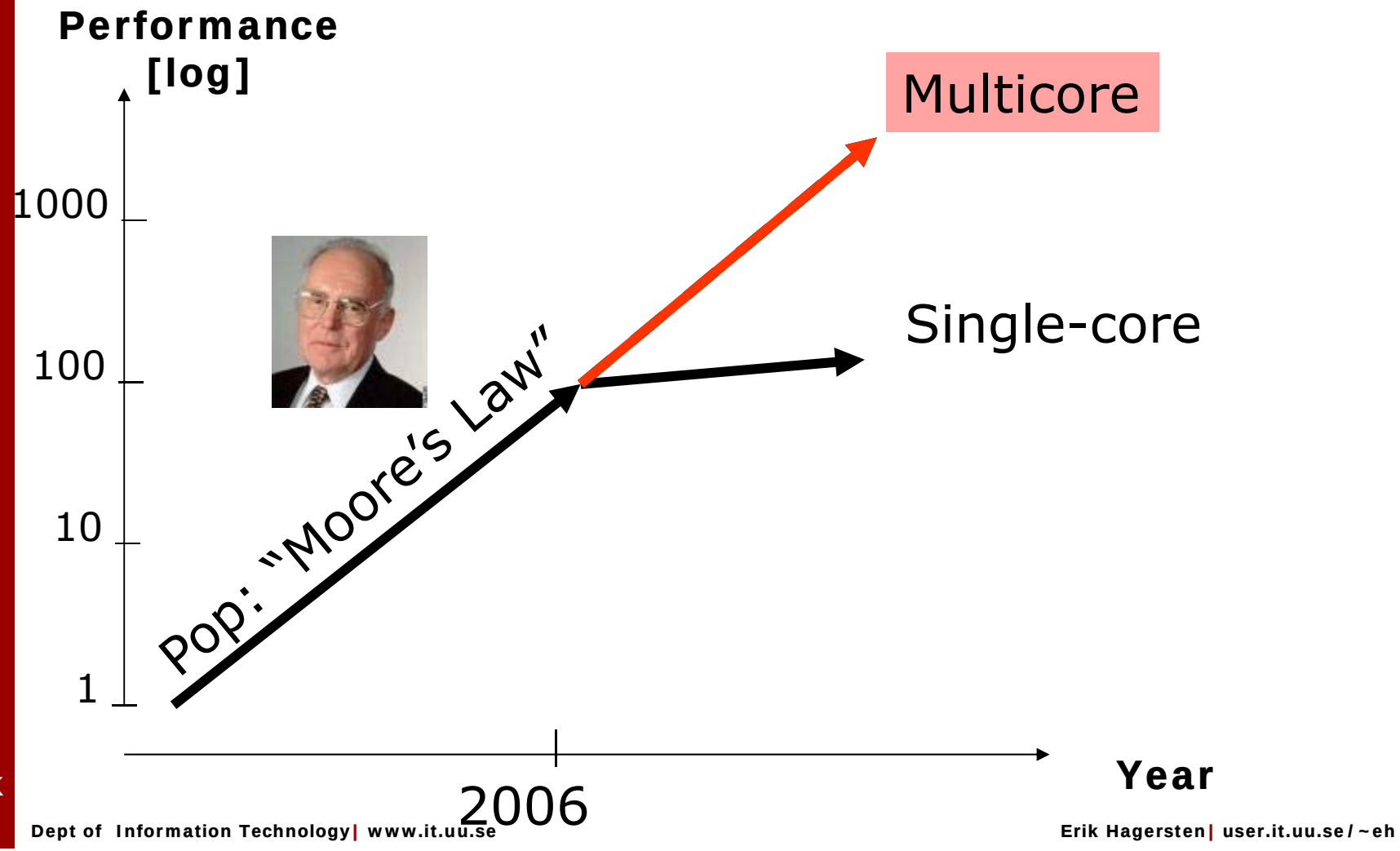
UPPSALA UNIVERSITET

65 years of "improvements"

- Speed
- Size
- Price
- Price/performance
- Reliability
- Predictability
- Energy
- Safety
- Usability….

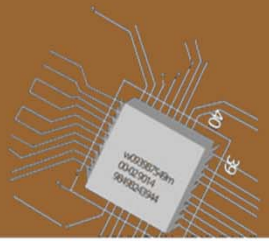
”Moore’s Law”

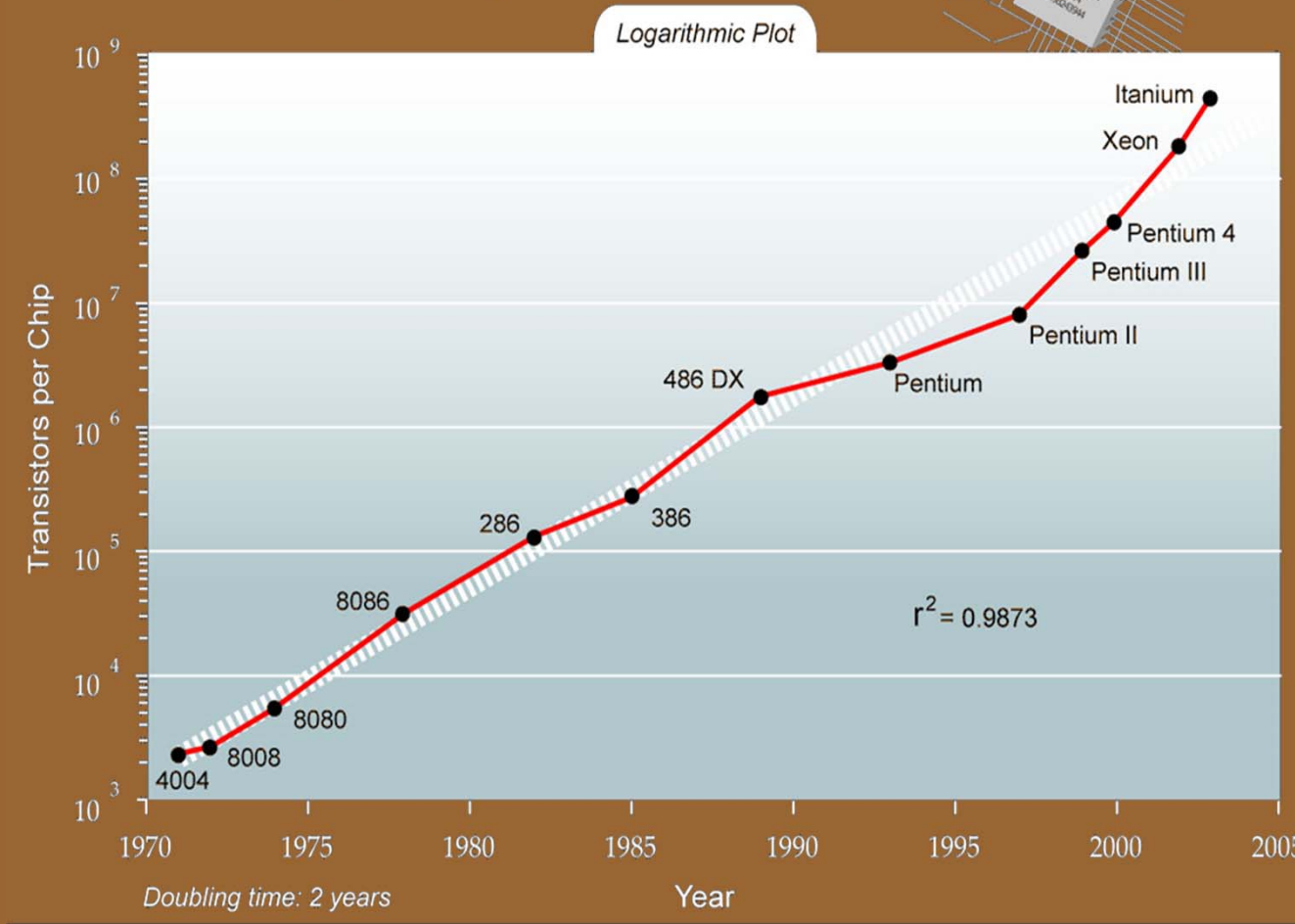
Pop: Double performance every 18-24th month



