

Technische Lösungsansätze für Rechnerarchitekturen
auf Grundlage vergegenständlichter Abstraktionen

- Ein Arbeitsbericht -

Von: Dr.-Ing. Wolfgang Matthes

Stand: 0.1 (unfertig)

Vom: 19. 7. 1989

Inhalt

1. Einführung
2. Prozessorkerne
 - 2.1. Ziele
 - 2.2. Grundlagen
 - 2.3. Prozessorkern 1 (PK 1)
 - 2.3.1. Aufbau und Wirkprinzipien
 - 2.3.2. Übersicht über die Ressourcen
 - 2.3.3. Speicherzugriffsprinzipien
 - 2.3.4. Befehlsliste
 - 2.4. Prozessorkern 2 (PK 2)
 - 2.4.1. Aufbau und Wirkprinzipien
 - 2.4.2. Übersicht über die Ressourcen
 - 2.4.3. Speicheranschluß- und -zugriffsprinzipien
 - 2.4.4. Befehlslisten
3. Operationswerke
4. Speicherstrukturen
5. Systemstrukturen
6. Hinweise zur Schutzrechtsarbeit
7. Literaturverzeichnis

1. Einführung

Diese Schrift bezeichnet den Beginn erfinderischer Bemühungen um technische Lösungen für leistungsoptimierte Rechnerstrukturen bzw. -architekturen im Rahmen der Vorstellungen, wie sie in der Denkschrift /1/ dargelegt wurden.

Der Zweck der Schrift besteht darin, die ersten Ansätze übersichtsweise zusammenzufassen, um eine Grundlage für Planungsarbeiten, Patentrecherchen usw. zu geben.

Die Schrift enthält Ausschnitte des Erkenntnis- bzw. Bearbeitungsstandes, wie er zum Termin gemäß Titelblatt gegeben ist. Sie unterliegt nicht dem Änderungsdienst. Die weiteren Arbeitsergebnisse werden in den künftigen Architekturbeschreibungen bzw. Schutzrechtsanmeldungen dargelegt werden.

Alle Bemühungen sind auf das Ziel gerichtet, Universalrechner höchsten Leistungsvermögens zu schaffen. Hier geht es vorerst um den einzelnen Rechner, dessen Strukturen und Wirkprinzipien zweckgerecht ausgestaltet werden sollen. Theoretische Grundlagen dazu sind in /2/ dargestellt worden. Nachfolgend wird daraus der methodische Ansatz wiedergegeben:

1. Für bedeutsame Anwendungsgebiete der Rechentechnik werden grundsätzliche und leistungsbestimmende Abstraktionen (Datentypen + Operationen + Ablaufprinzipien) gesucht.
2. Diese Abstraktionen werden exakt beschrieben.
3. Es werden Hardwarestrukturen entwickelt, um diese Abstraktionen so effektiv wie möglich vergegenständlichen zu können.
4. Auf Grundlage dieser Strukturen werden Rechnerarchitekturen definiert, die die erforderliche Universalität gewährleisten.
5. Der Auswahl der Abstraktionen und der Ausgestaltung der Schaltungslösungen wird die Tiefenstruktur der jeweiligen Prozesse der Informationsverarbeitung zugrunde gelegt und nicht eine Oberflächenstruktur, wie sie durch eine konkrete Programmiersprache, eine Betriebssystem-Umgebung, einen Standard für die Datendarstellung u. dergl. gegeben ist.
6. Für jede einzelne Abstraktion wird versucht, die Schaltmittel zu deren Implementierung so leistungsfähig wie möglich auszulegen; im Rahmen der technischen Beherrschbarkeit wird Hardware im Interesse des Leistungsvermögens rücksichtslos eingesetzt: sowohl hinsichtlich der Aufwendungen als auch der strukturellen Gestaltung.

Die eigenen Vorstellungen gehen von 3 Ansätzen aus:

1. Optimierung der Maschinenbefehle (Stichworte: RISC-Prinzipien, Prinzipien der Mikroprogrammsteuerung, schaltungstechnische Unterstützung höherer Programmiersprachen)

2. Nutzung des innewohnenden Parallelismus in Programmen (z. B. VLIW- oder Datenflußprinzipien)

3. Entwurf von Spezialmaschinen (Einfügen solcher Schaltmittel in die Struktur von Universalrechnern; Nutzung einschlägiger Entwurfsprinzipien und Erfahrungen für die Auslegung von Universalrechner-Schaltungslösungen).

Die Erfahrungsbasis für das Auffinden von Abstraktionen ist zunächst durch die eingeführten Maschinenarchitekturen und Programmiersprachen gegeben.

Es gehört zum methodischen Prinzip, Fragen der "Kompatibilität" nicht jenen Rang einzuräumen, den sie derzeit in der Praxis innehaben; schließlich soll erkundet werden, welche Leistungssteigerungen im Rahmen von Universalrechnerkonzepten überhaupt noch zu erwarten sind. Wenn es um das Ausführen technischer Lösungen geht, so dürfen natürlich solche Fragen nicht vernachlässigt werden. Sie werden dann folgendermaßen Berücksichtigung finden:

Wenn neue Grundsatzlösungen Leistungsvorteile versprechen, die eine Verwirklichung rechtfertigen, so sind weitere erfinderische Bemühungen notwendig, um die Paßfähigkeit zu gegebenen System- bzw. Anwendungsumgebungen sicherzustellen. Das betrifft sowohl Software als auch Hardware. Erweisen sich hingegen neue Grundsatzlösungen als nicht ausreichend überlegen, so sind künftige erfinderische Bemühungen eher auf Verfeinerungen gegebener Rechnerarchitekturen zu richten, beispielsweise dadurch, daß versucht wird, die gewonnenen theoretischen Erkenntnisse auf die Implementierung solcher Architekturprinzipien anzuwenden.

Die folgenden Skizzen technischer Lösungsvorstellungen umreißen nicht das gesamte Vorhaben; sie sind vielmehr als Ausdruck des derzeitigen Erkenntnisstandes anzusehen (für Vorstellungen zu Planabläufen sei auf /1/ verwiesen, besonders auf Anhang 8)..

Die Darstellung soll drei Zwecken dienen: Sammlung technischer Ideen, Vorbereitung der Schutzrechtsarbeit und Grundlegung für künftige geschlossene Darstellungen eigener Rechnerarchitekturen. Somit sind gewisse Ungenauigkeiten, Überschneidungen usw. nicht zu vermeiden.

2. Prozessorkerne

2.1. Ziele

Es sind 2 Prozessorkerne vorgesehen:

● PK 1 als Prozessorkern eines Hochleistungsrechners (Arbeitsbezeichnung: System.016). Dabei ist absolutes Leistungsvermögen bei technischer Beherrschbarkeit das wichtigste Entwicklungsziel. Es ist nicht entscheidend, daß PK 1 mit zuhandenen Technologien auf einem Schaltkreis implementiert werden kann. Erste Vorstellungen:

- Aufbau mit mehreren Standardzellen- Schaltkreisen
- Verarbeitung von 96-bit-Maschinenworten mit zusätzlichen TAG- Bits
- 96-bit-Steuerworte (Resource Control Words RCW); davon können intern mehrere gleichzeitig aktiv sein
- Mehrfachverzweigungen
- Anschluß an mehrstufige heterogene Speicheranordnungen (Datenbasis-, Laufzeit-, Steuer- und Operandenspeicher; davon manche aktiv wirkend; s. auch Abschnitt 4)
- parallele Verarbeitung von 64 bit
- schaltungstechnische Unterstützung der objektorientierten Verarbeitung.

