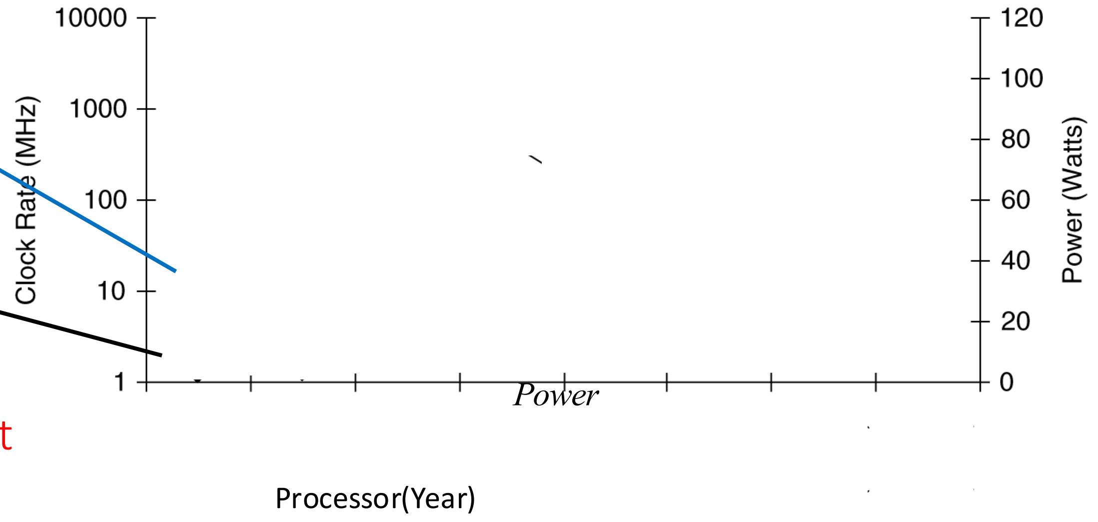
Topic V12

Power, Multiprocessor,
SPEC, Amdahl’s Law
Readings: (Section 1.7-1.10)

Power Trends

Clock Rate

Power



Integrated Circuit

In CMOS IC technology

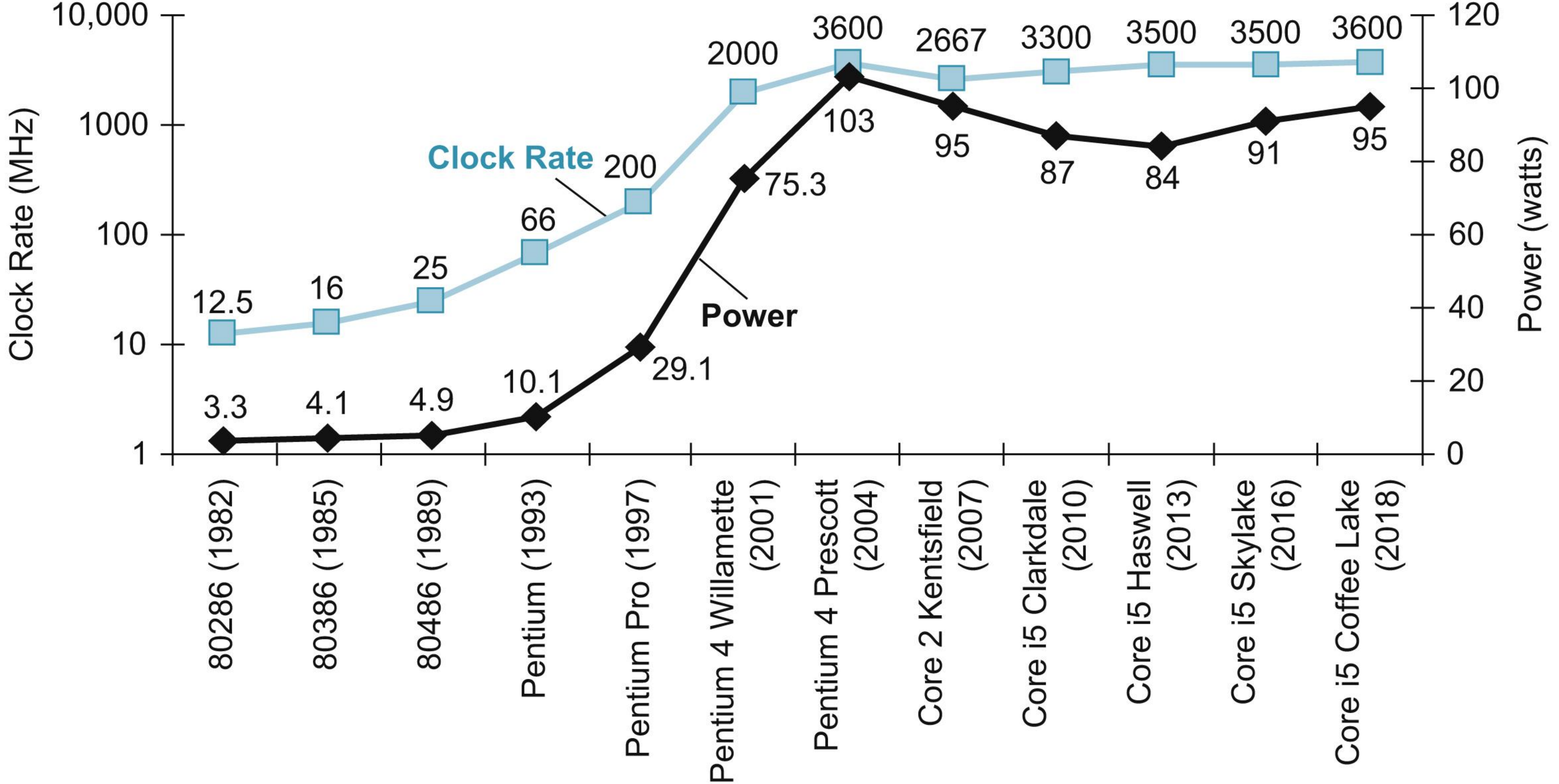
$$\text{Power} \propto 1/2 \text{ Capacitive load} \times \text{Voltage}^2 \times \text{Frequency}$$