Ray Kurzweil pictures
www.KurzweilAI.net/pps/WorldHealthCongress/

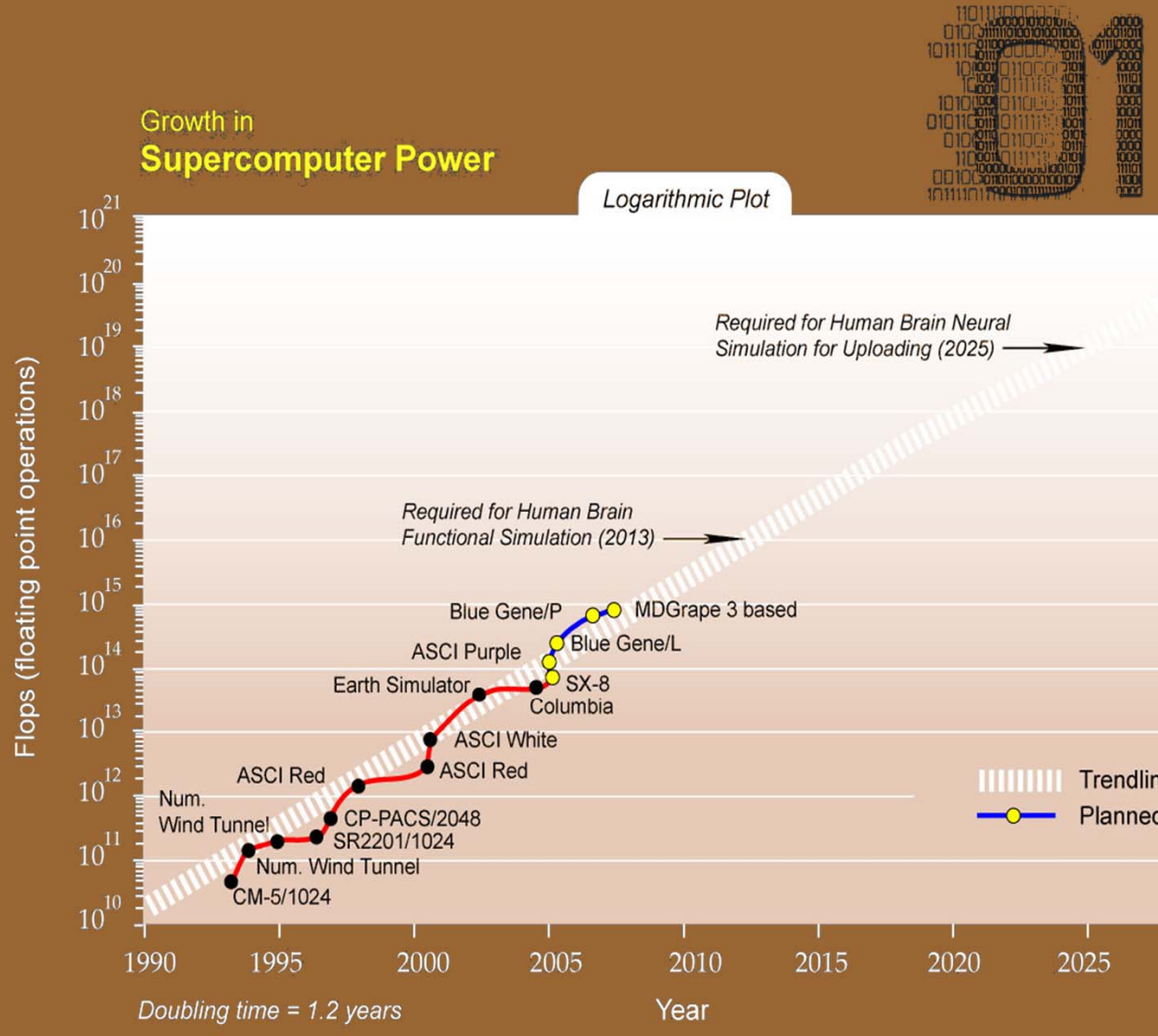
Ray Kurzweil pictures
www.KurzweilAI.net/pps/WorldHealthCongress/

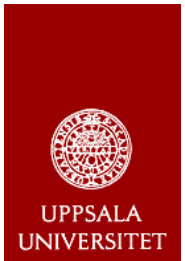




Ray Kurzweil pictures

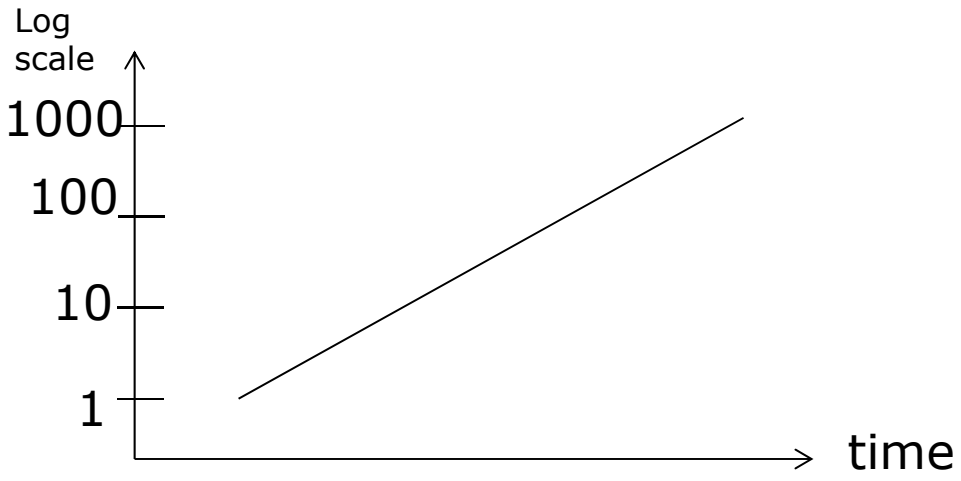
www.KurzweilAI.net/pps/WorldHealthCongress/