● PK 2 als schneller, vielseitiger und kostengünstiger Mikrocontroller- Schaltkreis bzw. als einheitliches Architektur- und Schaltungskonzept für verschiedene Ausführungen, die im Rahmen von Zellenbibliotheken bereitgestellt werden können (Arbeitsbezeichnung: System.015). Damit soll versucht werden, für den Bereich vom bisher üblichen Einchip- Mikrorechner bis zum 32-bit-Mikrocontroller eine einheitliche, rechtsmängel- freie und leistungsmäßig überlegene Architektur zu schaffen.

Das anwendungstechnische Hauptziel besteht darin, anstelle der bisherigen Baukastenlösungen auf Grundlage von Leiterplatten und standardisierten Bussystemen etwas Neues bereitzustellen: einen Baukasten aus Standardzellen mit untereinander genormten Anschlüssen, so daß die einzelnen Anwendungslösungen durch Zusammenfügen von Standardzellen auf ASIC- Schaltkreisen, erforderlichenfalls durch Zusammenschalten mehrere solcher Schaltkreise auf einer Leiterplatte zusammengesetzt werden können.

Einsatzfälle sind z. B.:

- Mikrocontroller in Steuerungssystemen (programmierbarer Kern entsprechender ASIC- Schaltkreise)
- Baustein komplexer verteilter Steuerungssysteme
- Baugruppe umfangreicher informationsverarbeitender Systeme; namentlich für das System.016, z. B. als Service-, Speicherverwaltungs- und Peripherieprozessor
- eigenständiges Informationsverarbeitungssystem mit standardisierten Schnittstellen (Beispiel: Accelerator- Baugruppe für PC).

Dazu sind folgende Anforderungen zu erfüllen:

- der PK 2 muß auf einen Schaltkreis passen; die Grundausführung sollte nicht mehr als 50 000 Transistoren umfassen
- einfache, leicht verifizierbare interne Wirkprinzipien
- Nutzung im Rahmen herkömmlicher Adressierungs- und Speicherkonzepte (Stichwort: 32-bit-Verarbeitung)
- variable Anordnung von Speicheradaptern und on-chip-Speichermitteln, Brauchbarkeit auch bei externen 8-bit-Datenwegen
- variable Nutzung aller chip-internen Ressourcen je nach Einsatzfall
- internes Multitasking (hardware-unterstützt) für maximal 16 Tasks
- sehr schnelle Kontextumschaltung
- sehr flexible direkte E-A-Steuerungsmöglichkeiten
- Vorkehrungen für mehrfache serielle Anschlüsse (ähnlich Transputer)
- Vorkehrungen für Selbstprüfung, "debugging" und Aufbau fehlertoleranter Systeme
- architekturseitige Datenstrukturdefinitionen im üblichen Rahmen (16, 32, 64 bit; ohne zusätzliche TAG-Bits); interne Verarbeitungsbreite: 32 bit.

Es wird nicht als notwendig angesehen, daß beide Architekturen zueinander "voll kompatibel" (wie etwa die Modelle des System/370) sind. Sie sind aber nach einheitlichen Grundprinzipien auszuarbeiten; gleiche Operationen auf gleiche Datenstrukturen müssen gleiche Wirkungen hervorbringen. PK 2 muß in der Lage sein, die Architekturdefinition von PK 1 programmseitig zu emulieren, und zwar mit deutlicher Überlegenheit gegenüber 80386. Die Architektur von PK 2 sollte gleichsam als maßstäbliche Verkleinerung jener des PK 1 erscheinen.

2.2. Grundlagen

Die Prinzipien sind ausführlich in /2/ dargestellt. Sie werden hier nur auszugsweise skizziert.

Datenstrukturen

Die einzige elementare Datenstruktur ist der Binärvektor. Ein Binärvektor kann sowohl als numerische als auch als nichtnumerische Angabe interpretiert werden. Es wird ausschließlich die binäre Codierung verwendet, und zwar wahlweise je nach Zweckmäßigkeit:

- durch Aneinanderreihen von Bits in einem Binärvektor
- durch Angabe einer binär codierten Ordinalzahl in eine endliche Menge
- durch Kombination dieser Möglichkeiten ("Gleitkommadarstellung")
- durch Strukturbeschreibung
- durch ternäre Vektoren.

Eine binär codierte Ordinalzahl von n bits kann ein Element aus einer Menge der Mächtigkeit 2^n auswählen. Ein solcher

Binärvektor wird als Codon der Länge n bezeichnet.
Alle numerischen Angaben werden in letzter Konsequenz auf natürliche Binärzahlen zurückgeführt. Ganze Zahlen werden durch natürliche Zahlen mit vorangestelltem Vorzeichen ausgedrückt ("Sign-Magnitude"-Darstellung).

Schaltungsstrukturen

Grundgedanke ist die Auffassung der Schaltmittel des Prozessorkernes als strukturierte Menge von Ressourcen, gekennzeichnet durch die Speichermittel (Flipflops, Register), die in ihrer Gesamtheit den Ressourcenvektor bilden, durch die Verknüpfungsschaltungen und die Informationspfade. Das Ziel muß darin bestehen, diese Ressourcen so zweckmäßig wie möglich nutzen zu können.

Numerische Operationen

Grundsätzlich sind die 4 Grundrechenarten über Binärvektoren von 32 bit (PK 2) bzw. 64 bit (PK 1) vergegenständlicht, die als natürliche oder vorzeichenbehaftete ganze Binärzahlen interpretiert werden.

Hinweis: Während an sich angestrebt wird, alle Zuordnungen in einem Zyklus auszuführen, können Multiplikation und Division sequentiell implementiert sein. Gründe für diese Festlegung (anstelle einer programmseitigen Emulation, wie in RISC-Konzepten): konzeptionelle Einheitlichkeit für alle technischen Ausführungen und Geschwindigkeit bei vergleichsweise geringen Mehraufwendungen (nur Steuerschaltungen für die recht einfachen Abläufe; Verknüpfungsschaltungen und Datenwege sind ohnehin vorhanden).

Es wird Wert darauf gelegt, Rechenoperationen mit natürlichen Binärzahlen soweit zu beschleunigen, wie aufwandsseitig vertretbar. Weiterhin sind alle Konzepte auf den effizienten Umgang mit Binärvektoren variabler Länge (bis aufs Bit adressierbar) ausgerichtet. Auf dieser Grundlage lassen sich alle anderen arithmetischen Verknüpfungen (Gleitkomma- und rationale Zahlen) programmtechnisch implementieren.

Für leistungsfähigere arithmetische Verarbeitungsschaltungen s. Abschnitt 3.

2.3. Prozessorkern 1 (PK 1)

2.3.1. Aufbau und Wirkprinzipien

Der PK 1 ist für höchste Verarbeitungsleistung und die Zusammenarbeit mit heterogenen aktiven Speicheranordnungen vorgesehen.

