

## Grundlegendes zur Speicherhierarchie

Claus Schünemann, IBM Böblingen

### 1. EINLEITUNG

Das Thema dieses Beitrags ist die konkrete Daten-Speicherung und -Adressierung unter Zugrundelegung eines hierarchischen Aufbaus des Speichersystems.

Soweit Datenbankaspekte dabei berührt werden, sind sie aus der Sicht der Hardware-Implementierung und vorwiegend unter Leistungsgesichtspunkten gesehen.

Heutige Computer-Speichersysteme sind bereits weitgehend hierarchisch strukturiert. Dabei soll unterschieden werden zwischen einer lediglich durch Kapazitätsabstufung gekennzeichneten und einer strengen Hierarchie, bei der auf jeder Stufe wahlfreier Zugriff möglich ist und der Datenfluß keine Stufe überspringt.

Die Kombination Hauptspeicher - Pufferspeicher stellt eine strenge Hierarchie dar, bei der der Hierarchiebegriff überhaupt erst ins Bewußtsein gerückt wurde [1]. Der Pufferspeicher (Cache) ist für die Maschinenarchitektur transparent und paßt die Geschwindigkeit des Hauptspeichers an die noch höhere des Prozessors an. Ebenso ist die Folge Hauptspeicher - Magnetplattenspeicher als strenge Hierarchie anzusprechen, auch wenn diese Betrachtungsseite (mit Ausnahme von Programm-Paging im Rahmen des virtuellen Speichers) bislang nicht im Vordergrund stand und der Plattenspeicher mehr als Ein/Ausgabegerät aufgefaßt und so von der Maschinenarchitektur behandelt wurde.

Der Magnetbandspeicher ist wegen seiner langen Zugriffszeit (incl. Bandladen) nicht mehr im strengen Sinne zur Hierarchie zu rechnen.

Ansätze, die große und billige Bandspeicherkapazität als echte oberste Datenfluß-Hierarchiestufe zu integrieren, sind mit der jüngeren Entwicklung von automatischen Bandtransportsystemen, wie z.B. beim IBM 3850-Kassettenpeicher, sichtbar geworden. Dabei könnte beispielsweise dem Bandspeicher die Funktion eines Archivs und dem Plattenspeicher die Funktion eines Arbeitsspeichers großer Kapazität zugeordnet werden, wobei der Inhalt ganzer virtueller Plattenstapel automatisch auf Verlangen auf das Plattensystem übertragen wird [2]. In Abbildung 1 ist das Schema dieses Hierarchiekonzepts skizziert.

Der schwache Punkt der gegenwärtigen Speicherhierarchie ist das Verhältnis der Zugriffszeiten des Hauptspeichers zum Plattenspeicher von mehr als 1:10000, die sog. Zugriffslücke. Auch ein Dazwischenschalten von Trommelspeichern bzw. Plattenspeichern mit festem Lesekopf ändert die Situation nicht wesentlich. Man versucht daher bekanntlich, das Mißverhältnis durch Programmumschaltung im Rahmen von Multiprogrammierung zu überbrücken. Mit fortschreitender Prozessor- und Hauptspeichergeschwindigkeit, aber gleichbleibender Zugriffszeit der mechanisch arbeitenden Massenspeicher, muß der Multiprogrammierungsgrad, die Hauptspeichergröße und die Zahl der Plattenspindeln immer größer werden. Damit entfernt man sich vom Kostenoptimum, außerdem steigen die Anforderungen an das steuernde Betriebssystem und seine Komplexität, bei abnehmender Effizienz.

Im Folgenden wird versucht, für das gesamte Hierarchiespektrum die Speicherparameter nach einheitlichen Gesichtspunkten zu klassifizieren und anhand solcher Parameter die Leistungsfähigkeit der Hierarchie zu diskutieren, mit besonderer Blickrichtung auf das Problem der Zugriffslücke. Die Anforderungen des Datenbankbetriebes werden kurz angesprochen.

## 2. TECHNOLOGIE- UND OPERATIONSPARAMETER

Es sind zahlreiche Technologien bekannt, die unter Ausnutzung verschiedenster physikalischer Effekte zu sehr unterschiedlichen Speichereigenschaften führen. Am verbreitetsten ist heute die Halbleitertechnologie für die schnellen elektronischen Matrix-Speicher mit wahlweisem Zugriff und die Magnetschichttechnologie für die langsameren und billigen Massenspeicher, hauptsächlich in den Ausführungen Platten- und Bandspeicher.

Eine weitere Gruppe, die aber noch nicht das Stadium breiter Produktreife erreicht hat, ist die der optischen und mit Elektronenstrahl operierenden

Speicher [3,4]. Auch die diversen Schieberegistertechnologien wie CCD (Charge Coupled Device) [5,6] oder Magnetblasen (Bubbles) [7] machen vorerst nur tastende Schritte im kommerziellen Einsatz. Die spezifischen Arbeitsweisen der einzelnen Speicherfamilien sollen hier nicht diskutiert werden, vielmehr wird das gesamte Speicherspektrum einheitlich durch einen Satz von invarianten technologischen und operativen Parametern beschrieben, Tabelle I.

Die beiden wichtigen Operationsparameter, mittlere Zugriffszeit und Bitkosten, stehen in einer gewissen reziproken Relation zueinander. Sie bestimmen den Standort einer Technologie innerhalb des Gesamtspektrums. Im Diagramm Abb. 2 sind heutige typische Werte in Abhängigkeit des gewichtigsten Technologieparameters, Bitzahl pro Schreib/Lesestation, dargestellt [8].

Die Zugriffszeit setzt sich zusammen aus der Zugriffszeit im engeren Sinne, einer Art Totzeit vor der Übertragung des ersten Bit, und der Datenübertragungszeit. Die Übertragungszeit ist abhängig von der Datenrate, gegeben durch Taktfrequenz und interne Bitbreite, und der gewählten übertragenen Blocklänge. Zusätzliche Verzögerungen durch den externen Übertragungskanal sind in der Übertragungszeit mitenthalten.

Unter Modularität ist die Unterteilbarkeit eines Speichers bzw. einer Hierarchiestufe in Module mit eigenem parallelen Zugriff verstanden. Dadurch wird die Zugriffsrate erhöht. Die Fähigkeit zur modularen Aufteilung nimmt im allgemeinen ab mit dem Technologieparameter "Bitzahl pro Schreib/Lesestation". Bei mechanischer Entkopplung zwischen Lesen/Schreiben und dem Datentransport kann die Zugriffsrate durch Überlappung weiter erhöht werden. So wird beim Bandkassettenpeicher IBM 3850 die nächste Kassette schon transportiert, während die vorhergehende sich noch in der Lese/Schreibstation befindet.

Weitere Beispiele für asynchronen Parallelbetrieb sind die Konfiguration mehrerer Plattenspeicher in einer DV-Anlage wie auch die Unterteilung des Hauptspeichers in unabhängig und parallel arbeitende Module.

Auch die Bitkosten bestimmen sich in erster Linie aus der Bitzahl pro Lese/Schreibstation. Sie sind außer von den spezifisch technologisch-konstruktiven Faktoren vom allgemeinen Miniaturisierungsstand der Technik abhängig. Abb. 3 zeigt beispielsweise die historische Entwicklung der Bitdichte beim Magnetplattenspeicher. Entsprechend sind die Zahlenangaben

in Abb. 2 nur zeitbezogen zu verstehen. Die relativen Zuordnungen dürften hingegen weitgehend invariant zum allgemeinen Stand der Technik sein, da fortschreitende Miniaturisierung allen Technologien zugute kommt. Die Speicherkapazität pro Hierarchiestufe ergibt sich in einer ausgewogenen Konfiguration nach einer Art reziproker Funktion der jeweiligen Bitkosten

Ein weiterer operativer Parameter ist die Zuverlässigkeit des Speichers, d.h. die mittlere Zahl von gelesenen Bits pro fehlerhaftem Bit. Dieses Merkmal ist eine Funktion der natürlichen Fehlerfreiheit des Mediums, des Sortierungsgrades nach guten Einheiten und des Aufwands an gezielter Redundanz mit nachfolgender Fehlerkorrektur. Die Fehlerdichte des Mediums nimmt naturgemäß mit der Homogenität ab. Typische Zuverlässigkeitswerte sind (nach entsprechendem Sortierprozess) z.B. beim fabrikneuen Plattenspeicher  $10^9$  und  $10^{12}$  nach erfolgter Korrektur.