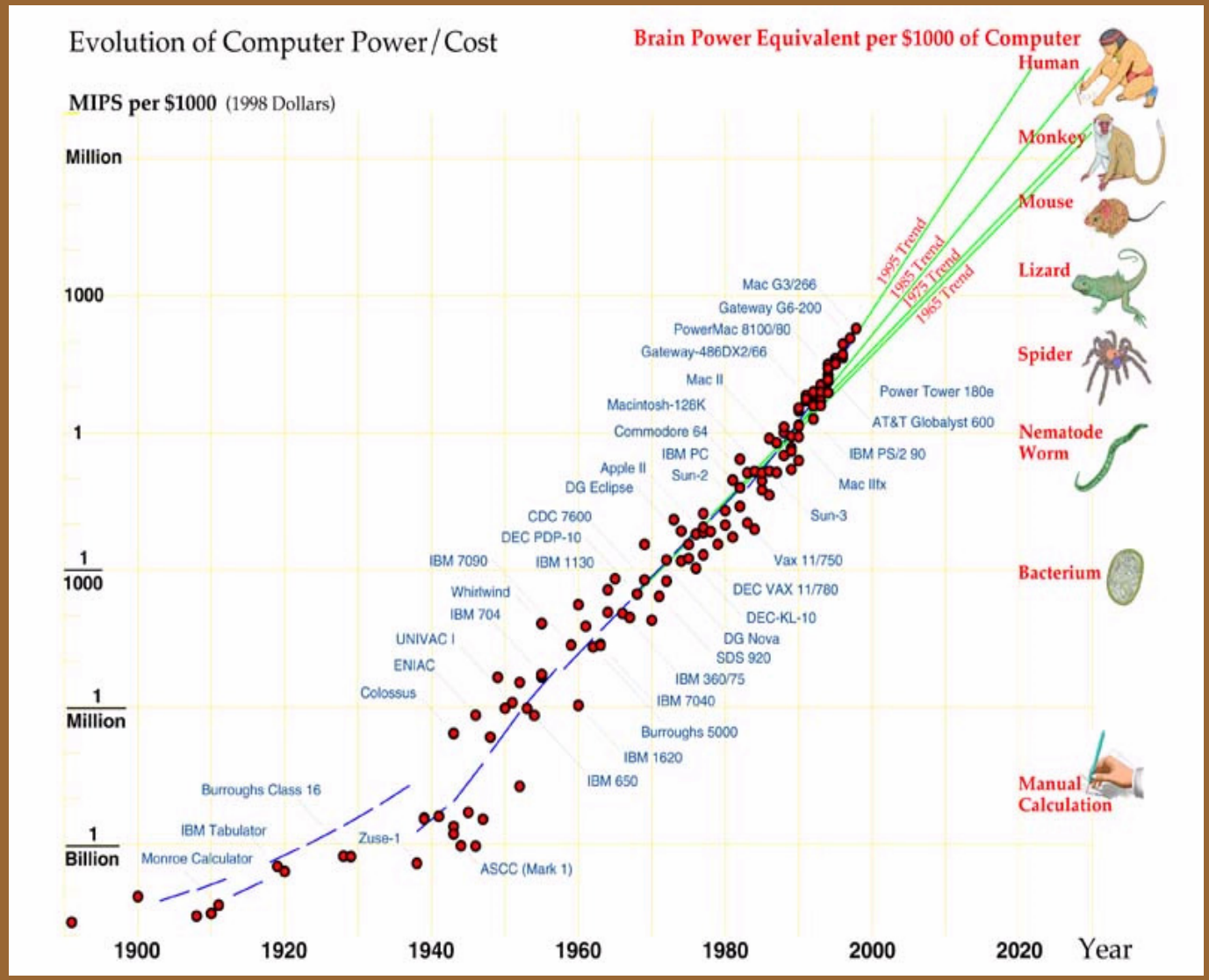
Exponentiell utveckling: Doublerings/halverings-tider (according to Kurzweil)

- **Dynamic RAM Memory (bits per dollar)** **1.5 years**
- **Average Transistor Price** **1.6 years**
- **Microprocessor Cost per Transistor Cycle** **1.1 years**
- **Total Bits Shipped** **1.1 years**
- **Processor Performance in MIPS** **1.8 years**
- **Transistors in Intel Microprocessors** **2.0 years**



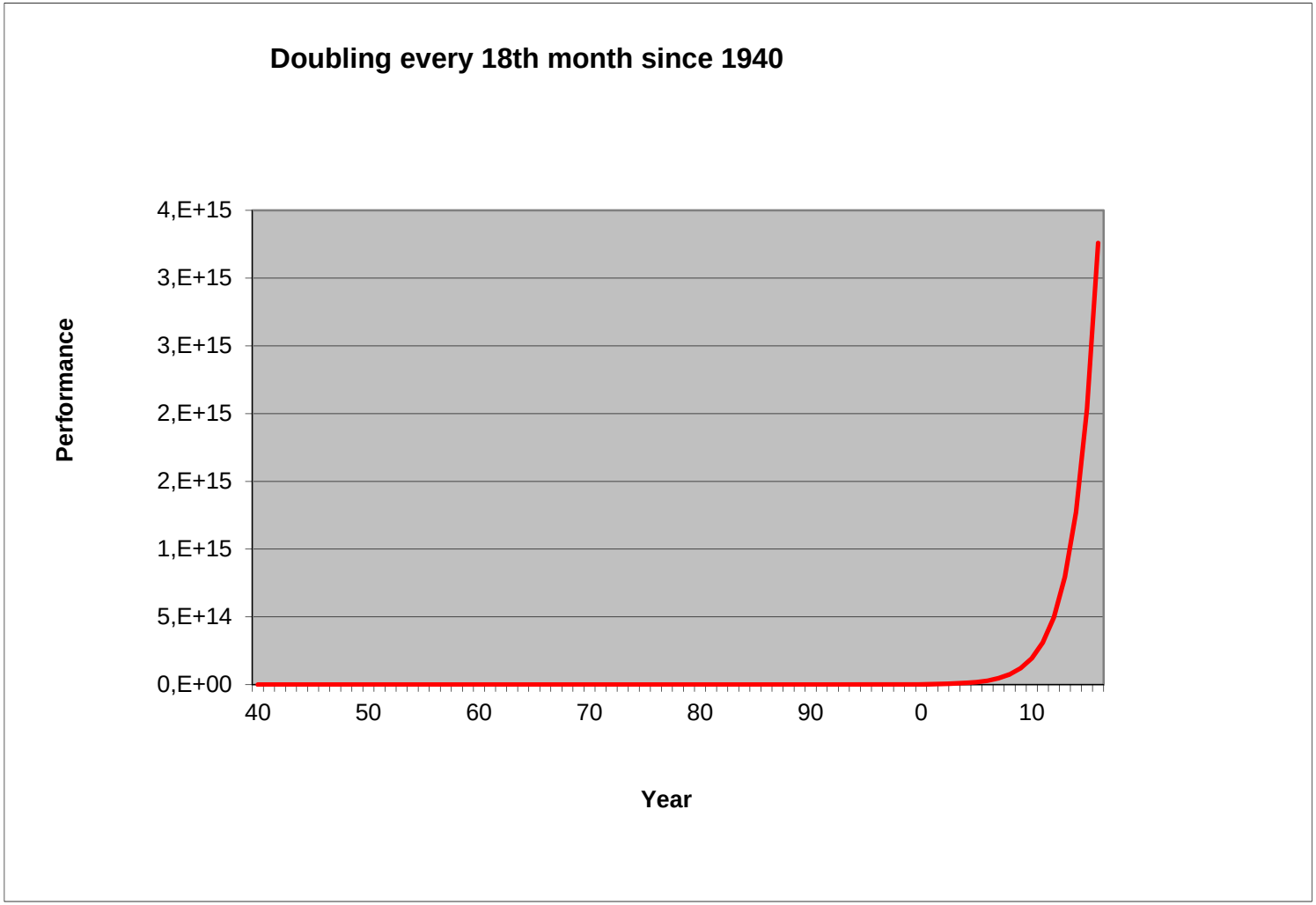
UPPSALA UNIVERSITET

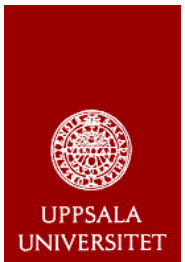
Ray Kurzweil pictures
www.KurzweilAI.net/pps/WorldHealthCongress/



