




Historische Entwicklung

- Computer History Museum
Mountain View, CA, USA
- <https://computerhistory.org/>
-  Explore/Timelines
-  Timeline of Computer History
-  Filter by Category Computers

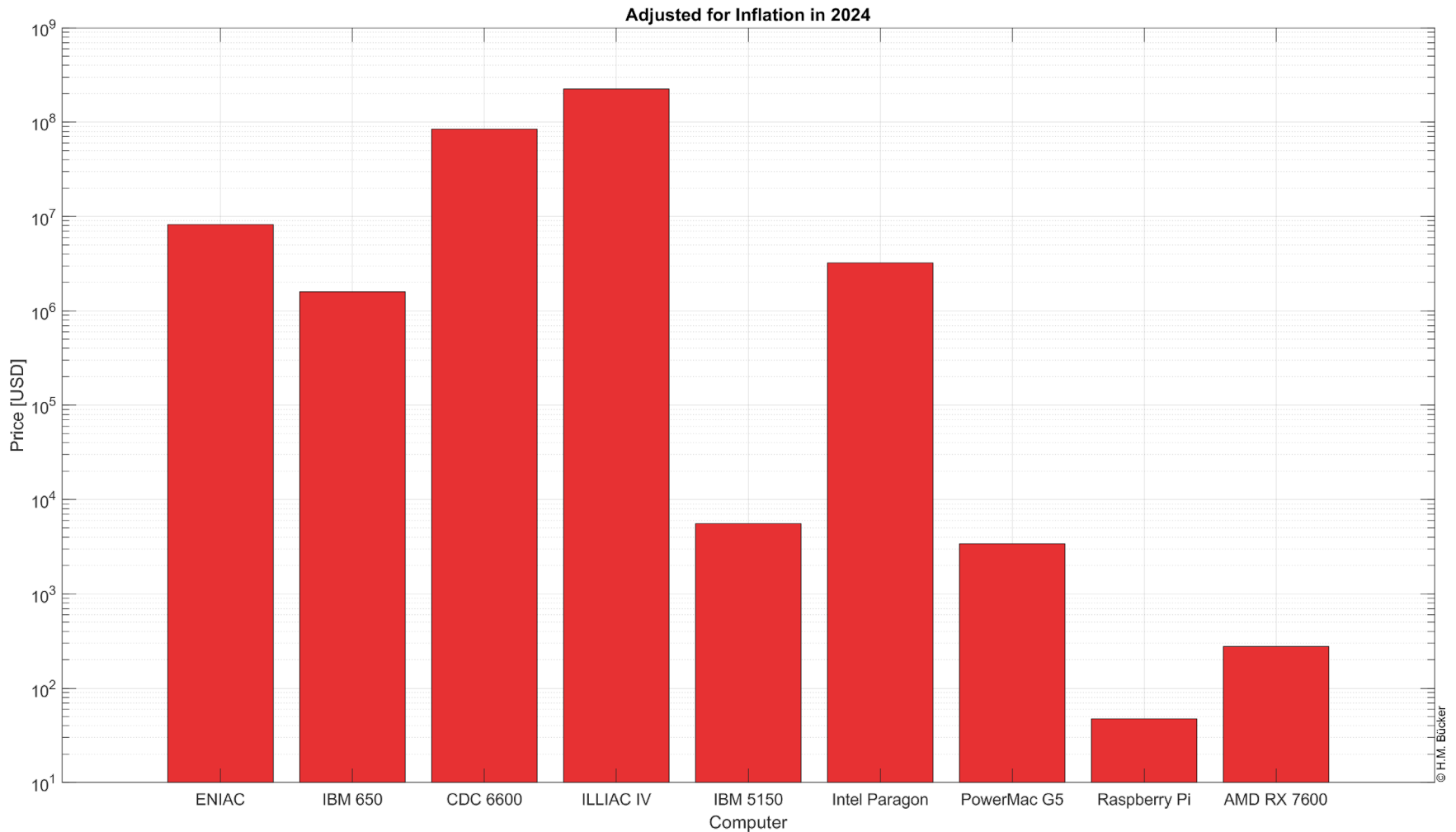
Datensammlung

Computer	Year	Price [USD]		FLOPS
		Introductory	2024	
ENIAC	1946	486 804	8 249 375	357
IBM 650	1954	150 000	1 595 260	100
CDC 6600	1964	8 500 000	84 839 628	$3.3 \cdot 10^9$
ILLIAC IV	1972	30 000 000	2 25 121 890	$50.0 \cdot 10^6$
IBM 5150	1981	1 565	5 548	$47.0 \cdot 10^6$
Intel Paragon	1993	1 495 000	3 233 404	$5.0 \cdot 10^9$
PowerMac G5	2003	2 000	3 395	$4.7 \cdot 10^9$
Raspberry Pi	2012	35	48	$41.0 \cdot 10^6$
AMD Radeon RX 7600	2023	269	277	$21.5 \cdot 10^{12}$

