

## Prozessoren und Architekturen

### Begriffe und Abkürzungen aus der Welt der Industrierechner und Embedded Systeme - Teil 1

#### Liebe Leserinnen und Leser,

in unserer Branche werden viele Begriffe und Abkürzungen verwendet, deren Bedeutung durchaus unterschiedlich interpretiert werden kann. Um etwas Klarheit in diesen Dschungel zu bringen, werden Tom Weber von roboblogs und Helmut Artmeier, Geschäftsführer der EFCO Electronics GmbH, ab dieser Ausgabe von PC&Industrie immer wieder Begriffe erklären.

Das Glossar wird in loser Folge weitergeführt und neben der print/ePaper-Ausgabe auch einen festen Platz auf unserer homepage erhalten.

Ab sofort finden Sie also unter <https://www.beam-verlag.de/glossar/> eine pdf-Datei, die vorlaufend ergänzt wird. In dieser können Sie blättern, suchen oder sie kostenlos downloaden!

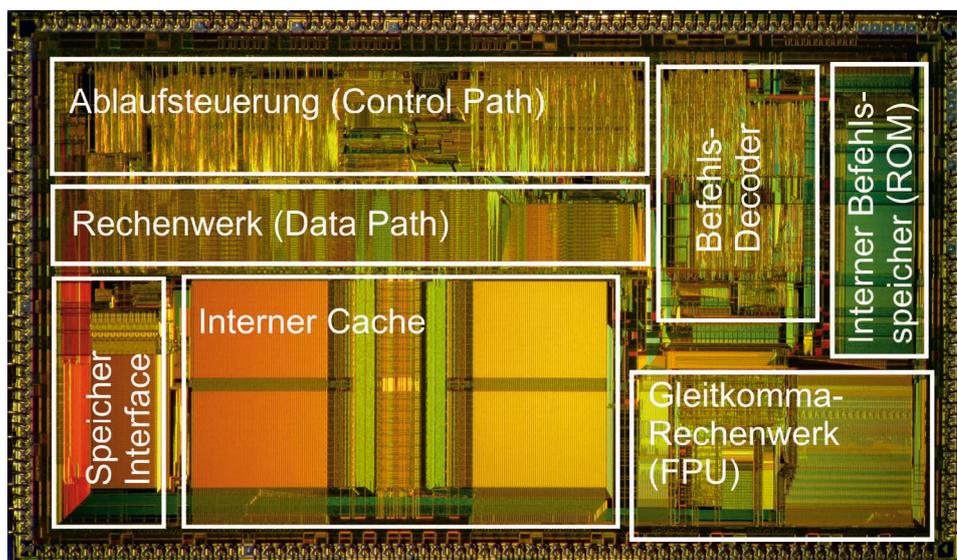
**x86** Die Bezeichnung x86 oder x86-Architektur wird in aller Regel synonym für die Prozessoren der Hersteller Intel und AMD verwendet. Der Urahn ist der 16-bit-Prozessor 8086, der 1978 vorgestellt wurde. Seit 1985 verwenden die Prozessoren eine 32-bit-Architektur; die 64-bit-Variante wurde 2005 eingeführt.

Lange Zeit war x86 die weltweit am meisten verbreitete Befehlssatzarchitektur; auf ihr basieren alle PC-Computer. x86 gilt als sehr leistungsfähig - aber nicht besonders energieeffizient. Seit etwa der Jahrtausendwende haben die ARM-Prozessoren mit RISC-Architektur stark aufgeholt. Sie benötigen deutlich weniger Leistung, so dass energie-sensible Anwendungen heute praktisch vollständig zu ARM übergewechselt sind. Allerdings sind direkte Vergleiche nicht einfach, da x86- und ARM sehr unterschiedliche Befehlssätze verwenden. So sind etwa x86-Prozessoren bei Gleitkomma-Operationen in aller Regel effizienter.

Die x86-Architektur erlaubt es, den Speicher in einzelne Segmente aufzuteilen. Der Vorteil: Programme können nur auf ein bestimmtes Speichersegment zugreifen; es gibt keine Auswirkungen in einem anderen Speichersegment. Zudem können die Prozessoren in unterschiedlichen Betriebsarten arbeiten.

Moderne x86-Prozessoren nutzen mehrere Rechenkerne auf dem gleichen Chip - sogenannte Cores - und können damit mehrere Aufgaben gleichzeitig ausführen. Dies erhöht die Leistung des Gesamtsystems. Durch das gezielte Zu- und Abschalten und die Steuerung der Taktfrequenz der einzelnen Cores kann der Energiebedarf eines x86-Prozessors deutlich reduziert werden.

Ein großer Vorteil der x86-Architektur ist ihre Abwärtskompatibilität: Anwendungen und Betriebssysteme, die für ältere Prozessorgenerationen geschrieben wurden, laufen in aller Regel ohne größere Probleme auf neueren Prozessoren.



