

HP PRECISION ARCHITECTURE

EINE NEUE PERSPEKTIVE



 HEWLETT
PACKARD

HP PRECISION ARCHITECTURE

EINE NEUE PERSPEKTIVE



Copyright © 1986 Hewlett-Packard Company.

Alle Rechte vorbehalten. Diese Broschüre darf ohne schriftliche Zustimmung von Hewlett-Packard weder kopiert noch übersetzt werden.

Inhaltsverzeichnis

Vorwort	Zu dieser Einführung	iv
Kapitel 1:	Einleitung	v
	Der Befehlssatz	1
	Computer mit komplexem Befehlssatz	2
	Der Beitrag der Wissenschaft	3
	Computer mit vereinfachtem Befehlssatz	4
	Ein völlig neuer Ansatz	4
Kapitel 2:	Die bisherige Entwicklung der Rechnerarchitekturen	5
	Die ersten elektronischen Rechner (1946-1962)	5
	Die ersten Computer mit Mikroprogrammierung (1963-1969)	7
	Mikroprogrammierte Computer mit Cache-Speicher (1970-1984)	12
	Computer mit vereinfachtem Befehlssatz (1985-....)	15
Kapitel 3:	Die neue HP Precision Architecture	17
	Die Grundzüge des RISC-Konzepts	17
	Zusätzliche Elemente runden das RISC-Konzept ab	20
	Alles fügt sich zusammen	23
	Eine zukunftsweisende Architektur	23
Glossar	27

Vorwort

Wie wählen Sie ein Computersystem aus? Woher wissen Sie, ob es flexibel genug ist, um auch wachsenden Anforderungen gerecht zu werden?

Sie haben ein Problem zu bewältigen und suchen nach brauchbaren Lösungen, von denen Sie, als anspruchsvolle Führungskraft, auch ein günstiges Preis-Leistungs-Verhältnis erwarten. Sie brauchen Kompatibilität bei Hardware und Software und suchen ein System, das nicht nur Ihren heutigen Ansprüchen genügt, sondern auch flexibel genug ist, sich zukünftigen Anforderungen anzupassen zu können.

Sie haben sich viel vorgenommen. Denn es ist nicht leicht, ein System oder eine Systemfamilie zu finden, die allen diesen Anforderungen genügt. Sie wissen, daß Ihre Firma neue Wege einschlägt und Ihre Abteilung bestens auf die neuen Probleme vorbereitet sein muß, und daß in der heutigen, vom starken Wettbewerb geprägten Geschäftswelt die Auswahl eines unzulänglichen Computersystems unabsehbare Folgen haben kann.

Die Frage lautet also: an welchen Maßstäben soll das neue System gemessen werden? Wie können Sie sicher sein, daß es mit Ihren steigenden Ansprüchen mitwachsen wird? Wie lange werden Sie mit dem Produkt, das Sie heute anschaffen, auch in Zukunft rechnen können? Und wie steht es mit der Hardware- bzw. Software-Kompatibilität? Wer sagt Ihnen,

daß ein System, das Sie heute kaufen, auch die Programme von morgen bewältigen wird? Kurz, wie können Sie sicher sein, daß der Aufwand an Mühe, Zeit und Geld, den Sie in ein Computersystem investieren, Ihnen auch den erwarteten Nutzen bringen wird? Diese Fragen sind leichter zu beantworten, wenn man die sogenannte „Architektur“ eines Computersystems selbst versteht und beurteilen kann.

Mit der Architektur eines Computers wird die Struktur des Rechners, wird Aufbau und Funktion der Maschine festgelegt. Wenn Sie einen Computer anschaffen, entscheiden Sie immer für eine Architektur. Die Architektur stellt gewissermaßen die Seele des Rechners dar.

Lange Zeit konnten die Computerwissenschaftler über den Zusammenhang zwischen der Architektur eines Rechners und seiner Leistung nur vage Vermutungen anstellen. Der Rechnerentwurf entbehrt praktisch jeder verlässlichen theoretischen Grundlage. Sogenannte Innovationen kamen nur dadurch zustande, daß bereits vorhandene Architekturen etwas abgeändert wurden. Da man mit dieser Art von Neuerungen anscheinend zufrieden war, wurden sie als selbstverständlich hingenommen

und blieben auch dann noch im Einsatz, wenn sie längst nicht mehr zweckmäßig waren.

Diese Situation hat sich jedoch geändert. Heute weiß man mehr über die Arbeitsweise von Computern und über die Auswirkungen der Rechnerarchitektur auf die Informationsverarbeitung. Die Forscher an den Hochschulen und in der Industrie verfügen nun über Erkenntnisse, auf die sie ihre Entscheidungen beim Entwurf eines neuen Computers gründen können.

Das Ergebnis dieser Forschungsarbeit bei Hewlett-Packard ist das Konzept der neuen HP Precision Architecture. Sie wurde entwickelt, um ganz bestimmte Ziele zu erreichen. Die Entscheidungen, die zu diesem Konzept geführt haben, waren bei HP keine Sache des Zufalls, sondern eines grundlegenden Verständnisses der Arbeitsweise des Computers. Das Ergebnis ist eine völlig neue Rechnerarchitektur, die Maßstäbe für Wirtschaftlichkeit und Leistung setzt, und die wegen ihrer Flexibilität geeignet ist, den wechselnden Anforderungen der Datenverarbeitung heute und in Zukunft gerecht werden wird.

Zu dieser Einführung

Sie richtet sich an alle, deren Aufgabe es ist, Computersysteme zu bewerten oder anzuschaffen. Sie verschafft Ihnen einen Überblick über Rechnerarchitektur im allgemeinen und über die HP Precision Architecture im besonderen.

- In Kapitel 1 werden zwei Grundkonzepte des Rechnerentwurfs gegenübergestellt: das Prinzip des Rechners mit komplexem Befehlssatz (CISC=Complex Instruction Set Computer) und das des Rechners mit vereinfachtem Befehlssatz (RISC=Reduced Instruction Set Computer).
- Kapitel 2 liefert einen geschichtlichen Abriss der Entwicklung in der Rechnerarchitektur. Es beschreibt die Fortschritte in der Hardware- und Software-Technik und die Konsequenzen die sich daraus für die Rechnerarchitektur ergaben.
- Kapitel 3 führt Sie in die Grundlagen der neuen HP Precision Architecture ein. Es beschreibt die wichtigsten Merkmale der Architektur und erklärt, warum gerade diese Computer in ihrer Preisklasse Spitzenleistung bieten.

Das Ergebnis der Forschungsarbeit ist eine Architektur, die neue Standards hinsichtlich Effizienz und Leistungsfähigkeit steht.

Technische Neuerungen haben viele Ursachen. Oft sind sie das Ergebnis gezielter Überlegungen, beispielsweise wenn sich ein Erfinder anschickt eine bessere Mausefalle zu konstruieren. Manchmal sind sie nur das Ergebnis eines Zufalls. So zum Beispiel, wenn ein Erfinder während seiner Arbeit auf etwas äußerst Interessantes stößt, das im Grunde aber ganz und gar nichts mit seiner eigentlichen Aufgabenstellung zu tun hat. Große Entdeckungen werden oft dadurch gemacht, daß Wissenschaftler einer ganz bestimmten Idee nachjagen und beiläufig auf etwas ganz anderes stoßen.

In der noch jungen Geschichte der Computerwissenschaft sind die interessantesten Neuerungen dadurch entstanden, daß sich die Ingenieure bemühten, zwei zentrale Probleme zu lösen: die Entwicklung einer Maschine, die Informationen schneller verarbeiten kann, und die einer Maschine, die in Sprachen programmiert werden kann, die der natürlichen Sprache möglichst ähnlich sind.

Neuerungen auf dem Gebiet der Computertechnik zeigen sich bei drei wichtigen Bereichen: bei der Konstruktion der Computer eingesetzten Technologie, den bereitgestellten Programmiersprachen und bei der eigentlichen Konzeption der Maschinen. Als die Schaltkreis-

technik sich entwickelte und sich höhere Programmiersprachen immer mehr durchsetzten, änderte sich, den neuen Möglichkeiten im Bereich Hardware und Software entsprechend, auch die Art und Weise, in der Maschinen Informationen verarbeiten.

Die ersten elektronischen Rechner hatten einen einfachen und übersichtlichen Aufbau. Das war hauptsächlich auf den niedrigen Entwicklungsstand der Technik zurückzuführen, die in den Maschinen zur Anwendung kam. In den letzten 30 Jahren führten jedoch immer neue technische Errungenschaften und Erkenntnisse zu einem immer komplexeren Rechnerkonzept.

Wegen des niedrigen technischen Entwicklungsstandards waren die ersten elektronischen Rechner einfach und übersichtlich.

Der Befehlssatz

Die Entwicklung zu immer komplexeren Rechnern spiegelt sich in der Entwicklung der in diesen Rechnern verwendeten Befehlssätzen wieder. Der Befehlssatz oder Befehlsvorrat ist für den Aufbau eines Rechners von so grundlegender Bedeutung, daß er oft in der Hierarchie mit der Rechnerarchitektur gleichgestellt wird.

Der Befehlssatz bestimmt die Grundfunktionen, die ein Rechner für eine nutzvolle Arbeit benötigt. Die Maschinenbefehle lassen sich mit den Tasten eines Taschenrech-

ners vergleichen. Ein Taschenrechner hat Tasten für die Addition, Subtraktion, Multiplikation und Division, aber auch für die Prozent- und Wurzelrechnung. Darüber hinaus sind viele Taschenrechner mit Tasten für spezielle Anwendungszwecke ausgestattet. Finanzrechner verfügen zum Beispiel über Funktionen für die Berechnung des Nettoarwerts oder der Abschreibung. Aus diesen Beispielen können Sie erahnen, wie durch die Bestimmung des Befehlsatzes eines Rechners die Operationen festgelegt werden, die er durchführen kann.

Die Entwicklung der Computer zeichnet sich durch immer komplexe Befehlssätze aus; die Befehle wurden immer zahlreicher und die einzelnen Befehle immer spezieller. Während die ersten Rechner mit einem Befehlssatz von nicht mehr als sieben Befehlen auskamen, arbeiten Rechner heute oft mit mehreren hundert Befehlen.

Computer mit komplexem Befehlssatz

Immer mehr Computerwissenschaftler setzten auf den Einsatz von komplexen Befehlssätzen, so daß in den letzten zwanzig Jahren fast ausschließlich Computer mit komplexem Befehlssatz (CISC) konzipiert und hergestellt wurden.

Mit zunehmendem Umfang des Befehlssatzes wurde es jedoch für die Compiler, die ja die Programme aus den höheren Programmiersprachen in die MaschinenSprache übersetzen müssen, immer schwieriger, die ganz speziellen Befehle nutzbringend einzusetzen. Stellen Sie sich einen Taschenrechner mit 300 verschiedenen Tasten vor, und Sie sehen sofort, wo das Problem liegt. Ein solcher Taschenrechner würde natürlich Tasten für die verschiedensten Sonderfunktionen bieten, nur hätten Sie wahrscheinlich große Schwierigkeiten,

die gerade benötigte Taste zu finden. Selbst um einfachste Rechenfunktionen auszuführen, müßten Sie wahrscheinlich lange suchen. Das gleiche Problem stellt sich für einen Computer, wenn er mit einem immer größeren Befehlsvorrat arbeiten muß. Dabei ist die Suche des richtigen Befehls weniger problematisch als die Entscheidung, welcher Befehl nun genau der richtige ist. Und bei so vielen speziellen Befehlen ist der genau passende Befehl oft schwer zu finden.

CISC-Computer arbeiten zwar mit komplexen Befehlen, das heißt aber noch lange nicht, daß sie sie auch sofort verstehen. Die komplexen Befehle müssen vielmehr zuerst in kleinere Befehle zerlegt werden, bevor sie von der Maschine ausgeführt werden können.

Eigentlich arbeitet der Rechner auch nicht anders als wir Menschen. Jede komplexe mathematische Formel, wie zum Beispiel die zur Berechnung des Nettoarwerts, läßt sich im allgemeinen in eine Folge von Multiplikationen und Divisionen zerlegen. Um also eine derart komplizierte Formel berechnen zu können, müßten wir viele Einzelberechnungen nacheinander durchführen.