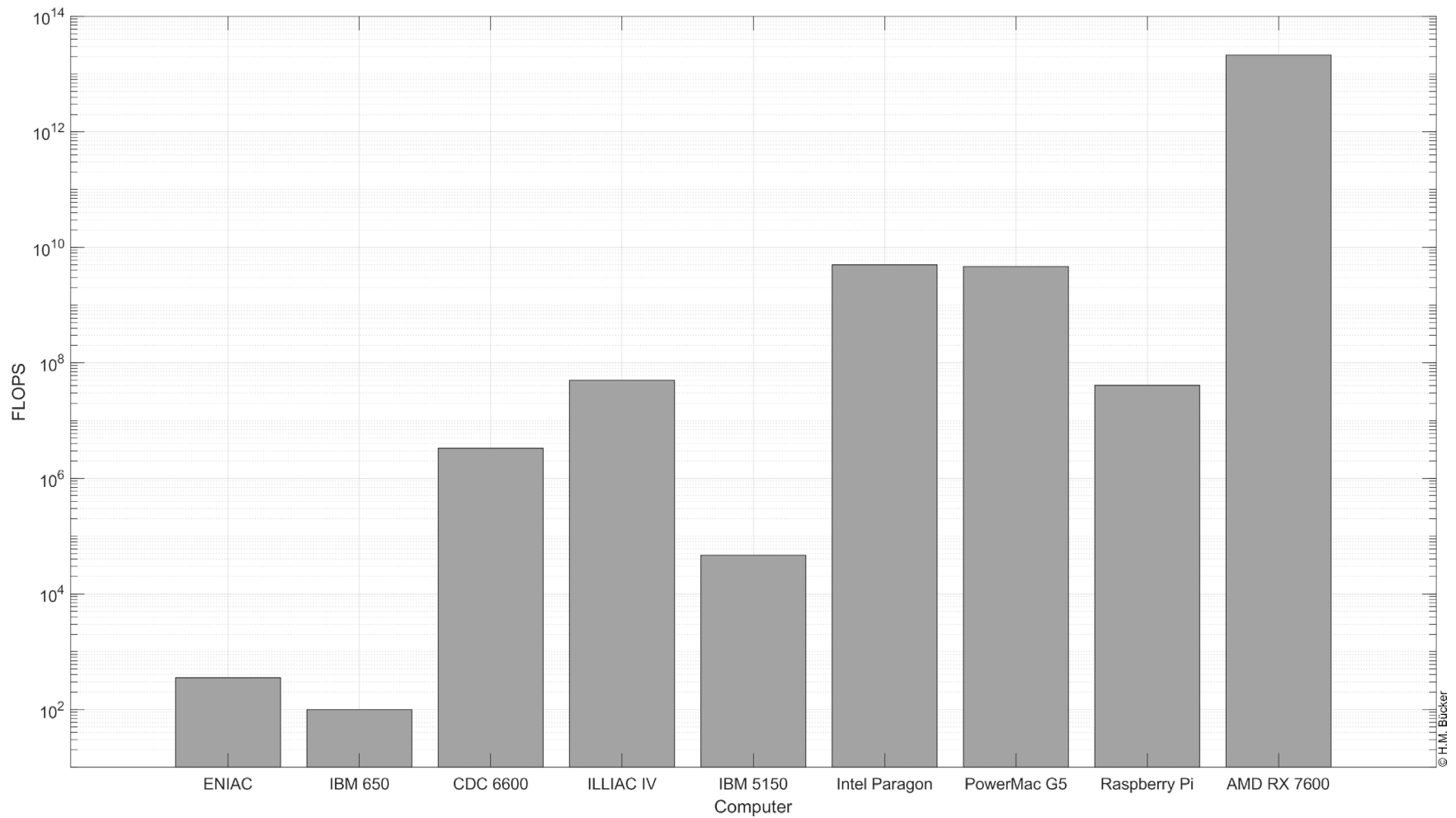
Quellen für Datensammlung

- Goldstine, Goldstine, 1966. doi: 10.1109/85.476557
- Martin, 1995. doi: 10.1109/44.476631
- Weik, 1961. www.jstor.org/stable/45363261
- Nelson, 2005. doi: 10.1002/cpe.890
- Ibbett, 1982. doi: 10.1007/978-1-4757-6715-5_6
- Spicer, 2000. [web.archive.org/web/20170605132007/
http://www.drdobbs.com/control-data-6600-the-supercomputer-arri/184404102](http://web.archive.org/web/20170605132007/http://www.drdobbs.com/control-data-6600-the-supercomputer-arri/184404102)
- Hockney, Jesshope, 1988. ISBN: 0852748116
- Slotnick, 1982. 10.1109/MAHC.1982.10003
- Qin et al., 2018. doi: 10.1007/978-981-13-0701-0_1
- Cooper, 2021. doi: 10.1093/itnow/bwab033
- www.computerhistory.org/collections/catalog/X1644.99
- en.wikipedia.org/wiki/Power_Mac_G5
- top500.org/system/173225
- en.wikipedia.org/wiki/Raspberry_Pi#History
- en.wikipedia.org/wiki/FLOPS
- Chien, 2022. doi: 10.1017/9781009000598