*Foto des Chips eines Intel 80486 DX2-66 Prozessors. Der Autor hat das ursprüngliche Bild aus dem Jahr 2013 um die Zuordnung der Funktionen ergänzt. © Wikipedia/Pauli Rautakorpi*

---

**ARM** ARM ist eine RISC-basierte Prozessor-Architektur, die ursprünglich vom britischen Unternehmen Acorn entwickelt wurde. Seit 1990 wird sie von dessen Tochterunternehmen Acorn Risc Machines lizenziert. Im Gegensatz zu Intel stellt ARM selbst keine Chips her. Anwender erwerben Lizenzen für das Nutzen der Entwicklungen und des geistigen Eigentums von ARM. Neben diesem Lizenzmodell bietet die ARM-Architektur einige spezifische Vorteile. Daher sind ARM-basierte Mikroprozessoren und Mikrocontroller heute weltweit verbreitet. ARM setzte von Anfang an auf 32-bit-Prozessoren; ab etwa 2013 gab es die ersten 64-bit-Ausführungen.

Bekannte Lizenznehmer von ARM sind etwa AMD, Apple, Cypress, Google, IBM, Infineon, Intel, Microsoft, Qualcomm, Texas Instruments, NXP, Mediatek oder Microchip.

Einer der Hauptgründe für die Verbreitung dieser Architektur ist der vergleichsweise geringe Energiebedarf bei hoher Leistungsfähigkeit. Entsprechend basieren nahezu alle Smartphones und Tablet-Computer auf ARM-Prozessoren, so etwa das iPhone oder die meisten Android-Geräte. Zudem ist die ARM-Architektur deutlich besser skalierbar als x86 und kann daher recht einfach an unterschiedlichste Anforderungen angepasst werden, von einfachen Mikrocontrollern bis hin zu leistungsfähigen Servern.

Sowohl ARM als auch x86 unterstützen mehrere Rechenkerne auf dem gleichen Chip (Multicore) und bieten Potenzial für weitere Entwicklungen. Wenn allerdings - wie etwa bei mobilen Geräten - ein optimales Verhältnis von Leistung und Energieeffizienz im Vordergrund steht, bieten ARM-basierte Designs bessere Möglichkeiten.

---

**RISC** Die Abkürzung bedeutet Reduced Instruction Set Computer - also Rechner mit reduziertem Befehlssatz. Diese Mikroprozessor-Design-Philosophie wurde 1980 von David A. Patterson und Carlo H. Séquin entwickelt. Ihnen war aufgefallen, dass die Befehlssätze der Mikroprozessoren ihrer Zeit immer komplexer geworden waren, um die Arbeit der Programmierer zu erleichtern und zu beschleunigen. Aus dieser Erkenntnis entwickelten sie einen radikal anderen Ansatz: Ein überschaubarer Satz einfach zu dekodierender und schnell ausführbarer Befehle. Häufig ist die Befehlsausführung in einem Takt möglich, womit RISC-Prozessoren erheblich höhere Befehl-per-Takt-Raten erzielen, als x86-Architekturen.

Heute hat diese Unterscheidung in reduzierte oder komplexe Befehlssätze weitgehend an Bedeutung verloren; moderne RISC-Prozessorfamilien nutzen ebenfalls komplexe Befehlssätze. Andererseits wurden RISC-Komponenten auch in das Design anderer Mikroprozessoren übernommen, um diese besser skalierbar zu machen.

Mit zu dieser Entwicklung beigetragen haben die Fortschritte in der Halbleitertechnik. Das Dekodieren von Befehlen nimmt heute bei allen Architekturen nur noch einen kleinen Teil der Chipfläche ein.

---

**RISC-V** RISC-V (gesprochen „risc five“) ist ein offener RISC-Standard, welcher der BSD-Lizenz unterliegt. Entsprechend ist RISC-V nicht patentiert oder durch Copyright geschützt und darf frei verwendet werden. Im Prinzip ist es also jedem erlaubt, RISC-V Mikroprozessoren zu entwerfen, herzustellen, weiterzuentwickeln und zu verkaufen. Entsprechend bieten zahlreiche Unternehmen RISC-V-basierte Hardware an.

Der RISC-V Befehlssatz wurde so ausgelegt, dass er für eine Vielzahl von Anwendungen geeignet ist, die in aller Regel jeweils nur einen Teil der Möglichkeiten nutzen. Die Architektur bietet eine variable Datenwortbreite bis 128 bit. Eingesetzt werden RISC-V-basierte Chips allgemein in Embedded-Systemen, aber auch in PCs oder Hochleistungsrechnern.

