Ein Demultiplexer zur Unterstützung von Parallelität in einem ATM-Transportsystem

Torsten Braun
Institut für Telematik, Universität Karlsruhe, Zirkel 2, 76128 Karlsruhe
Telefon: 0721/608-3982, Telefax: 0721/388097
E-mail: braun@telematik.informatik.uni-karlsruhe.de

Kurzfassung

Zur Realisierung hochleistungsfähiger Kommunikationssysteme werden diese zunehmend auf parallelen Architekturen implementiert. In einer Implementierung des Transportsystems PATROCLOS übernimmt eine Demultiplexer-Komponente die Aufgabe, von einem ATM-Netz empfangene Zellen in Abhängigkeit der Art der darin enthaltenen Protokollinformation parallelen Verarbeitungseinheiten zuzuordnen.

Ein paralleles ATM-Transportsystem

Neuartige Anwendungen aus dem Multimedia-Bereich fordern leistungsfähige Kommunikationssysteme, die hohen Durchsatz bei niedrigen Verzögerungen erreichen. Das hohe Leistungspotential neuer Netze (z.B. ATM) kann jedoch dem Benutzer von heutigen Kommunikationssubsystemen nicht zur Verfügung gestellt werden. Eine Möglichkeit zur Realisierung von Hochleistungs-Kommunikationssystemen besteht im Einsatz paralleler Implementierungsarchitekturen. In zahlreichen Ansätzen [1] wurden dabei Standardprotokolle auf parallelen Implementierungsarchitekturen realisiert, während im Rahmen des PATROCLOS-Projekts [2, 3] ein speziell auf die parallele Verarbeitung hin zugeschnittenes Transportsystem implementiert wird.

Die Protokollarchitektur von PATROCLOS wurde speziell für zellenbasierte Hochgeschwindigkeitsnetze (z.B. ATM, DQDB) ausgelegt und besteht zur Unterstützung der parallelen Verarbeitung aus einer Menge parallel arbeitender, erweiterter endlicher Automaten (extended finite state machines, FSMs), von denen jeder eine bestimmte Protokollfunktion erfüllt. Jeder Automat innerhalb einer Instanz hat einen Partnerautomaten in der Partnerinstanz, mit dem er ein individuelles Protokoll abwikkelt. Das gesamte Protokoll zwischen den Partnerinstanzen setzt sich aus den verschiedenen Protokollen der Automaten zusammen.

Zur Adressierung der einzelnen Automaten wird eine Erweiterung der Transportinstanzen-Adresse, die sogenannte Automatenadresse verwendet, wobei für jeden Automaten ein Bit zur Verfügung steht. Die Adreßerweiterung ist im Kopf einer PATROCLOS-Protokolldateneinheit (P-Rahmen) und damit in der ersten ATM-Zelle eines P-Rahmens enthalten. Das Adreßformat erlaubt insbesondere eine einfache und daher effiziente Zuordnung der P-Rahmen, bzw. Zellen, zu den Prozessormodulen, auf denen die Automaten implementiert werden. Bild 1 zeigt das Format der ersten Zelle eines P-Rahmens.

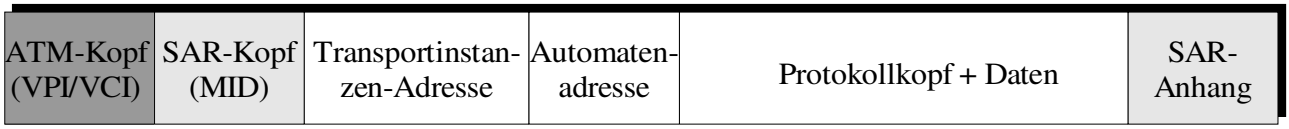

Bild 1: Format einer BOM-Zelle

Die parallele Implementierungsarchitektur

Die für PATROCLOS entwickelte Implementierungsarchitektur (Bild 2) setzt sich aus Universalprozessoren (Transputern) sowie dedizierten Hardwareeinheiten zusammen und bildet damit eine hybride Multiprozessorarchitektur. Die Hardwareeinheiten haben die Aufgabe, Funktionen auf Zellenebene wie Segmentieren und Reassemblieren (gemäß AAL 3/4) zu beschleunigen sowie die parallele Protokollverarbeitung innerhalb des Transportsystems zu unterstützen. Die Automaten sind demgegenüber auf den Universalprozessoren in Software implementiert. Da die Automaten weitgehend unabhängig voneinander arbeiten, lassen sie sich leicht auf ein Multiprozessorsystem abbilden. Dabei