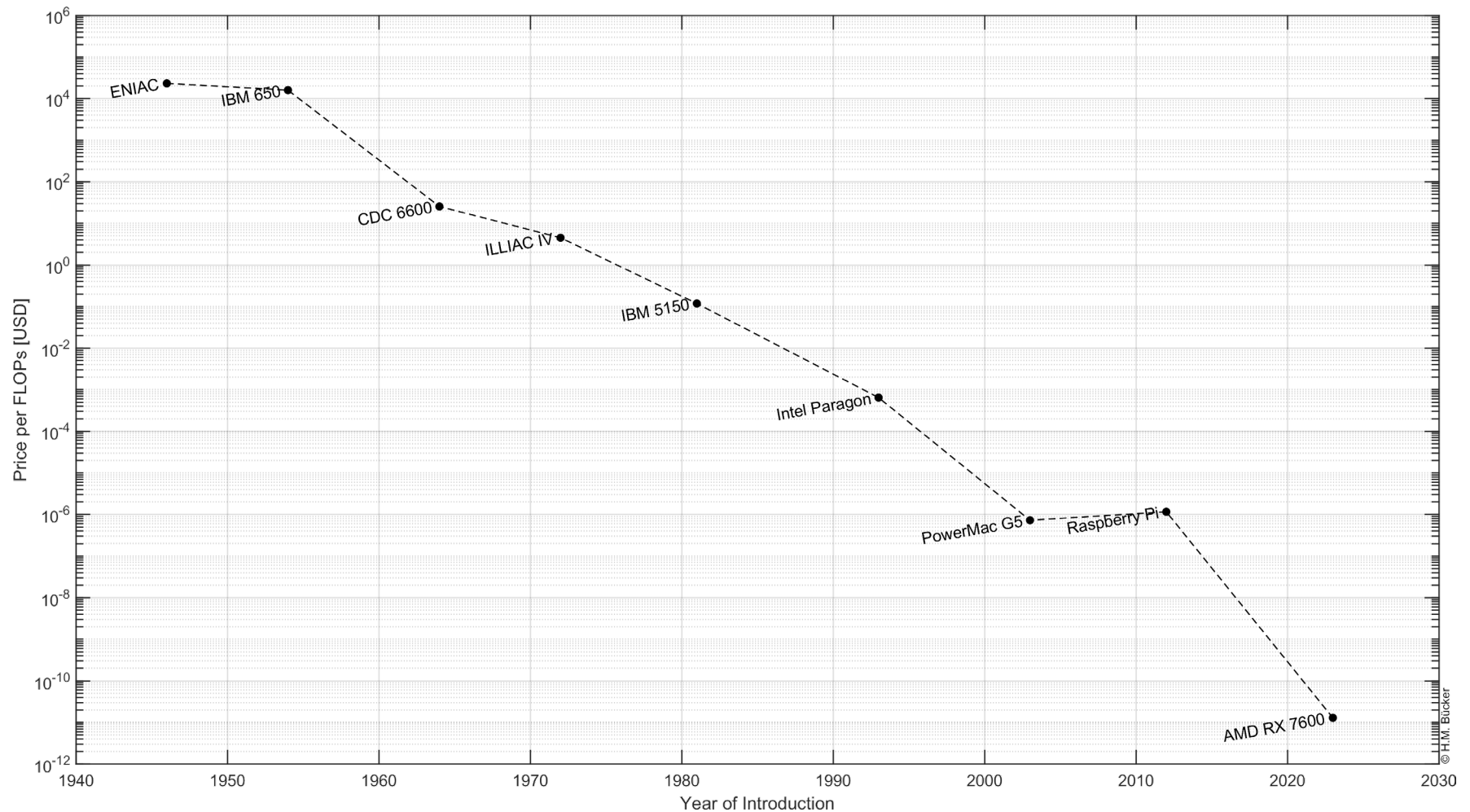
Price



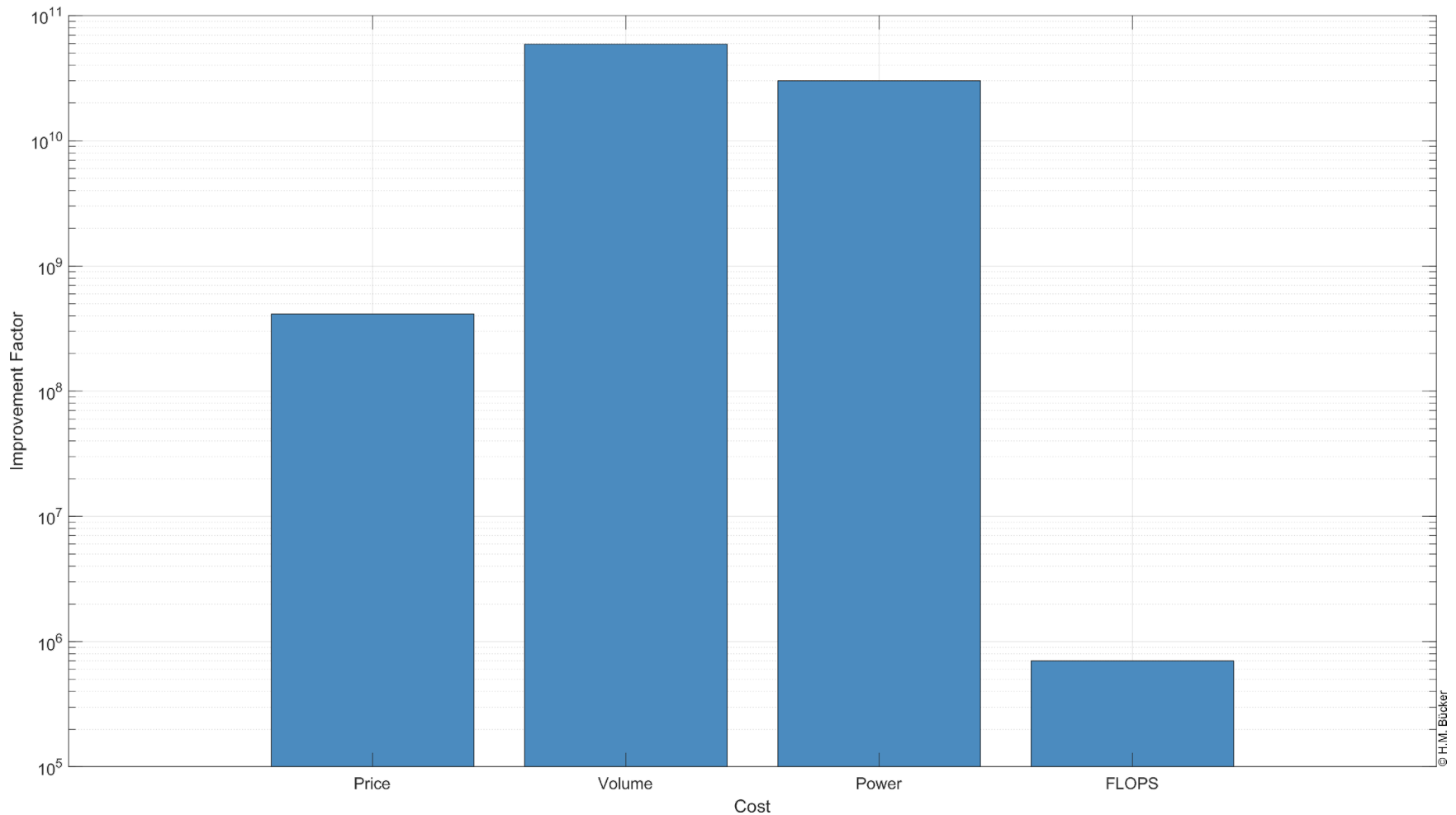
FLOPS



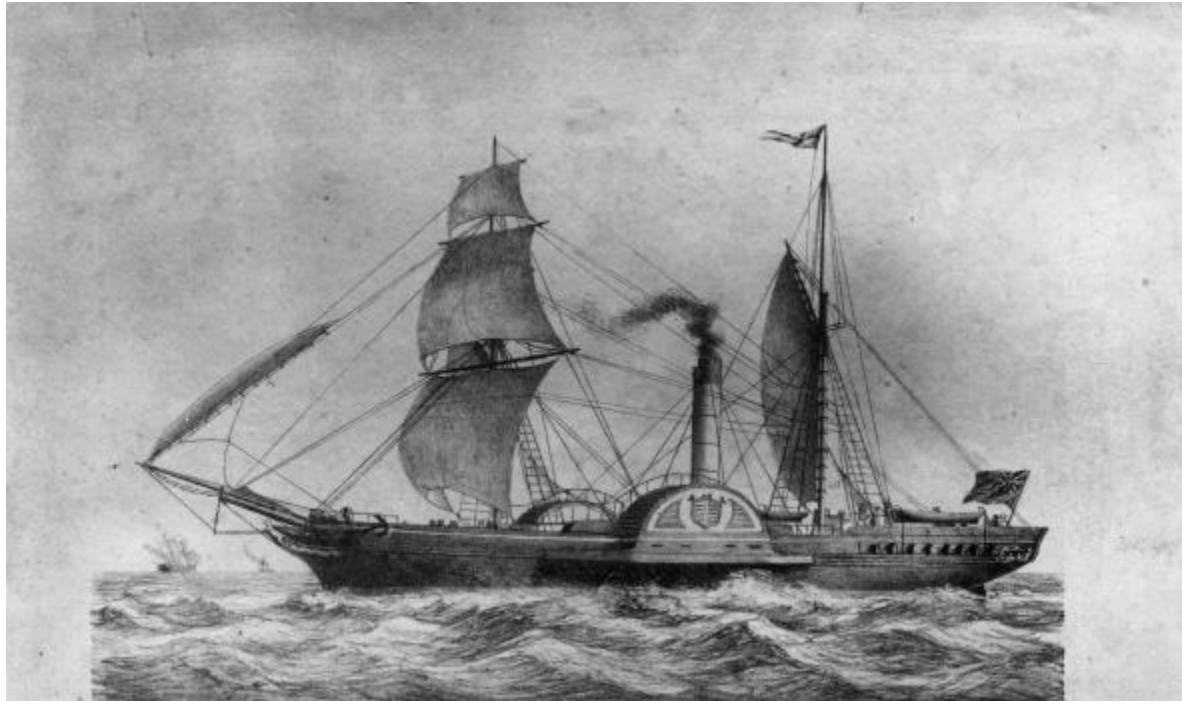
Price per FLOPS



ENIAC (1946) vs ARM Cortex M3 (2004)



Dampfschiff Sirius



[Wikimedia Commons](#)

- 4. April 1838: Cork, Irland
- 23. April 1838: Manhattan, USA
- Dauer: 18 Tage und 4 Stunden (= 436 h)

Überschallflugzeug Concorde

$$436/3,5 = 125$$