×30

5V → 1V

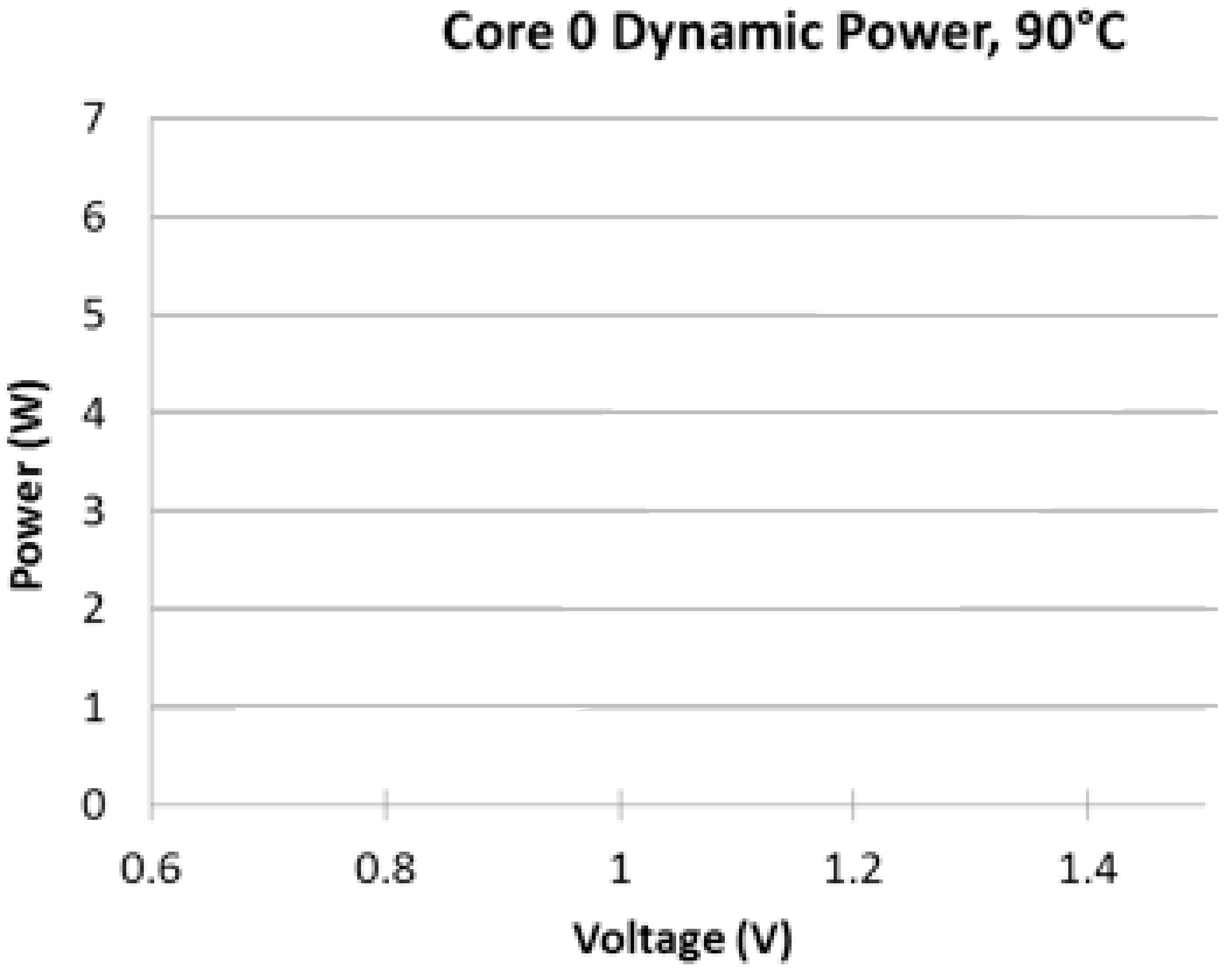
×1000

Complementary Metal Oxide Semiconductor



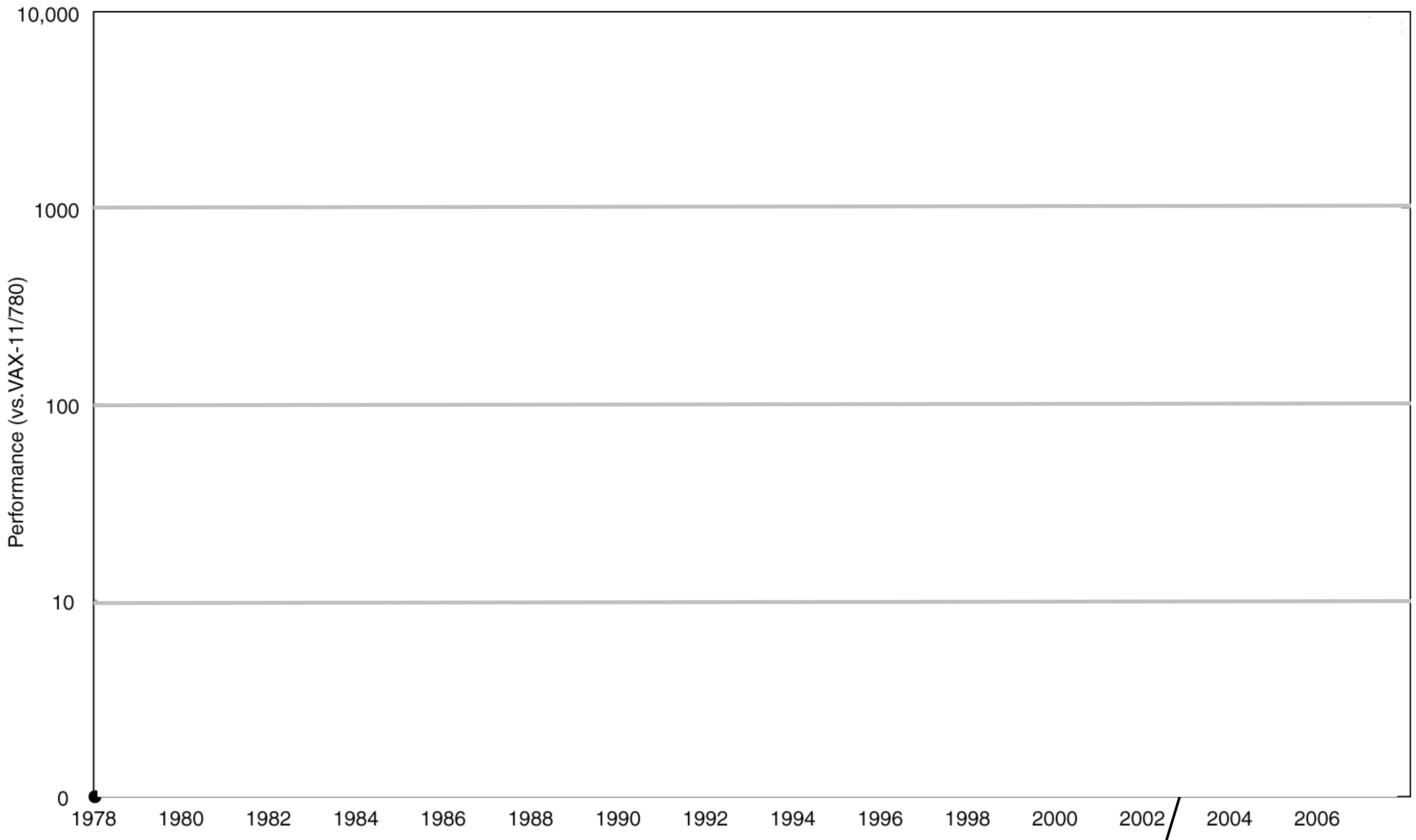
Clock Rate (MHz)
Power (watts)
Clock Rate
Power
10,000
1000
100
10
1
120
100
80
60
40
20
0
12.5
16
25
66
200
2000
3600
2667
3300
3500
3500
3600
3.3
4.1
4.9
10.1
29.1
75.3
103
95
87
84
91
95
80286 (1982)
80386 (1985)