Er ist nicht für Multitasking und für die direkte Ein- und Ausgabesteuerung eingerichtet. Vielmehr wird erwartet, daß er ausschließlich in Systemen eingesetzt wird, in denen andere Schaltmittel für solche Zwecke vorgesehen sind.

Die Architektur beruht auf einem Verbund von Hardwareregistern und unmittelbar nachgeordneten Verknüpfungsschaltungen, d. h. auf aktiven Registeranordnungen. Die Steuerworte haben eine Länge von 96 bit, erweitert um einige TAG-Bits. Es werden stets 4 Steuerworte parallel gelesen (eine solche Ganzheit wird als Quad bezeichnet). Es gibt verschiedene Steuerwortformate, die jeweils durch die TAG-Bits eindeutig gekennzeichnet werden. Die Steuerworte können parallel Verknüpfungs- und Verzweigungsangaben enthalten. Innerhalb eines Quad ist die Auswahl des jeweils aktiven Steuerwortes durch Vierfach-Spätverzweigung möglich (2 unabhängig wählbare Bedingungen, die auch als Ergebnis der aktuellen Verknüpfungen auftreten können). Das nachfolgende Quad wird gleichzeitig mit der Verknüpfung adressiert.

Alle Register-Flipflops werden als Ressourcenvektor betrachtet, und es ist möglich, beliebige Teilvektoren zu selektieren, untereinander zu verknüpfen und wieder in den Ressourcenvektor einzufügen. Auch die Steuerregister sind Bestandteil des Ressourcenvektors, so daß Steuerworte im aktuellen Quad intern modifiziert werden können. Es gibt Steuerregister für 2 Quads: ein Registersatz für das Maschinenbefehlsniveau und einer für das Emulatorenniveau. Damit läßt sich eine mikroprogrammtechnische Emulation gewährleisten (Rückführung komplexer Operationen auf elementare), wobei das aufrufende Quad zur Parameterübergabe für das emulierende verwendet werden kann.

Unterbrechungen laufen so ab, daß in die Steuerregister des Emulatorenniveaus zwangsweise bestimmte Steuerwortfolgen eingespeist werden. Auf sinngemäße Weise ist die Ausnahmebehandlung organisiert (die betreffenden Steuerregister sind standardmäßig mit einem entsprechenden Quad belegt, das die Ausnahmebehandlung startet; andere Emulatorroutinen müssen die Ausnahmen selbst behandeln).

Alle Verknüpfungen sind funktionelle Zuordnungen; die Verknüpfungsergebnisse beliebiger Registerbelegungen sind nach der entsprechenden Zykluszeit ständig verfügbar.

Die Parameterübergabe ist durch Adressierung auf verschiedenen Ebenen und durch Direktwertübergabe gewährleistet. Dafür gibt es auch Formate, die 2 Steuerworte umfassen (32 bit Verknüpfungs- und Verzweigungs- bzw. Bestimmungssteuerung, 2 x 80 bit Direktwert-Argumente). Ansonsten sind zusammengehörende Angaben in einem Steuerwort codiert, es gibt aber auch Steuerwortformate mit 3 aufeinanderfolgenden Befehlen zu je 32 bit.

2.3.2. Übersicht über die Ressourcen

1. Verarbeitung

a) Addition

Die Inhalte der 64-bit-Register A1, A2 ergeben eine Summe von 64 bit sowie ein Übertragsbit.

b) Subtraktion

Die Inhalte der 64-bit-Register S1, S2 ergeben eine Differenz von 64 bit sowie ein Übertragsbit.

c) Multiplikation

Die Inhalte der 64-bit-Register M1, M2 ergeben ein Produkt von 128 bit.

d) Division

Der Inhalt des 128-bit-Registers D1 wird durch den Inhalt des 64-bit-Registers D2 dividiert; das ergibt einen Quotienten sowie einen Rest von je 64 bit.

e) Akkumulation

Der Inhalt des 80-bit-Registers C1 wird zum Inhalt des 128-bit-Registers C2 addiert, das Resultat wird automatisch nach C2 zurückgeführt. Zusätzlich entsteht ein Übertragsbit.

f) Logische Verknüpfungen

Es sind die 64-bit-Register E1, V1 und E2, V2 vorgesehen. Zwischen deren Inhalten ist eine Vielzahl von binären und ternären Verknüpfungen vorgesehen.

2. Binärvektortransporte

Aus dem Ressourcenvektor sind beliebige Teile selektierbar. Es können parallel jeweils 64 bit verarbeitet werden. Für jeden Subvektor wird ermittelt:

- die Lage in einem Intervall, das durch 2 Grenzregister LB, HB gegeben ist
- die Position der 1. Eins
- die Position der letzten Eins
- die Anzahl der Einsen
- die Position in einem Intervall, dessen Grenzen durch 2 Register LB, HB gegeben sind.

Register, die in bestimmten Abläufen nicht für die funktionellen Zurechnungen benötigt werden, sind im Sinne von Universalregistern beliebig als Informationsspeicher nutzbar.

3. Adressierung

Aus logischer Sicht ist der PK 1 mit 12 Zugriffspfaden versehen:

1. 1 Pfad für Steuerwortzugriffe (Quad-Adresse in einem Schnellspeicher)

2. 4 Pfade zu Zugriffstabellen in Schnellspeichern. Es sind 4 Ebenen vorgesehen: rufendes Programm, aktuelles Programm, gerufenes Programm und System. Jede solche Zugriffstabelle (ART) hat maximal 4k Einträge zu 96 bit (um TAGs erweitert).

3. 4 Pfade zum Laufzeitspeicher

4. 3 Pfade zu Operanden- Schnellspeichern

Ein Programm kann maximal 4k Quads umfassen; die Adressierung ist für 64 k Quads ausgelegt.

Jeder Operanden- Schnellspeicher kann 16k Worte zu 64 bit (mit je 16 Zusatzbits) haben.

Der Laufzeitspeicher kann sehr groß sein. Er wird auf der Grundlage von Eimern zu 96 bit adressiert. Im allgemeinen Fall wird eine Angabe von 64 bit geliefert, die wahlweise als logische Adresse oder als technische Behälterangabe interpretiert werden kann.

Die Pfade zum Laufzeitspeicher sind auch zum Übermitteln von Steuerinformation an das gesamte Speicher- Subsystem vorgesehen, im besonderen an den übergeordneten Speicher der Wissensbasis.

Die Registerstruktur ist folgende:

1. für die Befehlszugriffe: 4 Basisregister BC1...BC4 zu 16 bit und 4 Grenzregister LC1...LC4 zu 12 bit

2. für die ART- Zugriffe je Niveau 1 Basis- und ein Grenzregister (BA1...BA4 zu 16 bit + 2 Niveau- Bits, die automatisch verwaltet werden, LA1...LA4 zu 12 bit). Je Port ist eine Selektor- Abstraktion der Form Registerwahl + 12- bit Offset mit Grenzprüfung vorgesehen.