Eduard Marmet via Wikimedia Commons

- 1976 – 2003: Paris, New York
- Dauer: 3,5 h

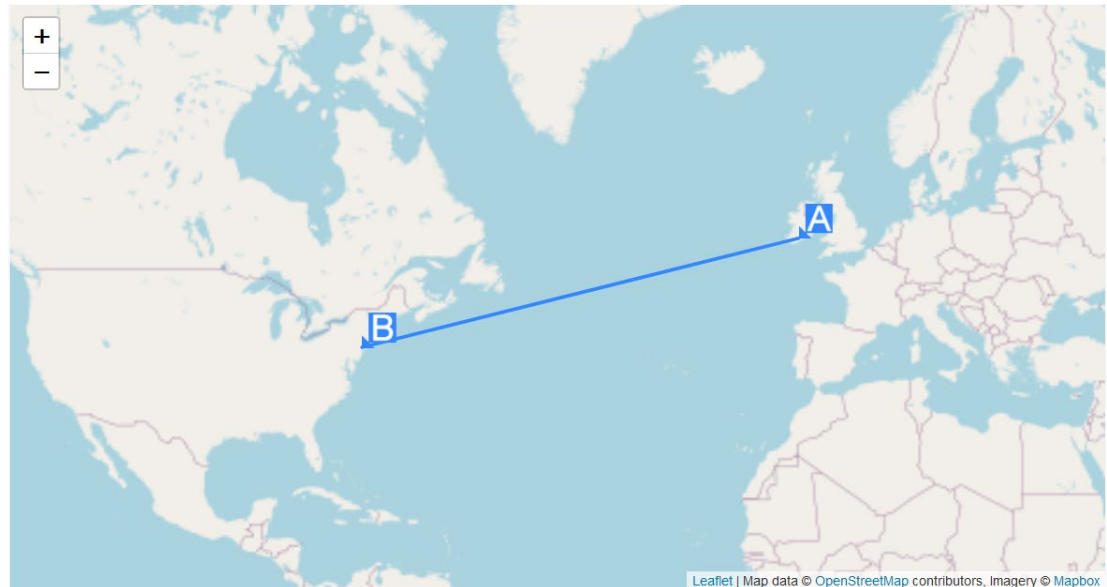
Long-Term Advanced Propulsion Concepts and Technologies

- www.esa.int/techresources/lapcat
 - Totale Kosten: 7,1 M Euro
 - Start: 15. April 2005
 - Dauer: 36 Monate
-
- www.esa.int/techresources/lapcat II
 - Totale Kosten: 10,4 M Euro
 - Start: 1. Oktober 2008
 - Dauer: 48 Monate