Die physikalische Natur der Speicherung bestimmt den Grad der Flüchtigkeit der eingeschriebenen Information. Bei einem Arbeitsspeicher kann man eine gewisse Flüchtigkeit mit periodischem Wiederauffrischen zulassen, bei einem Archiv- oder Journalspeicher muß natürlich ein dauerhaftes Speichern gefordert werden.

In gewisser Verwandtschaft zur Flüchtigkeit steht die Eigenschaft des ON-line oder OFF-line Einschreibens, letzteres auch allgemein unter ROM verstanden. Bei verschiedenen Anwendungen, z.B. Speicherung von Dokumenten mit geringer Änderungsfrequenz, kann der ROM-Speicher durchaus sinnvoll und, da entsprechend billig, von Interesse sein. Ein Übergang zwischen dem normalen schreibbaren Speicher und dem ROM stellt der PROM bzw. EAROM (Programmable bzw. Electrically Alterable Read Only Memory) dar. Der ROM-Speicher wird hier nicht weiter behandelt.

Der letzte Operationsparameter ist die adressierbare Einheit, die im Verein mit der eigentlichen Zugriffszeit die Komplexität der Zugriffsmethode und Effizienz des Datensuchens bestimmt.

Man unterscheidet zwischen Orts- und Inhaltsadressierung. Die Ortsadressierung ist auf Hauptspeicherebene die dominierende Adressierungsart: Die physische Lokation jedes Datenelementes ist vom Programm definiert und wird über die Adresse direkt gefunden. Dieses Konzept ist auf den höheren Speicherebenen für das Aufsuchen von Datensätzen nicht mehr zweckmäßig, wenn die Sätze z.B. in Form einer Datenbank organisiert,

programmunabhängig und vielen Benutzern verfügbar sein sollen. Sie müssen also letztlich durch ihren Inhalt, gegeben durch ein oder mehrere Merkmale, gekennzeichnet sein. Innerhalb eines Satzes sind die Daten im allgemeinen wieder formatiert, d.h. ihre semantische Bedeutung ist durch ihren relativen Ort bestimmt.

Die heutige Suchtechnik bei inhaltsadressierten Datensätzen bedient sich Indextabellen, in denen z.B. die Hauptmerkmale numerisch oder alphabetisch geordnet und die reale Speicheradresse direkt zugeordnet ist. Beim Vorliegen weiterer (Neben-) Merkmale können diese in eigenen Tabellen gelistet werden, wobei die Speicheradressen aller Sätze, die dieses Merkmal enthalten, wieder zugeordnet werden. Mit diesen invertierten Listen kann bekanntlich der Prozess des Suchens nach mehrfachen Merkmalen schnell, d.h. ohne alle Sätze sequentiell prozessieren zu müssen, durchgeführt werden. Mit Hilfe der Indextabellen wird also die Inhaltsadresse eines Datensatzes in eine Ortsadresse umgewandelt. Letztere wird dann beim Speichern mit wahlfreiem Zugriff schnell und direkt angesteuert.

Das Durchsuchen der Indextabellen nach dem gewünschten Merkmal stellt in sich nun wiederum einen Prozeß mit sequentieller Schrittfolge dar. Ein weiteres Parallelisieren wäre das Abspeichern der Indextabellen in Assoziativspeichern, mit folgenden Vorteilen:

- Fortfall der numerischen oder alphabetischen Merkmalsordnung.
- Dadurch einfache Aufarbeitung durch direktes Zufügen/Entfernen neuer Indizes.
- Fortfall der invertierten Listen, da gleichzeitig auf mehrfache Merkmale assoziiert werden kann.
- Direktes gleichzeitiges statt sequentielles Suchen.

Die Eigenart des Assoziativspeichers, eine Formatierung der Daten zu verlangen, wäre in diesem Fall kein Nachteil.

Ein Sonderfall der Ortsadressierung ist die Adressierung mit Zeigern. Dabei wird auch eine Entkopplung von Benutzerprogramm und Datenadresse erreicht. Nachteilig ist das sequentielle Durchlaufen der Zeigerkette.

Die einzelnen Speichertechnologien unterscheiden sich nun hinsichtlich der Größe der hardware-mäßig adressierbaren Einheit. Diese ist z.B. ein

Byte beim (Halbleiter-) Matrixspeicher, ca. 10-20 KBytes beim Plattenspeicher und Millionen von Bytes beim konventionellen Bandspeicher. Wenn diese adressierbare Einheit nun gleich oder kleiner als die gewünschte zu übertragene Blocklänge ist, soll von wahlfreiem Zugriff gesprochen werden.

Der Plattenspeicher hat nur einen semi-wahlfreien Zugriff, da seine Adressiereinheit (die Spur) um ein Vielfaches größer als eine bequeme logische Satzlänge bzw. eine für diese Hierarchiestufe optimale Blocklänge ist. Der konkrete Block muß dann wieder sequentiell auf der Spur gesucht werden.

Die sogenannten Zugriffsmethoden, also die praktischen Prozeduren zum Aufsuchen von Datensätzen spiegeln die jeweils zugrundeliegenden technologischen Adressierparameter wider.

Ein Beispiel ist die index-sequentielle Zugriffsmethode für "direkten wahlfreien" Zugriff zum Plattenspeicher: Dabei sind die Hauptmerkmale der Datensätze in einer Indextabelle nach aufsteigender Ordnungszahl geordnet. Die Tabelle ordnet jeweils einer Gruppe von Sätzen die zugehörige Spuradresse auf der Platte zu. Auch die Sätze selbst sind nach der gleichen Ordnungszahl geordnet, um im Falle sequentiellen Zugriffs die große Zugriffszeit für jeden individuellen Satz zu eliminieren. Beim Rotieren der Platte werden die ausgelesenen Satzmerkmale mit dem Suchmerkmal verglichen, bis Übereinstimmung herrscht. Beim Aufarbeiten, z.B. Zufügen eines weiteren Satzes in die möglicherweise physisch lückenlose Satzfolge, weist ein Zeiger zu einer neuen Spuradresse auf einer Überlaufspur. Die Methode kombiniert also die Suchelemente Indextabelle, sequentielles Suchen und Zeigertechnik zu einer den spezifischen Plattenspeicherbedingungen angepaßten Prozedur, Abb. 4a.

Bei einem anderen Speicher mit auch homogenem Medium, dem Elektronenstrahl-Speicher, ist die Adressiereinheit frei wählbar zwischen einem und Zehntausenden von Bytes. Das Zugriffsverfahren kann rein indexorientiert und entsprechend einfach gehalten werden: Das sequentielle Suchen entfällt. Ein Überlaufproblem existiert nicht. Dank der kurzen eigentlichen (elektronischen) Zugriffszeit kann auf eine sequentielle Satzordnung verzichtet und der Satz an beliebiger Stelle gespeichert werden, Abb. 4b.

Die größere Adressiereinheit, d.h. die geringere "Wahlfreiheit", bei

den kostengünstigen Technologien ist an sich kein prinzipieller Nachteil, da innerhalb einer Hierarchie ohnehin mit Blockübertragung gearbeitet wird. Ein gradueller Nachteil ist nur dann festzustellen, wenn wie beim Plattenspeicher optimale Blocklänge und technologische Adressiereinheit nicht übereinstimmen. Diese Diskrepanz schlägt sich dann in aufwendigen und zeitraubend ablaufenden "Zugriffsmethoden" nieder.

### 3. SPEICHERHIERARCHIE

Aufgabe eines Speichersystems ist neben der Speicherung, dem Prozessor die benötigten Daten in genügend kurzer Zeit und in der angeforderten Menge pro Zeiteinheit zur Verfügung zu stellen. Analog zu den Systemleistungsparametern Antwortzeit und Durchsatz läßt sich die Speicherleistung durch die Parameter Zugriffszeit und Zugriffsrate definieren. Wenn ein Speicher nur einen Zugriff gleichzeitig gestattet, kann die Zugriffsrate etwa gleich dem reziproken Wert der Zugriffszeit gesetzt werden. Bei gleichzeitig mehreren Zugriffen, d.h. Modularität größer als 1, erhöht sich die maximale Zugriffsrate entsprechend. Wie weit die maximale Zugriffsrate ausgenutzt werden kann, hängt von Parametern wie Systemsteuerung, Programmprofil, Multiprogrammierungsgrad und Zahl der Parallelprozessoren etc. ab.

