

Optimierte Architekturen  
für Hochleistungsrechner der 90er Jahre

- Disposition -

Was ist zu tun, um ab Mitte der 90er Jahre noch mit Universalrechnern oberhalb der PC- Szene Geld verdienen zu können? Welche technischen Innovationen sind dafür notwendig bzw. sinnfällig? Hat es überhaupt Zweck, sich weiterhin um grundsätzliche technische Verbesserungen des Universalrechners zu bemühen?

Dafür sind folgende Einzelfragen zu betrachten:

Wodurch ist der Stand der Technik gekennzeichnet?  
Wie ist er zu übertreffen (i. S. der Zielstellung)?  
Welche Möglichkeiten gibt es überhaupt?  
Schließlich sind Kosten und Nutzen zu bewerten.

Zur allgemeinen Methodenlehre: Es muß ein iterativer Prozeß in Gang gesetzt werden, der die Bewertung von Gegebenem, die Analyse von Anforderungen und innovative Vorschläge, die ihrerseits zu bewerten sind, umfaßt.

Man muß neue Architekturen ausarbeiten und damit gleichsam Fragen an die Technologie und an den Markt stellen:

Läßt sich das kostengünstig fertigen?

Besteht dafür ein Erfordernis bzw. wird es entstehen, wenn die Voraussetzungen einmal da sind (Marktaufbereitung)?

In diesem Sinne werden folgende Prinzipien als wesentlich angesehen:

1. das Ressourcen-Paradigma
2. das objektorientiertes Verarbeitungsmodell
3. das Prinzip der kontrollierten Kardinalität
4. das Prinzip der Vergegenständlichten Abstraktionen.

**Das Ressourcen-Paradigma**

Jede Rechnerstruktur läßt sich als Sammlung von Hardware-Ressourcen auffassen. Es ist einleuchtend, daß in erster Linie die Ressourcen an sich die Verarbeitungsleistung bestimmen: ist beispielsweise ein Multiplikationswerk vorhanden, so wird die Maschine schneller multiplizieren als mit "shift-add"- Folgen, gleichgültig nach welcher Lehre die Architektur ausgebildet ist. Sind mehrere Werke vorgesehen, so wird die Leistung entsprechend ansteigen, gesetzt den Fall, alle Werke können gleichzeitig mit nützlicher Arbeit beschäftigt werden. Die Verarbeitungsleistung einer jeden Rechnerstruktur kann prinzipiell folgenden Wert nicht übersteigen:

Anzahl der nutzbaren Ressourcen

-----  
Dauer des Maschinenzklus

Sind die Verarbeitungsressourcen an sich nach Art und Anzahl gegeben, so steht an zweiter Stelle die Gestaltung der Speichermittel und der Verbindungsstrukturen. Diese müssen so ausgebildet werden, daß eine weitestgehende Annäherung an ideale Betriebsverhältnisse erreichbar ist. Ideale Betriebsverhältnisse sind gegeben, wenn alle Ressourcen ständig mit nützlicher, also zum Lösen von Anwendungsaufgaben unmittelbar beiträgender Arbeit beschäftigt werden können.

Erst an dritter Stelle steht die Gestaltung der Befehlsliste, und zwar auch unter dem Gesichtspunkt, die Ressourcen weitestgehend mit nützlicher Arbeit beschäftigen zu können: es geht um eine zweckmäßige Codierung bzw. Verpackung (Formatierung) der gespeicherten Steuerangaben, die zur Steuerung der Ressourcennutzung und des Informationsflusses notwendig sind.

Von diesem Ansatz aus kann die Ausarbeitung einer Rechnerarchitektur als systematisch lösbare Aufgabe angegangen werden. Kennzeichnend für eine solche Rechnerarchitektur ist dann die Ressourcen-Algebra, nicht mehr die Befehlsliste (Hinweis: Ressourcen-Algebra als universelle Zwischensprache).

Die weithin eingeführten Architekturen sind durch eine gewisse charakteristische Ressourcen-Ausstattung gekennzeichnet (beschreibbar durch die Signatur der entsprechenden Ressourcen-Algebra). Die einzelnen Ressourcen selbst haben einen solchen Entwicklungsstand erreicht, der revolutionäre Veränderungen durch Zufallserfindungen kaum noch erwarten läßt. Wo sind also die Quellen für künftige weitere Verbesserungen zu suchen?

1. In der Auswahl der Ressourcen: Es muß nach geeigneteren Ressourcen gesucht werden, als in den üblichen Universalrechnern vorgesehen sind.

2. In der Anzahl der Ressourcen: Es sind mehr Ressourcen (z. B. Verarbeitungswerke) vorzusehen als in den eingeführten Architekturen; neue Architekturen sind von Anfang an dafür auszulegen.

3. In der Optimierung innerhalb der einzelnen Ressourcen.

#### Das objektorientierte Verarbeitungsmodell

Die Objektorientierung wird als zwingende Notwendigkeit angesehen, um große Softwarekomplexe beherrschen zu können. Die schaltungstechnische Unterstützung ist derzeit (nach dem spektakulären Mißerfolg des iAPX 432) aus der Mode gekommen. Es ist aber notwendig, diese Frage erneut anzugehen. Das ist lösbar, im besonderen wenn man von vornherein über den Aufwandsrahmen des Mikroprozessors hinausgeht. Ein entsprechend konzipiertes System wird herkömmlichen Architekturen deutlich überlegen sein.

#### Das Prinzip der kontrollierten Kardinalität

Mit diesem Prinzip wird die Endlichkeit jeglicher technischer Ressourcen explizit für die Architekturdefinition nutzbar gemacht. Damit werden der Software-Optimierung

festе Ziele geboten.