Hinweise:

Die Konventionen sind so gewählt, daß für Befehls- und ART- Speicher in denselben Schaltkreisen untergebracht werden können (bis 64 k x 4 Eimer). Auch wird erwartet, daß die ARTs 4 mal (redundant) gespeichert werden. Dazu werden die Ports nacheinander gemäß den Angaben in den Steuerworten belegt, so daß alle Zugriffe voll parallel erledigt werden können.

3. Für Zugriffe zum Laufzeitspeicher sind vorgesehen:

- a) 4 3- fache Iterator- Abstraktionen
- b) 8 1- fache Iterator- Abstraktionen.

Diese sind den Ports fest zugeordnet (jeder 3- fach- Iterator einem Port, 2 1- fach Iteratoren je Port); die Vermittlung ist Angelegenheit der Speicheranordnung. Jede Abstraktion ist durch einen 30- bit- Selektor- Offset erweitert.

Jede Abstraktion hat ein 64 bit- Basisregister und ein 48-bit-

Längenprüfregister sowie einige Steuerbits. Für jede Iterator-Stufe ist zusätzlich vorgesehen:

- 1 Anfangswertregister (64 bit)
- 1 Schrittweitenregister (32 bit, einschl. Vorzeichen)
- 1 Schrittzahlregister (32 bit)
- 1 Register für den aktuellen Wert (64 bit).

Hinweis: Die 3-fach-Iteratoren sind für Operandenzugriffe, die 1-fach-Iteratoren vorzugsweise für Stacks bzw. Warteschlangen vorgesehen. Auch ist in den Steuerworten angebbbar, ob Offset-Angaben lediglich zur aktuellen Adresse addiert oder mit ihr verrechnet werden sollen.

4. Für Zugriffe zu Operanden-Schnellspeicher sind je Port 2 2-fach-Iterator-Abstraktionen mit Längenprüfung, aber mit Basisadresse 0 vorgesehen, die bis aufs Bit wirken, d. h. jeweils
- 2 Längenprüfregister zu 19 bit
 - 2 Register für den aktuellen Wert zu 19 bit
 - 4 Schrittweitengregister zu 16 bit
 - 4 Schrittzahlregister zu 16 bit
 - 4 Anfangswertregister zu 19 bit.

2.4. Prozessorkern 2 (PK 2)

2.4.1. Aufbau und Wirkungsweise

Der PK 2 repräsentiert ein einheitliches Architektur- und Schaltungskonzept für vielfältige Ausführungsformen von Mikrocontrollern.

Er ist für den Multitasking-Betrieb eingerichtet und so ausgelegt, daß er unmittelbar - mit einem Minimum an zusätzlicher Anpassungs-Hardware - in "zu steuernden" Umgebungen eingesetzt werden kann.

Die Architektur beruht auf dem Universalregisterprinzip und der üblichen Wortlänge von 32 bit.

Programmseitig sind jeweils 32 Register zu 32 bit direkt zugänglich; diese bilden einen Ressourcenvektor von 1024 bit. Die Register sind universell nutzbar, bei manchen Abläufen ist aber eine implizite Nutzung vorgesehen. Die Universalregister können auch zur Speicherung von Befehlen oder für Stack-Bereiche (Stack Cache) verwendet werden. Jedes Universalregister ist um einige TAG-Bits erweitert, die in manchen Befehlen zusätzliche Steuerwirkungen veranlassen, z. B. Adressenzählung, Ein- oder Ausgabe usw.; in den TAGs sind auch Schutzangaben (Zugriffsprivilegien) codiert. Unterbrechungen und Task-Umschaltungen werden auf das Umschalten von Registersätzen zurückgeführt. Für jede Task sind mindestens 8 Register vorzusehen. Der Registerspeicher kann auf den einzelnen Schaltkreisen verschieden groß sein. Untergrenze: 64 Register (2 kbit); Obergrenze: 512 Register (16 kbit). Die installierte Anzahl ist programmseitig abfragbar. Grundsätzlich ist durch Umladen eines Task-Steuervektors jedes Register zugänglich. Die Art und Weise der Task-Umschaltung ist einstellbar: programmiert, durch externe Signale, zyklisch. Bevorzugt wird die zyklusweise Umschaltung zwischen den Tasks, so daß eine gewisse Anzahl virtueller Maschinen (Richtwert: 8...16) mit jeweils entsprechend verminderter Leistung nutzbar ist. E/A-Steuerung, Ausnahmebehandlung, Unterbrechungsannahme und Diagnose sind im Rahmen dieses Konzepts lösbar. Im besonderen können in einzelnen Tasks spezifische E/A-Funktionen emuliert werden, z. B. serielle Anschlüsse, parallele Interfaces, Zähler- und Zeitgeberfunktionen usw.

Der PK 2 ist für den Einsatz in kritischen Anwendungsfällen, die eine hohe Zuverlässigkeit bzw. Fehlertoleranz verlangen, ausdrücklich vorgesehen. Dazu tragen folgende Eigenschaften bei:

- verhältnismäßige Einfachheit der Hardware, so daß sie problemlos verifizierbar ist und in sicheren und zuverlässigen Technologien gefertigt werden kann
- Mehrversionsprogrammierung im Rahmen des Multitasking (mit besonders geeigneten Befehlen für Resultatvergleich und -auswahl)
- Master/Checker-Prinzip an den Schaltkreisanschlüssen
- Inhaltsskontrolle der Speicher durch Parität oder ECC
- eingebaute "debugging"- und Diagnose-Vorkehrungen
- praxisgerechte Auslegung der E/A- und Speicheranschlüsse, so daß nur sehr wenig Zusatzbeschaltung notwendig ist.

Der PK 2 ist für vielfältige Aufgaben einsetzbar, und es ist vorgesehen, je nach Einsatzgebiet zwischen verschiedenen wohldefinierten Grundvorstellungen bzw. -modellen der Informationsverarbeitung zu wählen. Diese Grundvorstellungen werden als Paradigmata bezeichnet. Im einzelnen sind vorgesehen:

- 3 Anwendungsparadigmata AP 1 - AP 3
- 3 Befehlsparadigmata IP 1 - IP 3
- 4 Adressierungs- bzw. Selektionsparadigmata SP 1 - SP 4.

Das Anwendungsparadigma wird der Herstellung des jeweiligen Schaltkreises zugrunde gelegt. Das Selektionsparadigma SP 1 ist stets vorgesehen; ein weiteres Selektionsparadigma kann zum Herstellungszeitpunkt gewählt werden. Welche Befehlsparadigmata vorgesehen werden, hängt vom Anwendungsparadigma ab. Sind in einem Schaltkreis mehrere alternative Paradigmata vergänglich, so ist es möglich, programmtechnisch zwischen ihnen umzuschalten.

Die Ausstattung eines konkreten Schaltkreises ist programmseitig abfragbar.

1. Anwendungsparadigmata

AP_1

Der PK 2 wird unmittelbar in einer zu steuernden Umgebung eingesetzt. Es ist wesentlich, daß möglichst viele Anschlüsse für Ein- und Ausgabezwecke direkt verfügbar sind und daß diese so praxisgerecht ausgestaltet sind, daß auf spezifische Anpassungslogik weitgehend verzichtet werden kann, daß also nur noch die unumgänglich notwendigen Koppelstufen (z. B. Optokoppler) vorgesehen werden müssen.