Bei der Ausführung eines Befehls geht der Computer ähnlich vor. Er unterteilt den auszuführenden Befehl in kleinere, für ihn leichter verständliche Einheiten. Zunächst war dieser Decodervorgang, nämlich die Unterteilung komplexer Befehle (Makrocode) in kleinere, unmittelbar ausführbare Befehle (Microcode) eine optimale Lösung. Die Anzahl der für die Ausführung einer Operation bzw. eines Befehls notwendigen Maschinenzyklen wurde zwar erhöht, jedoch ohne nachteilige Auswirkungen auf die Rechnerleistung, denn die Zentraleinheit (CPU), die die Befehle decodieren mußte, war viel schneller als der Hauptspeicher, in dem die Befehle und Daten gespeichert wurden. Während der langsamere Hauptspeicher die nächsten Daten besorgte, hatte die wesentlich schnellere Zentraleinheit genügend Zeit, einen Befehl zu decodieren.

Der Beitrag der Wissenschaft

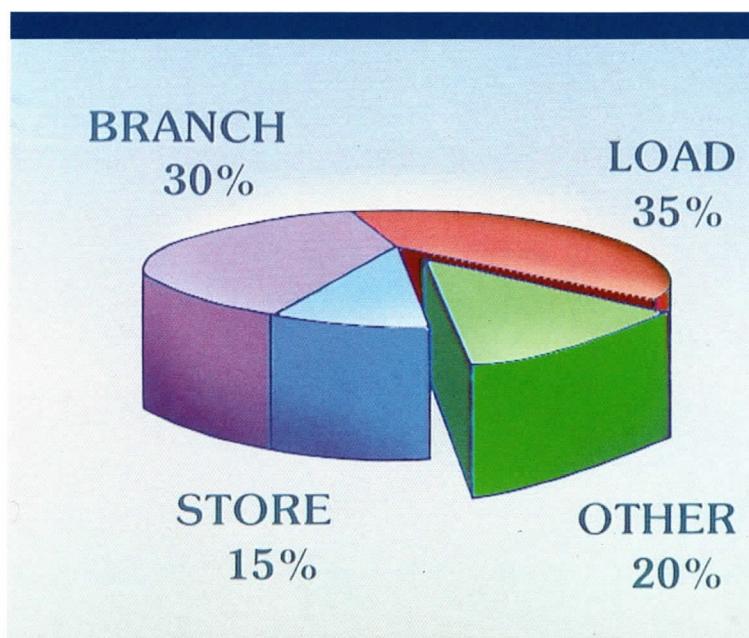
Bis vor kurzem wußte man noch sehr wenig über den Zusammenhang zwischen dem Befehlssatz und der Rechnerleistung. Ende der siebziger und Anfang der achtziger Jahre begannen die Wissenschaft-

ler, sich ein umfassenderes Bild von den Vorgängen zu verschaffen, die während der Ausführung eines Programms in einem Rechner abliefen. Die Ergebnisse überraschten die Fachwelt.

Es zeigte sich nämlich, daß komplexe Befehle nur sehr selten verwendet werden. Am häufigsten wurden einfache Befehle, zum Beispiel Lade-, Speicher- und Verzweigungsbefehle, ausgeführt. Nur zwanzig Prozent des verfügbaren Befehlssatzes belegten wurden in achtzig Prozent der Zeit aufgeführt. Dieses unerwartete Überwiegen der einfachen Befehle stellte die bisher geltenden, grundlegenden Annahmen der Rechnerkonstruktion auf den Kopf und eröffnete der Computerwissenschaft völlig neue Perspektiven.

Die Forschung zeigte, daß komplexe Befehle nur selten verwendet werden.

Einfache Befehle werden am häufigsten verwendet.



Entscheidungen über den Rechnerentwurf brauchen nicht mehr ad hoc und aufgrund wie Annahmen getroffen zu werden. Heute trifft man Entscheidungen aufgrund von Messergebnissen über die Effizienz der Informationsverarbeitung im Rechner.

Computer mit vereinfachtem Befehlssatz

Die Wissenschaftler bei Hewlett-Packard und anderen Herstellern versuchten natürlich sofort, diese neuen Erkenntnisse über die Arbeitsweise der Rechner technisch nutzbar zu machen. Eckstein des neuen Konzepts ist ein stark vereinfachter bzw. reduzierter Befehlssatz. Die Maschinen, deren Architektur sich auf dieses neue Konzept stützt, werden als RISC-Computer bzw. Computer mit vereinfachtem Befehlssatz (RISC = Reduced Instruction Set Computer) bezeichnet.

Ein einfacher Befehlssatz ermöglicht es, die Befehle direkt in die Rechner-Hardware einzubauen und damit unmittelbar die Schnelligkeit der neuen Hardware-Technologien zu nutzen. Da bei dem neuen Konzept die Decodierung komplexer Befehle entfällt,

wird die Belastung der Zentraleinheit erheblich vermindert. Ergebnis: Schnellere Ausführung der Befehle und damit höhere Gesamtleistung.

Ein völlig neuer Ansatz

Die Entscheidung für Computer mit vereinfachtem Befehlssatz stellt einen Meilenstein in der Entwicklung der Rechnerarchitektur dar und bedeutet einen neuen Ansatz für den Rechnerentwurf. Entscheidungen beim Rechnerentwurf brauchen nicht mehr aufgrund von Annahmen getroffen zu werden, sondern orientieren sich aus den gewonnenen Analyseergebnissen über die Verarbeitung der Information im Rechner im Vergleich zur Designvorgabe.

Diese neueste RISC-Technik spiegelt erstaunlicherweise genau die Grundprinzipien der ersten elektronischen Rechner wider. Warum das so ist und warum die RISC-Technik für die Entwicklung der Computer allgemein von entscheidender Bedeutung ist, wird erst dann klar, wenn man die bisherige Entwicklung der Rechnerarchitekturen Schritt für Schritt nachvollzieht.



RISC: Der entscheidende Schritt vom Komplexen zum Einfachen.

Kapitel 2: Die Evolution der Rechnerarchitekturen

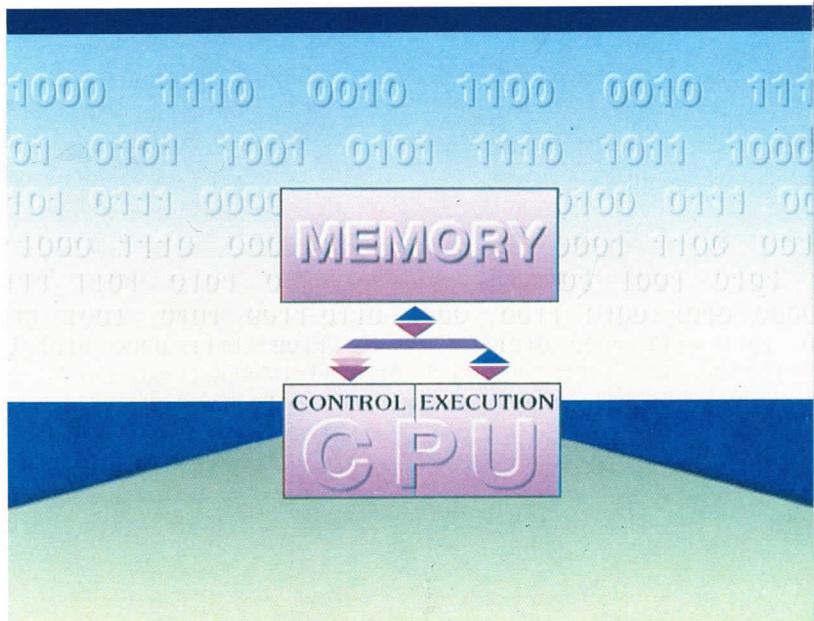
Die Entwicklung der Rechnerarchitekturen ist sehr eng mit den Fortschritten in der Hardware- und Software-Technik verknüpft. Die Entwicklung der Hardware-Technik, von der Elektronenröhre über die Halbleitertechnik bis hin zu den integrierten Schaltkreisen, wirkte sich unmittelbar auf die Schnelligkeit der Zentraleinheit aus. Dieser gewaltige technische Fortschritt brachte eine grundlegende Veränderung der Rechenarchitektur mit sich.

Die Fortschritte in der Hardwaretechnik wurden auch von Neuerungen in der Software-Technik begleitet. Eine wesentliche Neuerung war die Einführung benutzerfreundlicher Programmiersprachen. Die neuen Sprachen waren den natürlichen Sprachen ähnlicher und erforderten ihrerseits Änderungen in der Rechenarchitektur.

Erstaunlicherweise gelang es mit einem einzigen Konzept, dem der komplexen Befehle, sowohl die Hardware- als auch die Software-Probleme zu lösen. Das neue Konzept setzte sich ohne Schwierigkeiten durch und setzte für nahezu zwanzig Jahre den Standard für die Konstruktion von Computern.

Die ersten elektronischen Rechner (1946-1962)

Die ersten elektronischen Rechner lassen sich in zwei Worten beschreiben: einfach und ausgewogen. Ihr Aufbau stützte sich auf die von-Neumann-Maschine, die der berühmte Wissenschaftler und Computerpionier John von Neumann entworfen hatte. Das wohl bekannteste Beispiel für einen Computer der ersten Generation ist die 1950 gebaute UNIVAC I. Sie war damals zweifellos ein Wunderwerk der Technik, nahm jedoch einen ganzen Raum in Anspruch, kostete Millionen Dollar und hatte trotzdem nicht die Rechnerleistung unserer heutigen Personal Computer.



Die ersten elektronischen Rechner waren einfach und ausgewogen.

Die ersten Rechner arbeiteten mit einem einfachen Befehlsatz, der direkt in der Hardware implementiert war.

Einfache Befehlssätze

Die beiden Kernstücke jedes Computers sind Zentraleinheit und Hauptspeicher. Sie waren bei den ersten Rechnern in der gleichen Technik realisiert und deshalb auch gleich schnell. Der Befehlssatz war einfach und war direkt in der Hardware implementiert. Aufgrund dieses einfachen, ausgewogenen Aufbaus arbeiten sie etwa so wie ein Zwei-Takt-Ottomotor, das heißt, die Operationen wurden in zwei Takten bzw. Zyklen ausgeführt: Im ersten Takt oder Zyklus holte sich der Rechner die Befehle, und im zweiten führte er sie aus.

Einfache Hardware

Der Entwurf der ersten Rechner hing im wesentlichen Maße von den bescheidenen Möglichkeiten der Elektronenröhre ab. Man mußte zum Beispiel mit zwei Registern auskommen, denn die Elektronenröhre erlaubte nicht mehr. In Maschinen dieser Bauart wurden die Daten zur Verarbeitung in die Register geladen und das Ergebnis im Arbeitsspeicher abgelegt. Aufgrund der geringen Registerzahl mußten die Daten ständig zwischen der Zentraleinheit und dem Arbeitsspeicher hin- und herbewegt werden.

Elektronenröhren setzten nicht nur der Konstruktion sehr enge Grenzen, sie waren auch äußerst unzu-

verlässig. So wurde die mittlere störungsfreie Zeit (MTBF) der damaligen Rechner in Stunden gemessen. Heute dagegen gewährleisten einige Computerhersteller eine durchschnittliche Jahresverfügbarkeit von 99 %

Einfache Software

Damals steckte auch die Software-Entwicklung noch in den Kinderschuhen. Die Sprache des Programmierers und die Maschinensprache waren nahezu identisch, denn die Programme wurden in Maschinensprache, also mit lauter Einsen und Nullen, geschrieben. Das hatte natürlich den Vorteil, daß die Programme weder interpretiert noch übersetzt werden mußten. Die Einführung der Assembliersprache führte schließlich dazu, daß Programme nicht mehr in der Maschinensprache geschrieben werden mußten. Die Programmierung wurde dadurch zwar wesentlich erleichtert, der Dialog mit dem Computer mußte aber nach wie vor auf einer sehr maschinennahen Sprachebene geführt werden.

Kurz: die ersten Rechner waren einfach und ausgewogen. Bei jedem Befehlszyklus wurde auch jedesmal auf den Hauptspeicher zugegriffen. Dieses Rechnerkonzept funktionierte deshalb, weil die Zentraleinheit und der Arbeitsspeicher gleich schnell waren.

Die ersten Computer mit Mikroprogrammierung (1963-1969)

Im nächsten Jahrzehnt der Entwicklung der Rechnerarchitektur eröffneten neue Software- und Hardware-Techniken dem Konstrukteur neue Wege. Programme mußten nun nicht mehr in Assemblersprache geschrieben werden, und auch die unzulänglichen Elektronenröhren gehörten der Vergangenheit an. Das bedeutete jedoch nicht, daß die Probleme aus der Welt waren; sie wurden lediglich durch neue ersetzt. Aber die Computerwissenschaftler fanden auch hier eine, praktikable Lösung: die Mikroprogrammierung.