In einer Hierarchie ist eine gewisse Grundmodularität der einzelnen Stufen schon im Interesse eines gleichzeitigen Datenverkehrs nach oben und unten wünschenswert. Dies wird steuerungsmäßig z.B. auf Hauptspeicherebene durch das unabhängige Operieren von Prozessor und Kanälen erreicht. Für effektive Multiprogrammierung ist ausreichende Modularität der Plattenspeicherstufe zwingend Voraussetzung. Zweck der Multiprogrammierung ist es, die resultierende Zugriffsrate - gemessen an der Schnittstelle zum Prozessor - und damit den Systemdurchsatz zu erhöhen.

Bekanntlich liegt dessenungeachtet der Engpaß für den Durchsatz heutiger DV-Systeme immer noch bei der Zugriffszeit und Zugriffsrate des Plattenspeichers. Da weitere Geschwindigkeitsfortschritte für Prozessor und Halbleiterspeicher in Zukunft durchaus erwartet werden dürfen, die Plattenspeicher-Zugriffszeit aber kaum noch verbesserungsfähig ist, wird dieses Problem immer drängender: Eine Lösung über weitere Erhöhung des Multiprogrammiergrades, d.h. der Zahl der gleichzeitig operierenden Programme, mit entsprechender Erhöhung von Hauptspeichergröße und Plattenspeichermodularität erscheint aus Kosten- und Komplexitätsgründen

unpraktikabel. Außerdem leidet bei zu hohem Multiprogrammierungsgrad die Effizienz: Die Systemverwaltung nimmt relativ zur Wirkarbeit zu, die Chance, mit einer Plattenarmposition mehrfache Zugriffe abzudecken, nimmt ab usw.

Eine andere Lösung dieses Problems ist der weitere Ausbau des Speicherhierarchiekonzeptes, bei beschränktem Multiprogrammierungsgrad. Der (nicht realisierbare) ideale Speicher, d.h. der Speicher mit der Zugriffszeit des Pufferspeichers und den Kosten des Bandspeichers, läßt sich durch eine ausgewogene Hierarchie mit genügend feiner Stufung annähern.

Glücklicherweise verspricht die technologische Entwicklung Speicherprodukte, die leistungs- und kostenmäßig gerade das Gebiet der "Lücke" ausfüllen und sich so gut in das Spektrum einfügen. Mögliche Technologien für die "Lücke" sind z.B. der CCD-Schieberegisterspeicher, der Schieberegisterspeicher mit verschiebbaren magnetischen Blasen (Bubbles) sowie die Elektronenstrahlspeicherröhre, Abb. 5. Diese Technologien sollen im Folgenden elektronische Massenspeicher genannt werden.

### 3.1 Hierarchiemechanismus

Die Speicherhierarchie besteht also aus der Hintereinanderschaltung von Speicherstufen, wobei mit zunehmender Stufenordnungszahl die Zugriffszeit und Speicherkapazität zunimmt. Bei einem Speicherzugriff des Prozessors versucht dieser zunächst, die Daten auf der untersten schnellsten Ebene zu finden. Bei Mißerfolg wird zur nächsten Ebene zugegriffen und so fort. Bei einer Datenübertragung auf die jeweils niedrigere Ebene wird nun nicht nur das verlangte Wort oder Byte, sondern gleich ein ganzer Block übertragen. Auf jeder unteren Ebene wird ein Teil des Blocks abgelagert. Die Übertragungszeit ist bei den gewählten Blocklängen meist klein gegen die eigentliche Zugriffszeit. Das Wesen der Speicherhierarchie drückt sich also darin aus, daß unter Zulassung von geringfügig mehr Zugriffszeit (nämlich incl. Übertragungszeit) ganze Daten- oder Programmblöcke übertragen werden, in der Annahme, daß davon ein Teil in nächster Zukunft ohnehin zum Prozessieren angefordert wird. Es liegt also ein prophylaktischer Zugriff (look ahead) unter Ausnutzung der (gegen die eigentliche Zugriffszeit) kurzen Übertragungszeit vor. Unterstützt wird dieser Mechanismus dadurch, daß die Daten oftmals in kurzem Zeitraum mehrfach zugegriffen werden, z.B. bei Programmschleifen, aber auch beim Operieren



auf häufig benutzte Arbeitsdaten wie Indextabellen, Kataloge usw.

Die Trefferrate, d.h. die Wahrscheinlichkeit, Daten auf der jeweils zugegriffenen Ebene anzufinden, nimmt zu mit der Speicherkapazität dieser Ebene, ferner im allgemeinen mit der Blocklänge. Davon unabhängig hängt sie natürlich vom jeweiligen Daten- und Programmprofil ab.

Das Freimachen von Speicherplatz auf einer gefüllten Hierarchiestufe erfolgt im einfachsten Fall selbstregelnd nach den gebräuchlichen Algorithmen wie FIFO oder LRU (Least Recently Used). Dieser Mechanismus kann selbstverständlich unterstützt werden durch residentes Einspeichern gewisser häufig gebrauchter Datenteile in untere schnelle Ebenen, z.B. Teile des Betriebssystems im Hauptspeicher usw. Auf den höheren Ebenen, bei denen jeder Zugriff in die Leistungsbilanz eingeht, ist die Steuerung software-implementiert und entsprechend "intelligenter".

Die Adreßsteuerung und das Suchen von Daten auf einer Ebene könnte konzeptuell am einfachsten über einen das Gesamtspeichersystem umfassenden Adressraum erfolgen.

Jede Hierarchiestufe enthielte dann eine Tabelle für die dynamische Zuordnung der virtuellen Gesamtspeicheradresse zur lokalen Ebenenadresse.

Aufgrund der historischen Entwicklung gibt es meist mehrere Adressräume in einer Speicherhierarchie: Auf Pufferspeicherebene wird die reale Hauptspeicheradresse einem bestimmten Platz im Pufferspeicher zugeordnet. Beim Hauptspeicher wird die heute meist virtuelle Adresse, die also bereits einen größeren Adressraum umfaßt, der realen Adresse zugeordnet. Bei den inhaltsadressierten höheren Hierarchiestufen übernehmen die vorerwähnten Indextabellen die Datenlokalisierung: Logisches und hierarchie-spezifisches Suchen wird identisch.

Die Zuordnungstabellen werden entweder auf der gleichen oder auf unteren Ebenen gespeichert. Beim (schnellen) Pufferspeicher wird die Tabelle in einem eigenen mehr oder weniger assoziativ arbeitenden Speicher gehalten.

Man kann sich so das gesamte DV-System vorstellen als die Kombination eines Archivspeichers, der alle Daten im ON-line Zugriff enthält, beispielsweise einen magnetischen Bandspeicher mit automatischem Bandtransport, und einem Prozessorsystem, das wiederum aus dem eigentlichen Prozessor und einer Hierarchie von Arbeitsspeichern besteht. Die ver-

schiedenen, teilweise im vorigen Abschnitt diskutierten Technologie- und Steuerungsparameter variieren entlang der Hierarchieachse wie in Abb. 6 skizziert.

### 3.2 Leistungsbetrachtung

Das wichtigste Kriterium der Speicherhierarchie ist die Gesamtzugriffszeit bzw. Gesamtzugriffsrate, absolut gesehen als auch kostenbezogen. Diese Zusammenhänge sollen im folgenden anhand eines sehr einfachen Modells diskutiert werden. Das Modell orientiert sich an "typischen" Werten für die verschiedenen Parameter und extrapoliert bei nicht bekannten Daten.

Wie das Technologiediagramm Abb. 2 bereits indiziert, scheint eine natürlich einfache Gesetzmäßigkeit zwischen den Bitkosten und der Spektrumsvariablen Zugriffszeit zu bestehen. Diese und die Zuordnung der Trefferrate und Speicherkapazität zur Zugriffszeit sind im Modellparameterdiagramm Abb. 7 aufgetragen. Die Kapazitätsverteilungskurve ist als Gerade (im log. Maßstab) angenommen, mit den Endpunkten Puffer- und Archivspeicher. Die gewählte Archivkapazität ist  $10^{12}$  b, die Pufferkapazität 200 Kb. Die auf der Geraden liegenden Punkte für Haupt- und Plattenspeicher entsprechen etwa realen Werten. Die Kapazitätsverteilungskurve ist an sich natürlich innerhalb des technologisch verfügbaren Spektrums frei wählbar. Mit wachsender Prozessorleistung und Datenmenge wird sie nach oben verschoben werden.

Für die Trefferrate im multiprogrammierten Stapelbetrieb liegen als Funktion der Kapazität und Blocklänge einige Erfahrungsdaten im Bereich Puffer - Hauptspeicher vor [9]. Typische Werte dafür wurden der Modellkurve zugrundegelegt. Zu den oberen Hierarchieebenen hin wurde extrapoliert.