kann ein Automat einen eigenen Prozessor benutzen oder sich mit anderen Automaten einen Prozessor teilen.

Ein oder mehrere Universalprozessoren (Transputer), eine Zellen-Reassembliereinheit, ein gemeinsamer Speicher und eine Buszugriffsteuerung zur Koordinierung des Zugriffs auf den gemeinsamen Speicher bilden zusammen ein Prozessormodul. Auf diese Prozessormodule werden die von einem ATM-Netz empfangenen Zellen für die weitere Verarbeitung verteilt. Die Auswahl eines Prozessormoduls wird von einem Demultiplexer vorgenommen. Die Zellen werden in den Prozessormodulen zu P-Rahmen reassembliert, so daß deren weitere Verarbeitung durch die Prozessormodule parallel zueinander erfolgen kann. Die Verarbeitung empfangener Protokolldateneinheiten bildet meist den Engpaß bei der Protokollverarbeitung. Dieser Engpaß soll bei der Implementierungsarchitektur insbesondere durch den mit Hilfe des Demultiplexers erzielten hohen Parallelitätsgrad vermieden werden.

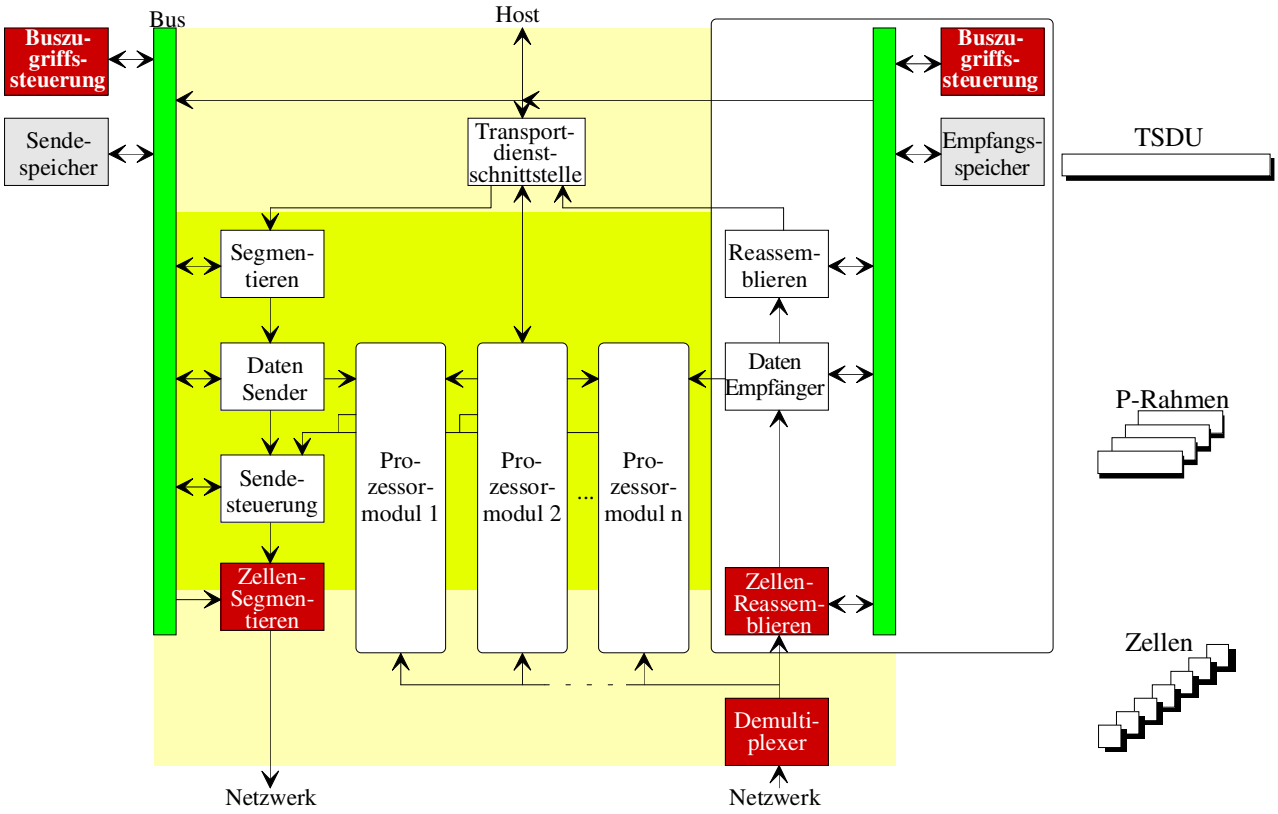

Bild 2: PATROCLOS-Implementierungsarchitektur

Der Demultiplexer zur Unterstützung paralleler Verarbeitung

Der Demultiplexer bildet die Schnittstelle zwischen dem ATM-Netzwerk und der für die Implementierung des Transportsystems verwendeten Multiprozessorarchitektur. Er verteilt die für das Endsystem bestimmten P-Rahmen an die für die weitere Verarbeitung