Grundsatz-Überlegung: Es gibt viele gute Prinzipien, die sich in überschaubarem Umfang sinnvoll verwirklichen lassen, in großem Maßstab aber stets irgendwelche Kompromisse erfordern. Die Größenordnungen sollen deshalb nach Maßgabe der Beherrschbarkeit festgeschrieben werden. Beispiel: Ein "crossbar"-Verbindungsnetzwerk läßt sich in der Größenordnung 16 x 16 technisch beherrschen, in der Größenordnung 256 x 256 hingegen nicht. Deshalb werden die Speichermittel so konfiguriert, daß zwischen jeweils 16 Modulen solche Verbindungsstrukturen vorgesehen sind. Das wird in der Architektur festgeschrieben. Somit kann sich jeder Systemprogrammierer oder Compilerautor auf die entsprechenden Eigenschaften der Hardware verlassen. (Dazu gehört auch, daß der Programmierer jeden Maschinenzklus unter Kontrolle hat. Im Gegensatz zu eingeführten Architekturen ist praktisch das Mikroprogramm-niveau von der Software direkt nutzbar. Die Absicht besteht darin, durch innovative Befehlsgestaltung die Geschwindigkeit von Mikroprogrammen und die Eleganz einer guten Befehlsliste zu vereinen.)

#### Das Prinzip der Vergegenständlichten Abstraktionen

Dieses Prinzip dient dazu, "geeignete" Ressourcen auszuwählen und Ziele für deren Optimierung zu setzen.

Bisher wurden Rechnerarchitekturen vorwiegend auf dem Befehlsniveau optimiert, wozu meist auf vorhandene Programme als Grundlage für statistische Auswertungen und meßtechnische Untersuchungen zurückgegriffen wurde.

Sich darauf zu beschränken heißt aber, herkömmliche Programmiergepflogenheiten auf neue Architekturen zu übertragen und somit weitere Innovationsgelegenheiten gar nicht zu erkennen.

Deshalb ist ein gleichsam ganzheitliches Herangehen zwingend geboten: die eigentlichen Anwendungserfordernisse sind in reiner Form, unbeeinflusst von den bisherigen Gepflogenheiten, zu betrachten, und die Konsequenzen sind bis in die Einzelheiten der Hardware zu verfolgen. Dafür sind grundlegende Konzepte der Informationsverarbeitung, der angewandten Mathematik, der Linguistik usw. systematisch abzusuchen, um aus den Grundlagen-Wissenschaften heraus allgemein anerkannte Operationen und Informationsstrukturen aufzufinden, die für eine Umsetzung in Schaltmittel ("Vergegenständlichung") in Frage kommen (ein Beispiel ist eine universelle Hochgenauigkeits-Numerik, die die Integer-, Gleitkomma- und BCD-Numerik eingeführter Architekturen ersetzen kann).

#### Folgerungen und Fragen

- Der Einzelprozessor ist nach wie vor sinnvolles Ziel für innovative Bemühungen. Dazu ist die Gesamtheit der zwischenzeitlich angesammelten Erkenntnisse aufzuarbeiten. Davon ausgehend sind die Möglichkeiten der modernen Schaltungstechnik auf innovative Weise zu nutzen.

- Die Leistungsgrenzen der v. Neumann-Maschine sind erkennbar. Parallelverarbeitungssysteme mit zuhandenen Prozessoren

sind aber keine guten, wirklich überlegenen Universalrechner. Neue Prozessoren müssen vielmehr: 1) in sich und 2) für den Einsatz in Parallelverarbeitungssystemen optimiert werden.

- Die Leistungsgrenzen von Mikroprozessoren sind an sich gegeben (durch die Ressourcensammlung als solche, die Zykluszeit und gewisse technische Nebenbedingungen). Es ist eine Angelegenheit der Optimierung im einzelnen, daß diese Grenzen in der praktischen Anwendung tatsächlich "ausgefahren" werden können (Ziel: durchschnittliche Skalarleistung muß der Maximalleistung nahekommen).

- Darüber hinaus ist noch mehr Leistung nur von neuartigen Ressourcensammlungen zu erwarten (mehr und besser optimierte Ressourcen). Grob abzuschätzen sind 10-fache Leistung mit 2-4 fachen Hardware- Systemkosten. Eine Verbesserung des Preis-Leistungs-Verhältnisses von 1:3 wird als möglich angesehen (erscheint durch ingenieurmäßige Bemühungen erreichbar, wesentlich mehr sollte man realistischerweise nicht erwarten). Das aber nur bei über-Bord-Werfen traditioneller Konzepte. Stichworte für Innovation: integrierte Wissensbasis statt file-System, universelle Hochgenauigkeits-Numerik, universelle Binärvektorverarbeitung für nichtnumerische Probleme, Eimerstruktur statt Bytestruktur, Behälterverwaltung statt Seitenverwaltung für die Speicherorganisation. Die Effekte der einzelnen Innovationen sind noch zu bewerten, auch ist das Innovationspotential keineswegs erschöpft.

- Fragen:

Wird so etwas akzeptiert werden?

Ist man bereit, für solche grundsätzlichen Verbesserungen Software neu zu schreiben (Wertumfang)? Welcher Softwarebestand ist unbedingt zu halten, erfordert also Kompatibilitäts-Vorkehrungen (die technisch grundsätzlich möglich sind)?

Wird das "dusty-deck"- Problem demnächst Folgen zeigen?

Ist ein massiver Nachholebedarf für wirklich innovative universelle Rechnerarchitekturen in den 90er Jahren zu erwarten: 1) als Konsequenz des dauernden Hinauszögerns eines Systemwechsels; 2) um die Softwarekrise zu überwinden; 3) weil wirklich nichts mehr "wie alt" geht usw., etwa vergleichbar mit dem Nachholebedarf der Zivilluftfahrt?