Long-Term Advanced Propulsion Concepts and Technologies

- Ziel: Brüssel – Sydney in 2 h bis 4 h
- Geschwindigkeit: 6 000 km/h
- Entfernung Cork – New York: 5000 km
- Geschätzte Dauer: 0,8 h

$$436/0,8 = 545$$



Instruction Set Architecture (ISA)

Examples of important ISA families

ISA	Supporters	Application Areas
x86	AMD, Intel	laptops, PCs, servers
ARM	Apple, ARM, MediaTek, Qualcomm, Samsung	smart devices, smartphones, IoT
RISC-V	RISC-V International	driven by collaborative community

Instruction Set Compatibility:
Programs run portably across many computers

RISC-V Greencard

Open RISC-V Reference Card ①

Base Integer Instructions: RV32I and RV64I					RV Privileged Instructions						
Category	Name	Fmt	RV32I Base		+RV64I		Category	Name	Fmt	RV mnemonic	
Shifts	Shift Left Logical	R	SLL	rd,rs1,rs2	SLLW	rd,rs1,rs2	Trap	Mach-mode trap return	R	MRET	
	Shift Left Log. Imm.	I	SLLI	rd,rs1,shamt	SLLIW	rd,rs1,shamt		Supervisor-mode trap return	R	SRET	
	Shift Right Logical	R	SRL	rd,rs1,rs2	SRLW	rd,rs1,rs2	Interrupt	Wait for Interrupt	R	WFI	
	Shift Right Log. Imm.	I	SRLI	rd,rs1,shamt	SRLIW	rd,rs1,shamt		MMU Virtual Memory FENCE	R	SFENCE.VMA rs1,rs2	
	Shift Right Arithmetic	R	SRA	rd,rs1,rs2	SRAW	rd,rs1,rs2	Examples of the 60 RV Pseudoinstructions				
	Shift Right Arith. Imm.	I	SRAI	rd,rs1,shamt	SRAIW	rd,rs1,shamt					
Arithmetic	ADD	R	ADD	rd,rs1,rs2	ADDW	rd,rs1,rs2	Branch = 0 (BEQ rs,x0,imm)		J	BEQZ rs,imm	
	ADD Immediate	I	ADDI	rd,rs1,imm	ADDIW	rd,rs1,imm	Jump (uses JAL x0,imm)		J	J imm	
	SUBtract	R	SUB	rd,rs1,rs2	SUBW	rd,rs1,rs2	MoVe (uses ADDI rd,rs,0)		R	MV rd,rs	
	Load Upper Imm	U	LUI	rd,imm	Optional Compressed (16-bit) Instruction Extension: RV32C		RETurn (uses JALR x0,0,ra)		I	RET	
Add Upper Imm to PC	U	AUIPC	rd,imm	Category			Name	Fmt	RVC	RISC-V equivalent	
Logical	XOR	R	XOR	rd,rs1,rs2	Loads	Load Word	CL	C.LW	rd',rs1',imm	LW	rd',rs1',imm*4
	XOR Immediate	I	XORI	rd,rs1,imm		Load Word SP	CI	C.LWSP	rd,imm	LW	rd,sp,imm*4
	OR	R	OR	rd,rs1,rs2		Float Load Word SP	CL	C.FLW	rd',rs1',imm	FLW	rd',rs1',imm*8
	OR Immediate	I	ORI	rd,rs1,imm		Float Load Word	CI	C.FLWSP	rd,imm	FLW	rd,sp,imm*8

<http://www.riscvbook.com/>

Sequenz von Befehlen

```
int foo(int x, int y, int z)
x in x2, y in x3, z in x4
foo:
add x5, x2, x3      # t = x+y
sub x6, x5, x4      # r = t-z
add x6, x6, x2      # r = r+x
add x6, x6, x3      # r = r+y
```

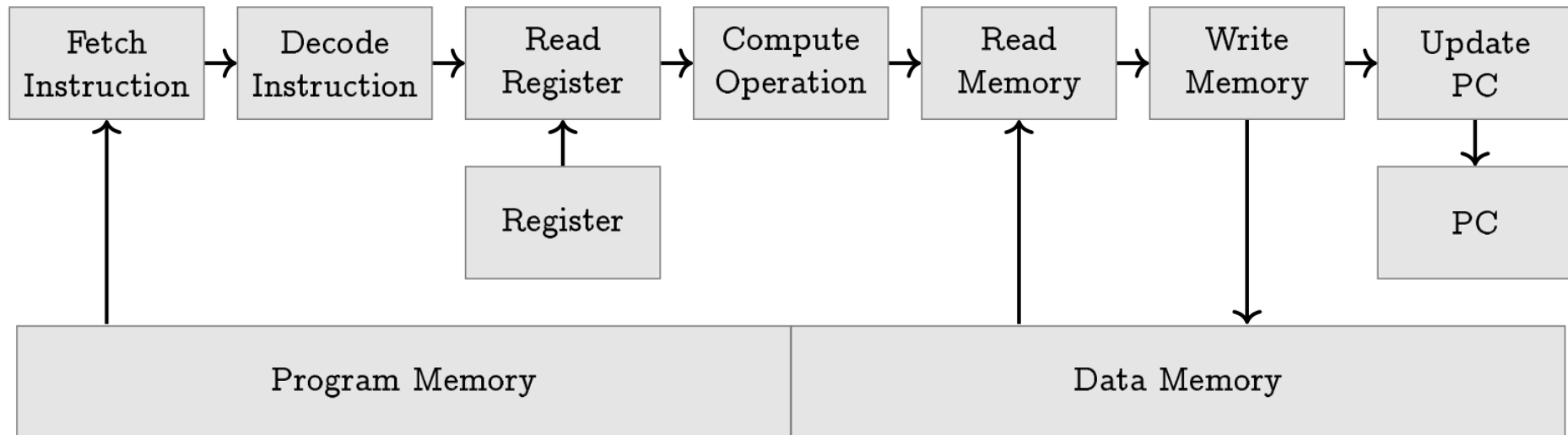
Befehlsausführung

n Anzahl Befehle

$k = 7$ Teilaufgaben einer Befehlsausführung

1. Befehl aus Speicher laden
2. Befehlstyp feststellen
3. Lade Operanden für Befehl
4. Führe Operation mit Operanden aus
5. Lade von Speicher, falls erforderlich
6. Schreibe Ergebnis in Speicher
7. Setze den Befehlszähler