Unterschiedliche Arbeitsgeschwindigkeiten durch verschiedene Hardware-Technologien

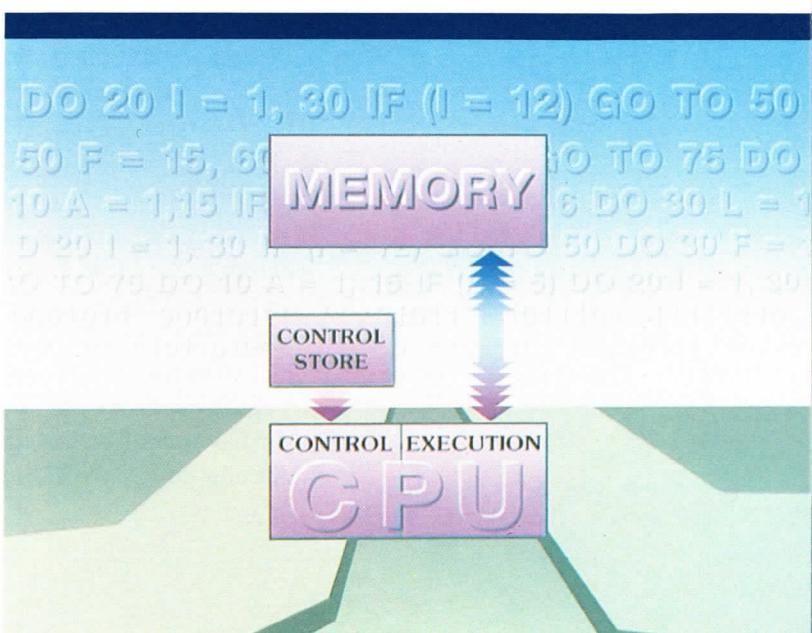
Die Entwicklung des Transistors und des integrierten Schaltkreises führten zu einer wesentlichen Erhöhung der Arbeitsgeschwindigkeit der Zentraleinheit. Ein bedeutender Fortschritt, der leider auch ein nicht weniger bedeutendes Pro-

blem aufwarf. Die Computer der neuen Bauart erforderten nämlich immer größere Hauptspeicher, die aus Kostengründen nicht, wie die Zentraleinheit, in derselben schnellen Schaltkreistechnik realisiert werden konnten. Man wählte auf kostengünstigere Bauarten, wie zum Beispiel Magnetkernspeicher, aus.

Das Ergebnis war, daß die Zentraleinheit nicht voll ausgelastet wurde, weil sie zeitweise auf die aus dem langsameren Hauptspeicher benötigten Daten warten mußte. In vielen Fällen war die Zentraleinheit zehnmal so schnell wie der Hauptspeicher. Maschinen dieser Bauart waren nicht mehr ausgewogen. Die Konstrukteure mußten diese Diskrepanz in den Arbeitsgeschwindigkeiten berücksichtigen.

Die Zentraleinheit war oft nicht ausgelastet, weil sie auf Daten aus dem langsameren Hauptspeicher warten mußte.

Mikroprogrammierte Computer nutzen Kon-trollspeicher, um die Dekodierung komplexer Befehle zu unterstützen.



Die Computerwissenschaftler waren sich einig, daß die „semantische Lücke“ geschlossen werden mußte.

Semantische Lücke durch höhere Programmiersprachen

Während die Hardware-Spezialisten mit der Diskrepanz der Arbeitsgeschwindigkeiten beschäftigt waren, entstand auf der Software-Seite ein ebenso schwerwiegendes Problem. Der immer häufigere Einsatz von Computern ließ den Ruf nach effizienteren Programmiersprachen laut werden, die der natürlichen Sprache möglichst ähnlich sein sollten.

Diese neuen Programmiersprachen, wie zum Beispiel FORTRAN und COBOL, wurden als höhere Programmiersprachen bezeichnet. Sie waren leichter zu lernen und anzuwenden und erhöhten dadurch die Produktivität der Programmierer. Die Bezeichnung „höherer Programmiersprachen“ deutet jedoch schon darauf hin, daß sie sich von dem Binärkode der Maschinensprache, mit seinen Einsen und Nullen, noch weiter entfernen.

Die in einer höheren Programmiersprache geschriebenen Programme mußten nun, vor ihrer Ausführung, zuerst in die Maschinensprache übersetzt werden. Diese Aufgabe fiel den sogenannten Compilern zu, einer neuen Art von Software. Zu Beginn der sechziger Jahre stand die Compiler-Technologie noch am Anfang ihrer Entwicklung.

Und es sollte eine mühevolle Entwicklung werden!

Die Diskrepanz zwischen den höheren Programmiersprachen und der Maschinensprache wurde als „semantische Lücke“ bezeichnet. Dieser Ausdruck wurde deshalb gewählt, weil der Bedeutungsinhalt der Programmieranweisungen einer höheren Programmiersprache viel umfassender ist als die Ausdrucksmöglichkeiten der Maschinensprache. Aus diesem Grund sind für die Verarbeitung eines Befehls einer höheren Programmiersprache mehrere Maschinbefehle nötig. Die Computerwissenschaftler waren sich darüber einig, daß diese Lücke geschlossen werden mußte und die höheren Programmiersprachen und die Maschinensprachen einander angeglichen werden sollten. Eine naheliegende Lösungsmöglichkeit bestand darin, den Bedeutungsinhalt der Maschinbefehle anzuhaben und sie damit den Befehlen der höheren Programmiersprache anzugeleichen.

Komplexe Befehle gleichen die Disparität der Arbeitsgeschwindigkeiten aus

Bei der Lösung des Problems, das sich aus den unverträglichen Arbeitsgeschwindigkeiten von Zentraleinheit und Hauptspeicher ergab, ging man den Weg des geringsten Widerstands. Man suchte nach einem Konzept, das die Zahl der erforderlichen Speicherzugriffe reduzierte.

So wurde die möglichst effektive Nutzung des Hauptspeichers zu einem primären Ziel. Man mußte versuchen, bei einem Speicherzugriff so viele Daten wie möglich zu holen. Damit konnte man die Zentraleinheit besser auslasten und der Computer konnte in der gleichen Zeit mehr Arbeit verrichten. Alles in allem wurde dadurch die Wirtschaftlichkeit erhöht und die Leistung gesteigert.

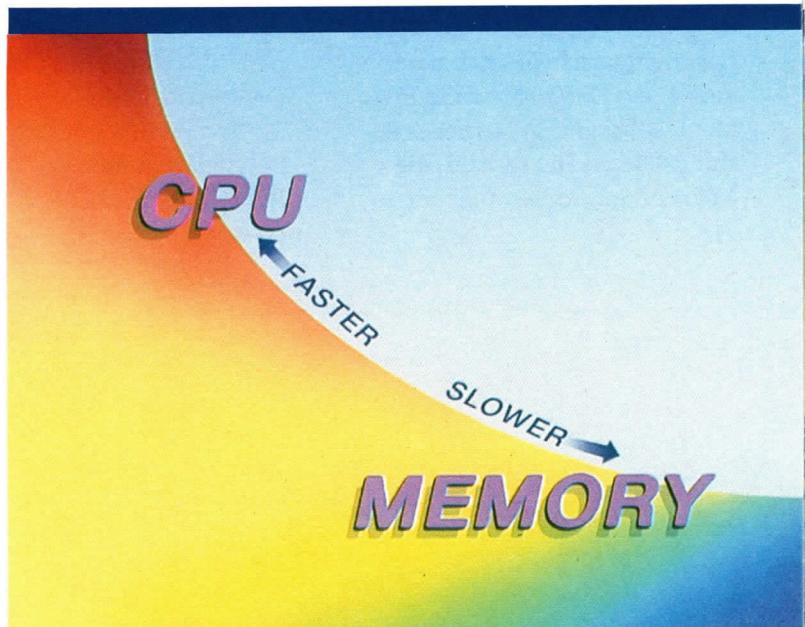
Die Computerkonstrukteure versuchten, das Problem durch die Verwendung komplexer Befehle zu lösen. Die Computer der ersten Generation arbeiteten bekanntlich mit einfachen Befehlen, mit denen lediglich Grundoperationen, wie zum Beispiel die Addition zweier

Zahlen, durchgeführt werden konnten. Bei den komplexen Befehlen jedoch handelt es sich um spezielle Befehle, die sich aus vielen Einzeloperationen zusammensetzen. Maschinen, bei denen dieses neue Konzept zur Anwendung kam, wurden als Computer mit komplexem Befehlssatz (CISC= Complex Instruction Set Computer) bezeichnet.

Ein kleines Beispiel soll verdeutlichen, wie durch komplexe Befehle die erforderlichen Speicherzugriffe reduziert werden. Der Ausdruck „ein Doppelrumpfboot aus zwei starr miteinander verbundenen Bootskörpern“ soll eine Folge von einfachen Befehlen darstellen. Computer der ersten Generation verarbeiteten den Ausdruck, indem sie zuerst das erste Wort „ein“, aus dem Arbeitsspeicher holten und es ausführten, anschlie-

Computerkonstrukteure verwendeten komplexe Befehle, um die Anzahl der Hauptspeicherzugriffe zu reduzieren.

Oft war die CPU zehn mal schneller als der Hauptspeicher.



Bend das zweite Wort („Doppelrumpfboot“) usw., bis der Ausdruck vollständig ausgeführt war. Bei dieser Art der Verarbeitung waren acht Speicherzugriffe, nämlich einer für jedes Wort, erforderlich.

Würde man dieses Verarbeitungsproblem mit einem komplexen Befehl lösen, dann würde man den Ausdruck durch ein einziges Wort ersetzen, das dem Ausdruck entspricht, also in unserem Fall das Wort „Katamaran“. Der Computer versteht dieses speziellere Wort und braucht deshalb nur einmal auf den Arbeitsspeicher zuzugreifen. Mit dieser Art der Verarbeitung können also sieben Hauptspeicherzugriffe eingespart werden.

Der Ansatz hatte aber einen großen Nachteil: Die Zentraleinheit hätte mit weiteren Schaltkreisen ausgerüstet werden müssen, um die komplexeren Befehle erkennen zu können. Dadurch wäre diese Lösung unerschwinglich teuer geworden. Das anstehende Problem wurde durch ein neues leistungsfähiges Instrument, die Mikroprogrammierung, umgangen.

Jetzt war es möglich, komplexe Befehle (Makrocode) in kleinere, einfache und für den Rechner direkt verständliche Befehle (Mikrocode) zu zerlegen. Dazu war eine eigenständige, sehr schnelle Speichereinheit in der Zentraleinheit erforderlich. Diese Steuereinheit, diente der Zentraleinheit gewissermaßen als Wörterbuch. Um bei unserem Beispiel zu bleiben:

Die Definition, das heißt die Einzelbefehle oder der Mikrocode, den die Zentraleinheit zur Ausführung eines komplexen Befehls benötigt, wurde in der Steuereinheit abgelegt. Durch die Zerlegung der Makrobefehle in kleinere, für den Computer verständliche Mikrobefehle erreichten die Computerwissenschaftler damals höhere Flexibilität. Allerdings waren für den Codiervorgang zusätzliche Maschinenzyklen erforderlich.

Bei den CISC-Computern (Complex Instruction Set Computer) werden die komplexen Befehle in die Steuereinheit geladen, wo sie bei Bedarf in mehrere einfache Befehle aufgeschlüsselt werden. Diese einfachen Befehle werden

dann von der Zentraleinheit ausgeführt. So gelang es, die einfachen, in der Hardware eingebauten Befehle der ersten Rechner durch komplexe Befehle zu ersetzen.

Aber auch diese Lösung hatte ihren Preis, nämlich die Zeit, die bei der Decodierung und Übersetzung der komplexen Befehle verlorenging. Da jedoch der Hauptspeicher sowieso viel langsamer war als die Zentraleinheit, fiel der Zeitverlust durch das Decodieren, im Vergleich zur gesamten Zeitsparnis, kaum ins Gewicht. Das Problem der unterschiedlichen Arbeitsgeschwindigkeiten von Zentraleinheit und Arbeitsspeicher wurde gelöst, indem man die Anzahl der Speicherzugriffe durch die Verwendung von komplexen Befehlen verringerte.