AP 1 wird weiter spezifiziert in der Form AP 1.c.m, wobei c die Betriebsweise der Systemkommunikation und m die Speicherausstattung kennzeichnet (c = 0 oder n; m = 0, 1, 2, 3).

c=0

Keine Systemkommunikation ("stand alone"- Betrieb).

c=n

Systemkommunikation über n bidirektionale serielle Kanäle (n = 1...16). Für diese Kanäle gibt es eine einheitliche interne Anschaltung. Es sind 2 Übertragungsverfahren vorgesehen:

1. gemäß "Transputer" T 800
2. eigenes Protokoll mit Fehlerbehandlung und automatischer Pfadsuche in einer Struktur untereinander verbundener PK 2.

Weitere serielle Prozeduren können explizit programmiert werden.

m=0

Keine externen Speicher. Nur chip- interne ROM- und RAM- Ausstattung, in den einfachsten Ausführungen nur ROM + Universal-

registersatz. Mit entsprechendem ROM-Inhalt läßt sich ein Anfangsladen über serielle Anschlüsse organisieren.

m=1

Externer Speicheranschluß für ROM und statischen RAM mit 1-Byte-Datenweg und 20-24 bit Adresse.

m=2

Externer Speicheranschluß für ROM (1 Byte-Datenweg) und statischen RAM (2 Byte-Datenweg, mit Paritätskontrolle); 20-24 bit Adresse.

m=3

Externer Speicheranschluß mit konfigurierbarem 32 bit-Datenweg und 32 bit-Adresse, die als 20 bit statische oder als "gemultiplexte" dynamische (für DRAM-Zugriffe aufbereitet, mit den zugehörigen Steuersignalen) geliefert wird. Der adressierte Speicher bestimmt selbst die Datenwegbreite sowie die Art der Kontrolle (keine, Parität oder ECC).

AP_2

Der PK 2 wird in einer informationsverarbeitenden Umgebung eingesetzt, beispielsweise in höheren Ebenen eines hierarchisch aufgebauten Steuerungssystems, als eigenständiger Rechner oder als "add in"-Baugruppe in einem Personalcomputer.

Serielle Kanäle zur Systemkommunikation sind wie bei AP 1 konfigurierbar.

Die Schaltkreisanschlüsse werden vorzugsweise für den Anschluß schneller und großer Speicher sowie für die Kopplung mit Bussystemen verwendet.

Es sind 2 unabhängige Speicherdatenpfade vorgesehen: einer für schnelle statische Speicher (32 bit mit Paritätsprüfung, 20 bit-Adresse), einer für dynamische Speicher (Multiplex-Adresse, 32 bit Datenweg, ECC).

(Ob ein weiterer unabhängiger Pfad für den Systembusanschluß vorgesehen werden kann, hängt von der Zahl der nutzbaren Schaltkreisanschlüsse ab.)

AP_3

Der PK 2 wird als Steuerprozessor an verschiedenen Stellen des System.016 eingesetzt (Serviceprozessor, E/A-Prozessor, Speicherwaltungsprozessor usw.).

Die Anschlüsse sind dazu zweckgerecht ausgebildet, im besonderen dafür, sowohl die Informationstrukturen der .016-Architektur (Grundlage: Eimer zu 96 bit) als auch die üblichen 32 bit- und Bytestrukturen (für Ein- und Ausgabe bzw. die Kopplung von Fremdsystemen) effektiv nutzen und verwalten zu können.

2. Befehlsparadigmata

IP_1

Es sind stack-orientierte Befehle von 16 bit Länge vorgesehen, die sich nur mittelbar über eine Zugriffstabelle (ART) auf die eigentlichen Informationsstrukturen beziehen. Position und Größe der ART sowie des Stackbereiches sind durch Registerinhalte definiert. In den Befehlen sind 2 Maximalgrößen der ART vorgesehen: 64 Einträge und 4k Einträge. Jeder Eintrag ist 32 bit lang. Er wird wahlweise interpretiert

- als Direktwert
- als Selektor für einen Binärvektor (1... 32 bit aus einem Ressourcenvektor von 1k bit...16 kbit
- als Adresse.

IP 1 soll für übliche Steuerungsanwendungen eine kompakte Codierung ermöglichen. Im besonderen sollen die Universalregister als Arbeitsspeicher nutzbar sein, in dem Steuerangaben, wie Einzelbits, Bitfelder, Adressenzeiger, Zählwerte usw. dicht gepackt untergebracht werden können. Alle Informationsstrukturen sind bis aufs Bit adressierbar, dadurch spart man die Verschiebe- und Verknüpfungsbefehle, die ansonsten zur Auswahl der Informationsstrukturen notwendig sind (**hier Beispiel in Bild!).

Im besonderen soll der PK 2 effektiv als Bitprozessor betreibbar sein. (Bitstack!)

IP_2

Die Befehle haben eine feste Länge von 32 bit. Ihre Struktur ist weitgehend an den üblichen RISC-Prinzipien orientiert. Grundsätzlich wird zwischen Verknüpfungs- und Verzweigungsbefehlen unterschieden. Verknüpfungsbefehle sind wie folgt in Gruppen eingeteilt:

- Operationsbefehle der Form Register - Register - Register
- Test- und Modifikationsbefehle der Form Register - Direktwert
- Befehle für Blockoperationen mit Registerangaben für 2 Quelladressen und eine Zieladresse
- Lade- und Speicherbefehle
- Direktsteuerbefehle
- E/A-Befehle
- Diagnosebefehle.

Ein Befehlsformat ist für Erweiterungen reserviert.

Die Verzweigungsbefehle sind wahlweise für Programmverzweigungen oder Unterprogrammaufrufe nutzbar, teils mit impliziter, teils mit expliziter Adressenrettung über die Universalregister. Es ist wenigstens ein 16-bit Displacement vorgesehen. Das bei RISC-Maschinen übliche Vorordnen der Verzweigungen vor ausführbaren Befehlen ist möglich (NULLIFY-Option).

Verzweigungen sind abhängig von:

- Bittests (Einzelbitabfragen)
- Bedingungsabfragen
- Registerinhaltsvergleichen.

Des weiteren ist eine Verzweigung über Tabellenangaben möglich (z. B. für Betriebssystemrufe).

IP_3

Zusätzlich zu den Befehlen gemäß IP 2 sind mikrobefehlsähnliche Steuerworte von 64, 96 oder 128 bit Länge vorgesehen. Ein Steuerwort enthält in den ersten 32 bit:

- eine Operationsangabe
- eine Angabe zum Format der Quellen- und Zielbestimmungen
- eine Verzweigungsangabe oder die (kurze) Zielbestimmung.

Quellen- und Zielbestimmungen können kurz oder lang sein. Kurze Bestimmungen (16 bit) enthalten entweder einen Binärvektor- Selektor für die Universalregister, eine Angabe der Form Register + kurzer Offset, einen kurzen Direktwert oder eine Angabe der Form Register + E/A- Steuerung.

Lange Bestimmungen (32 bit) enthalten einen Direktwert oder einen Binärvektor- Selektor für den Speicheradressenraum.

Die Formate sind so festgelegt, daß die häufigsten Abläufe in 64-bit- Worten codiert werden können (Operation, Verzweigung, 3 kurze bzw. 2 kurze und eine lange Bestimmung).