80486 (1989)
Pentium (1993)
Pentium Pro (1997)
Pentium 4 Willamette (2001)
Pentium 4 Prescott (2004)
Core 2 Kentsfield (2007)
Core i5 Clarkdale (2010)
Core i5 Haswell (2013)
Core i5 Skylake (2016)
Core i5 Coffee Lake (2018)

Power X Frequency X Voltage



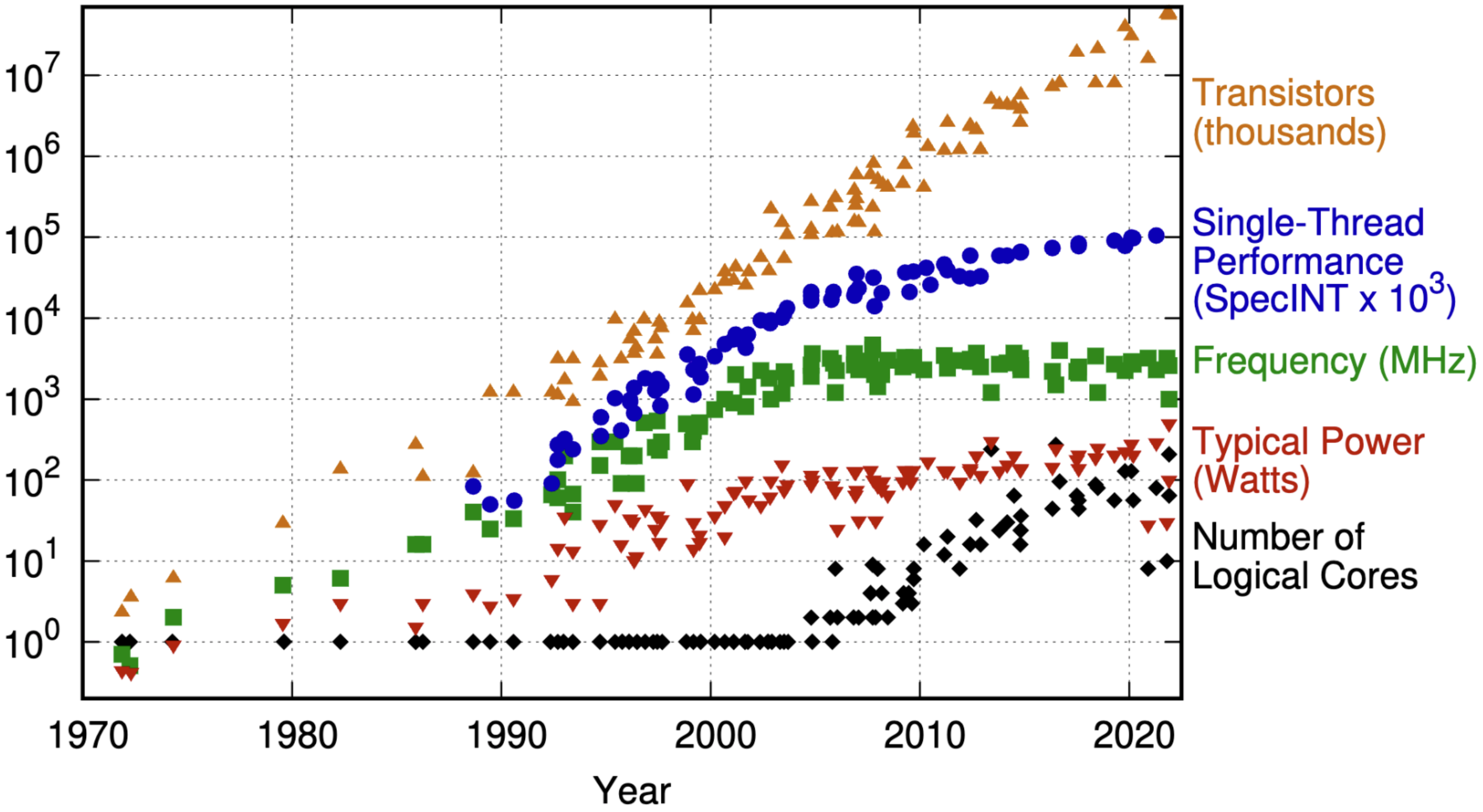
Source: http://chipdesignmag.com/lpd/pangrle/tag/amd/