Das Modell berücksichtigt nicht die gegenseitigen Abhängigkeiten von Blocklänge, Zugriffszeit, Trefferrate, Multiprogrammierungsgrad usw., sondern nimmt starr typische Werte an.

Die Gesamtzugriffszeit ist

$$t_{\text{ges}} = t_1 + (1-h_1)t_2 + (1-h_2)t_3 + \dots + (1-h_{n-1})t_n \quad \text{Gl. 1}$$

mit

$$\begin{aligned} t_n &= \text{Zugriffszeit der } n\text{-ten Stufe} \\ h_n &= \text{Trefferrate der } n\text{-ten Stufe} \end{aligned}$$

Die maximale Gesamtzugriffsrate, d.h. der Zugriffsfluß an der Schnittstelle zum Prozessor ist

$$\text{max. } Z_{\text{ges}} = \frac{1}{\frac{t_1}{p_1} + \frac{1-h_1}{p_2} t_2 + \dots + \frac{1-h_{n-1}}{p_n} t_n} \quad \text{Gl. 2}$$

mit

$$p_n = \text{Zugriffsparellität auf der } n\text{-ten Stufe.}$$

Die Zugriffsparellität entspricht in etwa der Modularität. Es wird angenommen, daß 50% der Zugriffsparellität sich jeweils in echter Erhöhung der Zugriffsrate durch Multiprogrammierung niederschlagen,  $p_{\text{eff}}$  also 0,5 p. Ferner, daß unterhalb der Plattenspeicherebene Programmumschaltung nicht mehr lohnt ( $p=1$ ) und schließlich, daß Einzel-Prozessorbetrieb vorliegt. Gl. 2 modifiziert sich dann entsprechend.

Einige Modellergebnisse auf der Grundlage realer Technologien sind in Tabelle II zusammengestellt. Unterschiedliche Speicherzugriffsraten schlagen sich in unterschiedlicher Prozessorauslastung nieder. Es wurde ein Modellprozessor mit 2 MIPS (Millionen Instruktionen pro Sekunde) und durchschnittlich 2 Zugriffen pro Instruktion gewählt. Dieser Prozessor kann seine volle Leistung nur entfalten, wenn das Speichersystem 4 Millionen Zugriffe pro Sekunde zuläßt.

Die schlechte Auslastung dieses 2-MIPS-Prozessors bei heutiger Konfiguration ohne Multiprogrammierung überrascht nicht. Auch mit Multiprogrammierung ist die Auslastung nur mäßig.

Erst die Einführung des elektronischen Massenspeichers erbringt eine Verbesserung auf eine vernünftige Größenordnung. Bei Multiprogrammierung verlagert sich jetzt der Engpaß für die Zugriffsrate vom Plattenspeicher (mit seiner hohen Modularität) zum Bandspeicher. Dieser Engpaß könnte überwunden werden durch weitere Erhöhung der Hierarchiestufenzahl, konkret durch Einbau einer Zwischenstufe zwischen Platten- und Bandspeicher.

Technologisch liegt eine solche Stufe im Bereich des Sichtbaren, nämlich über eine Modifizierung des konventionellen Plattenspeichers zu einem Satz von flexiblem Platten mit sehr hoher Bit-Volumendichte [9]. Die Zugriffsrate der Hierarchiekonfiguration liegt dann oberhalb von 4 Millionen pro Sekunde.

Die Ergebnisse aus Tabelle II werfen die Frage nach der optimalen Hierarchiestufung auf, bei festgehaltenen Endpunkten. Für diese Analyse wird ohne Bezug auf reale Technologien eine gleichmäßige Stufung vorgesehen und die Stufenzahl variiert. Multiprogrammierung wird jetzt nicht berücksichtigt. Ergebnisse sind in Abb. 8 aufgetragen: Bei ca. 16 Stufen stellt sich ein Sättigungswert für die Zugriffsrate ein (die in diesem einfachen Fall der reziproke Wert der mittleren Zugriffszeit ist). Diese Zugriffsrate ist nur etwa 2 mal kleiner als die der reinen Pufferspeicherstufe.

In Abb. 8 ist weiterhin die Preisleistungszahl, nämlich Zugriffsrate pro Gesamtbitkosten, aufgetragen.

Hier liegt das Optimum bei ca. 8-10 Stufen. Die Verbesserung gegenüber einer 4-stufigen Hierarchie ist größer als Faktor 6. Auf der Grundlage der realeren Daten in Tabelle II ist der Gewinn bei einem Schritt von heutigen 4 Stufen auf (die durchgespielten) 6 Stufen noch wesentlich höher, da dort nicht von einer gleichmäßigen Stufung ausgegangen wurde.

Ein weiterer Vorteil der feineren Hierarchiestufung ist die Verbesserung des Prozessor-"Wirkungsgrades": Die Zahl der Zugriffe zum Platten- und Bandspeicher nimmt ab. Damit nimmt auch die Zahl der prozessierten Instruktionen (der Zugriffsroutinen) pro Zugriff zur Speicherhierarchie ab, und der Prozessor-"Wirkungsgrad" nimmt zu. Schließlich kann das Betriebssystem einfacher gehalten werden.

In diesem Modell ist der Zuverlässigkeitsaspekt nicht enthalten, der mit wachsender Stufenzahl kritischer wird. Ebenso sind die Kosten der Steuerungen, Adresstabellen, etc. nicht berücksichtigt. Die Extrapolation der Trefferratenkurve ist völlig hypothetisch. All dessen ungeachtet dürfen die Modellergebnisse als Indiz dafür verstanden werden, daß eine feinere Hierarchiestufung noch erhebliches Leistungspotential enthält.

## 4. SPEICHERASPEKTE BEI DATENBANKBETRIEB

Auch der Datenbankbetrieb kann grundsätzlich in die bisherige Modellbetrachtung eingeordnet werden. Derjenige Parameter, der sich möglicherweise (in Richtung ungünstiger Werte) ändert, ist die Trefferrate, insbesondere auf den hohen Ebenen. Erfahrungen darüber müssen aber erst gewonnen werden, sodaß hier die Modellwerte beibehalten werden, zumal auch bei der Datenbank ein gewisses "Nachbarschafts"-Verhältnis von Anfragen festzustellen sein dürfte. Praktisch-anschaulich könnte man sich eine Funktionsverteilung auf die einzelnen Hierarchiestufen wie in Tabelle III skizziert, vorstellen. Datengruppen mit hoher professioneller Zugriffsrates müssen von der Archivstufe auf die Plattenspeicherstufe resident ausgelagert werden.

Der spezifische Datenbank-Leistungsparameter ist, neben der Datenmenge, die zulässige Anfragenrate. Diese sollte mit wachsender Datenbankkapazität auch ansteigen. Die folgende Überschlagsrechnung möge einige Veranschaulichung bringen:

Nach Tabelle II ist bei heutiger Hierarchie und Multiprogrammierung die Modellzugriffsrates .85 M/s. Wenn wir einen Programmablauf von durchschnittlich 100 K Instruktionen pro Datenbank-Anfrage annehmen, würde das System 4.25 Anfragen pro Sekunde erlauben. Dieser Wert dürfte bei einer Datenbank-Kapazität von  $10^{12}$  b nicht ausreichen. Nach Einführung des elektronischen Massenspeichers erhöht sich die Anfragenrate auf 14 pro Sekunde. Mit einer zusätzlichen Zwischenstufe zwischen Platten- und Bandspeicher erhöht sie sich auf ca. 30 pro Sekunde - entsprechende Prozessorleistung von ca. 3 MIPS vorausgesetzt.

Die letzten Endes interessierende Frage, wieviele Terminals an eine Datenbank dieser Größe bei befriedigender Bedienung angeschlossen werden können, hängt natürlich von der mittleren Anfragemlast pro Terminal ab. Bei einer angenommenen mittleren Last von einer Anfrage pro Terminal und Minute errechnet sich eine Terminalzahl von  $30 \cdot 60 = 1800$ . Diese Anschlußmöglichkeit pro  $10^{12}$  b Datenbankkapazität erscheint ausreichend.

Als Schlußfolgerung aus diesen Betrachtungen soll die Feststellung getroffen werden, daß Organisation und Technologie zukünftiger Speichersysteme das Potential haben, den Leistungsanforderungen eines breiten Datenbankbetriebes gerecht zu werden.