Komplexe Befehle schließen auch die semantische Lücke

Man fand sehr schnell heraus, daß die komplexen Befehle auch geeignet waren, die durch die höheren Programmiersprachen entstandene semantische Lücke zu schließen.

Durch zusätzliche komplexe Befehle, die den „Wortschatz“ des Rechners vergrößerten, konnte die semantische Ebene der Maschinenbefehle erweitert werden. Die inhaltliche Unverträglichkeit zwischen den Maschinenbefehlen und den Anweisungen der höheren Programmiersprachen konnte auf diese Weise verringert werden. Man glaubte sogar, weniger Übersetzungszyklen zu benötigen, wenn der Befehlssatz den Anweisungen der höheren Programmiersprachen möglichst nahekam. Diese Überzeugung setzte sich immer stärker durch, so daß die Programmierer von Compilern ihre „Lieblingsbefehle“ der höheren Programmiersprachen in den Befehlssatz der Computer implementiert haben wollten.

CISC-Computer wurden Standard

Mit dem komplexen Befehlssatz glaubten die Computerkonstrukteure die Lösung ihres Problems gefunden zu haben. Die komplexen Befehle ermöglichten es, die Anzahl der notwendigen Speicherzugriffe zu verringern und gleichzeitig die semantische Lücke weitgehend zu verkleinern. Nichts schien diese Überzeugung in Frage zu stellen, und so ist es auch kein Wunder, daß sie das Design der Computer der nächsten beiden Jahrzehnte entscheidend beeinflußte. Eine Folge war, daß die Befehlssätze immer umfangreicher und komplexer wurden; während Rechner der ersten Generation, wie die MARK-1, mit sieben Befehlen auskamen, arbeiten Computer mit komplexen Befehlssatz bereits mit über 300 Befehlen.

Im Gegensatz zu den elektronischen Rechnern der ersten Generation, die einfach aufgebaut sein mußten, weil die Elektronenröhre keine komplexere Bauart erlaubte, war die neue Computergeneration eigentlich nur noch durch die Anzahl der komplexen Befehle eingeschränkt, die in die Steuereinheit paßten.

Mit der Zeit wurden die Befehlssätze immer größer und komplexer.

Mikroprogrammierte Computer mit Cache-Speicher (1970-1984)

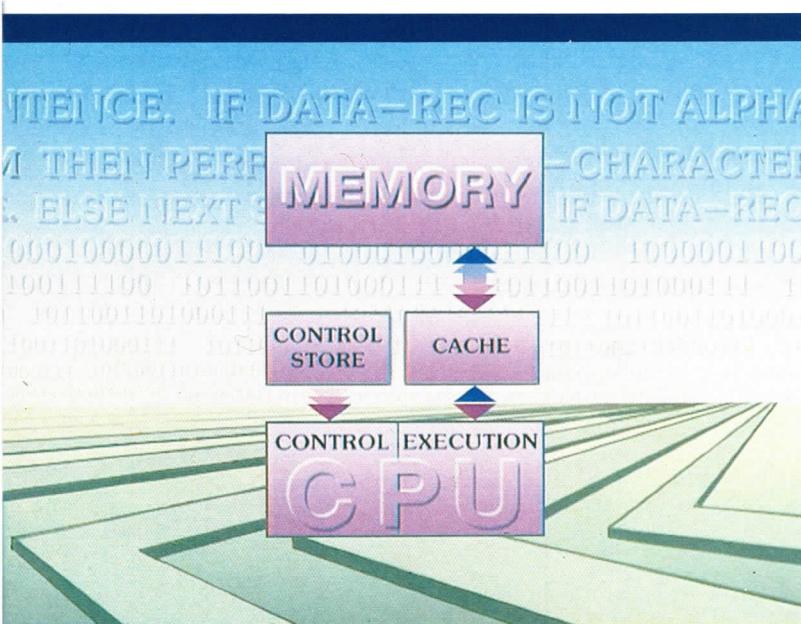
Die Mikroprogrammierung blieb lange Zeit unangefochtener Ansatz in der Rechenarchitektur. Dennoch versuchten die Computerwissenschaftler weiterhin, die immer noch unterschiedlichen Arbeitgeschwindigkeiten von Zentraleinheit und Hauptspeicher auszugleichen. Dabei kamen ihnen vor allem Fortschritte in der Schaltkreisentwicklung und im Compilerbau zugute. Ein entscheidender Durchbruch gelang jedoch erst mit der Entwicklung der Cache-Speicher.

Der Einsatz von Speichern mit integrierten Schaltkreisen

In den sechziger und siebziger Jahren, als integrierte Schaltkreise (ICs) immer billiger und leistungsfähiger wurden, war der Fortschritt in der Computertechnik kaum noch aufzuhalten. Die wesentlich billigeren integrierten Schaltkreise traten nun auch an die Stelle der langsameren Magnetkernspeicher. Damit war ein entscheidender Schritt getan, denn der Hauptspeicher und die Zentraleinheit waren nun in der gleichen Technik realisiert. Das Ergebnis waren wesentlich schnellere Hauptspeicher, die nun nach zwanzig Jahren mit der Arbeitgeschwindigkeit der Zentraleinheit mithalten konnten.

Cache-Speicher

Zusätzlich wurde die Schnelligkeit des Hauptspeichers durch den Einsatz von Cache-Speichern erhöht. Cache-Speicher sind kleine, schnelle Pufferspeicher zwischen dem Hauptspeicher und der Zentraleinheit. Cache-Speicher und Hauptspeicher sehen für die Zentraleinheit wie der Hauptspeicher aus. Wenn man jedoch häufig benötigte Daten im Cache-Speicher vorhält, kann man Zugriffe auf den Hauptspeicher minimieren.



Cache-Speicher erhöhen die Schnelligkeit des Speicherzugriffs.

Die Auswirkung des Cache-Speichers auf die Schnelligkeit des Hauptspeichers soll mit einem Beispiel genauer erläutert werden. Stellen Sie sich den Hauptspeicher als Bibliothek vor. Wenn Sie in eine Bibliothek gehen, um dort Material zu sammeln, arbeiten Sie im Normalfall mit mehreren Nachschlagewerken. Sie können nun Ihre Arbeit auf zweierlei Weise gestalten: Entweder Sie gehen jedesmal zum entsprechenden Regal, wenn Sie ein bestimmtes Buch brauchen, arbeiten es durch und holen sich das nächste, oder Sie holen sich die Bücher zu Ihrem Thema auf einmal und suchen sich einen entsprechenden Platz zum Arbeiten. Wenn Sie jedesmal aufstehen und zu einem Regal gehen, um ein Buch zurückzustellen oder ein neues zu holen, verlieren Sie auf Ihrem Weg zwischen Ihrem Arbeitsplatz und den Bücherregalen kostbare Zeit. Wenn Sie sich jedoch alle Ihre Bücher besorgen, bevor Sie zu arbeiten beginnen

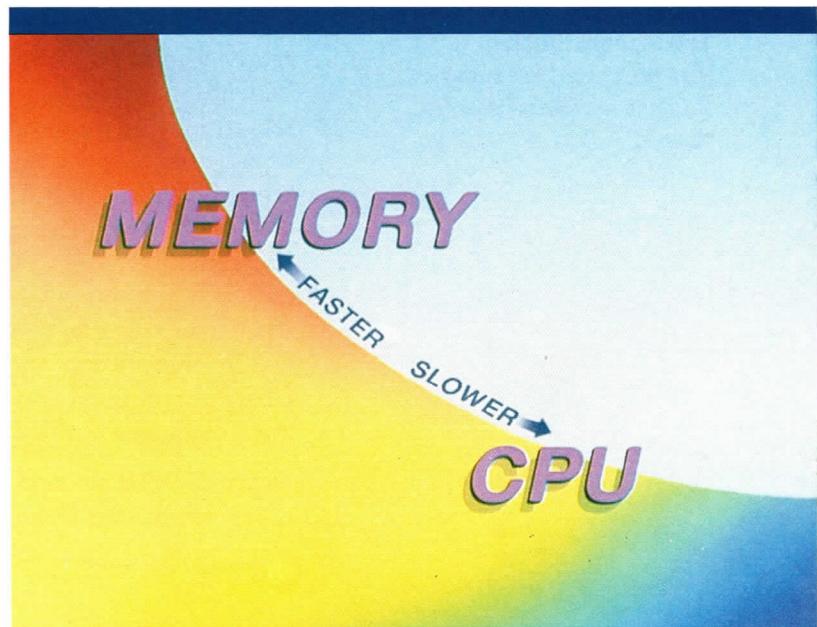
und sie geschlossen zu Ihrem Arbeitsplatz bringen, haben Sie die gewünschten Quellen schneller zur Hand und verlieren weniger Zeit mit dem Zurücktragen Ihrer Bücher. Die Bücherregale in unserem Beispiel entsprechen dem Hauptspeicher, Ihr Arbeitsplatz dem Cache-Speicher und das von Ihnen benutzte Buch den Registern des Computers.

Da Cache-Speicher relativ klein sind, war es wirtschaftlich, sie in der schnellsten Schaltkreistechnik bereitzustellen. Der Hauptspeicher d.h. der Cache-Speicher und die Zentraleinheit verfügten nun beide über die Vorteile der Schaltkreistechnik und konnten somit in der gleichen Geschwindigkeit arbeiten. Bald jedoch stellte man fest, daß nun der schnellere Cache-Speicher warten mußte, bis die Zentraleinheit die Makrobefehle in kleinere

Mikrobefehle zerlegt hatte. Vorher war die Zentraleinheit durch den Arbeitsspeicher gebremst worden; nun war es umgekehrt.

Da der Hauptspeicher und die Zentraleinheit bisher unterschiedlich schnell gewesen waren, hatten auch die komplexen Befehle gute Dienste geleistet. Da sie oft auf den viel langsameren Hauptspeicher warten mußte, war die Zentraleinheit längere Zeit ohne Arbeit. Während dieser Zeit konnte die Zentraleinheit komplexe Befehle decodieren, ohne dadurch einen Leistungsverlust hervorzurufen. Mit der Einführung der neuen integrierten Speicherschaltung war der Hauptspeicher jedoch genauso schnell geworden wie die Zentraleinheit, und das führte dazu, daß die zusätzlichen Decodierzyklen die Leistung des Computers nun eindeutig beeinflussen.

Das Ungleichgewicht der Arbeitsgeschwindigkeit wurde umgekehrt. Der schnellere Hauptspeicher mußte nun oft abwarten, bis die CPU-Zentraleinheit komplexe Befehle decodiert hatte.



Eine neue Generation von Compilern tritt auf den Plan

Während die Hardware-Technik immer größere Fortschritte machte, blieb auch bei der Software die Entwicklung nicht stehen. So wurden zum Beispiel für die Übersetzung der höheren Programmiersprachen immer raffiniertere Compiler entwickelt. Diese neue Generation von Compilern wurde aufgrund ihrer erstaunlichen Fähigkeiten „optimierende Compiler“ genannt.

Die Überzeugung „Des Guten kann nie zuviel sein“, führte zu spezifischen Befehlssätzen, daß man sie kaum noch sinnvoll ansehen konnte.

Die Übersetzung von Programmen in die Maschinensprache ist eine sehr problematische Aufgabe, da es keine für jedes Programm optimale Übersetzung gibt. Es kann durchaus vorkommen, daß Compiler verschiedene Übersetzungen generieren, die trotzdem die Bedeutung des ursprünglichen, in der höheren Programmiersprache geschriebenen Programms nahezu ohne Abweichungen im Maschinencode wiedergeben. Die optimierende Compiler können sehr viel mehr als die üblichen Compiler. Sie übersetzen nicht nur Programme der höheren Programmiersprache in den Maschinencode, sondern verkürzen darüber hinaus die Programmalaufzeit und sorgen dafür, daß das in Maschinensprache vorliegende Programm möglichst wenig Speicherplatz beansprucht.

Bei den Computern mit komplexem Befehlssatz standen Hunderte von Befehlen zur Wahl, von denen sehr viele eine sehr spezifische Bedeutung hatten. Es war beschwerlich und umständlich, einen kompletten Satz von hochspezifischen Befehlen jedesmal neu zu analysieren. Deshalb fingen die Programmierer von Compilern an, nur noch bestimmte Untermengen des Befehlssatzes zu verwenden, so daß Befehle mit sehr spezifischem Charakter oft überhaupt nicht mehr eingesetzt wurden.