Uniprocessor Performance



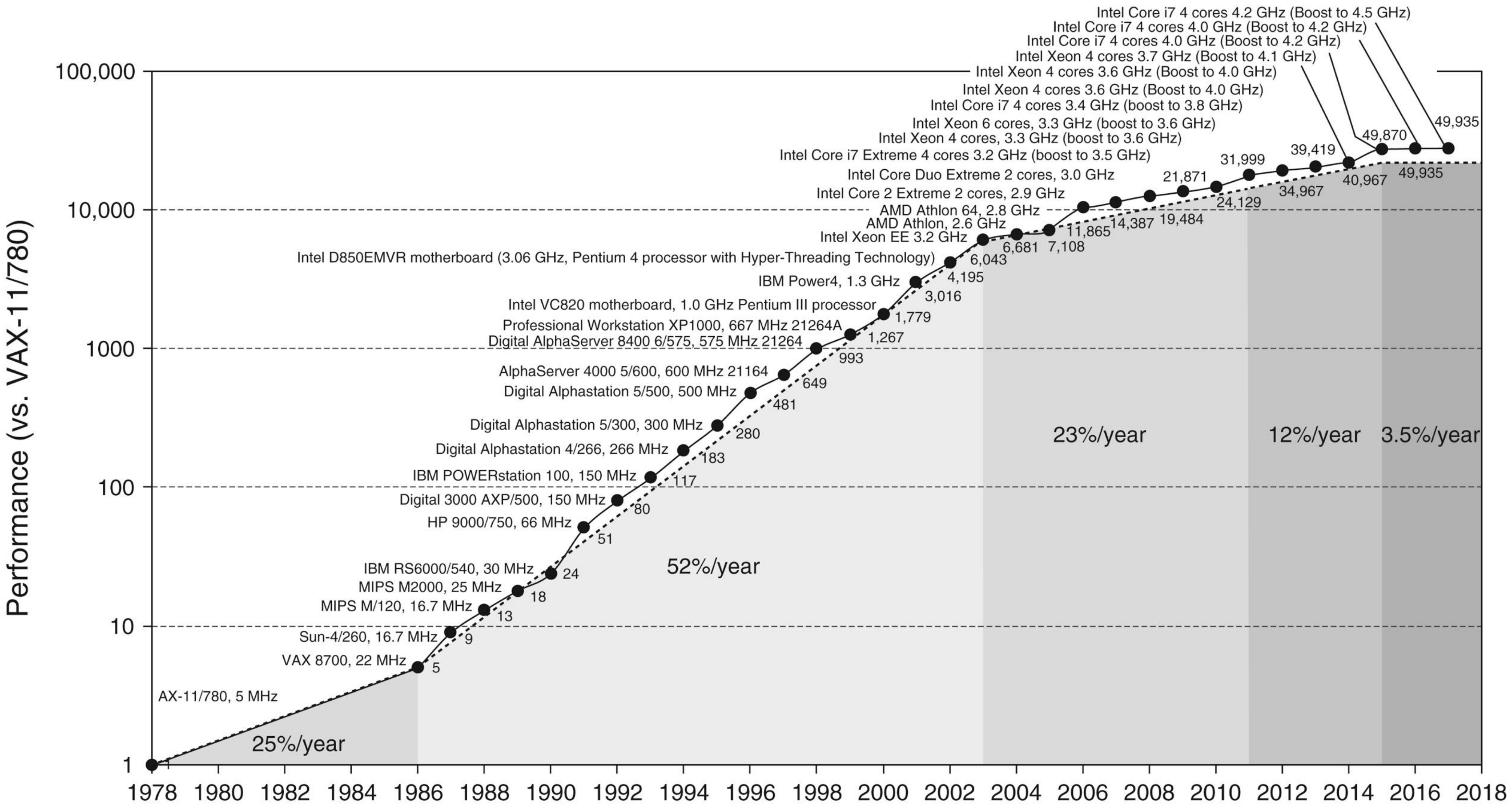
Constrained by power, instruction-level parallelism, memory latency

50 Years of Microprocessor Trend Data



Original data up to the year 2010 collected and plotted by M. Horowitz, F. Labonte, O. Shacham, K. Olukotun, L. Hammond, and C. Batten
New plot and data collected for 2010-2021 by K. Rupp

https://github.com/karlrupp/microprocessor-trend-data/blob/master/50yrs/50-years-processor-trend.pdf