zuständigen Prozessormodule (vgl. Bild 3). Der Demultiplexer leitet jedoch nicht komplette P-Rahmen, sondern die einzelnen Zellen an die verschiedenen Reassembliereinheiten der jeweiligen Prozessormodule weiter. Die Reassembliereinheiten, welche die Schnittstelle der Prozessormodule zum Demultiplexer darstellen, setzen dann die Zellen wieder zu P-Rahmen zusammen.

Eine Reassembliereinheit eines Prozessormoduls wird abhängig von der Automatenadresse, die im Kopf eines P-Rahmens enthalten ist, ausgewählt. Die Automatenadresse ist folglich nur in der ersten Zelle eines Rahmens enthalten, nicht jedoch in eventuell nachfolgenden Zellen. Aus diesem Grund muß der Demultiplexer die Automatenadresse für jede BOM-Zelle eines Rahmens speichern, um nachfolgende COM- und EOM-Zellen an das richtige Prozessormodul weiterleiten zu können. Der Demultiplexer verwendet dazu eine Tabelle, welche einem für jeden Rahmen eindeutigen Identifikator eine Automatenadresse zuordnet. Bei BOM-Zellen wird der Identifikator bestehend aus Kanal-

/Pfadidentifikator (VCI/VPI) und Nachrichtenidentifikator (MID) sowie die Automatenadresse (FSMA) in die Tabelle geschrieben, bei einkommenden COM- und EOM-Zellen wird aus dieser Tabelle die Automatenadresse zum zugehörigen Identifikator ermittelt. Für SSM-Zellen ist kein Schreib- oder Lesezugriff auf die Tabelle erforderlich.