Die komplexen Befehle waren von Computerwissenschaftlern eingeführt worden, denen es hauptsächlich darauf ankam, die zweckmäßigsten und besten Befehle zu verwenden. Auch bei der Entwicklung von Compilern wurden häufig ganz spezifische Befehle benötigt. Nie aber hatte man bisher bei der Auswahl eines Befehls auf die Häufigkeit seines Einsatzes oder ein anderes, allgemein anerkanntes Auswahlkriterium geachtet. Der Wahlspruch lautete einfach: Des Guten kann nie zuviel sein! Und auf die einzelnen Befehle angewendet, bedeutet das: Wenn „Katamaran“ ein guter Befehl ist, dann müßte der Befehl „roter Katamaran“ ein noch besserer sein. Und wenn man schon mit dem Befehl „roter Katamaran“ arbeitet, warum dann nicht auch mit „grüner“, „gelber“ oder „blauer Katamaran“? Am Ende der Entwicklung standen Befehle, die so spezifisch waren, daß sie kaum noch wirtschaftlich eingesetzt werden konnten.

Neue Erkenntnisse eröffnen neue Wege

Als man feststellte, daß Cache-Speicher die Rechnerleistung verbessern können, bemühte man sich, diese Wirkung weiter auszubauen, um die Effizienz der Cache-Speicher noch zu steigern. Zu diesem Zweck war es notwendig zu wissen, was in einem Computer während eines Programmelaufs vorging. Das Ergebnis dieser Untersuchungen war überraschend. Man stellte fest, daß die Computer meistens nur eine relativ kleine Anzahl der verfügbaren Maschinenbefehle benutzen.

Mit einem solchen Ergebnis hatte man überhaupt nicht gerechnet. Die komplexen Befehle, von denen man sich eine so starke Verbesserung der Rechnerleistung erhofft und sie deshalb als wichtigsten Bestandteil der CISC-Architektur betrachtet hatte, wurden in Wirklichkeit kaum benutzt. Die Untersuchungen ergaben vielmehr, daß der Computer während 80 % der Programmalaufzeit einfache Lade-, Speicher- und Verzweigungsbefehle ausführt.

Man hatte nicht damit gerechnet, daß die einfachen Befehle eine so entscheidende Rolle spielen, und man war nun gezwungen, den Nutzen komplexer Befehle neu zu überdenken. Welchen Sinn hatten in der Tat Befehle, die 80 % der

Steuereinheit in Anspruch nahmen und kaum eingesetzt wurden!

Zusammenfassend kann man sagen, daß das Ende dieser Computerära von einer Enttäuschung gekennzeichnet war, denn was man bisher hochgelobt hatte und was als glücklicher Versuch galt, den Computern ihre technische Ausgewogenheit zurückzugeben und die immer weiter auseinanderklaffende semantische Lücke zu schließen, war durch die fortschreitende Entwicklung in der Hardware- und Software-Technik überholt worden. Angesichts der modernen Compiler-und Hauptspeichertechnik waren komplexe, mikroprogrammierte Architekturen offensichtlich keine effiziente Lösung mehr.

Man wurde sich bewußt, daß die komplexe Mikroprogrammierung nicht mehr effizient war.

Computer mit vereinfachtem Befehlsatz (1985-....)

Diese neue Erkenntnis führte dazu, daß neue Architekturen auf der Grundlage eines kleinen Befehlsatzes häufig verwendeter Befehle entwickelt wurden. Das neue Konzept in der Computerkonstruktion hieß: Rechnen mit vereinfachtem Befehlssatz (RISC= Reduced Instruction Set Computer). Durch den Verzicht auf Steuereinheiten und den Tausch des komplexen Befehlssatzes gegen einen einfachen, erreichten die RISC-Computer erstaunliche Leistungen.

In der RISC-Technologie sind einfache Befehle die Grundelemente aller Operationen, einfacher und komplexer.

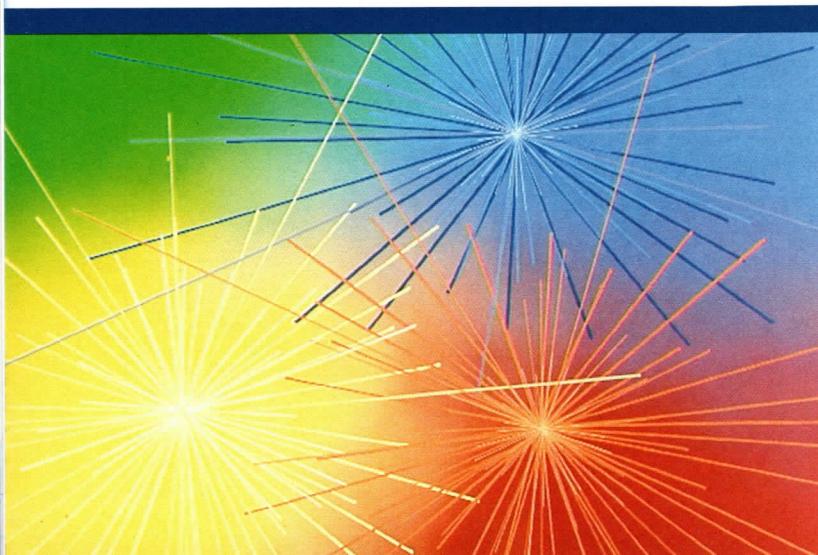
Zurück zum Einfachen

Durch die Rückkehr zu einem einfachen Befehlssatz war es nun wieder möglich, die Befehle hardwaremäßig zu implementieren. Außerdem wurde der Mikrocode überflüssig und damit auch die Decodierung komplexer Befehle. Nun konnte praktisch bei jedem Maschinenzyklus ein Befehl ausgeführt werden. Darüber hinaus ermöglichte die Einfachheit der RISC-Architektur höhere Taktfrequenzen für einen kürzeren Maschinenzyklus, was sich wiederum vorteilhaft auf die Leistung des Rechners auswirkte.

Der vereinfachte Befehlssatz ist der Kern jedes RISC-Computers. Aus den einfachen Befehlen können alle Operationen aufgebaut werden, seien sie nun einfacher oder komplexer Art. Ein CISC-Rechner

kann dadurch auf einem Rechner mit RISC-Architektur emuliert werden. Die Beispiele für dieses Modell sind zahlreich. Denken Sie an den Farbkreis: Alle auf dem Farbkreis enthaltenen Farben lassen sich auf drei Grundfarben rot, gelb und blau zurückführen. Diese drei Grundfarben bieten Ihnen alle nötigen Kombinationsmöglichkeiten, das heißt, wenn Sie sie miteinander mischen, können Sie jeden beliebigen Farbton erzeugen. Das besondere daran sind nicht die Kombinationsmöglichkeiten, sondern die Bauelemente selbst. In unserem Beispiel sind genau drei Grundfarben nötig; mit nur zwei Grundfarben würde das System nicht mehr funktionieren.

Bei den RISC-Computern gibt es keine goldene Regel für die zu verwendenden Befehle und kein allgemeines Konzept, das für die Konstruktion solcher Maschinen als Grundlage dienen könnte. Sie können, ebenso wie die CISC-Computer, auf verschiedene Art und Weise realisiert werden. Bei Hewlett-Packard, zum Beispiel, hat das RISC-Konzept eine entscheidende Rolle bei der Entwicklung der neuen HP Precision Architectur gespielt.



Mit einfachen Bauelementen zum vollen Spektrum.

Kapitel 3: Die neue HP Precision Architecture

Die Entwicklungsingenieure von Hewlett-Packard haben eine neue Computergeneration entwickelt, die sich auf das Konzept der Computer mit vereinfachtem Befehlsatz gründet. Nach jahrelangen Untersuchungen der Verarbeitungsvorgänge, die sich in einem Computer abspielen, haben sie dieses Konzept übernommen und neueste Entwicklungen in Hardware- und Software-Technologien realisiert. Das Ergebnis dieser Bemühungen ist die neue HP Precision Architektur.

Kernstücke der neuen HP Precision Architecture sind:

- ein vereinfachter Befehlssatz
 - Befehle mit fester Länge und festem Format
 - Speicherzugriff nur bei Lade- oder Speichervorgängen
 - fest verdrahtete Befehle
 - Einzyklusbefehlsbetrieb
 - optimierende Compiler

Alle diese charakteristischen Merkmale beschreiben das grundlegende HP RISC Konzept. Doch die eigentliche HP Precision Architecture geht noch weiter. Die HP-Entwicklungsingenieure haben die wichtigsten RISC-Elemente um weitere Eigenschaften und Funktionen ergänzt, um die Flexibilität und Leistung der Rechner noch weiter zu erhöhen.

Durch diese zusätzliche Optimierung ist Hewlett-Packard in der Lage, eine Computerfamilie vorzustellen, die ihre Anforderungen heute abdeckt und die bis weit in das kommende Jahrhundert Gültigkeit haben wird.

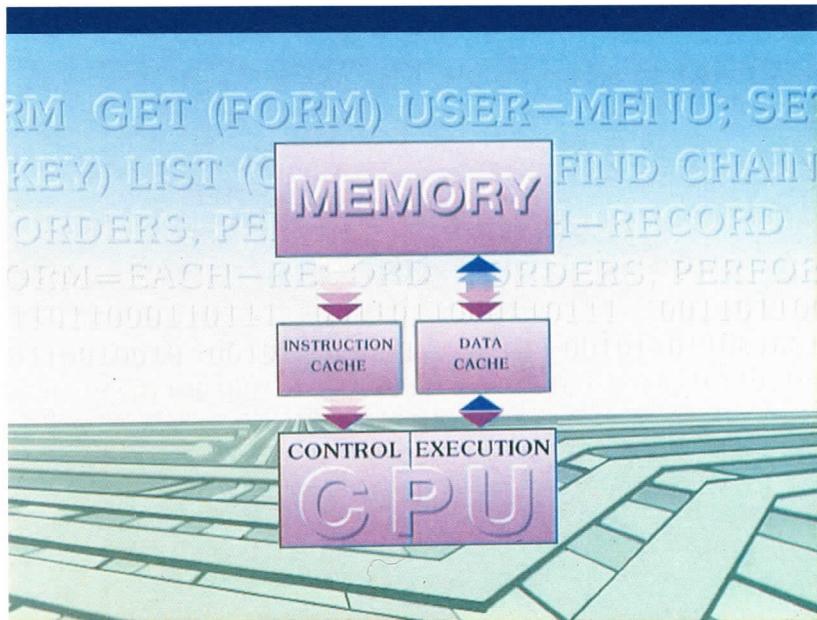
wahl der geeigneten Befehle investiert; sie haben eine Vielzahl von Programmen mit Billionen von Befehlen getestet, bevor sie sich für den typischen Befehlssatz der neuen HP Precision Architecture entschieden haben.

Die Grundzüge des RISC-Konzepts

Einfacher Befehlssatz

Ein einfacher, zweckoptimierter Befehlssatz ist das Merkmal jedes Computers. Es gibt jedoch keine goldene Regel, nach der man sich bei der Auswahl der zu verwendenden Befehle richten kann. Die Wissenschaftler von Hewlett-Packard haben viel Zeit in die Aus-

Die HP-Entwicklungsingenieure haben eine Vielzahl von Programmen mit Milliarden von Befehlen getestet, bevor sie sich für den typischen Befehlssatz der neuen HP Precision Architecture entschieden haben.



*Die neue HP Precision
Architecture sorgt für
Einfachheit und Ausgegli-
chenheit.*

Der Befehlssatz wurde von Grund auf neu aufgebaut. Bei der Auswahl der Befehle verließ man sich nicht auf vage Vermutungen. Jeder Befehl mußte sich seine Aufnahme in den Befehlssatz sozusagen verdienen, denn es wurden nur solche Befehle aufgenommen, die das Leistungsvermögen des Systems verbessern.

Bei Computern mit vereinfachtem Befehlssatz spielt die Zusammenstellung der Befehle deshalb eine entscheidende Rolle, weil die Befehle als Bauelemente für komplexere und speziellere Operatio-

nen dienen müssen. Die einfachen Befehle werden zu geeigneten Befehlsfolgen zusammengesetzt, um den jeweiligen besonderen Anforderungen entsprechen zu können. Damit schränken einfache Befehle den Betrieb des Rechners keineswegs ein. Dazu kommt noch der Vorzug einer nahezu unbegrenzten Flexibilität. Die Hardware wird nicht mehr durch selten benötigte Befehle belastet.