## Literatur

- [ 1] C.W. Pugh, "Storage Hierarchies: Gaps, Cliffs and Trends",  
IEEE Transactions on Magnetics, Vol. Mag-7, No. 4, Dez. 1971
- [ 2] C. Johnson, "IBM 3850-Mass Storage System",  
Nat. Comp. Conf. 1975, S. 509
- [ 3] J. Kelly, "The Development of an Experimental Electron-Beam-  
Addressable Memory Module",  
Computer, Februar 1975
- [ 4] W.C. Hughes et. al., "BEAMOS, A New Electronic Digital Memory",  
Nat. Comp. Conf. 1975, S. 5-41
- [ 5] G.F. Amelio, "Charge-Coupled Devices for Memory Application",  
Nat. Comp. Conf. 1975, S. 515
- [ 6] W.S. Boyle et. al., "Charge-Coupled Devices - A New Approach to  
MIS Device Structures",  
IEEE Spectrum, Juli 1971, S. 18
- [ 7] A.H. Bobeck et. al., "A New Approach to Memory and Logic: Cylindri-  
cal Domain Devices",  
Proc. AFIPS Conf., Vol. 35, 1969
- [ 8] R.R. Martin et. al., "Electronic Disks in the 1980's",  
Computer, Februar 1975, S. 24
- [ 9] D.H. Gibson, "Considerations in Block-Oriented Systems Design",  
AFIPS Proc., Vol. 30, SJCC 1967, S. 75-80

TECHNOLOGIE PARAMETER	<ul style="list-style-type: none"> <li>- SPEICHERMEDIUM (HOMOGENITÄT, BITDICHTHE)</li>   <li>- BITZAHL PRO SCHREIB-LESE-STATION (MATRIX-/SEQUENTIELLE ANORDNUNG)</li>   <li>- DATENTRANSPORT</li> </ul>
OPERATIONS- PARAMETER	<ul style="list-style-type: none"> <li>- ZUGRIFFSZEIT</li> <li>- ÜBERTRAGUNGSZEIT = F(ÜBERTRAGUNGSBREITE, BLOCKLÄNGE, TAKTFREQUENZ)</li> <li>- MODULARITÄT → ZUGRIFFSRATE</li> <li>- BITKOSTEN → KAPAZITÄT</li> <li>- ZUVERLÄSSIGKEIT</li> <li>- FLÜCHTIGKEIT</li> <li>- ADRESSIERBARE EINHEIT (BYTE/BLOCK-ADRESSIERUNG)</li> </ul>

TABELLE I

SPEICHERPARAMETER

KONFIGURATION	$t_{[\mu s]}/p_{eff}$						$t_{ges}$ [ $\mu s$ ]	max. $Z_{ges}$ [ $10^6/s$ ]	Prozessor Auslastung [%]	Gesamt- Kosten [ $10^6 \text{ \$}$ ]	$Z_{ges}$ [ $\frac{1}{\text{g}\cdot s}$ ]
	P	H	E	SP	FP	B					
P+H+SP+B	0,03	0,04	9	0,3	0,37	0,11	2,8	1,27	0,084		
P+H+SP+B Multiprogr.	0,03	0,04	0,9	0,2	(1,17)	0,85	21	1,27	0,67		
P+H+E+SP+B	0,03	0,04	0,075	0,9	0,535	1,87	47	1,32	1,4		
P+H+E+SP+B Multiprogr.	0,03	0,04	0,075	0,009	(0,354)	2,82	70	1,32	2,1		
P+H+E+SP+F+B Multiprogr.	0,03	0,04	0,075	0,009	0,015	(0,17)	100	1,82	3,2		

TABELLE II

Modellhierarchie-Leistungsparameter (Prozessor 2 MIPS, 2 Zugriffe/Instruktion)

P	Pufferspeicher
H	Hauptspeicher
E	Elektronischer Massenspeicher
SP	Starre Platte
FP	Flexible Platte
B	Band



HIERARCHIEEBENE NR.	TECHNOLOGIE	TYP. KAPAZITAT	FUNKTION
1	BIP PUFFER- SPEICHER	4-16K BYTES	SCHNELLER ARBEITSSPEICHER FÜR VERKNÖPFUNG VON DATEN MIT PROGRAMMEN
2	FET HAUPT- SPEICHER	$10^5$ - $10^7$ B	BEREITSTELLUNG VON PROGRAMMEN UND DATEN FÜR ÜBERSCHAUBAREN OPERATIONSZEITRAUM
3	SCHIEBERE- GISTER- BZW E-STRAHL- SPEICHER	$10^7$ - $10^9$ B	HALTEN VON HÄUFIGEN PROGRAMMEN Z.B. BETRIEBSSYSTEM UND AR- BEITSDATEN Z.B. INDEXTABELLEN, DESKRIPTOREN, KATALOGE, ZEIGER- NETZE USW.
4	PLATTEN- SPEICHER	$10^8$ - $10^{10}$ B	DATEIEN FÜR PROFESSIONELLE BENUTZUNG, DATENSICHERUNG
5	BAND- SPEICHER (AUTOMAT, BANDTRANS- PORT)	$10^{10}$ - $10^{13}$ B	DOKUMENTEN-DATENBANK DATENSICHERUNG, ARCHIVIERUNG