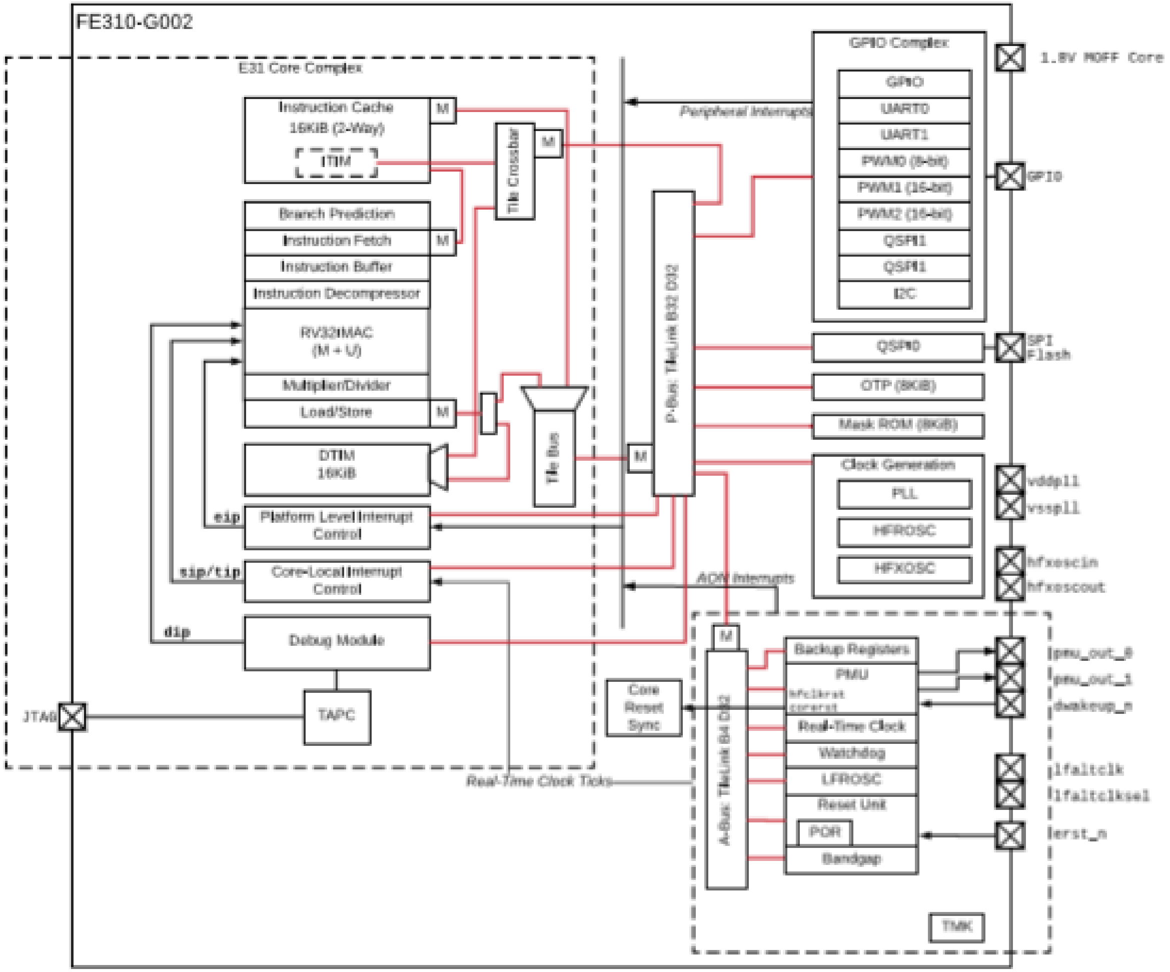


Performance (vs. VAX-11/780)
100,000
10,000
1000
100
10
1
1978
1980
1982
1984
1986
1988
1990
1992
1994
1996
1998
2000
2002
2004
2006
2008
2010
2012
2014
2016
2018
25%/year
52%/year
23%/year
12%/year
3.5%/year
AX-11/780, 5 MHz
VAX 8700, 22 MHz
Sun-4/260, 16.7 MHz
MIPS M/120, 16.7 MHz
MIPS M2000, 25 MHz
IBM RS6000/540, 30 MHz
HP 9000/750, 66 MHz
Digital 3000 AXP/500, 150 MHz
IBM POWERstation 100, 150 MHz
Digital Alphastation 4/266, 266 MHz
Digital Alphastation 5/300, 300 MHz
Digital Alphastation 5/500, 500 MHz
AlphaServer 4000 5/600, 600 MHz 21164
Digital AlphaServer 8400 6/575, 575 MHz 21264
Professional Workstation XP1000, 667 MHz 21264A
Intel VC820 motherboard, 1.0 GHz Pentium III processor
IBM Power4, 1.3 GHz
Intel D850EMVR motherboard (3.06 GHz, Pentium 4 processor with Hyper-Threading Technology)
Intel Xeon EE 3.2 GHz
AMD Athlon, 2.6 GHz
AMD Athlon 64, 2.8 GHz
Intel Core 2 Extreme 2 cores, 2.9 GHz
Intel Core Duo Extreme 2 cores, 3.0 GHz
Intel Core i7 Extreme 4 cores 3.2 GHz (boost to 3.5 GHz)
Intel Xeon 4 cores, 3.3 GHz (boost to 3.6 GHz)
Intel Xeon 6 cores, 3.3 GHz (boost to 3.6 GHz)
Intel Core i7 4 cores 3.4 GHz (boost to 3.8 GHz)
Intel Xeon 4 cores 3.6 GHz (Boost to 4.0 GHz)
Intel Xeon 4 cores 3.6 GHz (Boost to 4.0 GHz)
Intel Xeon 4 cores 3.7 GHz (Boost to 4.1 GHz)
Intel Core i7 4 cores 4.0 GHz (Boost to 4.2 GHz)
Intel Core i7 4 cores 4.0 GHz (Boost to 4.2 GHz)
Intel Core i7 4 cores 4.2 GHz (Boost to 4.5 GHz)
1
5
9
13
18
24
51
80
117
183
280
481
649
993
1,267
1,779
3,016
4,195
6,043
6,681
7,108
11,865
14,387
19,484
21,871
24,129
31,999
34,967
39,419
40,967
49,870
49,935
49,935