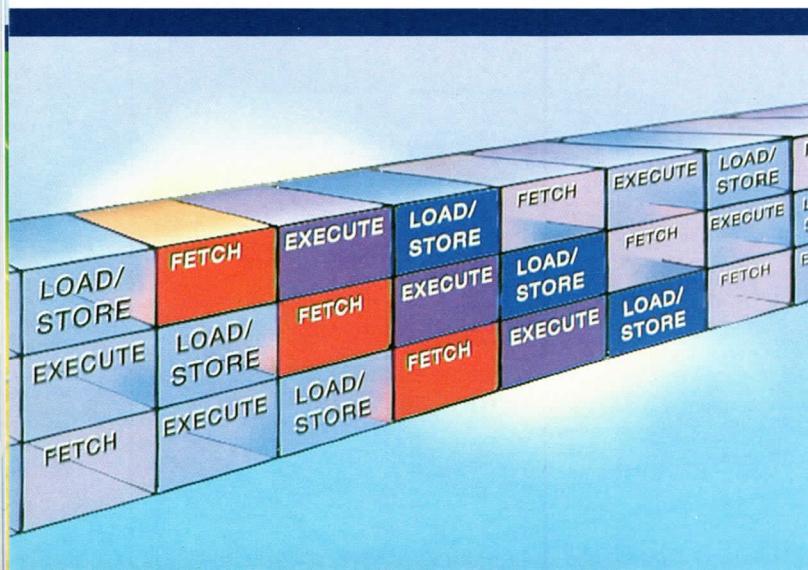
Befehle mit fester Länge und festem Format

Die neue HP Precision Architecture kommt nicht nur mit weniger Befehlen aus, sondern benutzt darüber hinaus auch noch mit Befehlen gleicher Länge und gleichen Formaten. Bei Befehlen mit festem Format stehen der Opera-

tionscode, als Beschreibung der vom Befehl bewirkten Operation, und die Operanden immer an der gleichen Stelle.

Damit wird die segmentweise Parallelverarbeitung von Befehlen (Pipelining) wesentlich erleichtert. Das besondere der Pipelinings besteht darin, daß sich die einzelnen Befehls-Ausführungsphasen zeitlich überlappen, das heißt, ein Befehl kann bereits ausgeführt werden, obwohl die Ausführung eines anderen noch nicht beendet ist. Die Leistung des Rechners wird dadurch erheblich gesteigert.

Es gibt auch im täglichen Leben eine Vielzahl von Prozessen, die nach dem Pipelining Prinzip ablaufen. Denken Sie zum Beispiel an eine Autowaschstraße, ein Fließband oder an die Bearbeitungsdurchläufe in einem Büro. Die Funktion eines solchen Systems kann jedoch nur dann aufrechterhalten werden, wenn Verzögerungen und Engpässe vermieden werden. Jede Prozeßstufe muß jederzeit voll ausgelastet sein, und die einzelnen Operationen dürfen sich nicht gegenseitig behindern.



Pipelining ist eine Technik, bei der sich die Befehlsverarbeitung überlappt.

Einfache Befehle mit fester Länge und festem Format ermöglichen einen wirkungsvollen Einsatz des Pipelining. Da alle Befehle gleich lang sind, entstehen bei ihrer Verarbeitung keine Lücken. Außerdem kann man leichter abschätzen, wo ein Befehl beginnt und wo er endet. Und man kann Informationen innerhalb eines Befehls, aufgrund des festen Befehlsformats, leichter wiederfinden.

Lade- und Speicherkonzept

Die neue HP Precision Architecture beinhaltet ein Lade- und Speicherkonzept. Die Ladebefehle übertragen Daten aus dem Cache-Speicher in die Register der Zentraleinheit, und Speicherbefehle übertragen die Daten von den Registern wieder zurück in den Cache-Speicher.

Dieses Lade- und Speicherkonzept entspricht voll und ganz dem Konzept des Einfachen in der RISC-Architektur, denn es stellt die einfachste Steuerungsart dar. Da nämlich alle Daten in die Hochgeschwindigkeitsregister geladen werden müssen, werden die Operationen zwischen den Registern oder zwischen einem Register und

einer im Befehl enthaltenen Konstante durchgeführt. Kombiniert man diese Technik mit optimierenden Compilern, dann kann man sicher sein, daß häufig benutzte Daten in den Registern gespeichert bleiben und die Anzahl der erforderlichen Cache-Zugriffe weiter verringert wird.

Fest verdrahtete Befehle

Da es sich um einfache Befehle handelt und der Rechner den Cache-Speicher nur für Lade- und Speichervorgänge beansprucht, können die Befehle direkt in der Hardware implementiert werden. Das heißt, daß die Befehle direkt in der Zentraleinheit implementiert sind.

Einfache, in die Hardware fest verdrahtete Befehle sind schneller als komplexe Befehle, die erst noch decodiert werden müssen. In der neuen HP Precision Architecture existiert deshalb keine Steuereinheit, die die komplexen Befehle decodiert. Folglich können die Maschinenzyklen besser ausgenutzt werden.

Einzyklusbefehlsablauf

Durch die Verwendung von einfachen, in der Hardware fest verdrah-

teten Befehlen und die Möglichkeit des Pipelining kann bei der neuen HP Precision Architecture praktisch jeder Befehl in einem Maschinenzyklus ausgeführt werden.

Beim Einzyklusbefehlsablauf gehen also keine Maschinenzyklen mehr verloren, und Sie können die Leistungsfähigkeit Ihres Rechners voll ausnutzen. Eine enorme Leistungssteigerung, wenn man bedenkt, daß pro Impuls der Systemuhr (Taktgeber) ein Befehl ausgeführt werden kann.

Durch die Verwendung von einfachen, in der Hardware fest verdrahteten Befehlen und die Möglichkeit der Pipelining kann bei der neuen HP Precision Architecture praktisch ein Befehl pro Maschinenzyklus ausgeführt werden.

Optimierende Compiler

Aufgrund des einfachen Befehlsatzes der neuen HP Precision Architecture kann durch den Einsatz von optimierenden Compilern die Rechnerkapazität noch effektiver ausgenutzt werden.

Optimierende Compiler sorgen für eine bestmögliche Abstimmung zwischen höheren Programmiersprachen und Maschinenbefehlen. Optimierende Compiler analysieren das Verhalten bzw. den Ablauf des Programms als Ganzes und sorgen dafür, daß die Befehle in der günstigsten Reihenfolge ausgeführt werden. Dadurch können die Vorteile des Pipelinings voll ausgeschöpft werden.

Um dieses noch effizienter zu gestalten, machen die optimierenden Compiler von der in der neuen HP Precision Architecture gegebenen Möglichkeit der »verzögerten Verzweigung« (delayed branches) Gebrauch. Verzweigungen z.B. Prozeduraufrufe gehören zu den am häufigsten verwendeten Befehlen. Sie haben jedoch bei den meisten Systemen mit Parallelverarbeitung den Nachteil, daß während der Ausführung eines Verzweigungsbefehls der nachfolgende Befehl zwar geladen, aber nicht ausgeführt wird. Ein Maschinenzyklus wird also vergeudet. Bei der HP Precision Architecture sorgen die optimierenden Compiler für die Ausführung anderer Befehle, während einer Verzweigung.

Dadurch ist ein noch effizienteres Pipelining gewährleistet was in einer kürzeren Programmausführungszeit resultiert.

Der optimierende Compiler sorgt nicht nur für eine reibungslose Ausführung der einzelnen Befehle, sondern weist auch Register zu. Damit kontrolliert er direkt die Hardware. Alle Berechnungen werden nämlich auf Registerebene durchgeführt, das heißt, Daten müssen nicht mehr in den Registern gespeichert und von dort wieder zurückgeholt werden; die Befehlausführung wird einfacher und nimmt weniger Zeit in Anspruch.

Die RISC-Architektur wird noch leistungsfähiger

Die HP-Ingenieure haben sich mit den Standardvorteilen der RISC-Architektur nicht zufriedengegeben. Um die durch die neue HP Precision Architecture gegebene Leistung und Flexibilität noch weiter zu verbessern, haben sie das RISC-Grundprinzip erweitert und zusätzliche Funktionen eingebaut, wie zum Beispiel:

- erweiterter Adreßraum
- Unterstützung von Mehrprozessorbetrieb und Coprozessoren
- im Hauptspeicher abgebildetes Ein-/Ausgabe-Subsystem (E/A-Speichermapping)

Erweiterter Adreßraum

Eines der wichtigsten Leistungsmerkmale der neuen HP Precision Architecture ist ihr erweiterter Adreßraum. Damit sind die HP-Systeme für die Anforderungen nicht nur von heute, sondern auch von morgen gerüstet.

Bei der HP Precision Architecture werden wahlweise drei verschiedene virtuelle Adreßräume unterstützt: 2^{32} , 2^{48} und 2^{64} . Das heißt, daß bei der neuen HP Precision Architecture bis zu 16 Giga-Giga Byte (!!) adressiert werden können!

Um sich die ungeheuere Größe dieser Zahl einmal vor Augen zu führen, soll ein Vergleich zwischen dem Adreßraum und der Weltbevölkerung angestellt werden.

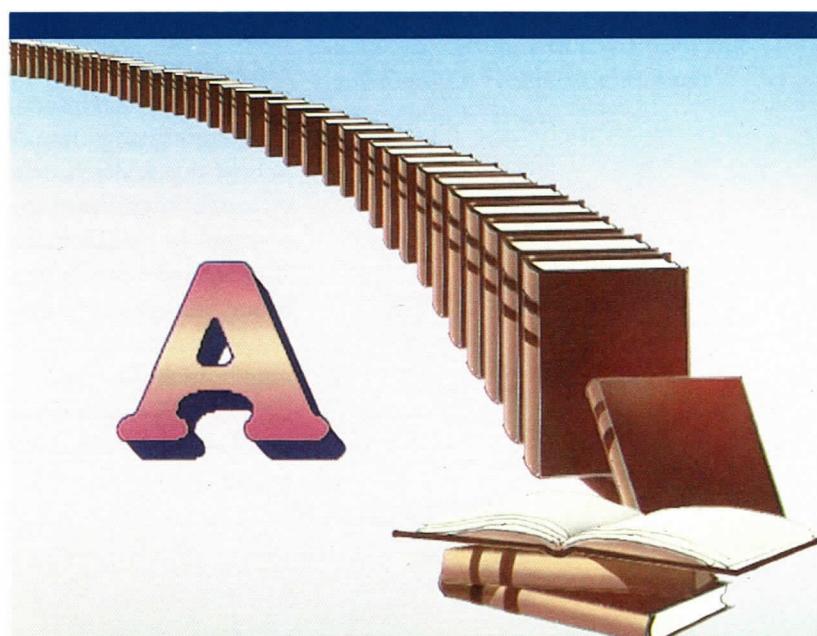
Heute leben ungefähr 5 Milliarden Menschen auf der Erde. Der Adreßraum vieler heutiger 32-bit-Computer würde gerade ausreichen, um jedem Menschen eine Speicherstelle zuzuweisen. Der Adreßraum der neuen HP Precision Architecture ist dagegen so groß, daß jeder Mensch 4 Milliarden Speicherplätze erhalten würde.

Das ist ungefähr das gleiche, als würde man diesem Menschen einerseits einen einzigen Buchstaben des Alphabets und andererseits 500 Bände einer Enzyklopädie zugestehen.

Experten sind der Meinung, daß sich die Adreßraumanforderungen alle zwei Jahre verdoppeln. HP-Kunden können also dank des gigantischen Adreßraums der HP Precision Architecture gelassen in die Zukunft blicken.

Bei der HP Precision Architecture können darüber hinaus die Adreßumwandlungen aus der Hardware erfolgen, so daß der Hauptspeicher bei diesem Vorgang keinen Engpaß mehr darstellt. Ein zusätzlicher Schritt also in Richtung Leistungssteigerung!

Der Unterschied im Adreßraum zwischen vielen 32-bit-Computern und der neuen HP Precision Architecture.



Mehrprozessorbetrieb und Coprozessoren

Die HP Precision Architecture kann dank der Unterstützung von Mehrprozessorbetrieb und Coprozessoren den unterschiedlichsten Anforderungen an Rechnerleistung und Flexibilität angepaßt werden.