TABELLE III

FUNKTIONSVERTEILUNG BEI DATENBANKBETRIEB

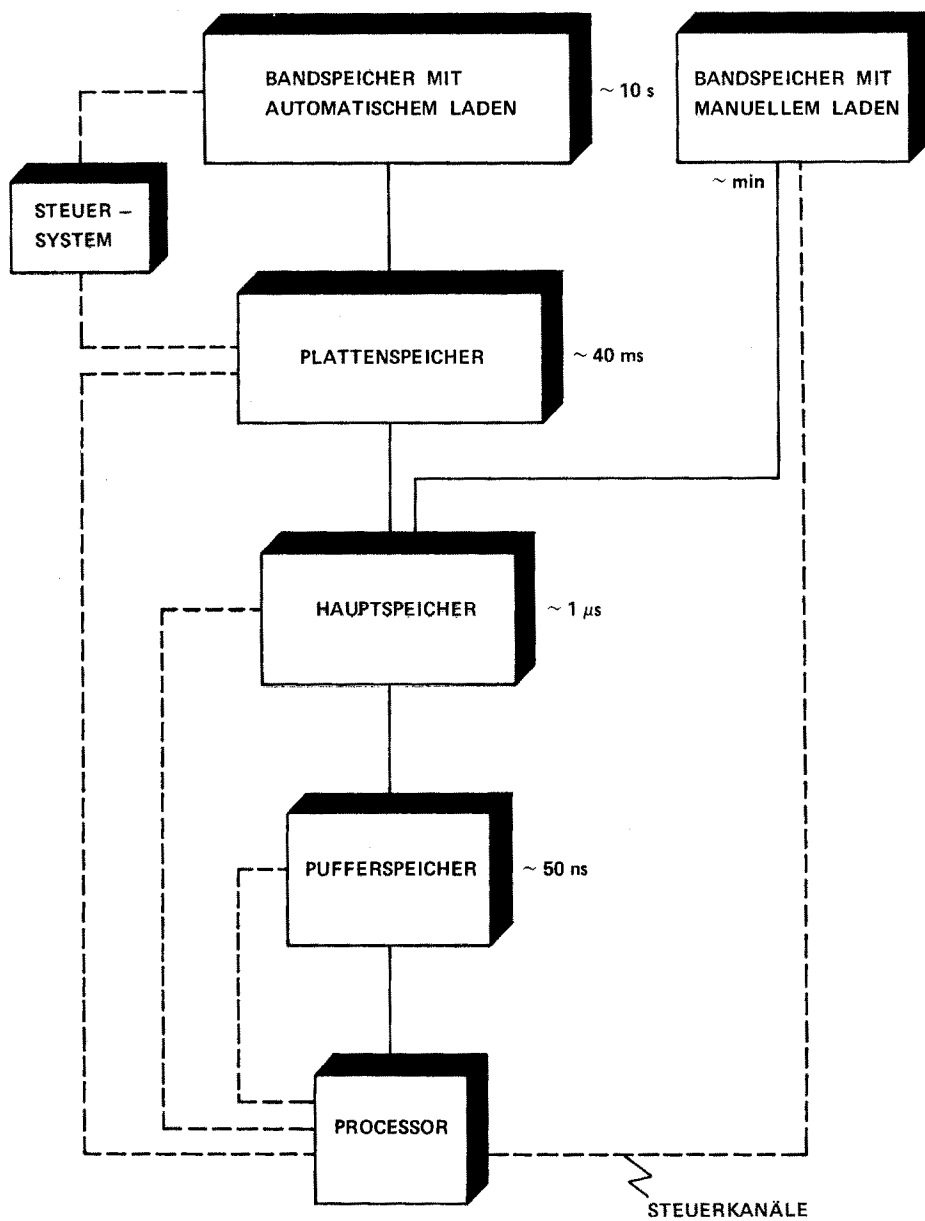


Abb. 1 SPEICHERHIERARCHIE HEUTE

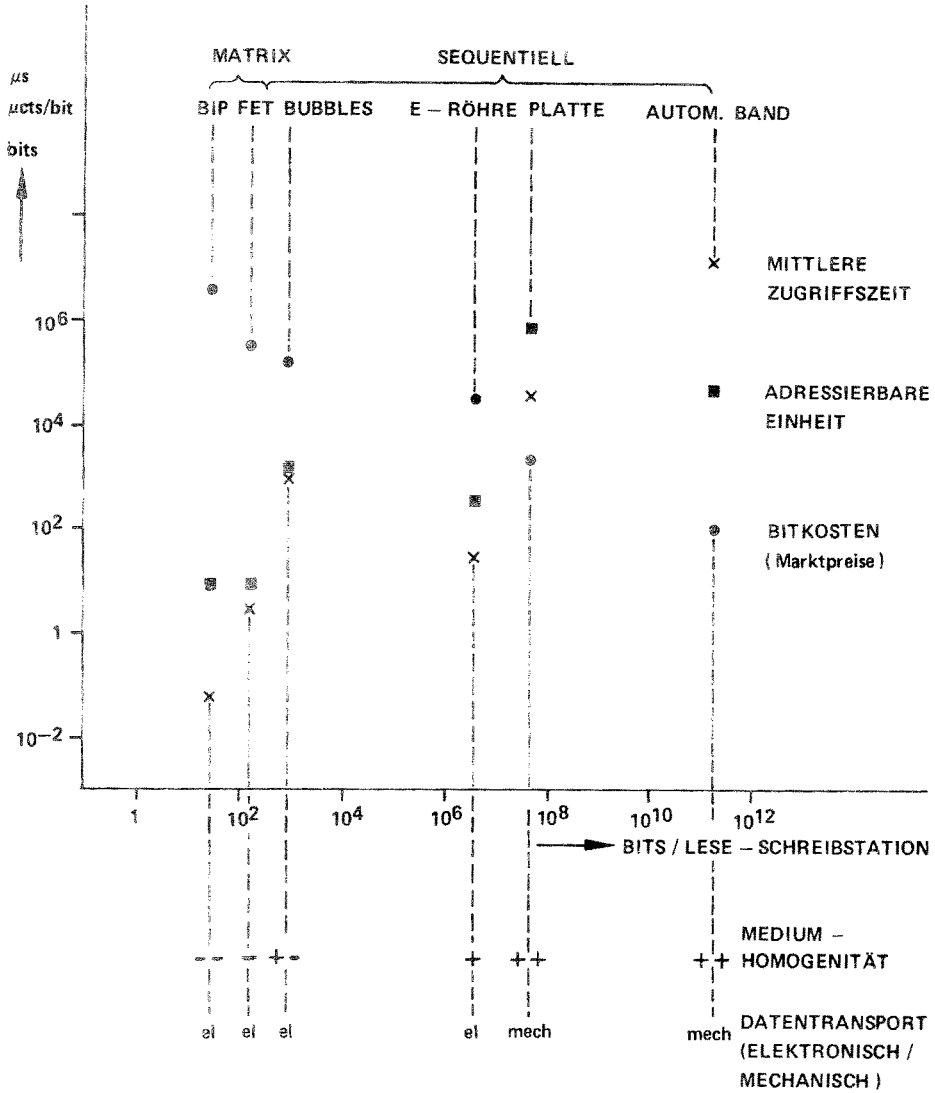
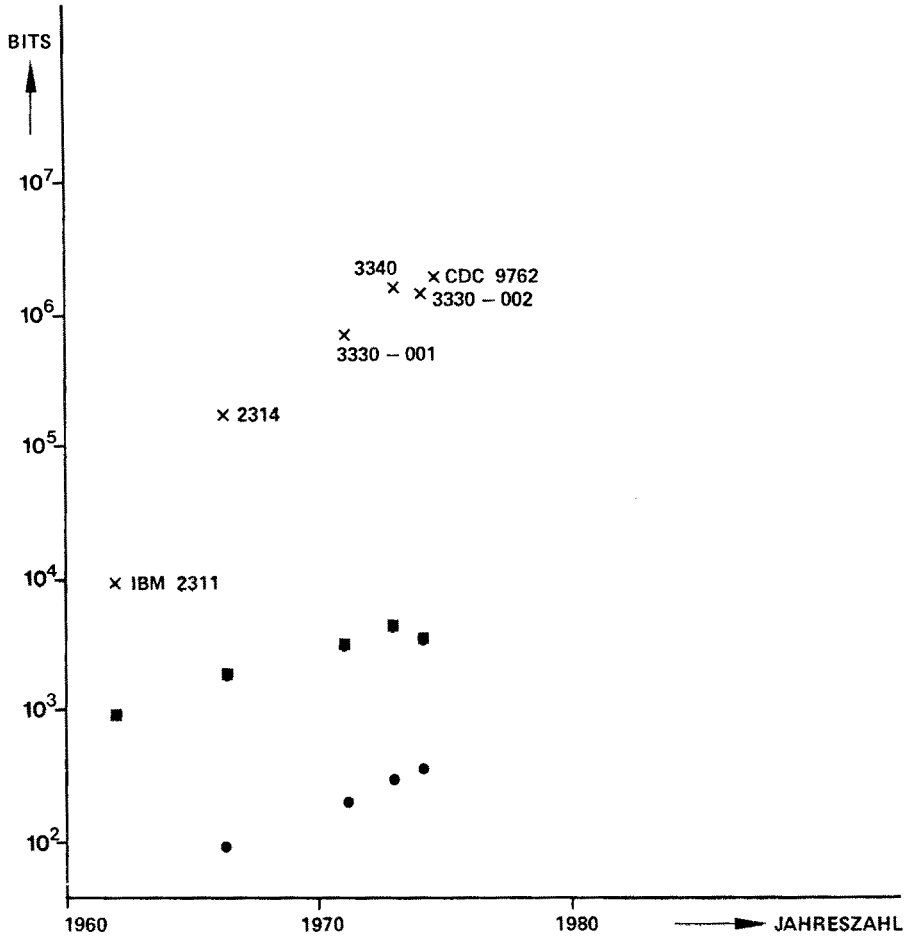