AVDARK 2013

Linear scale 1940 ➔ 2017 (2x performance every 18th month)





Exponential growth

Example: Doubling every 2nd year

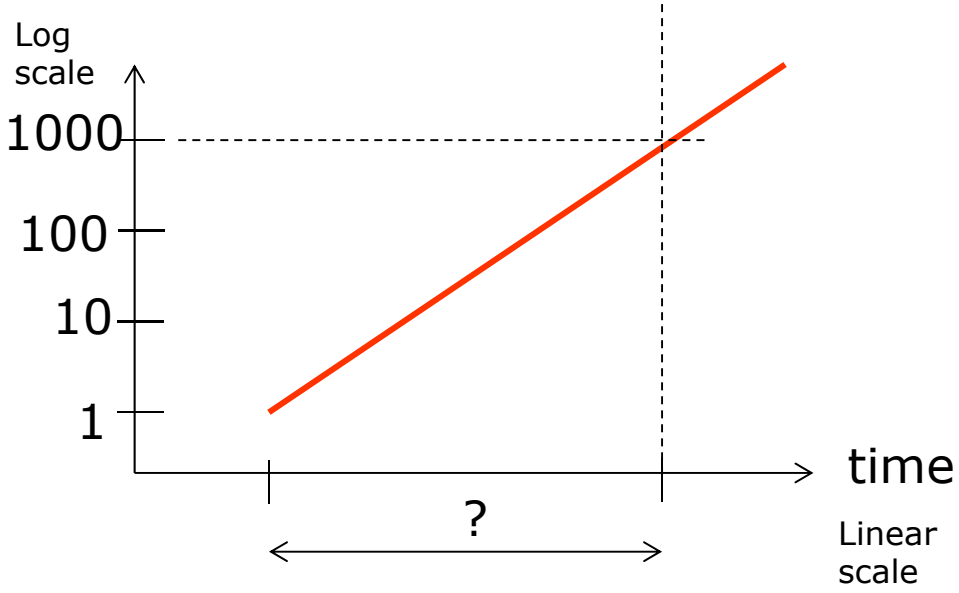
Q: How long does it take for 1000x improvement?

A: 20y

Example: Doubling every 18th month

Q: How long does it take for 1000x improvement?

A: 15y



UPPSALA UNIVERSITET

Looking Forward

Three rules of wisdom:

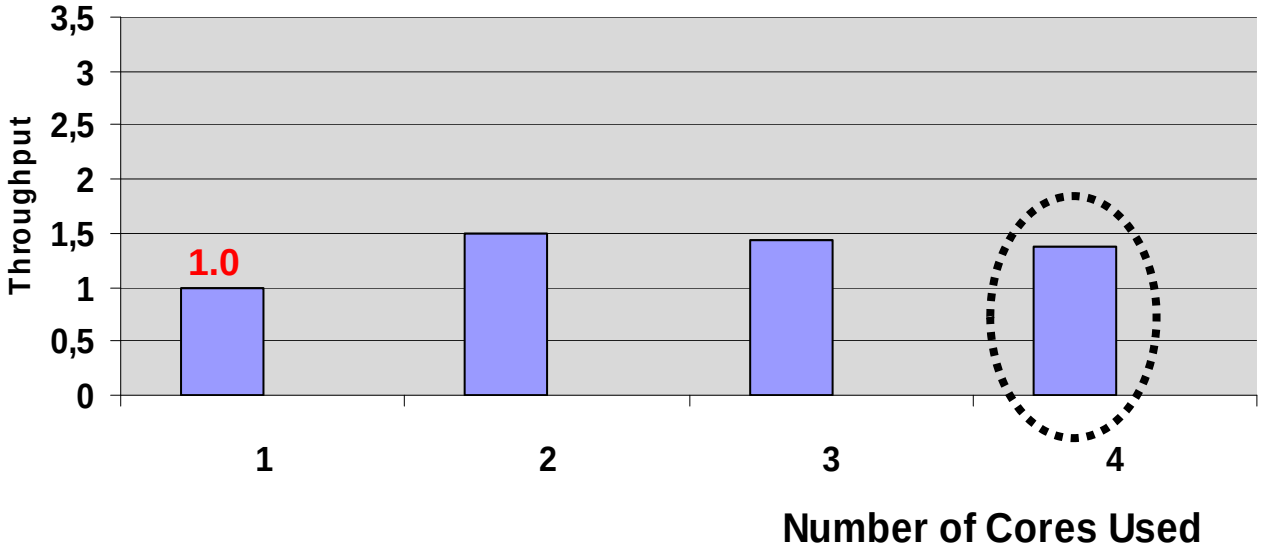
- Do not bet against exponential trends
- Do not bet against exponential trends
- Do not bet against exponential trends

But, is it possible to continue ”Moore’s Law”?

- Are there show-stoppers?
- Can we utilize an exponential growth of #cores?

Not everything scales as fast!

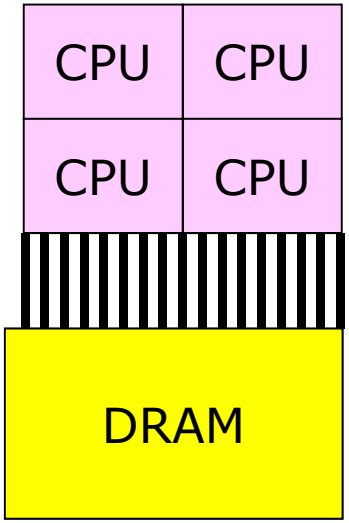
Example: 470.LBM
"Lattice Boltzmann Method" to simulate incompressible fluids in 3D



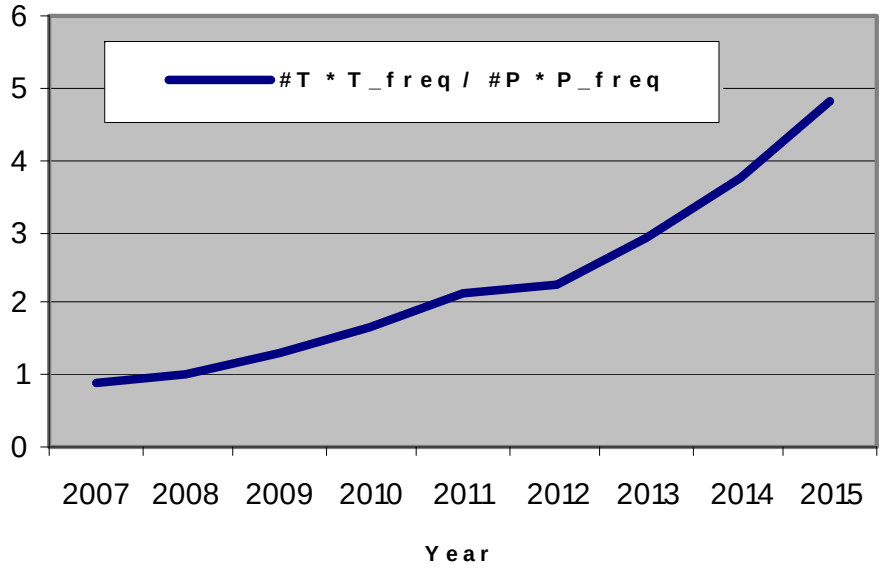
Throughput (as defined by SPEC):
Amount of work performed per time unit when several instances of the application is executed simultaneously.
Our TP study: compare TP improvement when you go from 1 core to 4 cores

Remember: It is getting worse!