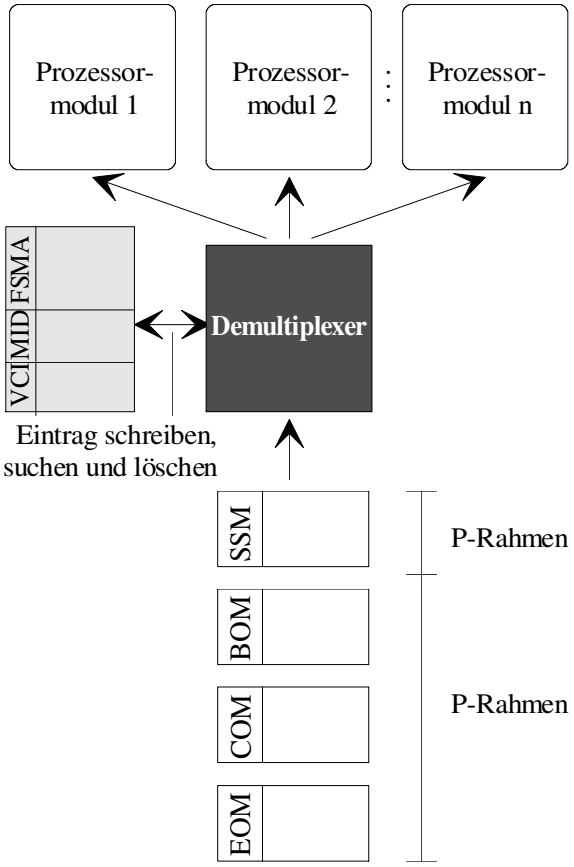

Bild 3: Funktion des Demultiplexers

Realisierung des Demultiplexers

Bild 4 zeigt die Realisierung des Demultiplexers. Kernteil ist die Demultiplex-Steuerung, die den gesamten Ablauf kontrolliert und dazu die anderen Komponenten ansteuert. Die Zellendaten werden mit Hilfe der von einem Zähler erzeugten Signale byteweise von der Netzwerkzugangseinheit in den FIFO-Speicher übernommen. Über einen Filter werden die für das Demultiplexen notwendigen Informationen aus dem Zellenkopf (Kanal-/Pfad-, Nachrichtenidentifikator) und der Nutzlast (Automatenadresse, Zellentyp) extrahiert. Der Filter setzt die Informationen in eine für den Assoziativspeicher (content addressable memory, CAM) geeignete Form (Suchinformation) zusammen und signalisiert der Demultiplex-Steuerung den Zellentyp. Die Automatenadresse einer Zelle wird ermittelt und dem Auswahlgenerator mitgeteilt, der die Automatenadresse in Auswahlsignale für die Reassembliereinheiten der Prozessormodule umsetzt. Die Umsetzung hängt dabei von der Verteilung der Automatenprozesse auf die Prozessormodule ab. Die Automatenprozesse sind in der Regel einem Prozessormodul fest zugeordnet.

Eine weitere Komponente erzeugt zusätzlich Steuersignale zur Ansteuerung der Reassembliereinheiten. Die Prozessormodule werden nur dann angesteuert, wenn alle ausgewählten Reassembliereinheiten ausreichend Speicherplatz in ihrem Eingangs-FIFO-Speicher besitzen, um eine weitere Zelle aufzunehmen. Ansonsten wird die Zelle vernichtet. Das Auslesen der Zellendaten vom FIFO-Speicher in einen Ausgabepuffer wird durch einen weiteren Zähler unterstützt.