Durch die Verwendung von Coprozessoren kann Ihr System für den von Ihnen gewünschten Verwendungszweck maßgeschneidert

Mit Coprozessoren können Sie Ihren Rechner nach Ihren besonderen Anforderungen ausrüsten.

werden. Wenn Sie zum Beispiel umfangreiche numerische Berechnungen durchführen müssen, können Sie durch den Einbau eines Gleitkomma-Coprozessors eine auf diesen Bereich zugeschnittene Leistungssteigerung Ihres Systems erreichen. Sie können Ihr System aber auch genauso gut auf die Vektorrechnung oder die grafische Datenverarbeitung ausrüsten.

Kurz: Das Grundsysteem von Hewlett-Packard kann jederzeit durch Zukauf eines Coprozessors mit Sonderfunktionen für eine ganz besondere Aufgabe ausgerüstet werden. Sie brauchen nicht gleich einen neuen Computer zu kaufen, denn durch die Verwendung von Coprozessoren hält Hewlett-Packard eine Lösung auch für ganz besondere Datenverarbeitungsanforderungen bereit. Und wenn sich Ihre Anforderungen ändern, paßt sich Ihr System problemlos an.

Ein weiterer Vorteil der neuen HP Precision Architecture: Die Rechnerleistung kann Schritt für Schritt durch die Aufrüstung mit weiteren Prozessoren gesteigert werden. Auch hier bleibt Ihnen Ihre Investition erhalten, denn Sie brauchen sich nicht gleich ein neues System zu kaufen, wenn Sie eine höhere Rechnerleistung benötigen. Ihr System paßt sich steigenden Leistungsanforderungen flexibel an.

E/A-Mapping

Die HP Precision Architecture wurde durch die Verwendung eines im Arbeitsspeicher abgebildeten Ein-/Ausgabe-Systems noch weiter optimiert. Das heißt, daß bestimmte Hauptspeicherbereiche speziell für die Ein-/Ausgabeverarbeitung reserviert sind. Da bei der HP Precision Architecture die Abwicklung der Ein-/Ausgabeoperationen durch einfache Lade-/Speicher-Befehle erfolgt, wird die Systemzeit eingespart, die bei den CISC-Systemen durch die dort üblichen speziellen E/A-Befehle für die Ein- und Ausgabeverarbeitung verbraucht wird. Die HP Precision Architecture sorgt für einen unkomplizierten und reibungslosen Informationsaustausch zwischen der Zentraleinheit und den E/A-Geräten und damit auch für einen größeren Systemdurchsatz.

Darüber hinaus wird eine höhere Flexibilität hinsichtlich der Sicherheit bei der Benutzung von Ausgabegeräte erreicht. Jede Ein-/Ausgabeoperation unterliegt nämlich den gleichen Sicherheitsvorkehrungen wie die Daten selbst. Dadurch wird den Benutzern ohne nachteilige Auswirkungen auf die Systemsicherheit der kontrollierte Zugriff auf die Ein-/Ausgabegeräte gewährt.

Alles fügt sich zusammen

Jedes einzelne Merkmal der neuen HP Precision Architektur hat seine besonderen Vorteile. Aber aus der Kombination dieser Merkmale und Funktionen ergeben sich weitere Vorteile. Hier leistet das organische Ganze mehr als die Summe seiner Bestandteile:

- Die HP Precision Architecture eignet sich hervorragend für die Implementierung in VLSI-Technik (Höchstintegration elektronischer Bauelemente auf einen Chip).
- Die HP Precision Architecture ermöglicht den bedarfsgerechten Einsatz verschiedener leistungsfester Rechner dieser Computerfamilie.
- Die HP Precision Architecture gewährleistet Kompatibilität für alle, die bereits mit einem HP 3000 System arbeiten.
- Die HP Precision Architecture ermöglicht eine in dieser Preisklasse besondere Leistungsfähigkeit.

Ideales für die VLSI-Technik

Der Befehlssatz der HP Precision Architecture ist in sich geschlossen, einfach und in der Hardware fest verdrahtet. Daraus ergeben sich für die Realisierung in VLSI-Technik zwei wichtige Aspekte.

Erstens: Durch einen einfachen, in der Hardware fest verdrahteten Befehlssatz wird auf dem Chip Platz für andere Funktionen frei. Bei einem Computer mit komplexem Befehlssatz werden bereits 50 % der verfügbaren Chipkapazität allein durch den Steuerspeicher beansprucht. Fällt die Steuereinheit weg, stehen diese 50 % der Chipkapazität für andere Funktionen, wie zum Beispiel Schaltkreise für das Pipelining oder für einen Cache-Speicher zur Verfügung.

Wenn die Chipkapazität besser ausgenutzt wird, werden in einem Rechner weniger Chips und Platten benötigt. Die Herstellungskosten können dadurch gesenkt und die Zuverlässigkeit gesteigert werden, denn je geringer die Zahl der Bauelemente, umso geringer die Störanfälligkeit.

Zweitens: Bei einem einfacheren Aufbau kann auch mit kürzeren Entwicklungszeiten gerechnet werden. Daraus ergibt sich, daß technische Neuerungen schneller und kostengünstiger in die Konstruktion mit aufgenommen werden können. Dank der kürzeren Entwicklungszeit können neue technische Errungenschaften sofort nutzbar gemacht werden.

Bedarfsgerechte Rechnergröße

Kernstück der HP Precision Architecture ist ein einfacher Befehlssatz, der alle in einem Computer benötigten Grundfunktionen ermöglicht. Da der Befehlssatz einen relativ geringen Umfang hat, kann er auch auf Rechnern der untersten Leistungsklasse eingesetzt werden. Andererseits ermöglicht es die besondere Leistungsfähigkeit der HP Precision Architecture, denselben Befehlssatz auch auf den Rechnern der obersten Leistungsklasse einzusetzen. Der Befehlssatz ist trotz seiner hervorragenden Eigenschaften nicht besonders umfangreich und kann deshalb mit wesentlich weniger Chips als bisher üblich auch auf besonders leistungsstarken Rechnern realisiert werden. Weniger Chips bedeuten aber auch weniger Abwärme, und damit kann auf die bei den meisten CISC-Systemen erforderlichen ausgeschütteten Kühlvorrichtungen verzichtet werden.

Sie brauchen Ihre Anwendungsprogramme nicht neu zu schreiben, wenn Sie die Vorteile der HP-Architektur nutzen wollen.

Die HP Precision Architecture ist somit das charakteristische Merkmal der gesamten HP-Produktlinie, vom Einplatzsystem bis zu den Großanlagen für mehrere hundert Benutzer.

Kompatibilität

HP Precision Architecture ist kompatibel mit im Einsatz befindlichen HP-Systemen. Da der vereinfachte Befehlssatz dem Mikrocode der Computer mit komplexem Befehlssatz sehr ähnlich ist, können CISC-Rechner im Bedarfsfall auf einem RISC-Rechner problemlos emuliert werden. Eine beruhigende Nachricht für alle Hewlett-Packard-Kunden: Bei der Entwicklung der neuen HP Precision Architecture stand die Frage der Kompatibilität mit an erster Stelle.

Sie brauchen Ihre alten Anwendungsprogramme nicht einmal umzuschreiben kommen trotzdem

in den Genuss der Vorzüge der HP precision Architecture. Auf der HP 3000 einzusetzen genügt es bereits, wenn Sie die Programme neu komplizieren. Sie werden dann von der Leistungsfähigkeit Ihrer alten Programme überrascht sein. Ohne zusätzlichen Programmieraufwand können Sie von der leistungssteigernden Wirkung der HP Precision Architecture profitieren.

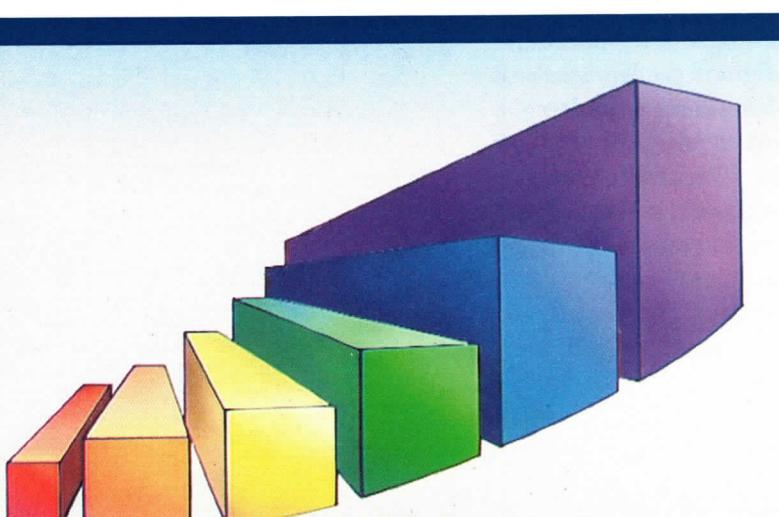
Preis-/Leistungsverhältnis

Unter dem Strich steht immer das Preis-/Leistungsverhältnis. Die HP Precision Architecture garantiert nicht nur hohe Leistung, sondern ist darüber hinaus auch noch eine Lösung mit günstigem Kosten-Nutzungs-Verhältnis.

Bei der HP Precision Architecture wurde stets auf ein einfaches Konzept Wert gelegt. Dieses Konzept der Einfachheit spart dreifach Kosten: in der Design- und Entwicklungszeit, dann bei der Herstellung und schließlich durch die hohe Zuverlässigkeit.

Ein einfaches Konzept ist leichter zu realisieren. So macht zum Beispiel ein einfacher Befehlssatz den Mikrocode überflüssig. Bei der Entwicklung müssen weniger Komponenten in Betracht gezogen werden, was sich günstig auf die Entwicklungszeit auswirkt.

*Einfachheit und Leistung
HP Precision Architecture für
eine ganze Computerfamilie*



Wofür man früher Jahre brauchte, kann man heute in Monaten erledigen, und die bei Forschung und Entwicklung erzielten Einsparungen können an Sie weitergegeben werden.

Der Herstellungsprozeß wird durch eine geringere Anzahl von Komponenten vereinfacht und beschleunigt. Je einfacher der Aufbau der Maschine, umso leichter und schneller ist sie herzustellen, und umso zuverlässiger arbeitet sie dann später. Weniger Komponenten bedeuten auch geringere Störanfälligkeit.

Das Ergebnis all unserer Bemühungen ist: Sie erhalten die Leistung, die Sie brauchen, zu einem Preis, der Ihrem Budget entspricht.

Eine zukunftsweisende Architektur

Bei der Lösung der heutigen Datenverarbeitungsanforderungen durch die HP Precision Architecture hat Hewlett-Packard die Zukunft nicht aus den Augen verloren. Die HP-Entwicklungsingenieure haben das Konzept der HP Precision Architecture aus der Summe ihrer Erfahrungen gegründet, dabei aber auch für die Flexibilität gesorgt, die für die Lösung zukünftiger Probleme nötig ist. Die HP Precision Architecture ist so konzipiert, daß sie jederzeit auch mit den neuesten Technologien realisiert werden kann und somit

Ihren wechselnden Anforderungen auch morgen entsprechen wird. Flexibilität ohne Verzicht auf Leistung ist durch ein unkompliziertes Konzept möglich geworden.

Einfachheit als Konstruktionsprinzip ist die Grundlage bei vielen erfolgreichen Systemen. Selbst der Aufbau der lebenden Organismen läßt sich auf eine ganz einfache Formel zurückführen. Ihre erstaunliche Vielfalt röhrt vom »einfachen Befehlssatz« der DNS her, die aus nur 21 einfachen Proteinen (als »Befehlen«) besteht. Die zahllosen physikalischen Merkmale der Lebewesen sind auf die unendliche Zahl von Aminosäurenkombinationen zurückzuführen. Es gibt jedoch keine komplexen, spezifischen Befehle. Denn wenn dem so wäre, wären die Organismen hochspezifisch und starr und könnten sich nicht an die wechselnden Umweltbedingungen anpassen und sich weiterentwickeln.

Bei der Entwicklung dieses neuen Systems hat Hewlett-Packard ein einfaches Konzept zugrunde gelegt und damit die Flexibilität geschaffen, die es den HP-Produkten ermöglicht, mit Ihren Anforderungen mitzuwachsen. Mit der neuen HP Precision Architecture setzt Hewlett-Packard heute Maßstäbe für die Datenverarbeitung von morgen und übermorgen.

Durch die Entwicklung eines Systems auf der Basis von Einfachheit hat Hewlett-Packard die Flexibilität erreicht, die notwendig ist, um Ihnen Computerlösungen zu bieten, die auch noch im nächsten Jahrhundert Ihren Anforderungen entsprechen.