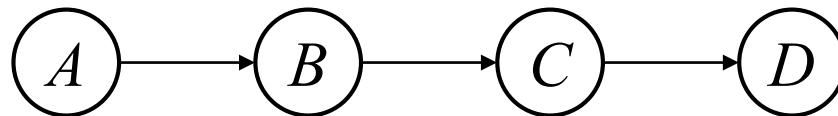
Hardware zur Befehlsausführung



© H.M. Bucker

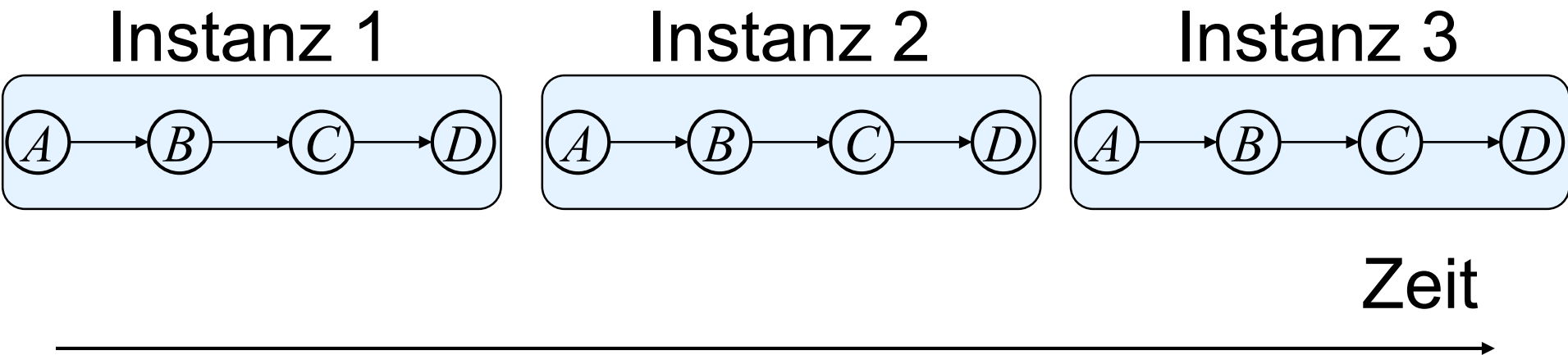
Pipelining (Teil 1)

- Sei $G = (V, E)$ ein gerichteter Graph.
- Knoten $v \in V$ repräsentiert Teilaufgabe.
- Kante $u \rightarrow v \in E$ repräsentiert Abhängigkeit: Teilaufgabe u muss beendet sein, bevor Teilaufgabe v begonnen werden kann.



Pipelining (Teil 2)