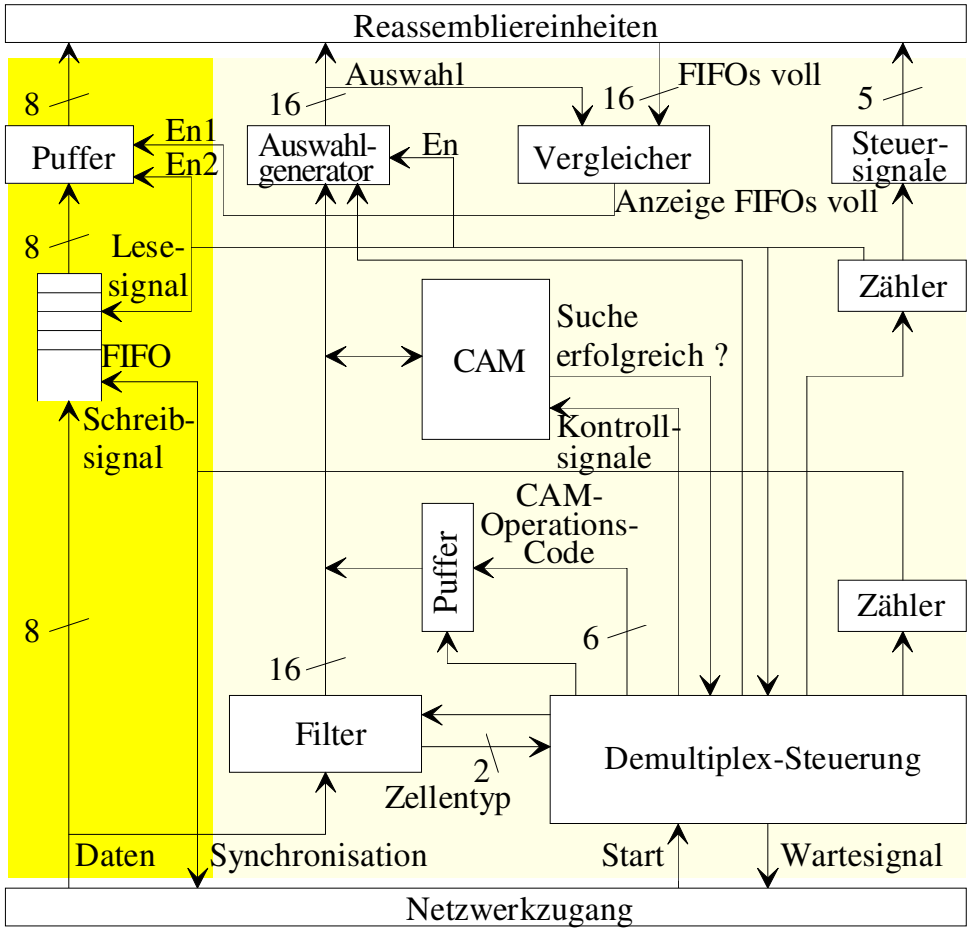

Bild 4: Demultiplexer

Die Demultiplex-Steuerung [4] (s. Bild 5) steuert den Ablauf der gesamten Demultiplexer-Hardware. Beim Eintreffen einer neuen Zelle veranlaßt sie den Filter durch entsprechende Signale, die notwendigen Informationen aus den Zellendaten zu filtern und der Demultiplex-Steuerung den Zellentyp mitzuteilen. Im einfachsten Fall, dem Eintreffen eines aus einer einzigen Zelle (SSM-Zelle) bestehenden P-Rahmens, wird die Automatenadresse direkt und ohne weitere Aktionen an den Auswahlgenerator weitergeleitet. In allen anderen Fällen wird die Suchinformation in ein CAM-Register geladen. Bei BOM-Zellen wird zunächst überprüft, ob die enthaltene Suchinformation bereits in der Tabelle enthalten ist und der Eintrag gegebenenfalls aus der Tabelle gelöscht. Danach wird die neue Suchinformation in den CAM-Speicher geschrieben und die Automatenadresse direkt an den Auswahlgenerator weitergeleitet. Bei COM- und EOM-Zellen muß die Automatenadresse durch einen Suchvorgang im CAM-Speicher bestimmt werden. Bei erfolgloser Suche wird die Zelle aus dem FIFO-Speicher gelöscht, indem sie ausgelesen wird, ohne sie an den Ausgabepuffer weiterzugeben. Ansonsten wird der für das Auslesen der Zellen zuständige Zähler gestartet und der Ausgabepuffer aktiviert. Bei EOM-Zellen muß außerdem vor dem Auslesen des FIFO-Speichers die Suchinformation aus dem CAM-Speicher gelöscht werden.

Durch die separaten Einheiten zum Ein- und Auslesen der Zellen in den FIFO-Speicher und zur Bestimmung der Automatenadresse können diese Funktionen innerhalb des Demultiplexers nahezu unabhängig voneinander und daher parallel ausgeführt werden, was sich schließlich positiv auf die erzielbare Leistungsfähigkeit auswirkt. Aufgrund der parallelen Verarbeitung werden nur 55 Taktzyklen zur Verarbeitung einer 53-Byte-Zelle benötigt, d.h. zusätzlich zum byteweisen Lesen und Schreiben der 53 Bytes einer Zelle sind lediglich zwei weitere Taktzyklen erforderlich. Mit einem 40 MHz-Takt wird dabei ein Durchsatz von mehr als 300 Mbit/s erreicht. Die prototypische Realisierung des Demultiplexers besteht aus 2 MACH™210 Bausteinen für die Demultiplexer-Steuerung, einigen PLDs (GAL22V10, Treiber), einem FIFO (256 * 8 byte), und einem AMD 99C10A-CAM (256*48 bit).