#Cores ~ #Transistors



Computation vs Bandwidth



Source: Internatronal Technology Roadmap for Semiconductors (ITRS)

From Karlsson and Hagersten. *Conserving Memory Bandwidth in Chip Multiprocessors with Runahead Execution*. IPDPS March 2007. [graph updated with more recent data]

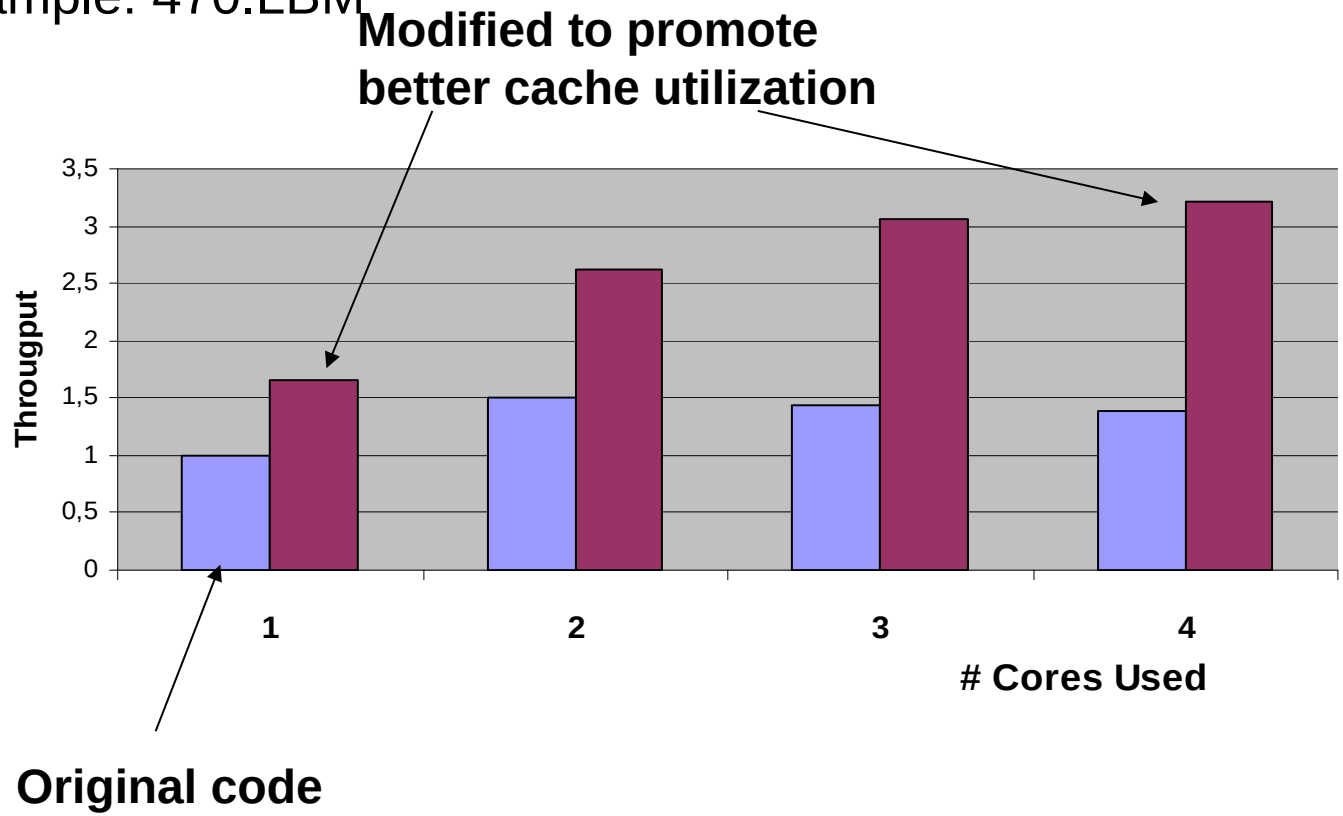
HPCwire Feb 2011 [cites Linley Gwennap and Justin Rattner]
Without Silicon Photonics, Moore's Law Won't Matter

HPCwire Feb 2011
Growing Data Deluge Prompts Processor Redesign

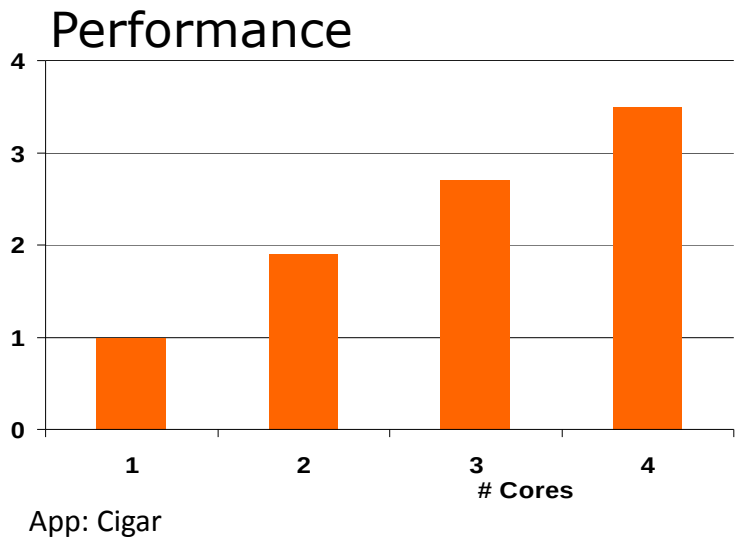
Case study: Limited by bandwidth

➔ Better Memory Usage!