Single Core Processor

Instruction-Level Parallelism (ILP)

- Hardware executes multiple instructions at once
- ILP is hidden from the programmer



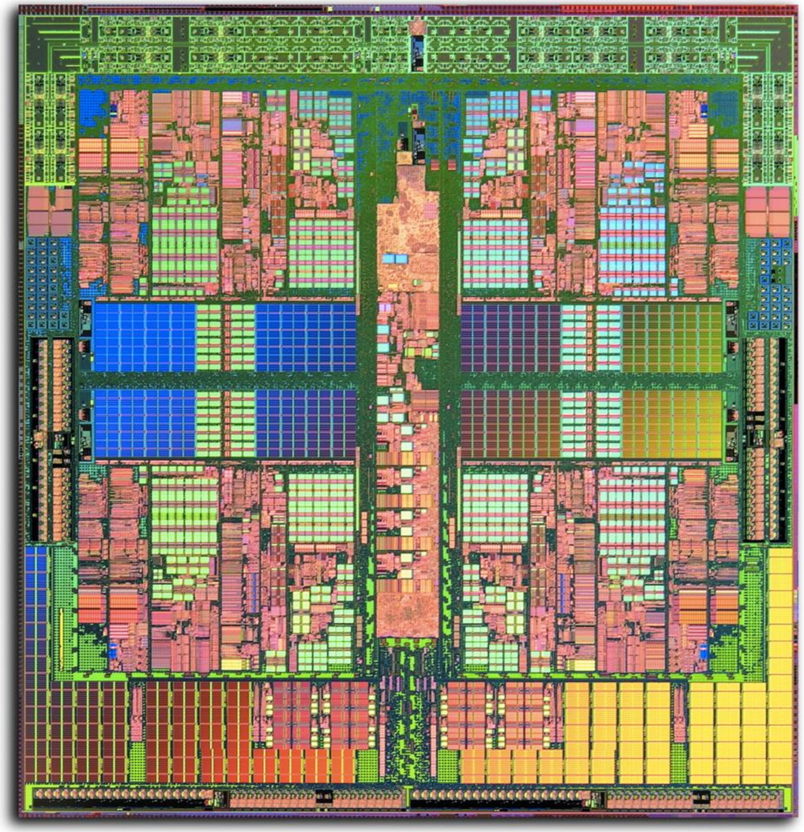
RISC-V FE310-G002 top-level

Multiprocessors

Multicore microprocessors

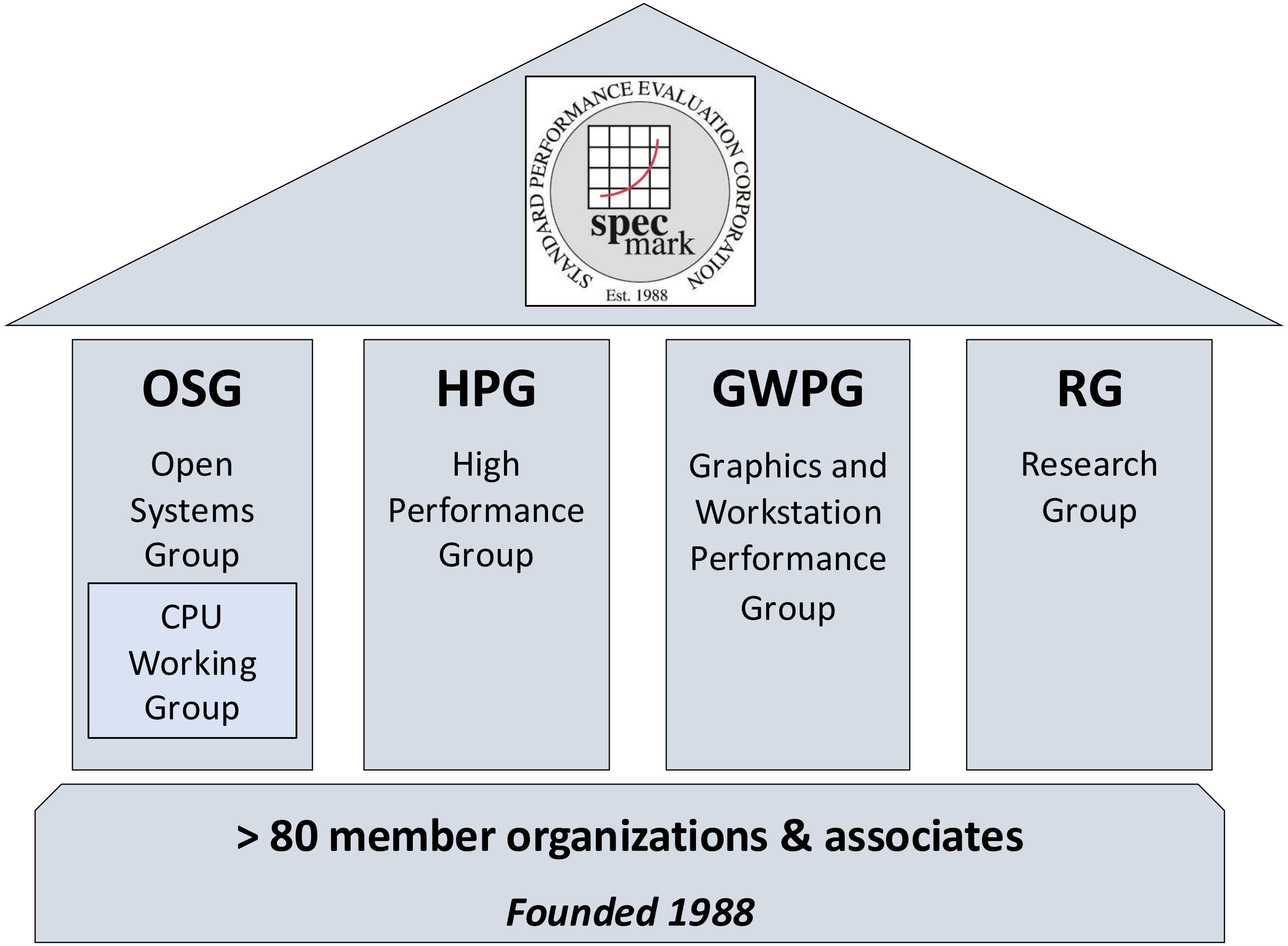
Requires explicitly parallel programming