---

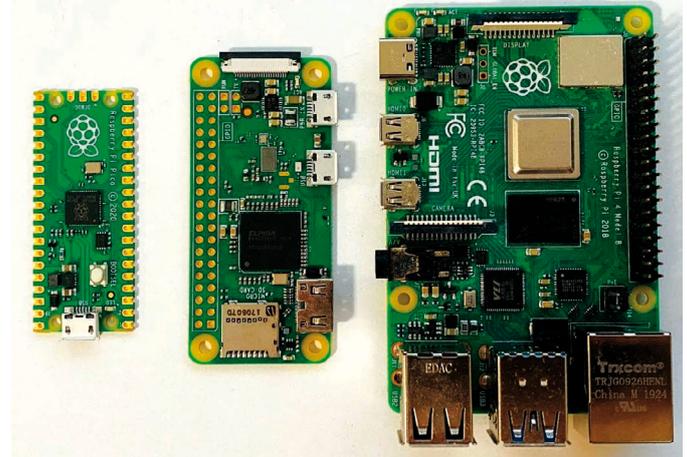
**OpenRISC** Das OpenRISC-Projekt greift die Philosophie freier Hardware auf. Das Ziel dabei: Eine frei verfügbare CPU zu entwickeln, auf der Linux als Betriebssystem läuft. Bisher verfügbar ist die OpenRISC-1000-Familie (kurz OR1k), sowohl als 32- als auch als 64-Bit-Version. Das Design wurde in der Hardwarebeschreibungssprache Verilog unter der GNU Lesser General Public License (GNU LGPL) veröffentlicht. Firmware und Microcode zum Prozessor wurden unter der GNU General Public License (GNU GPL) bereitgestellt.

Der Unterschied zu RISC-V liegt im komplexeren Befehlssatz - und der fehlenden skalierbaren Datenwortbreite. Diese ist bei OpenRISC entweder auf 32 oder 64 bit begrenzt.

## Energie-Effizienz

Vergleich x86 und RISC bei einer Aufgabe aus dem Bereich Machine Learning. Während der spartanisch ausgestattete Raspberry Pi Pico die geringste Leistungsaufnahme bietet, benötigt er um Größenordnungen mehr Zeit, um die gestellte Aufgabe abzuarbeiten.

Der ARM Cortex M72 im Raspberry Pi 4B hingegen benötigt lediglich 3 x so viel Zeit wie der Desktop-PC - bei weniger als 5 % dessen Stromverbrauch.



**Größenvergleich Raspberry Pico (links) und Raspberry 4 B (rechts)**  
© TH Deggendorf

System	Raspberry Pi Pico	Raspberry Pi 4B	Desktop PC
CPU	ARM Cortex M0+	ARM Cortex M72	Intel i7-4770
Takt [GHz]	0,133	1,5	3,4
Kerne	2	4	8
RAM [GB]	0,264	4	16
Leistungsaufnahme [W]	0,5	2,5	60
Rechenzeit insgesamt [µs]	392.930	0,12	0,04
Rechenzeit pro sample [ns]	20.390	0,06	0,02

Quelle: TH Deggendorf

## Power Architecture

Die gemeinsame Entwicklung von IBM und Freescale (früher Motorola) ist heute die am weitesten verbreitete RSC-Architektur im oberen Leistungssegment. Typische Einsatzbereiche sind Drucker oder Router, Workstations bis hin zu Supercomputern.

## GNU GPL

Die GNU General Public License (kurz GNU GPL oder GPL; aus dem Englischen wörtlich für allgemeine Veröffentlichungserlaubnis oder -genehmigung) ist eine Lizenz, welche Nutzern sehr weitreichende Rechte einräumt: Produkte und Software unter GNU GPL dürfen ausgeführt, analysiert, verändert und verbreitet werden (kopieren).

GPL bedeutet aber auch, dass der Urheber von Veränderungen alle Rechte weitergeben muss, auch das Recht, den Code zu analysieren, zu verändern und weiter zu verbreiten. Aus Software, die unter der GNU GPL Lizenz steht, kann also niemals ein proprietäres Produkt entstehen. Insofern umfasst GNU GPL auch das Copyleft.

## GNU LGPL

Das erste "L" in der GNU Lesser General Public License (kurz GNU LGPL oder LGPL; aus dem Englischen wörtlich für weniger allgemeine Veröffentlichungserlaubnis oder -genehmigung) hat entscheidende Bedeutung für Software-Entwickler: LGPL erlaubt das Verwenden und Einbinden Software in eigene (sogar proprietäre) Software und zwingt nicht zu der für die kommerzielle Verwertung entscheidenden Copyleft-Anforderung, nämlich den Quellcode der eigenen Software-Teile offenzulegen. Damit steht LPGL irgendwo zwischen GPL und BSD.

In der Praxis werden die Teile der Software, die unter der LGPL stehen, meist in Form dynamischer Programm-Bibliotheken (z. B. DLL) eingebunden, um auf diese Weise die erforderliche Trennung zwischen proprietären und quelloffenen LGPL-Teilen zu ermöglichen. Für Anwender ist in diesem Zusammenhang wichtig, dass ausschließlich die LGPL-Teile das Recht zur Modifizierung einräumen, nicht aber etwaige proprietäre Software-Teile.