Example: 470.LBM



Example 2: A Scalable Parallel Application

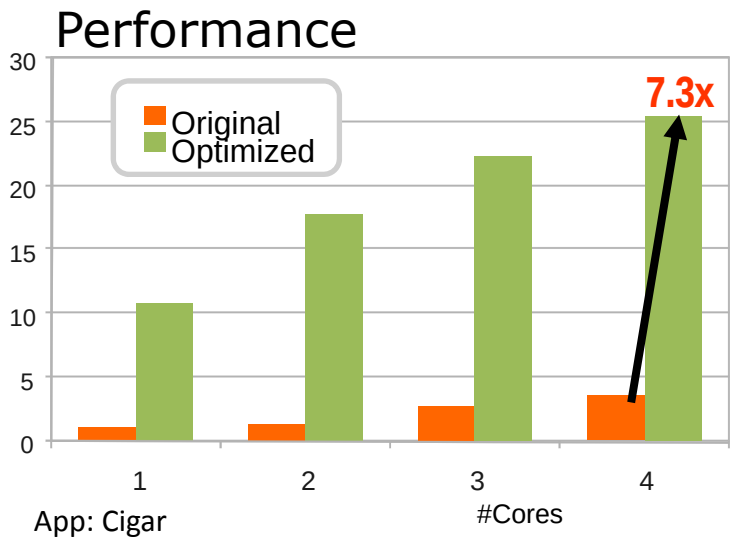


App: Cigar

Looks like a perfect scalable application!

Are we done?

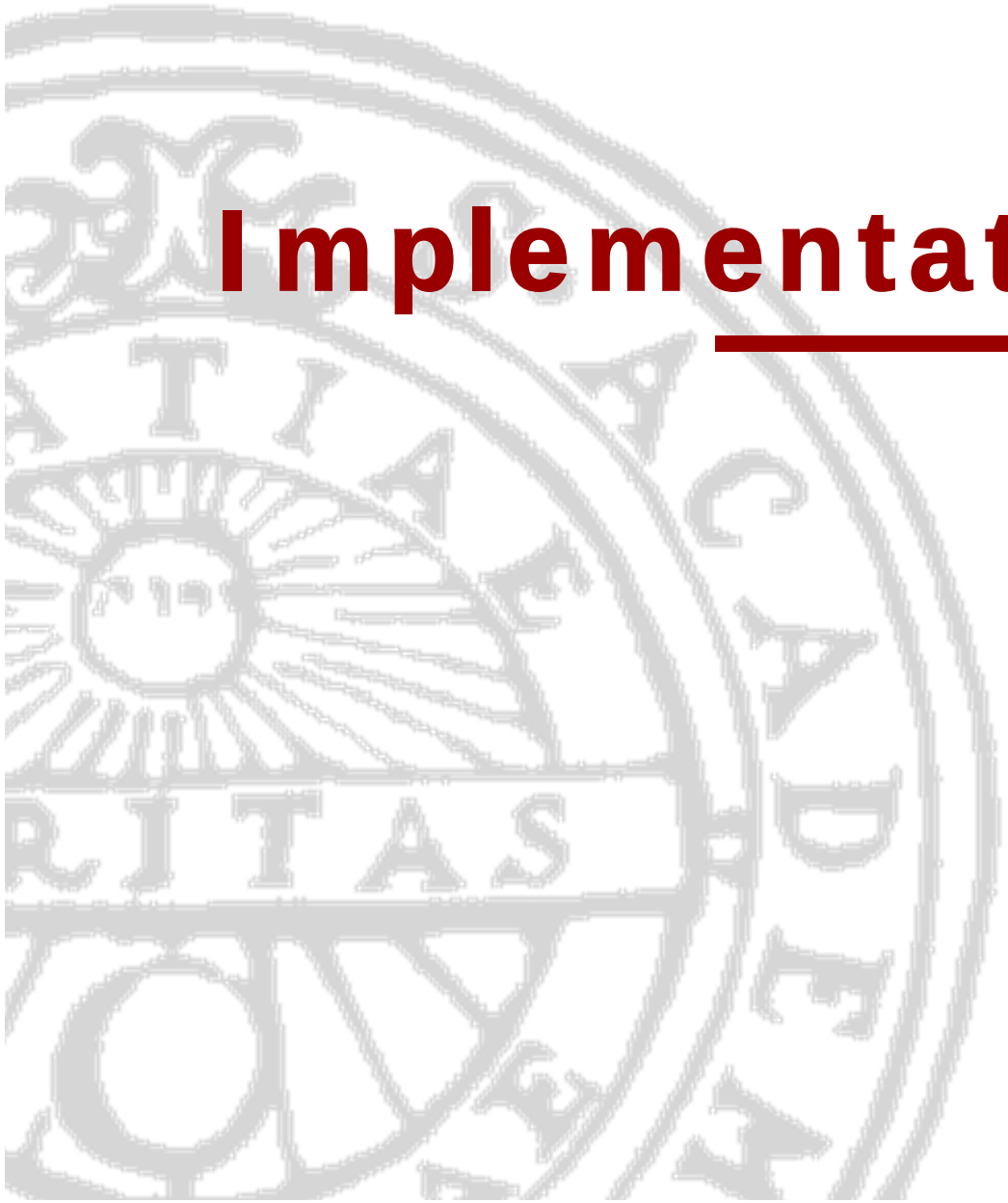
Example 2: The Same Application Optimized



Looks like a perfect scalable application!

Are we done?