Abb. 2 OPERATIONSPARAMETER ALS FUNKTION DER TECHNOLOGIEPARAMETER



x BITFLÄCHENDICHTE    BITS / INCH<sup>2</sup>  
 ■ BITSPURDICHTE      BITS / INCH  
 ● SPURDICHTE         SPUREN / INCH

Abb. 3    PLATTENSPEICHER – BITDICHTE

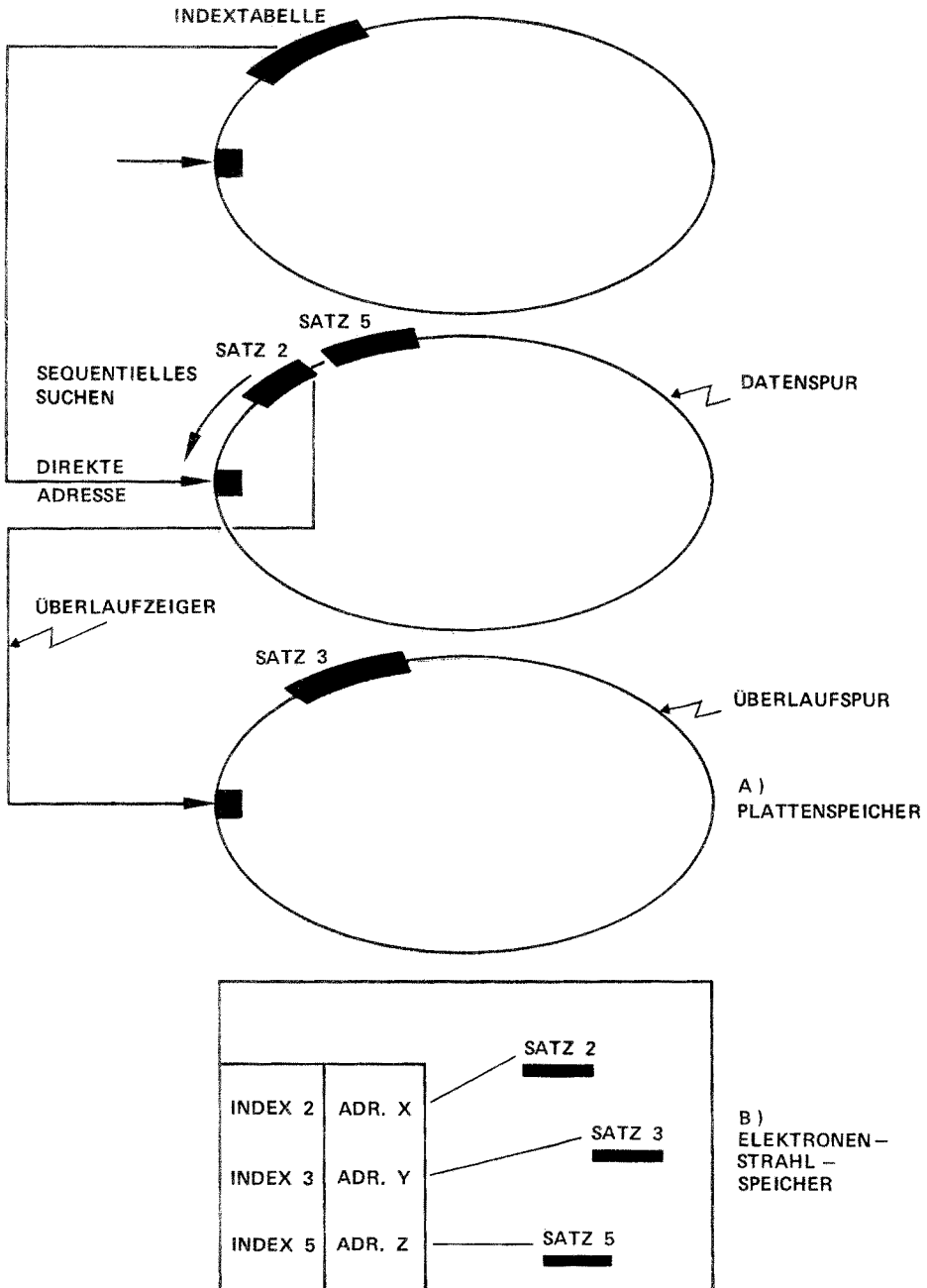


Abb. 4 ADRESSIERUNGSSYSTEME

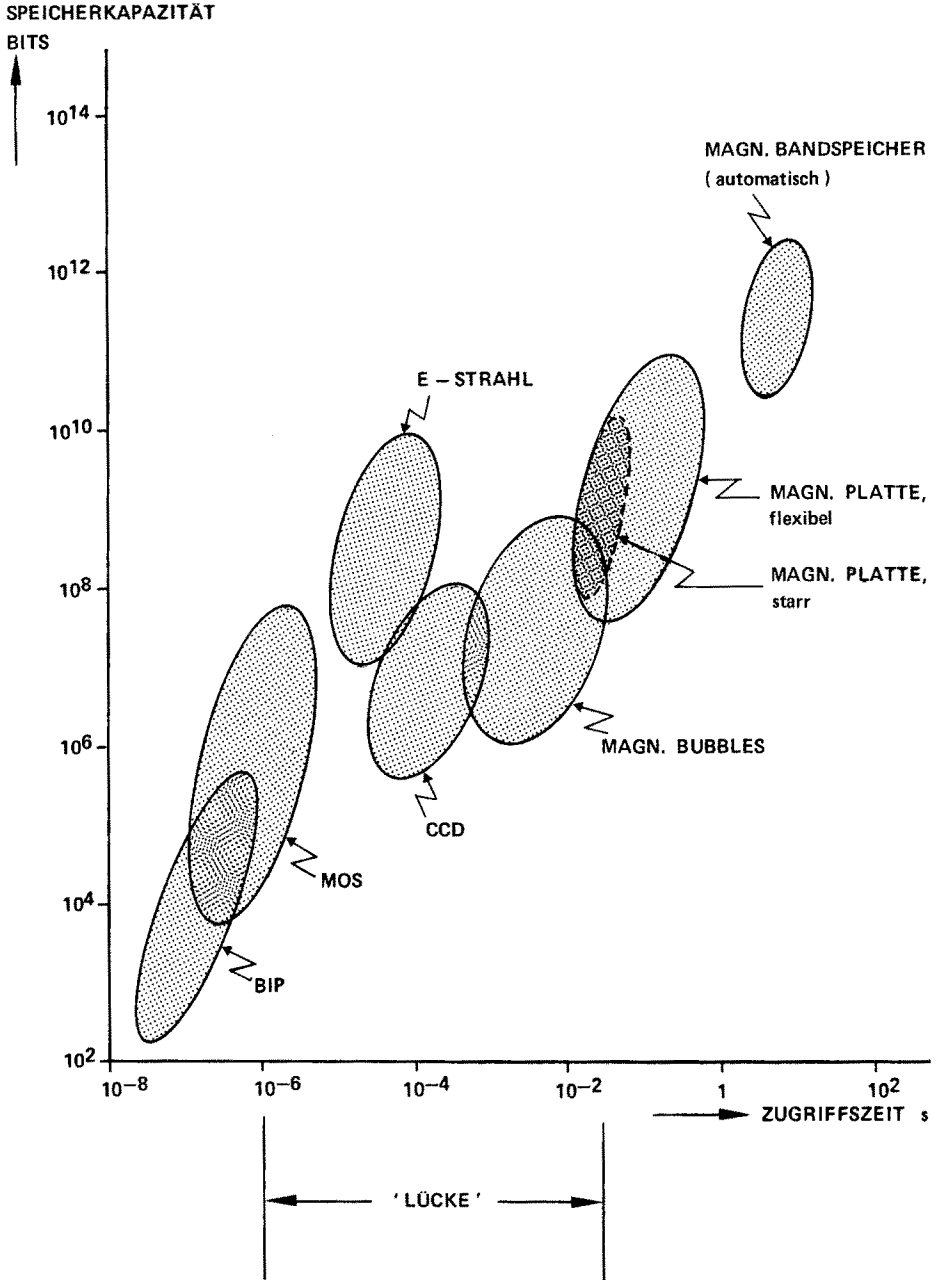


Abb. 5 TECHNOLOGIE – ÜBERSICHT ( ohne opt. Techn.)