- Hard to do
  - Programming for performance
  - Load balancing
  - Optimizing communication and synchronization



Quad-Core AMD Opteron - 2007

Standard Performance Evaluation Corporation



Spec Research Group

spec Research

KIT Karlsruhe Institute of Technology

IBM

SAP

AMD

Cisco Systems

intel

TU Delft
Delft University of Technology

Berkeley University of California

fortiss

hp

vmware

U C

McGill

TECHNISCHE UNIVERSITÄT DARMSTADT

PURDUE UNIVERSITY

ORACLE

SIEMENS

Universität Stuttgart

berlin

UNIVERSITÄT MANNHEIM

THE UNIVERSITY OF TEXAS AT AUSTIN

Imperial College London

NICTA

MITRE

UH

UNF UNIVERSITY of NORTH FLORIDA

OFFIS

UNIVERSITY OF ALBERTA

UNIVERSITEIT GENT

redhat

UNIVERSITY of HOUSTON

LINCOLN LABORATORY
MASSACHUSETTS INSTITUTE OF TECHNOLOGY

salesforce

C A U
Christian-Albrechts-Universität zu Kiel

SAN FRANCISCO STATE UNIVERSITY

DELL

財團法人資訊工業策進會
INSTITUTE FOR INFORMATION INDUSTRY

POLITEHNICA UNIVERSITATEA DIN BUCURESTI 1818

IIT Bombay

UNIVERSITY OF MINNESOTA
Driven to Discover℠

浙江大學
ZHEJIANG UNIVERSITY

HPI Hasso Plattner Institut
IT Systems Engineering | Universität Potsdam

SPEC CPU Benchmark

Programs used to measure performance

- Supposedly typical of actual workload

Standard Performance Evaluation Corp (SPEC)

- Develops benchmarks for CPU, I/O, Web, …

SPEC CPU2006 and SPEC CPU2017

- Elapsed time to execute a selection of programs
  - Negligible I/O --- focuses on CPU performance
- Normalize relative to reference machine
- Summarize as geometric mean of performance ratios
  - CINT2006 (integer) and CFP2006 (floating-point)

$$\sqrt[n]{\prod_{i=1}^{n} \text{Execution time ratio}_i}$$

A Sample SPEC CPU 2017 int Result Publication

Hardware		Software	
CPU Name:	AMD EPYC 7742	OS:	SUSE Linux Enterprise Server 15 SP1 (x86_64) Kernel 4.12.14-195-default
Max MHz:	3400	Compiler:	C/C++/Fortran: Version 2.0.0 of AOCC
Nominal:	2250	Parallel:	Yes
Enabled:	64 cores, 1 chip, 2 threads/core	Firmware:	Version 0302 released Aug-2019
Orderable:	1 chip	File System:	xfs
Cache L1:	32 KB I + 32 KB D on chip per core	System State:	Run level 3 (multi-user)
L2:	512 KB I+D on chip per core	Base Pointers:	64-bit
L3:	256 MB I+D on chip per chip, 16 MB shared / 4 cores	Peak Pointers:	32/64-bit
Other:	None	Other:	jemalloc: jemalloc memory allocator library v5.1.0
Memory:	256 GB (8 x 32 GB 2Rx4 PC4-3200AA-R)	Power Management:	--
Storage:	1 x 1 TB SATA SSD		
Other:	None		

https://www.spec.org/cpu2017/results/res2019q3/cpu2017-20190830-17266.pdf

Benchmark	Base						
	Threads	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio
600.perlbench_s	64	368	4.82	375	4.74	**371**	**4.78**

Bold indicates the median of the three measurements.

https://www.spec.org/cpu2017/results/res2019q3/cpu2017-20190830-17266.pdf

$$\text{SPECratio} = \frac{\text{Ref time}}{\text{Exec time}}$$

Benchmark	Base						
	Threads	Seconds	Ratio	Seconds	Ratio	Seconds	Ratio
600.perlbench_s	64	368	4.82	375	4.74	**371**	**4.78**
602.gcc_s	64	399	9.97	400	9.95	**400**	**9.95**
605.mcf_s	64	305	15.5	**305**	**15.5**	305	15.5
620.omnetpp_s	64	317	5.15	315	5.17	**316**	**5.16**
623.xalancbmk_s	64	152	9.33	**152**	**9.35**	151	9.37
625.x264_s	64	**137**	**12.8**	137	12.8	138	12.8
631.deepsjeng_s	64	286	5.00	**286**	**5.00**	286	5.00
641.leela_s	64	393	4.34	393	4.34	**393**	**4.34**
648.exchange2_s	64	179	16.5	179	16.4	**179**	**16.5**
657.xz_s	64	296	20.9	**296**	**20.9**	296	20.9

SPECspeed®2017_int_base = 8.98

Bold indicates the median of the three measurements.