➔ Duplicate one data structure



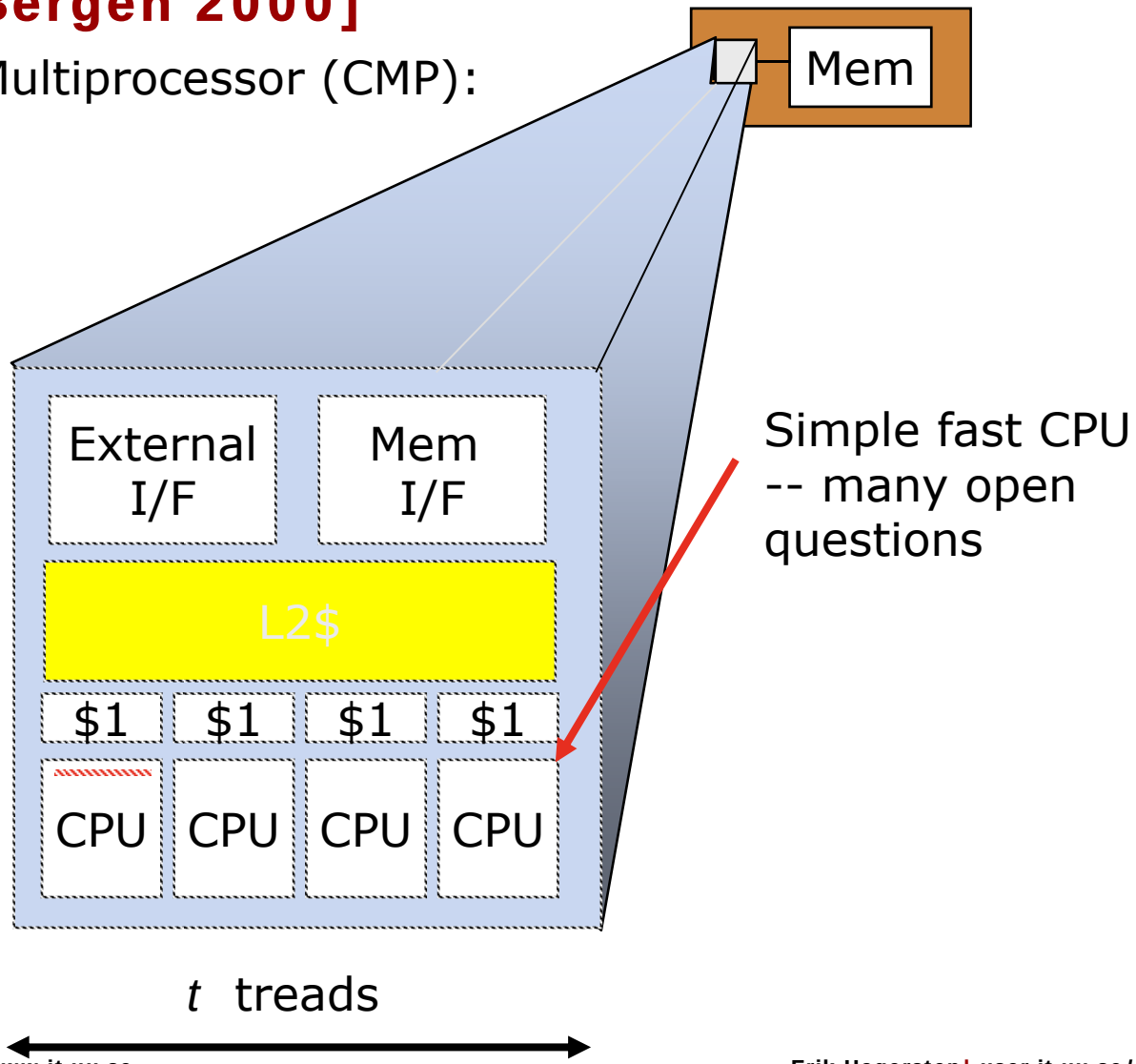
Implementation Trends

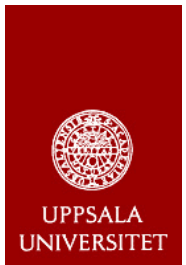
Predicting the future is hard

Predicting: "Chip Multiprocessor"

[from PARA Bergen 2000]

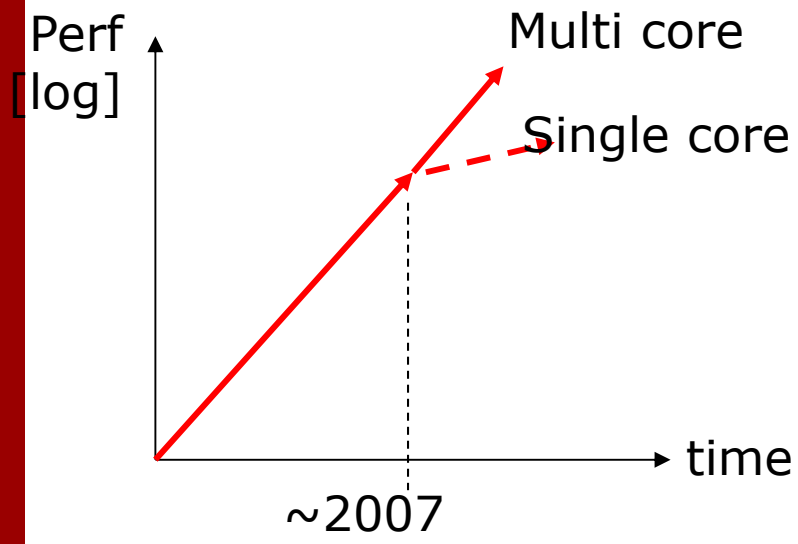
Chip Multiprocessor (CMP):





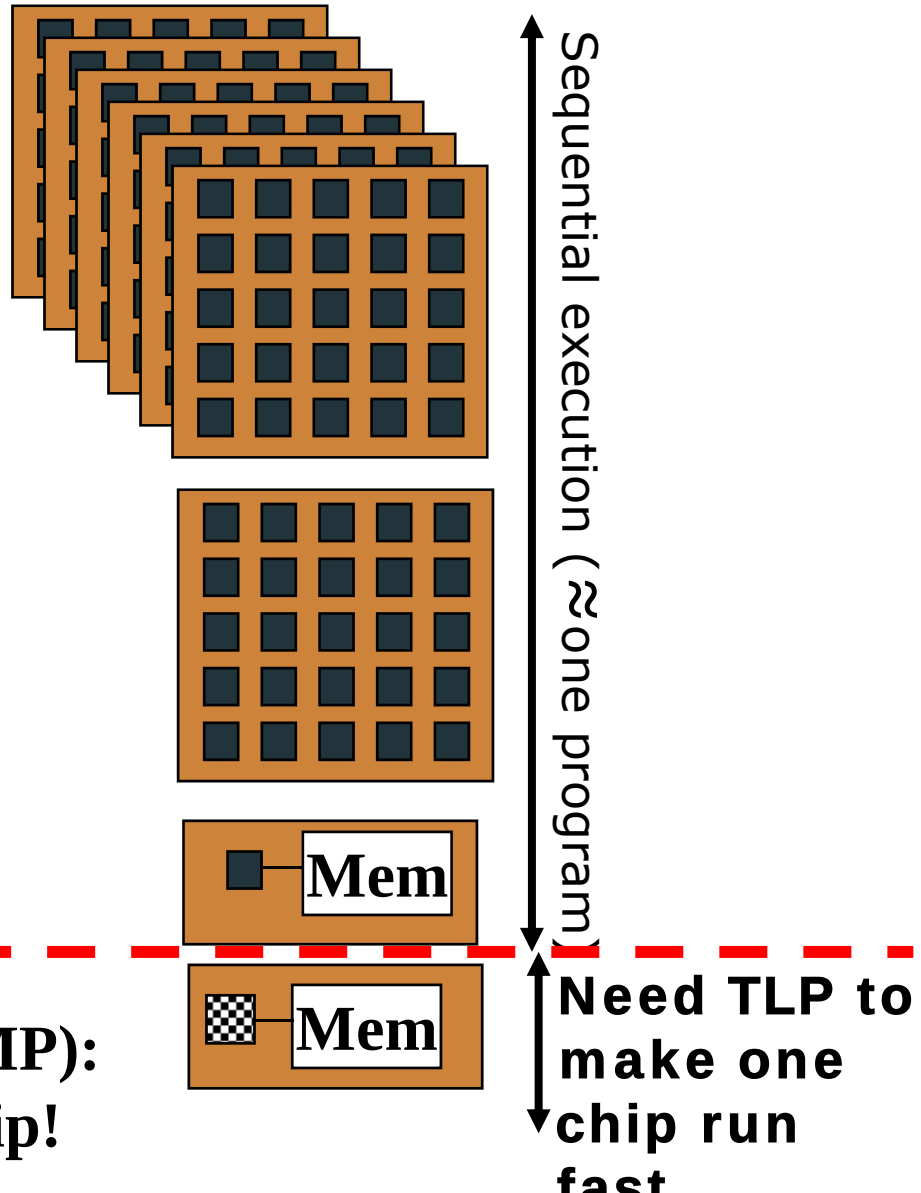
Why Multicores Now?

-- Hur Mår ”Moore’s Lag”? --



1. Not enough ILP/MLP to get payoff from using more transistors
2. Signal propagation delay » transistor delay
3. Power consumption $P_{dyn} \sim C \bullet f \bullet V^2$

Darling, I shrunk the computer



Paradigm Shift

Chip Multiprocessor (CMP):
A multiprocessor on a chip!

HPC in the Rear Mirror…

AMD Fusion,
New Phi,
ARM server,
ARM+nVIDIA
???