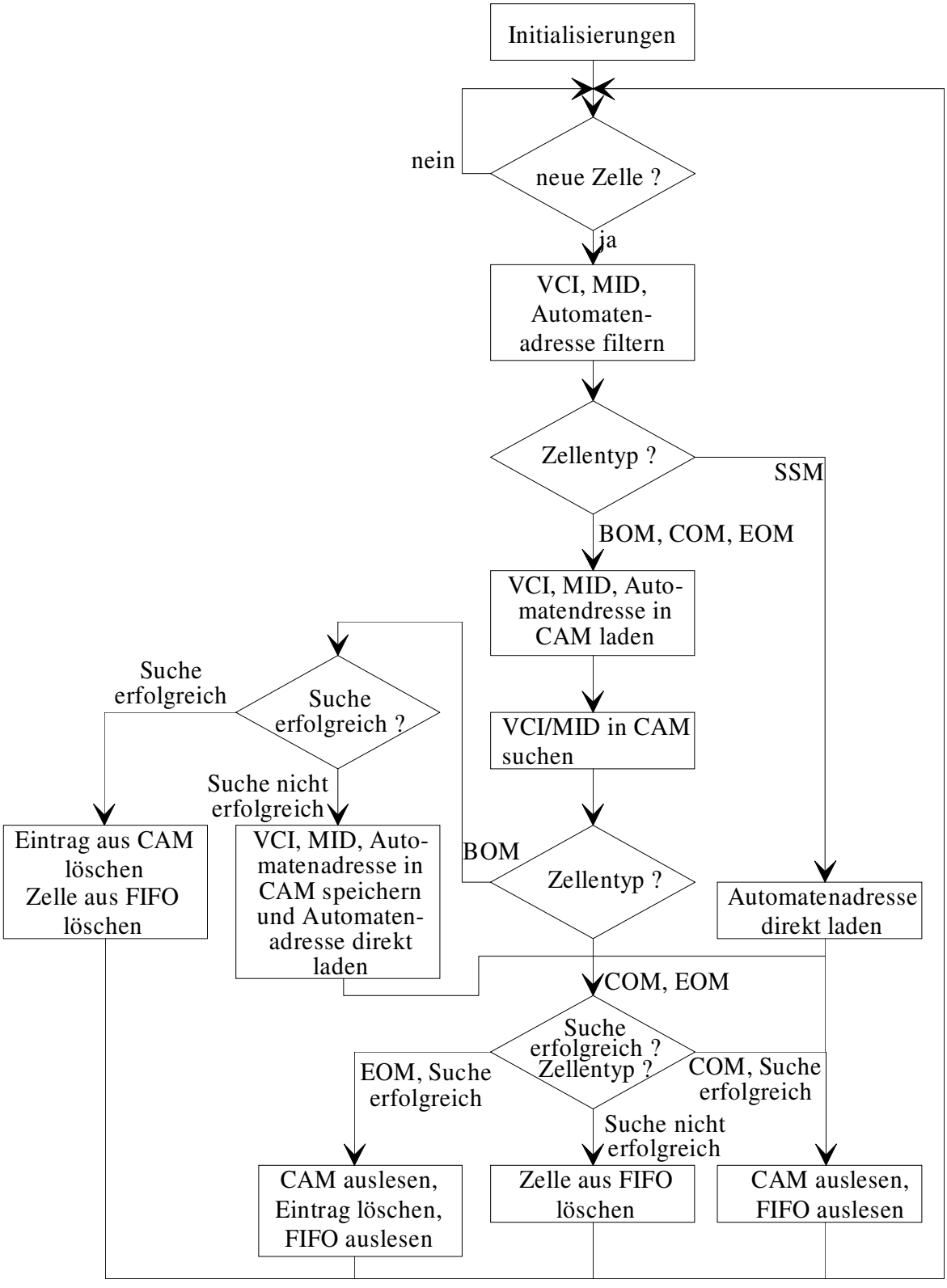

Bild 5: Demultiplex-Steuerung

Zusammenfassung und Ausblick

Der vorgestellte Demultiplexer separiert unterschiedliche Protokollinformationen in einem Transportsystem und ordnet diese parallelen Verarbeitungseinheiten zu. Er trägt entscheidend zu einer effizienten parallelen Verarbeitung von empfangen Protokolldateneinheiten innerhalb der PATROCLOS-Implementierung bei. Komponenten mit ähnlicher Funktionalität des Demultiplexers werden in verschiedenen anderen modernen Architekturen und Implementierungen von ATM-Kommunikationssystemen benötigt. In einer VLSI-Implementierungsarchitektur [5] für parallele Protokolle sind verschiedenen Prozessoren bestimmte Verbindungen zugeordnet. Ein Demultiplexer leitet die eintreffenden Daten an die jeweils dafür zuständigen Prozessoren weiter. Der sogenannte A-Chip separiert in der in [6] vorgestellten ATM-Workstation isochronen und nicht-isochronen Verkehr anhand der VPI/VCI-Werte. Bei der F-CSS-Kommunikationsarchitektur [7] wird ein Demultiplexer benötigt, um verschiedene Datenströme (z.B. Audio, Video, Dateien usw.) auf unterschiedliche Protokollmaschinen zu verteilen, die wiederum auf separate Verarbeitungseinheiten abgebildet werden können. Der Protokollfilter des Multimedia-Protokoll-Adapters [8] bestimmt anhand der Instanzenadresse eine Verbindungskennung in Netzwerk oder Transportprotokollen. Er benutzt ebenfalls einen CAM-Speicher und arbeitet nach einem ähnlichen Prinzip wie der PATROCLOS-Demultiplexer.