$$\text{SPECratio} = \frac{\text{Ref time}}{\text{Exec time}}$$

https://www.spec.org/cpu2017/results/res2019q3/cpu2017-20190830-17266.pdf

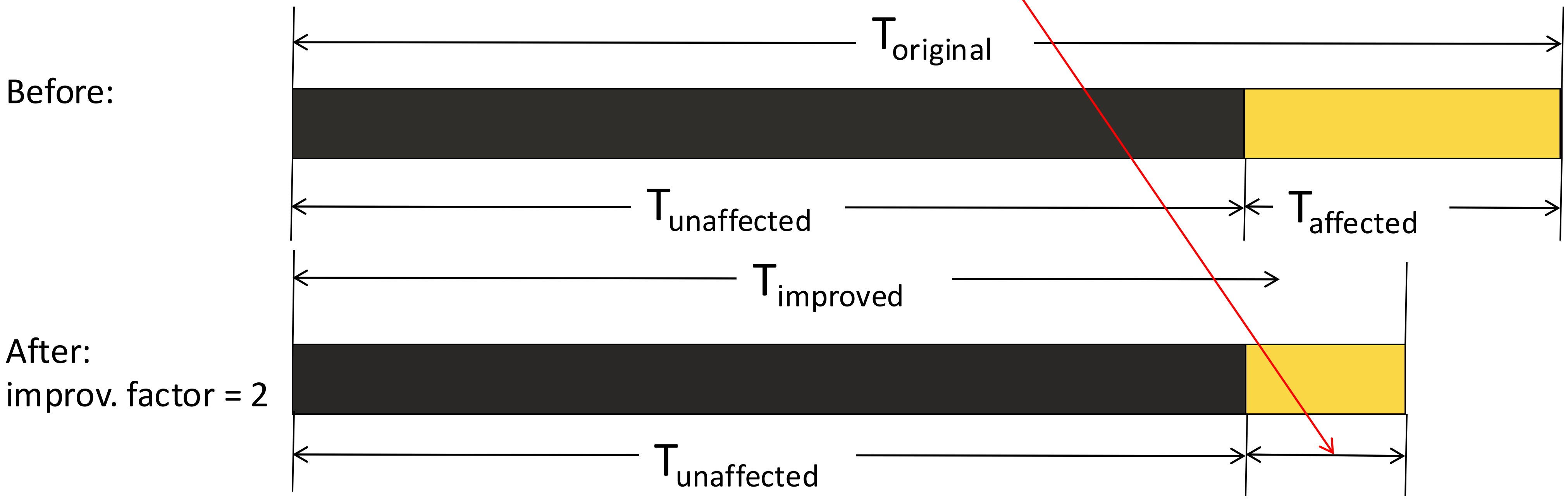
The Alberta Workloads for the SPEC CPU 2017 Benchmark Suite

José Nelson Amaral*, Edson Borin†, Dylan Ashley*, Caian Benedicto†, Elliot Colp‡,
João Henrique Stange Hoffmam†, Marcus Karpoff*, Erick Ochoa*, Morgan Redshaw¶, Raphael Ernani Rodrigues§
* Department of Computing Science, University of Alberta, Edmonton, AB, Canada.
† Instituto de Computação, Universidade de Campinas, Campinas, SP, Brazil.
‡ Bioware, Edmonton, AB, Canada.
§ Microsoft, Redmond, WA, USA.
¶ DeepMind, London, UK.

Pitfall: Amdahl's Law

Improving a single aspect of a computer does not lead to a proportional improvement in overall performance

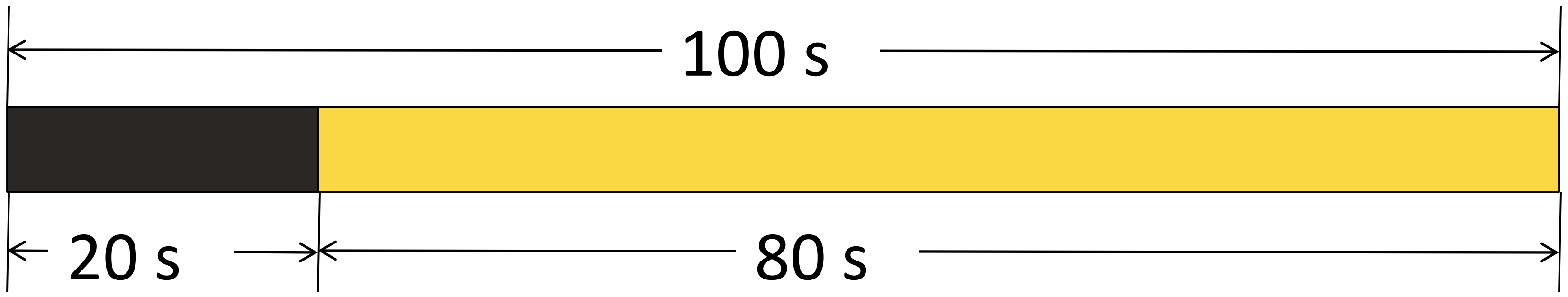
$$T_{improved} = \frac{T_{affected}}{\text{improvement factor}} + T_{unaffected}$$



Pitfall: Amdahl's Law

Example: In a 100-second program, multiplication accounts for 80 seconds

How much improvement in multiplication performance do we need to get 5× overall improvement?



5× improvement ⇒ new total time = 20 seconds

$$20 = \frac{80}{n} + 20$$

Can't be done!

Pitfall: MIPS as a Performance Metric

- MIPS: Millions of Instructions Per Second
  - Doesn't account for
    - Differences in ISAs between computers
    - Differences in complexity between instructions

$$\text{MIPS} = \frac{\text{Instruction Count}}{\text{Execution Time} \times 10^6}$$

$$= \frac{\text{Instruction Count}}{\frac{\text{Instruction Count} \times \text{CPI}}{\text{Clock rate}} \times 10^6} = \frac{\text{Clock rate}}{\text{CPI} \times 10^6}$$

- CPI varies between programs on a given CPU

Concluding Remarks

Cost/performance is improving

Due to underlying technology development

Hierarchical layers of abstraction

In both hardware and software

Instruction-set architecture

The hardware/software interface

Execution time: the best performance measure

Power is a limiting factor

Use parallelism to improve performance