* Promise of performance

MC + Accelerators

* Forced by technology

† Hard to Use?

MC Clusters

† Too power Hungry?

* COTS cost

Beowulf x86 Linux Clusters

†COTS perf management

* UNIX Commercial computing

Killer Micro SMPs

†High cost, Bad scaling

* Scalability Naive view

Nifty Parallel

†Hard to use No standards

Vector

†Not general Expensive

1980 1990 2000 2010 ????

UPPSALA UNIVERSITET

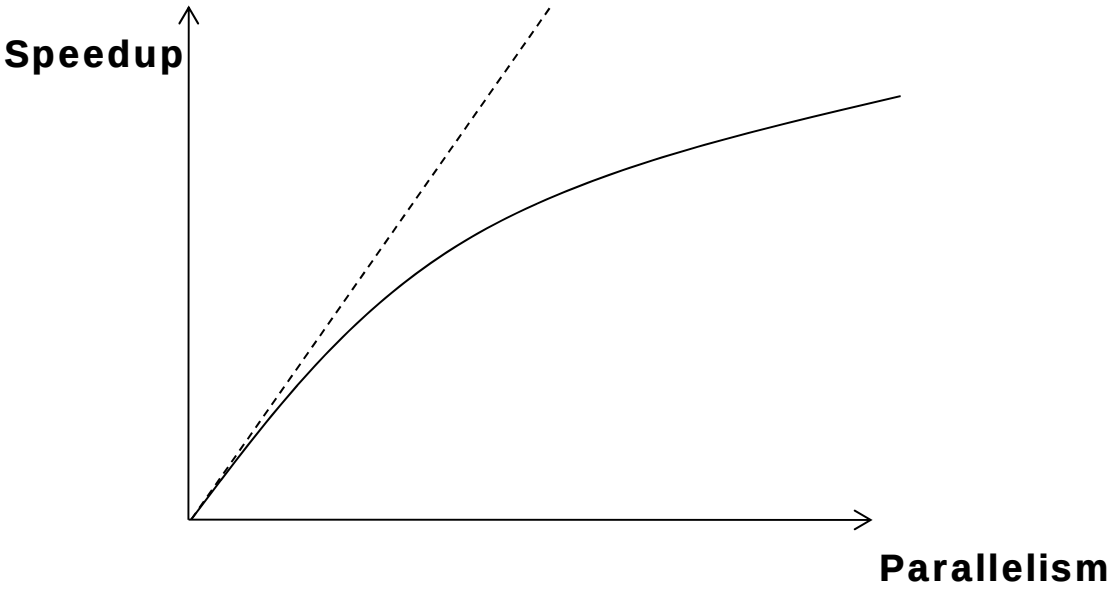
Parallelism can be used to hide memory latency

- Intel ”Hyper Threading”
- T1 Niagara, MIC, … (4 threads per core)
- GPUs
- …

- Is this a good idea?
- It cannot hide the need for bandwidth!

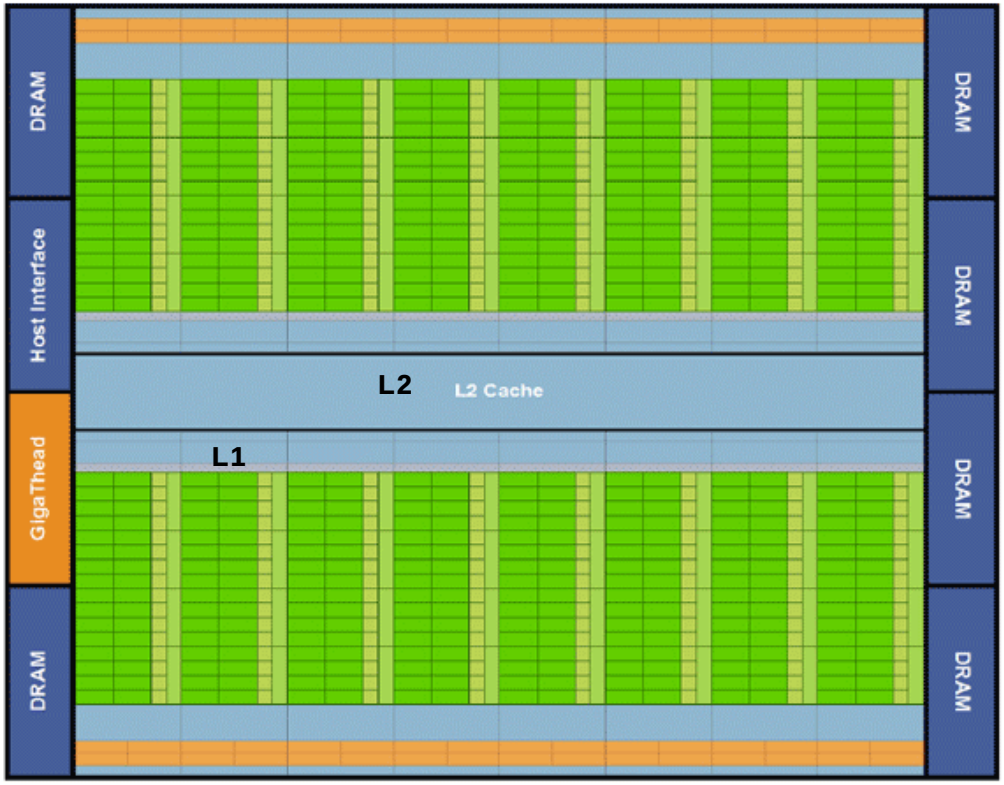
UPPSALA UNIVERSITET

Parallelism is a Hard Currency



Remember Amdahl's Law?

Do you have 1000 threads to spare?



Fermi's 16 SM are positioned around a common L2 cache. Each SM is a vertical rectangular strip that contain an orange portion (scheduler and dispatch), a green portion (execution units), and light blue portions (register file and L1 cache).

- 512 "cores" (***C***)
- 16 ***C***/StreamProcessor (***SP***)
- **SP** is SIMD-ish (sort off)
- Full DP-FP IEEE support
- 64kB L1 cache /***SP***
- 768kB global shared cache
  (less than the sum of L1:s)
- Atomic instructions
- ECC correction for DRAM
- Debugging support
- Synch within SP efficient
- Giant chip/high power
- …

UPPSALA UNIVERSITET

Trends for 2016

No major revolution of the Multicore magnitude

Challenge: Will the number of cores double every 2 years?

Moving towards MIMD+SIMD "fusion"

Architecture complexity grows

Bumpy memory/communication costs

Heterogenious architectures (e.g., ARM: big.LITTLE)

Memory bandwidth is a major bottleneck

Energy is a first-class citizen (and bottleneck)

Users are getting less computer-savvy (and ideally should not have to be)

UPPSALA UNIVERSITET

Implications

- One size will not "fit all"
- SIMD parallelism will be more prominent, but the jury is still out about how this will be done
- New memory technology is needed (stacked DRAM, memristors, phase-change memory …)
- More heterogeneous arch (size, mem, isa)
- Diversity: Different applications will need different "heterogeneous configurations"
- More parallelism needed, but ... memory/power to become the bottleneck anyhow
- **Even harder to use resources efficiently**

UPPSALA UNIVERSITET

HiPEAC Roadmap -- High Performance Embedded Architecture and Compilers