Es gibt verschiedene Modifikationen für die Verzweigung einschließlich Unterprogrammaufruf und (alternativ) 4fach- Spätverzweigung.

3. Adressierungs- bzw. Selektionsparadigmata

IP_1

Der PK 2 liefert Byteadressen von maximal 28 bit zuzüglich 4 bit für die technische Speicherauswahl. Weviele Bits davon tatsächlich außerhalb des Schaltkreises nutzbar sind, ist schaltkreisabhängig. Es sind aber wenigstens 20, und der jeweilige Speicheradapter muß die überschreitung der jeweiligen installierten Maximaladresse erkennen.

Die technische Speicherauswahl betrifft:

- internen ROM
- internen RAM
- Universalregister relativ
- Universalregister absolut
- externen ROM (bootstrap ROM)
- externer Speicher unspezifiziert
- Fremdkommunikation (z. B. über einen Universalbus)
- externen ROM, volle Aufrufbreite
- externen RAM statisch
- externen RAM, dynamisch

(bei den letzten beiden noch Angabe der Kontrollmaßnahmen)

keine, Parity, ECC).

SP_2

Es wird eine 31-bit-Wortadresse geliefert (32. Bit ist immer 0). Deren technische Interpretation wird dem Speicheradapter überlassen.

SP 1 ist weiterhin nutzbar; ob ein Registerinhalt gemäß SP 1 oder SP 2 interpretiert wird, wird durch TAG-Bits bestimmt.

SP_3

Es wird eine 63-bit-Wortadresse geliefert (64. Bit ist immer 0). Deren technische Interpretation wird dem Speicheradapter überlassen.

SP 1 ist weiterhin nutzbar; ob ein Registerinhalt gemäß SP 1 oder SP 3 interpretiert wird, wird durch TAG-Bits bestimmt.

SP_4

Es wird eine 64-bit-Angabe gemäß den „016-Architekturprinzipien geliefert, die wahlweise als Adresse eines 96-bit-Eimers oder als behälterorientierte Selektionsangabe interpretiert wird.

SP 1 ist weiterhin nutzbar; ob ein Registerinhalt gemäß SP 1 oder SP 4 interpretiert wird, wird durch TAG-Bits bestimmt.

Hinweise:

1. Bei AP 1 gilt immer SP 1.
2. SP 4 gilt nur bei AP 3.
3. Bei IP 1 gilt nur SP 1.
4. Bei AP 2 kann zwischen SP 2 und SP 3 gewählt werden.
5. SP 2, 3 und 4 werden zur Herstellungszeit bestimmt; sie sind intern nicht umsteuerbar.

2.4.2. Übersicht über die Ressourcen

Der physische Ressourcenvektor umfaßt konzeptionell 512 Register zu 32 bit, die um TAG-Bits erweitert sind.

Ihrer Nutzung liegt an sich das Prinzip der aktiven Register zugrunde. Im Gegensatz zum PK 1 ist dies aber nicht in reiner Form zu verwirklichen; um die grundsätzlichen Anforderungen erfüllen zu können, sind vielmehr bei Aufwands-Leistungs-Abwägungen eher Kompromisse in Hinblick auf geringen Aufwand einzugehen. In den meisten PK 2-Schaltkreisen werden deshalb die Register als RAM-Array ausgeführt sein, und es sind Maßnahmen unumgänglich, diese Ressourcen so vielseitig wie möglich, also universell, nutzen zu können.

In den Befehlen ist normalerweise ein Ressourcenvektor von 32 Registern ansprechbar, der in 2 Subvektoren zu 16 Registern eingeteilt ist. Das höchstwertige Bit eines 5-bit-Auswahlcodons bestimmt die Auswahl. (1. Bereich: CURRENT TASK BASE CTB, 2. Bereich: WORKSPACE EXTEND BASE WEB). Die gesamte Registeradressierung bezieht sich auf den Tasksteuervektor TCV, der für jeden der beiden Bereiche CTB, WEB die erste Registerposition enthält (9-bit-Angabe, zu der die jeweilige 4-bit-Angabe addiert wird; CTB ist nur mit 6 bit codiert, die verbleibenden 3 werden zu 0 angenommen). In besonderen Befehlen bzw. Auswahlcodons ist eine 10-bit-Registerangabe vorgesehen: das höchstwertige Bit bestimmt die Bereichsauswahl, die verbleibenden 9 Bits werden zum Bereichswert im TCV addiert, so daß eine absolute Registeradresse modulo 512 gebildet wird. Der TCV selbst belegt die Register 0 und 1 der jeweiligen Task (er enthält darüber hinaus den Befehlszähler, Flags usw.). Nach dem Hardware-Rücksetzen wird Task 0 gestartet, wobei deren TCV den physischen Registern 0 und 1 zugeordnet wird. Die ersten 16 Register (CTB) sind fest der Task zugeordnet, der zweite Adressenbereich (WEB) kann beliebig umgeladen werden.

Aus dem TCV erschließt sich die Belegung der weiteren Register.

Bei IP 1 enthält Register 2 den Bitstack, Register 3 den externen Stackpointer, Register 4 die Stackbegrenzung, Register 5 die Stackbasis, Register 6 den ART-Pointer, Register 7 weitere Steuerinformation einschließlich der internen Stackpointer. Der interne Wortstack beginnt ab Register 8. Die verbleibenden Register sind frei benutzbar, aber nicht für Befehle. Es kann mit 3 Stacks gearbeitet werden: dem externen Stack, dem Wortstack ab Register 8 und dem Bitstack in Register 2. Die internen Stacks haben maximal 32 Positionen. Im ACCESS CONTROL WORD ACW (Register 7) wird bestimmt, mit welchem Stack gearbeitet wird und welche Maßnahmen bei Überlauf eingeleitet werden. Für Stacks (bzw. Angaben zum externen Stack) vorgesehene Register sind bei Nichtnutzung des jeweiligen Stack anderweitig belegbar.

Bei IP 2 ist Register 3 das implizite Rettungsregister (das über TAG-Bits als Stack konfiguriert werden kann), wobei nachfolgende Register je nach Adressenparadigma noch mit belegt sein können. Alle anderen Register sind frei belegbar, nur bei manchen Befehlen (z. B. Blocktransporten), werden zu einer Registerangabe noch Folgeregister benutzt.

Bei IP 3 sind die Register 10...15 implizit Bestimmungs- oder Zielregister für die Verknüpfungsoperationen (für jede Operation genau bestimmt), und der Bereich 2 ist von Register 0 an mit Steuerworten belegbar (Vielfache zu 64 bit). Das erlaubt es, kurze Steuerwortfolgen, beispielsweise zur Ansteuerung eines Interfaces, vollkommen im Registerspeicher zu halten, so daß alle Zyklen der Externspeicherzugriffe für die Datenübertragung verfügbar sind; und es ist auch möglich, daß mehrere Tasks dieselben Steuerwortfolgen nutzen (reentranter Code!).

Bei SP 3 und SP 4 ist für alle Adressierungsangaben immer das Folgeregister mit eingeschlossen. Für Längenkontrollen ist jeweils ein weiteres Register vorgesehen.