Architektur Die Gesamtheit der Maschinenbefehle (Befehlssatz) und Adressierungsmöglichkeiten, durch deren Festlegung das Arbeitskonzept eines Computers definiert wird.

Assemblersprache Eine Programmiersprache, bei der jede von der Zentraleinheit auszuführende Operation durch einen symbolischen Befehl dargestellt wird. Mit Hilfe der Assemblersprache kann die Maschinensprache in einer für den Menschen besser verständlichen Form dargestellt werden. Ein als Assembler bezeichnetes Programm übersetzt die in der Assemblersprache geschriebenen Befehle in die Befehle der Maschinensprache.

Ausführungseinheit Der Teil der Zentraleinheit, der das Rechenwerk und die Register umfaßt. Die Register dienen zur Speicherung der Daten, die im Rechenwerk verarbeitet werden.

Befehlssatz Die festgelegte Gesamtheit der Maschinenbefehle, die ein Computer verstehen und ausführen kann. Mit der Definition des Befehlssatzes ist gleichzeitig auch die Architektur des Computers festgelegt.

Cache-Speicher Ein kleiner, sehr schneller Pufferspeicher zwischen dem Hauptspeicher und der Zentraleinheit, der ein auszuführendes Programm und die dazu benötigten Daten entweder ganz oder zum Teil für den schnellen Zugriff durch die Zentraleinheit bereithält.

COBOL (Common Business Oriented Language) Eine dem Englischen ähnliche, höhere Programmiersprache, die hauptsächlich zur Programmierung von kommerziellen Anwendungsprogrammen dient.

Computer mit komplexem Befehlssatz (CISC= Complex Instruction Set Computer) Ein Computer, dessen Architektur sich auf die Mikroprogrammierung und die Verwendung komplexer Befehle gründet.

Coprozessor Ein zusätzlicher Prozessor mit Sonderfunktionen, der die Zentraleinheit bei der Ausführung von speziellen Operationen, wie zum Beispiel Gleitkommarechnung und grafische Datenverarbeitung, unterstützt.

Compiler Ein spezielles Programm, das die in einer höheren Programmiersprache geschriebenen Programme in die Maschinensprache übersetzt. Der Compiler ist darüber hinaus in der Lage, Programme zu diagnostizieren und auf Fehler im zu übersetzen den Anwendungsprogramm hinzuweisen.

E/A (Ein-/Ausgabe Subsystem) Die Gesamtheit der Geräte, die die Ein- und Ausgabe von Informationen in und aus der Zentraleinheit ermöglichen. Ein-/Ausgabegeräte sind zum Beispiel Arbeitsplatzrechner, Personal Computer, Band-/Plattenlaufwerke, Drucker und Plotter.

FORTRAN Eine der ersten höheren Programmiersprachen, in der hauptsächlich technische Anwendungsprogramme geschrieben werden.

Höhere Programmiersprache Programmiersprachen, wie FORTRAN und COBOL, die auf einer höheren sprachlichen Ebene angesiedelt sind als die aus »Nullen« und »Einsen« bestehende binäre Maschinensprache.

Hauptspeicher Eine Komponente des Computers, in der Daten in Binärform gespeichert werden können.

Implementierung Die Realisierung eines bestimmten Computerkonzepts mit Hilfe der geeigneten Hardware. Eine bestimmte Architektur kann mit sehr verschiedenen Schaltkreistechniken implementiert werden.

In der Hardware fest implementierte Befehle Bei festverdrahteten Computern, in denen der Befehlssatz direkt in den Registern der Zentraleinheit untergebracht ist, sind die Maschinenbefehle in Festspeichern permanent aufgezeichnet.

Integrierter Schaltkreis Ein einzelnes Halbleiter-Bauelement mit einer Vielzahl von Schaltkreisen. Die Dichte der auf einem Chip enthaltenen Schaltkreise wird als Grad der Integration bezeichnet. Die beiden wichtigsten Integrationsdichten sind: LSI (Large Scale Integration = Hochintegration) und VLSI (Very Large Scale Integration = Höchstintegrierung).

Kompatibilität Programme sind kompatibel, wenn sie auf einer Maschine entwickelt und auf einer anderen eingesetzt werden können.

Leitwerk Der Teil der Zentraleinheit, der die Ausführungseinheit steuert und den Ablauf der Befehlszyklen überwacht.

LOAD (Laden) Ein Maschinenbefehl, der bewirkt, daß die Zentraleinheit Daten aus dem Speicher holt und in einem Register ablegt.

Makrobefehle Bei Rechnern mit Mikroprogrammierung müssen die Maschinenbefehle in eine Reihe von Mikrobefehlen zerlegt werden, die der Rechner dann unmittelbar ausführen kann. Die (komplexen) Maschinenbefehle werden als Makrobefehle oder, zusammenfassend, als Makrocode bezeichnet.

Maschinensprache Der für die Zentraleinheit verständliche Binärkode. Alle in höheren Programmiersprachen geschriebenen Programme müssen in die Maschinensprache übersetzt werden, bevor sie von einem Computer verarbeitet werden können.

Maschinenzyklus Die Zeit, die ein Rechner benötigt, um eine elementare Operation auszuführen.

Mikrobefehle Bei Computern mit Mikroprogrammierung müssen die Makrobefehle vor ihrer Ausführung durch die Zentraleinheit zuerst in Befehle einer niedrigeren Ebene umgewandelt werden. Diese Befehle der niedrigeren Ebene werden als Mikrobefehle oder, zusammenfassend, als Mikrocode bezeichnet.

Mikroprogrammierung Ein Konzept, in dem eine komplexe Rechnerarchitektur in Form einer Folge von in einem Steuerspeicher gespeicherten Mikrobefehlen implementiert ist.

Multiprozessoren Ein Prozessor, der zum Hauptprozessor hinzugefügt werden kann, um die Rechnerleistung zu steigern.

OPcode (Operationscode) Der Teil eines Maschinenbefehls, in dem festgelegt ist, welche Operation die Zentraleinheit mit den Daten durchführen soll. Dem OPcode folgen normalerweise ein oder mehrere Operanden. Im OPcode sind Operationen, wie zum Beispiel LADEN, SPEICHERN und VERZWEIGEN.

Operand Der Teil eines Maschinenbefehls, auf den die im Operationscode vorgegebene Operation angewandt wird. Mit dem Assemblersprachenbefehl STORE A (SPEICHERN A) wird zum Beispiel festgelegt, daß die Speicherung im Hauptspeicher an der mit A bezeichneten Stelle erfolgen soll.

Optimierender Compiler Ein hochentwickelter Compiler, der bei der Übersetzung von Programmen der höheren Programmiersprachen Mängel beseitigt und unnötige Befehle entfernt. Programme, die mit einem optimierenden Compiler übersetzt werden, laufen schneller und nehmen weniger Speicherkapazität in Anspruch.

Parallelverarbeitung von Befehlen (Pipelining) Ein Prozeß, bei dem mehrere Maschinenbefehle gleichzeitig (simultan) abgearbeitet werden.

Programm Eine Befehlsfolge, die einem Computer die Ausführung bestimmter Aufgaben vorschreibt.

Register Kleine, schnelle Rechnerkomponenten im Rechnerwerk der Zentraleinheit, in denen Informationen vorübergehend gespeichert werden.

RISC-Computer oder Computer mit vereinfachtem Befehlssatz (RISC = Reduced Instruction Set Computer) Ein Computer, dessen Architektur sich durch einen einfachen, in der Hardware fest verdrahteten Befehlssatz auszeichnet.

Semantische Lücke Die Diskrepanz zwischen dem Bedeutungsinhalt der für einen Computer verständlichen binären Maschinensprache und den für die Programmierung der Anwendungsprogramme verwendeten höheren Programmiersprachen.

Steuereinheit Ein spezieller, schneller Speicher, in dem die Mikrobefehle stehen, die in einer Architektur mit Mikroprogrammierung benötigt werden.

STORE (SPEICHERN) Ein Befehl, der bewirkt, daß die Zentraleinheit Daten aus einem Register holt und sie in den Speicher überträgt.

Transistor Ein Halbleiterbauelement, das als elektronischer Schalter dient.

Verzögerte Verzweigung Ein Maschinenbefehl, der die Reihenfolge von Befehlen, die in der Zentraleinheit ausgeführt werden sollen, regelt.

ZE (Zentraleinheit) Der Teil des Computers, in dem die Daten verarbeitet werden. Die Zentraleinheit besteht aus dem Rechenwerk und dem Leitwerk.

Hewlett-Packard Deutschland

Hauptverwaltung

Herrenberger Straße 130
7030 Böblingen
Tel. (07031) 14-0

Vertriebszentrale

Hewlett-Packard-Straße
6380 Bad Homburg v.d.H.
Tel. (06172) 400-0

Verbindungsstelle Bonn

Friedrich-Ebert-Allee 26
5300 Bonn 1
Tel. (0228) 234001

Werk I Böblingen

● Medizinelektronik
Herrenberger Straße 110
7030 Böblingen, Tel. (07031) 14-0

Werk II, III, IV Böblingen

● Elektronische Meßtechnik
● Techn.-wissenschaftl. Computersysteme
● Kommerzielle Informationssysteme

Herrenberger Straße 130
7030 Böblingen, Tel. (07031) 14-0

Werk Waldbronn

● Analytische Meßtechnik
Hewlett-Packard-Straße
7517 Waldbronn 2, Tel. (07243) 602-0

Support-Zentrum Ratingen

Berliner Straße 111
4030 Ratingen
Tel. (02102) 494-500

Reparatur-Zentrum Frankfurt

Berner Straße 117
6000 Frankfurt/Main
Tel. (069) 500001-22

Zentrale für Ersatzteile, Verbrauchs- material und Zubehör

Dornierstraße 7
7030 Böblingen
Tel. (07031) 14-0

Vertriebszentrum Nord

Kapstadttring 5
2000 Hamburg 60
Tel. (040) 63804-0

Geschäftsstelle Hannover

Heidering 37-39
3000 Hannover 61
Tel. (0511) 5706-0

Geschäftsstelle Berlin

Keithstraße 2-4
1000 Berlin 30
Tel. (030) 219904-0

Vertriebszentrum Ratingen

Berliner Straße 111
4030 Ratingen
Tel. (02102) 494-0

Geschäftsstelle Dortmund

Schleefstraße 28
4600 Dortmund 41
Tel. (0231) 45001-0

Vertriebszentrum Bad Homburg

Hewlett-Packard-Straße
6380 Bad Homburg v.d.H.
Tel. (06172) 400-0

Geschäftsstelle Mannheim

Rossauer Weg 2-4
6800 Mannheim 31
Tel. (0621) 7005-0

Vertriebszentrum Böblingen

Schickardstraße 2
7030 Böblingen
Tel. (07031) 645-0

Geschäftsstelle Karlsruhe

Ermlis-Allee
7517 Waldbronn 2
Tel. (07243) 602-0

Geschäftsstelle Neu-Ulm

Messerschmittstraße 7
7910 Neu-Ulm
Tel. (0731) 7073-0

Vertriebszentrum München

Eschenstraße 5
8028 Taufkirchen
Tel. (089) 61207-0

Geschäftsstelle Nürnberg

Emmericher Straße 13
8500 Nürnberg 10
Tel. (0911) 5205-0

Hewlett-Packard Schweiz

Hewlett-Packard (Schweiz) AG
Abteilung Information
Allmend 2
CH-8967 Widen
Tel. (057) 312111

Hewlett-Packard (Schweiz) AG
Département Informatique et Systèmes
7, Rue du Bois-du-Lan
CH-1217 Meyrin 1/Schweiz
Tel. (022) 831111

Schwamendingenstraße 10
CH-8050 Zürich
Tel. (01) 3158181

Hewlett-Packard Österreich

Hewlett-Packard Ges.m.b.H
Lieblgasse 1, A-1222 Wien
Tel. (0222) 2500-0

Verkaufsbüro Graz
Grottenhofstraße 94, A-8052 Graz
Tel. (0316) 291566-0

Sozialistische Staaten

Hewlett-Packard Ges.m.b.H
Lieblgasse 1, A-1222 Wien
Tel. (0222) 2500-0

Europa-Zentrale

Hewlett-Packard S.A.
150, route du Nant-d'Avril
1217 Meyrin 2-Genf/Schweiz
Tel. (022) 838111