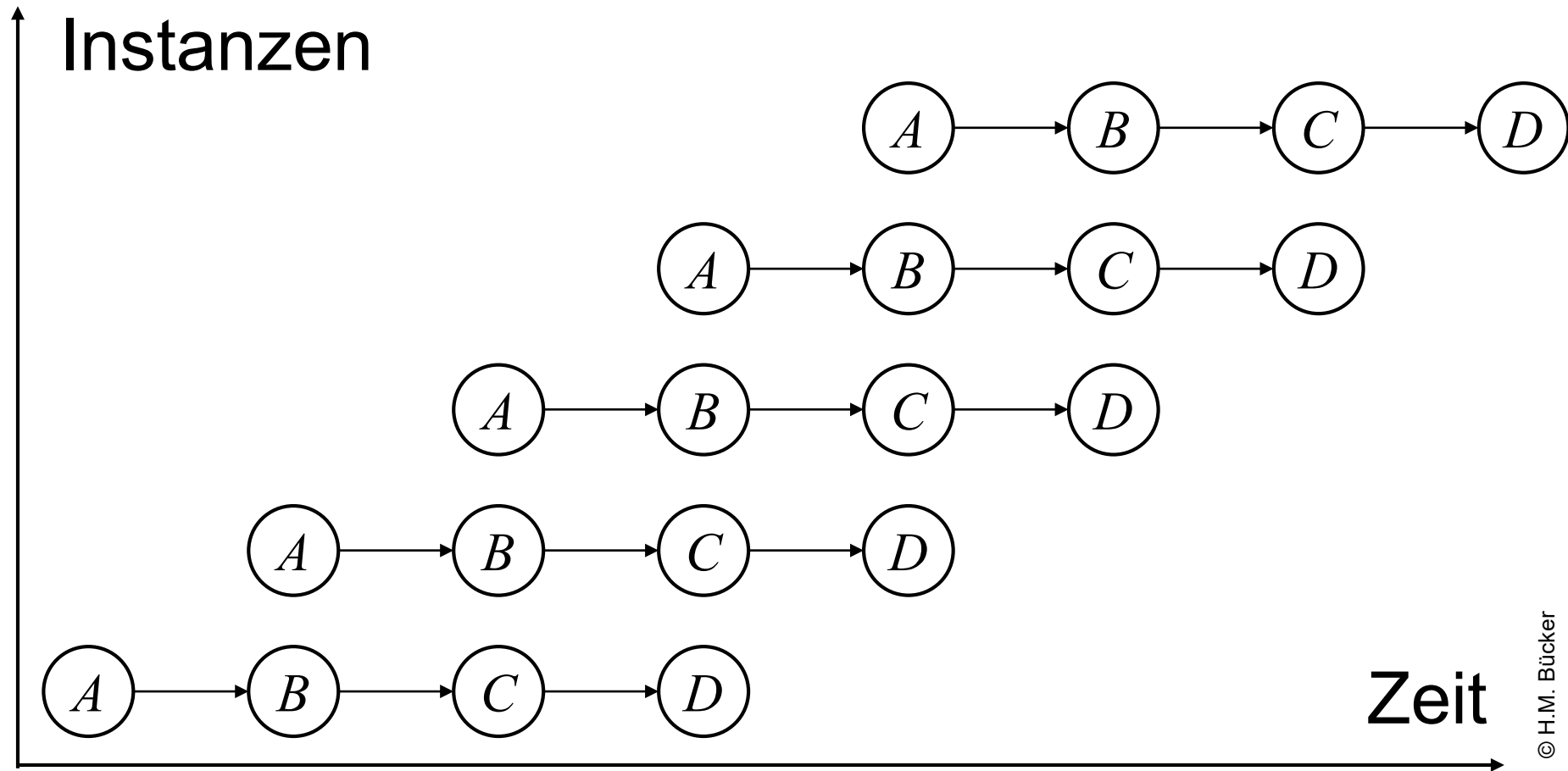
- Ohne Pipelining



Pipelining (Teil 3)

- n Instanzen eines sequentiellen G mit k Teilaufgaben müssen abgearbeitet werden.
- Alle Teilaufgaben benötigen die gleiche Zeit t zur Ausführung.
- Unterschiedliche Teilaufgaben von unterschiedlichen Instanzen können parallel ausgeführt werden.
- Die gleiche Teilaufgabe von unterschiedlichen Instanzen können nicht parallel ausgeführt werden.

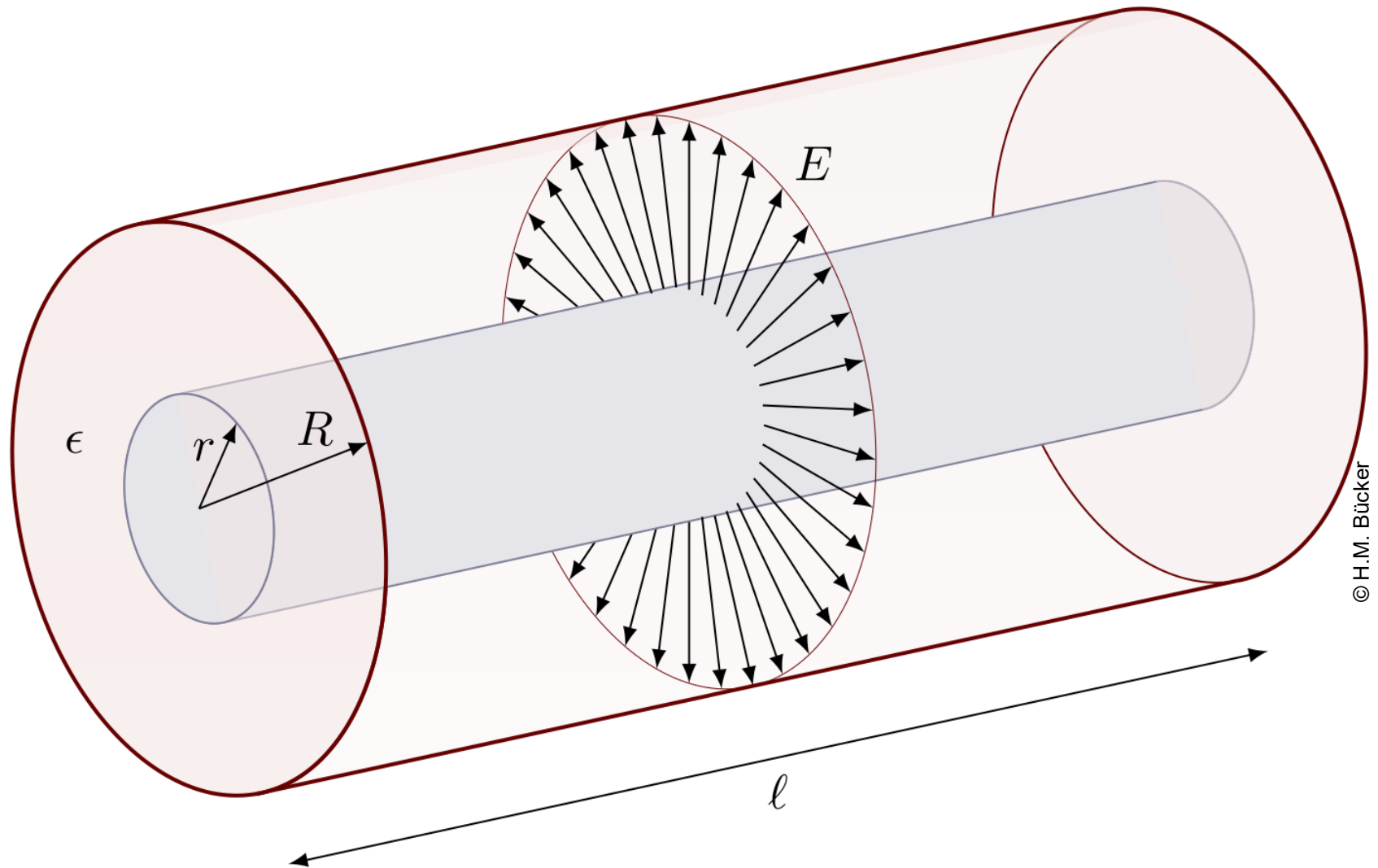
Pipelining (Teil 4)



Pipelining (Teil 5)

- Zeit für n Instanzen ohne Pipelining:
- Zeit für n Instanzen mit Pipelining:
- Beschleunigung:

Zylinderkondensator



Koaxialkabel im Querschnitt



[Руслан Измайлов via Wikimedia Commons](#)

Zylinderkoordinaten

$$x = r \cos \varphi$$

$$y = r \sin \varphi$$

$$z = z$$

$$\frac{\partial(x, y, z)}{\partial(r, \varphi, z)} = \begin{bmatrix} \cos \varphi & -r \sin \varphi & 0 \\ \sin \varphi & r \cos \varphi & 0 \\ 0 & 0 & 1 \end{bmatrix}$$

$$\left| \det \left(\frac{\partial(x, y, z)}{\partial(r, \varphi, z)} \right) \right| = r$$

$$dV = r \, dr \, d\varphi \, dz$$

Kugelkoordinaten

$$x = r \sin \theta \cos \varphi$$

$$y = r \sin \theta \sin \varphi$$

$$z = r \cos \theta$$

$$\frac{\partial(x, y, z)}{\partial(r, \theta, \varphi)} = \begin{bmatrix} \sin \theta \cos \varphi & r \cos \theta \cos \varphi & -r \sin \theta \sin \varphi \\ \sin \theta \sin \varphi & r \cos \theta \sin \varphi & r \sin \theta \cos \varphi \\ \cos \theta & -r \sin \theta & 0 \end{bmatrix}$$

$$\left| \det \left(\frac{\partial(x, y, z)}{\partial(r, \theta, \varphi)} \right) \right| = r^2 \sin \theta$$

$$dV = r^2 \sin \theta \, dr \, d\theta \, d\varphi$$

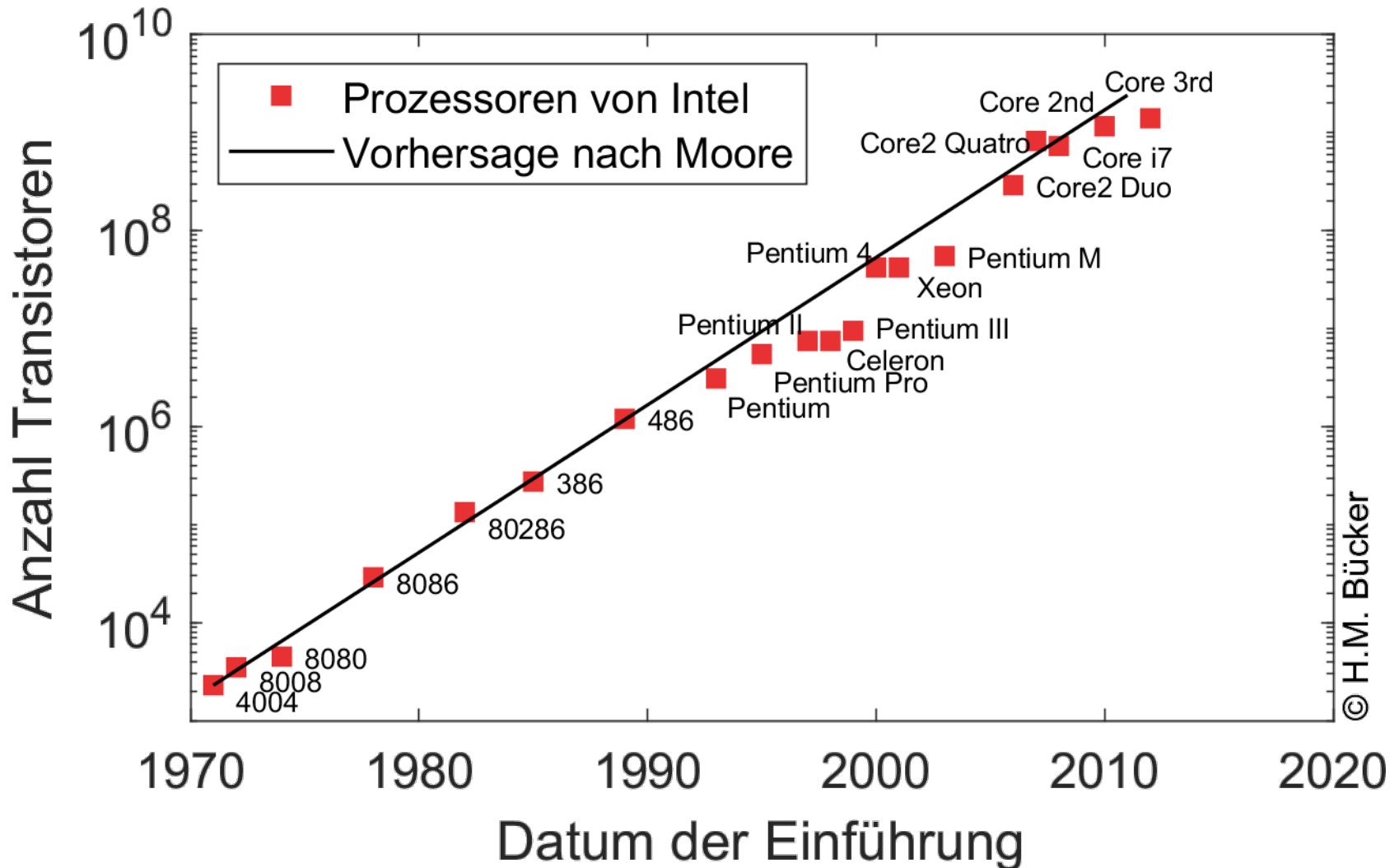
Miniaturisierung

- Kapazität eines Zylinderkondensators ist proportional zu seiner Länge:

$$C = \frac{Q}{U} = 2\pi\epsilon\ell \bigg/ \ln \frac{r_2}{r_1}$$

- Zylinderkondensator ist einfaches Modell, das in der Praxis nicht verwendet wird. Allerdings sind Eigenschaften realer Komponenten ebenso abhängig von der Länge.

Anzahl der Transistoren



Taktfrequenz