Für alle Verknüpfungsoperationen gibt es eine arithmetisch-logische Einheit mit 32 bit Verarbeitungsbreite.

***** muß noch genauer dargestellt werden ****

2.4.3. Speicheranschluß- und Zugriffsprinzipien

Bei SP 1 sind vielfältige technische Ausführungsformen von Speichern (externe und chip-interne) mit realen Byte-Adressen zugänglich. Explizite Lade- und Speicherfunktionen sind für Bytes, 16-bit-Worte und 32-bit-Worte vorgesehen. Bei den anderen Selektor-Paradigmata obliegt die Interpretation der gelieferten Adresse dem jeweiligen Speicheradapter. Es handelt sich hier um Wort- bzw. Eimeradressen.

Jede Adressenangabe kann durch einen Binärvektor-Selektor erweitert werden; damit ist praktisch jedes Bit direkt adressierbar. Das erfordert bei IP 1 2 aufeinanderfolgende ART-Positionen oder den Aufbau einer entsprechenden Angabe im Stack. Bei IP 2 ist der Selektor im Folgeregister anzugeben oder es ist eine indirekte Bestimmung in einem Register vorzusehen (mit entsprechenden TAG-Belegungen und YIELD-Angabe im Befehl). Bei IP 3 sind entsprechend lange Bestimmungen vorgesehen (Basisregister + 16-bit-Adressenoffset + Bitfeldangabe).

Ab IP 2 und SP 2 wird die objektorientierte Adressierung unterstützt. Parameter sind Ordinalzahlen der gewünschten Informationsstrukturen sowie die Basisadresse der ersten Descriptortabelle. Durch Hardware wird mit der ersten Ordinalzahl der entsprechende Eintrag in der ersten Tabelle aufgesucht, dieser liefert die Anfangsadresse der 2. Tabelle, zu der mit der 2. Ordinalzahl zugegriffen wird usw. (Bemerkung: Es ist notwendig, einen solchen Ablauf schaltungstechnisch zu unterstützen, da die üblichen RISC-Überlappungsschemata wirkungslos sind).

Bei SP 2 wird ein 64-bit-Tabellenformat ohne TAGs verwendet (32 bit Adresse, 30 bit Länge, 1 PRESENCE-Bit, 1 ESCAPE-Bit), bei SP 3 ein 128-bit-Tabellenformat (SP 4 entspricht der .016-Architektur: 96 bit mit TAGs).

Bei Ausnahmen (NOT PRESENT, Bereichsüberschreitung) wird der Ablauf abgebrochen (Ausnahmebehandlung oder abfragbare Bedingung).

2.4.4. Befehlsliste

6. Hinweise zur Schutzrechtsarbeit

Im folgenden werden einige Vorstellungen zu eigenen Schutzrechten kurz skizziert. Dabei sind jeweils Arbeitstitel, bzw. Erfindungsgegenstand sowie - falls bereits absehbar - wesentliche Kennzeichen angegeben. Die Aufteilung der technischen Ideen auf einzelne Schutzrechtstitel gilt für DDR-Anmeldungen: hier sollte - unter Maßgabe ausreichender Erfindungshöhe, die erst nach der gezielten Recherche beurteilbar sein wird - eine komplexe technische Lösung auf mehrere Anmeldungen verteilt werden, wobei die einzelne Schrift vergleichsweise knapp gehalten werden kann. Darüber hinaus sind unbedingt Auslandsanmeldungen anzustreben. Hier kommt es darauf an, mit vertretbaren Aufwendungen an Gebühren die technische Eigenleistung in der internationalen Fachwelt vorzustellen. Dazu sind mehrere technische Lösungen jeweils zu einer Schrift zusammenzufassen, die dann (nach den anderwärts üblichen Gepflogenheiten) recht umfangreich ausfallen kann. Schreib- und Zeichenarbeiten sind in diesem Zusammenhang als im Interesse der Sache notwendig einzuplanen.

1. Digitale Rechneranordnung mit erweiterten Universalregistern.

Dadurch gekennzeichnet, daß die Universalregister über die eigentliche Verarbeitungsbreite hinaus mit zusätzlichen Bitpositionen erweitert sind, daß diese Bitpositionen an Steuerschaltungen angeschlossen sind, daß die Aktivierungsleitungen dieser Steuerschaltungen an Positionen des Befehlsregisters angeschlossen sind... (TAG-Bits an den Registern, in denen der Gebrauch der Register sowie zusätzliche Steuerwirkungen für Adressierung bzw. E/A codiert sind; YIELD-Bits in den Befehlen, um diese Wirkungen bei den Zugriffen zu den Registern auszulösen oder nicht, man kann also wählen zwischen einem normalen Registerzugriff oder einer Adressenrechnung bzw. einer Ausgabeoperation usw. Ausführungsbeispiel für einfache 32-bit-Maschine (32-bit-Paradigma IP 2 des PK 2).)

2. Programmierbare digitale Verarbeitungseinheit mit aktiven Registern.

Kennzeichen: die Hardware-Register sind zu einem Ressourcenvektor zusammengefaßt, die meisten Register sind fest an Verarbeitungs-, Adressierungs- oder Steuerschaltungen angeschlossen, alle Registerinhalte sind aber bis aufs Bit selektierbar und über zusätzliche Schaltmittel untereinander unbeschränkt austauschbar, es sind mehrere Befehlsregister vorgesehen, die auch in das Prinzip mit einbezogen sind, damit auch selbstmodifizierende Befehle, Datenflußsteuerung ist wählbar (eine Verknüpfung wird - ohne besondere Angabe eines Operationscodes - sofort ausgeführt, sobald die Quellregister gültige Daten haben), Übergabe aller Argumente eines Befehls als Direktwerte im Befehl selbst (das vielleicht auch extra!). Hier wird der PK 1 in seiner Grundform vollständig beschrieben.

3. Universelle digitale Verarbeitungseinheit.

Kennzeichen: der Prozessor ist zwischen mehreren Programmier-Paradigmata umschaltbar, dazu gehören Elementarbefehle, die sich über ein Descriptorfeld auf die Universalregister beziehen, registerorientierte 32-bit-Befehle und längere Befehle, die Mikrobefehlen ähnlich sind, für die Adressierung gibt es ebenfalls verschiedene Paradigmata, mehrstufige indirekte bzw. objektorientierte Adressierung wird in der Hardware unterstützt, herkömmliches (v. Neumann-) Speicherkonzept, ohne TAGs, Universalregister mit TAGs, Befehle können auch in den Universalregistern residieren, hardwareseitiges Multitasking, Möglichkeit der rein chip-internen Emulation von E/A-Funktionen, verschiedene technische Speicherkonfigurationen. Hier wird der PK 2 vollständig beschrieben, und dabei werden Lösungen gemäß der Schutzrechte 1 und 2 mit eingesetzt.