Referenzen

[1] Zitterbart, M.
Funktionsbezogene Parallelität in transportorientierten Kommunikationsprotokollen
VDI Verlag, Reihe 10: Informatik/Kommunikationstechnik, Nr. 183

[2] Braun, T.:
Parallel High Performance Transport System for MANs
Computer Communications, Vol. 16, No. 2, Februar 1993, S. 108-115

[3] Braun, T.; Zitterbart, M.:
Entwurf eines parallelen Transportprotokolls
in Gerner, N; Hegering, H.G.; Swoboda, J. (Hrsg.): Kommunikation in Verteilten Systemen, ITG/GI-Fachtagung, München, 3.-5. März 1993, Informatik aktuell, Springer-Verlag, S. 144-158

[4] Lindner, M.
Hardware-Unterstützung für ein paralleles Transportsystem
Diplomarbeit am Institut für Telematik und am Institut für Hoch- und Höchstfrequenztechnik, Universität Karlsruhe, Februar 1993

[5] Schiller, J.; Braun, T.
VLSI-Implementation Architecture for Parallel Transport Protocols
IEEE Workshop on VLSI in Communications, Stanford Sierra Camp, Lake Tahoe, Kalifornien, U.S.A., 15.-17. September 1993

[6] Koufopavlou, O.; Tantawy, A.; Zitterbart, M.
An ATM Communication Workstation
2nd International Conference on Broadband Islands, Athen, 15.-16. Juni 1993, S. 51-58

[7] Zitterbart, M.; Stiller, B.; Tantawy, A.
A Model for Flexible High-Performance Communication Subsystems
Journal on Selected Aread in Communications, Mai 1993

[8] Rütsche, E.
Multimedia Communication Subsystems. Architecture, Interfaces and Implementation
IBM Research Report, RZ 2477 (#81993)