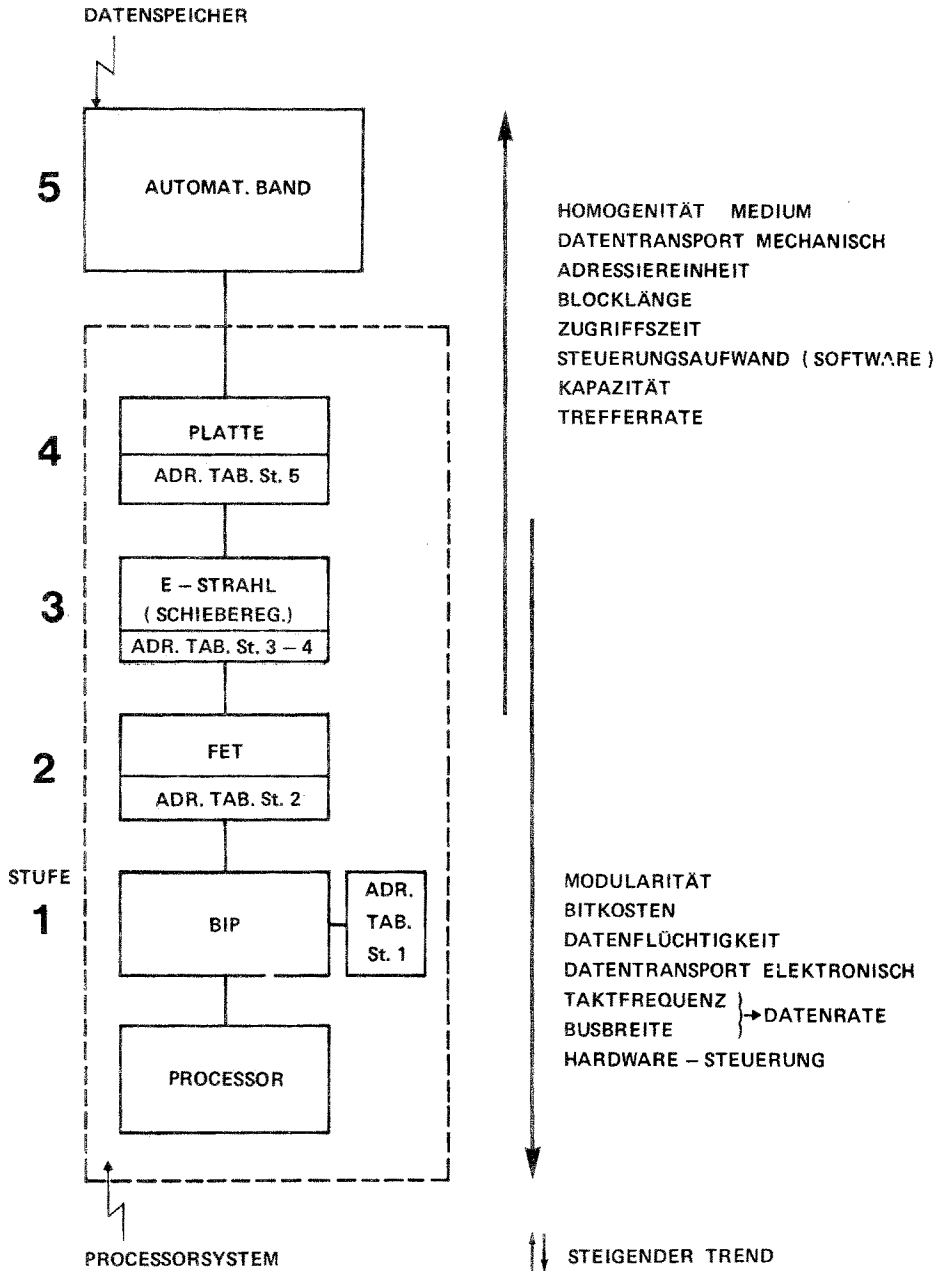
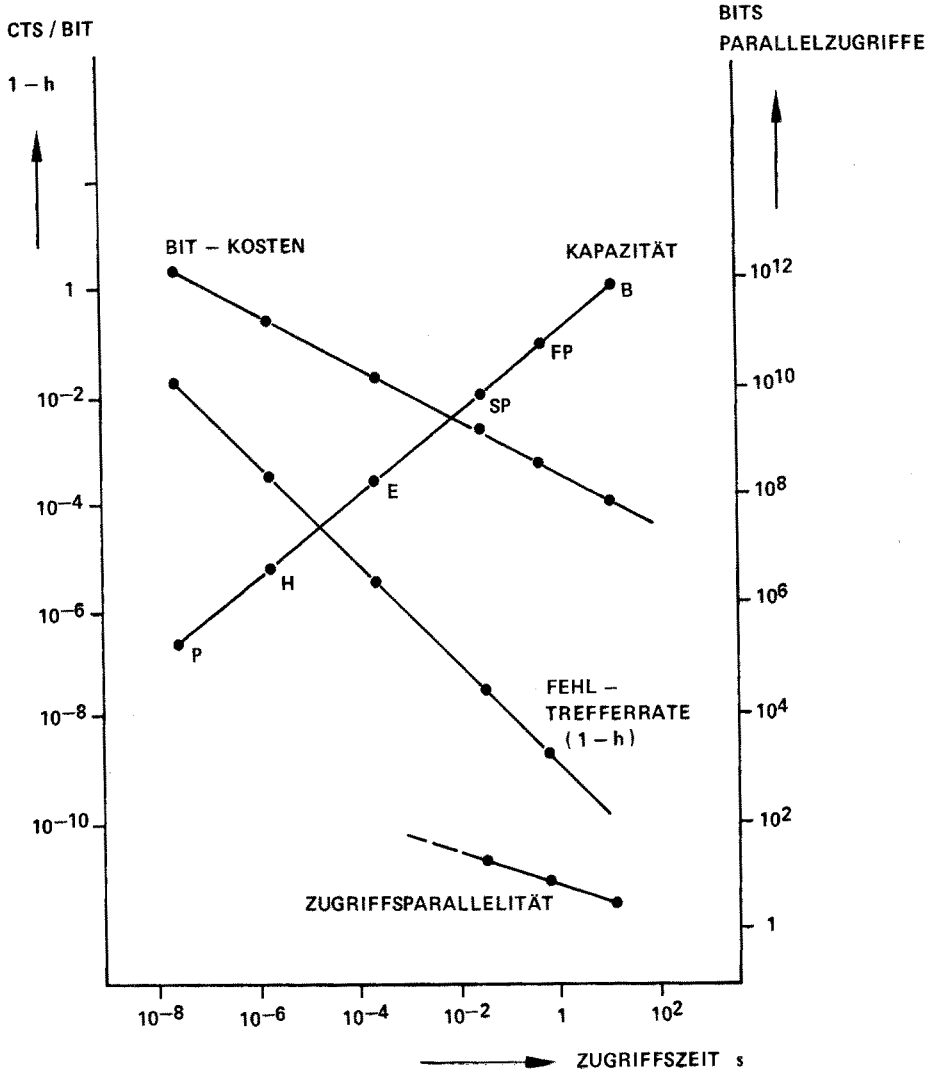


Abb. 6 PARAMETER TREND ÜBER HIERARCHIESPEKTRUM



- P PUFFERSPEICHER
- H HAUPTSPEICHER
- E ELEKTRON. MASSENSPEICHER
- SP STARRE PLATTE
- FP FLEXIBLE PLATTE
- B BAND

Abb. 7 MODELLPARAMETER



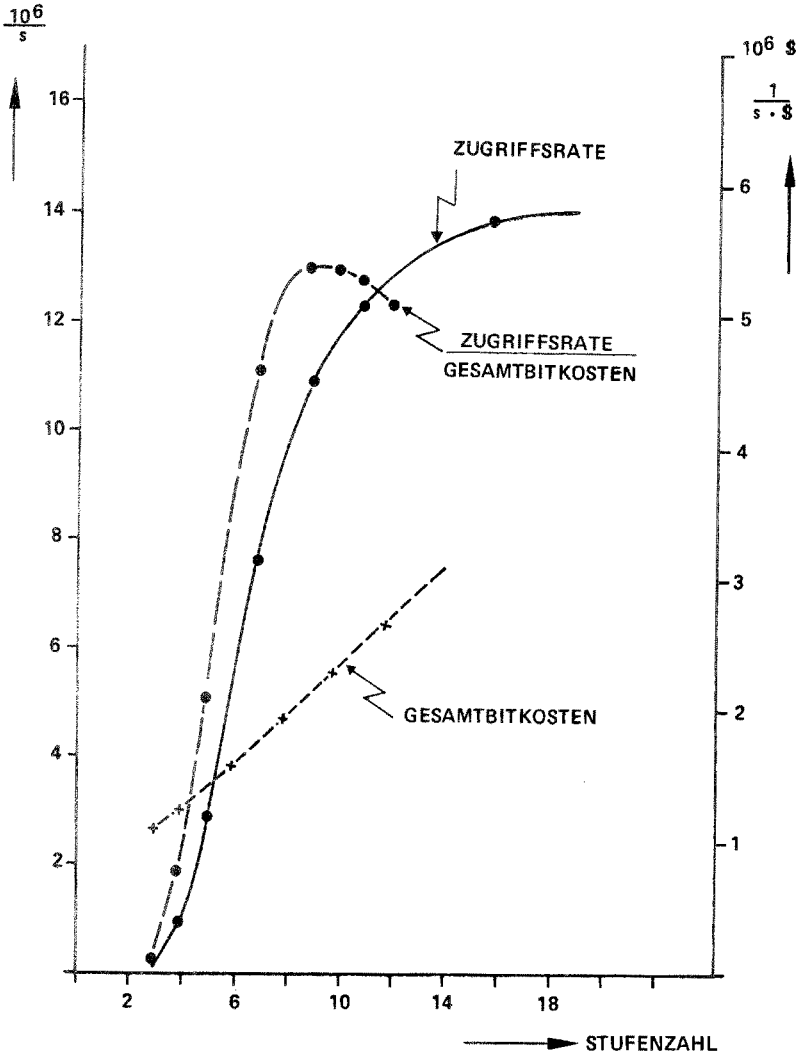


Abb. 8 MODELLERGEBNISSE  
GLEICHMÄSSIGE STUFUNG ( im log. Maßstab )