4. Einrichtung zur Verarbeitung von Ternärvektoren.

Kennzeichen: Nutzung der 4. Belegung (01) der Ternärcodierung zur Kennzeichnung maskierter Positionen, Ausgestaltung der Verknüpfungsschaltungen, neue Operationen "Laden Maske" und "Laden maskiert". Hier evtl. die komplette Logik-Verarbeitungseinrichtung auf Grundlage ternärer Operationen beschreiben. Anmeldung muß 1 - 3 vorangehen. *neue Logikeinheit mit 2 weiteren Werken (o. 2 Anmeldungen: TVL; davon d. neue Logikeinheit.)*

5. Aktive Speicheranordnung, vorzugsweise zur Durchmusterung binär codierter Listen.

Kennzeichen: Adressen- und Datenweglogik, die die eigentlichen Speicherschaltkreise so dicht wie möglich umgibt, Durchmusterungen ternär - ternär und binär - ternär; ternäre Vektor-Abschnitte enthalten - unter Nutzung der 4. Codierung - Zeigerinformation, um don't care-Abschnitte überspringen zu können, insgesamt Ausrichtung auf relationale Durchmusterungen, Weiterbildung für Zeichenketten-Durchmusterungen und -transporte, auch "quer-Speicherung" von Zeichenketten, Auftragssteuerung für mehrere Zugriffe.

Je nach Recherche- und Entwurfs-Ergebnissen auch mehrere Anmeldungen.

Insgesamt geht es um das Aktive Memory Array (AMA).

6. Verarbeitungseinrichtung für numerische Daten hoher Genauigkeit. (Ggf. auch mit Verfahrensanspruch.)

Kennzeichen: hochgenaue Akkumulation mit herkömmlichen Schaltungsmitteln, abschnittsweises Aufaddieren in RAM-Doppelanordnung, Aufsammeln von Überträgen bei der Skalarproduktbildung, Kennzeichnung signifikanter Stellen, damit beschleunigte End-Akkumulation, Schaltungsmaßnahmen für das Löschen (z. B. während des Akkulierens - der gelöschte Speicher ist Voraussetzung für die jeweils nachfolgende Rechenoperation!), für Rundung, Vergleich usw. Voraussetzung für 7 und 2 (falls das dort aufgenommen werden soll; es könnte hier auch eine Verfeinerung des PK 1 als Ausführungsbeispiel beschrieben werden).

7. Anordnung aus mehreren numerischen Verarbeitungs- und Speicherwerken.

Kennzeichen: baumförmige Struktur aus 4 Werken, Top of Stack am 4. Werk, Speicher mit Datenaustauschschaltungen, schneller

*Akkumulation über 21
die 4 baumf. verschalteten Werke*

"bypass" in den Werken, um mathematische Grundoperationen effektiv auf das starre Datenflußschema abbilden zu können, Minimum/Maximum-, Bereichsgrenzen- und Delta-Detektoren, vielleicht auch hier schon bidirektionale Verbindungen zwischen den Werken (Alternative zu Bus bzw. Crossbar).

8. Aktive mehrstufige heterogene Speicheranordnung.

Kennzeichen: angepaßte, technisch heterogene Speichermittel mit aktiven Zusatzschaltungen und Verbindungsstrukturen für objektorientierte bzw. wissensbasierte Zugriffe, Maßnahmen für Diagnose und Fehlertoleranz, mehrere unabhängige Zugriffspfade.

Vielleicht auch als "Hochleistungsrechner mit ...Speicheranordnung", evtl. in Zusammenhang mit dem Einbau von Direktwerten in Befehlsfolgen zwecks schneller Ausführung bzw. Bildung von Datenfluß-Angaben.

9. Schnelles bidirektionales Verbindungssystem.

Ziel: wenig Leitungen, höchste Geschwindigkeit, Vorkehrungen für flexibles "Routing", um beim Hochleistungsrechner mit wenig Schaltkreis-Pins auskommen zu können, Sondermaßnahmen für Fehlertoleranz und Diagnose.

Die technischen Vorstellungen werden entweder separat angemeldet oder zur Hebung der Erfindungshöhe von 7 bzw. 8 verwendet.

10. Emulation von Fremdarchitekturen (I).

Vorstellungen: technische Vergegenständlichung der Befehle, die sich nach RISC-Prinzipien abarbeiten lassen, in einem Incarnated Interpreter Module, ggf. Ergänzung durch Zusatzbefehle, Anschluß an das aktive Speichersystem, Nutzung des Hochleistungsrechners als Accelerator, hardwaregestütztes wissensbasiertes Handling der Fremddaten im Speichersystem unter Einschluß des gelegentlich notwendigen Umpackens (ASCII \leftrightarrow EBCDIC, Rechtsadressierung \leftrightarrow Linksadressierung u. dergl.).

(Hinweis: Die Zerlegung des Befehlssatzes in RISC-ähnliche Befehle und andere ist als bekannt anzusehen. Eigene Ansprüche wären in Richtung auf die Einordnung in die eigene Architektur, auf wissensbasiertes "Handling" usw. geltend zu machen.)

11. Emulation von Fremdarchitekturen (II).

Prinzip: Aus den zu emulierenden Original-Programmen werden Steuerfolgen (der eigenen Befehlsliste) erzeugt, und zwar "live" beim Umladen in die Schnellspeicher (Cache). Entscheidend ist die Behandlung der Verzweigungen; dafür sind besondere Schaltmittel notwendig ("Befehlsstrukturspeicher" o. dergl.). (Hinweis: Es ist bereits bekannt, den einzelnen aktuellen Befehl vor der Verarbeitung umzuschlüsseln; hier geht es aber um ganze Programme.)

12. Ein- und Ausgabe-Anschlußschaltungen, vorzugsweise für Mikroprozessoren.

Vorstellungen: Nutzung der Speicherzugriffsleitungen und anderer Leitungen für Zwecke der Ein- und Ausgabe, so daß "außen" die Hardware so einfach wie möglich wird, Vordecodierung von

Adressen, Lösung der technischen Probleme durch Synchronisation mit der Speichersteuerung und evtl. Sperren der Speicheradressen in besonderen Treiberstufen.
Nutzung auch mit Bezug auf BSCC; Anmeldung müßte 1 voraufgehen. Wenn Erfindungshöhe nicht reicht, wird damit 1 aufgewertet.

13. Dreiwertige Informationsübertragung.

Ziel: Zwecks Klemmen- und Leitungs-Ersparnis mit 2 Leitungen 9 Werte übertragen (entspr. 3 bit + 1 Sonderwert, der vielleicht anderweitig nutzbar ist), auch Steuerfolgen 3-wertig. Nicht vordringlich!

14. Binärer serieller Akzeptor-Automat.

Ziel: Speichergesteuerter Akzeptor für serielle Bitströme, mit DMA, Fehlerprüfung, sehr flexible Protokoll-Handling, mehrere Bitströme unabhängig mit einer Einrichtung.
Sinngemäße Weiterbildung des BSCC-Prinzips. Nicht vordringlich!

15. Hardware zur Datenflußanalyse, um Compiler zu unterstützen. Noch keine eigenen Vorstellungen.

Erste Vorstellungen zu Europa-Anmeldungen:

- a) 1 und 3 zusammen sowie 11
- b) 2 mit 4, ggf. 6 einbeziehen
- c) 6 und 7
- d) 8 und 10
- e) Verfahrensanmeldung für das relationale Wissensbasis-System (S.014, s. auch /1/, Anh. 5) mit Anordnung gem. 5 zur Durchführung des Verfahrens.