

Hard- und Software-Entwicklung eines ATM- Testgenerator/Monitors

Herrmann, P., Keschull, U., Spruth, W.G.

**Universität Leipzig
Augustusplatz 10-11
04109 Leipzig**

Inhaltsverzeichnis

Abstrakt	4
1 Übersicht	5
2 Problemstellung	5
2.1 ATM-Übersicht	5
2.2 Test von ATM-Implementierungen	6
2.2.1 Teststrategien	6
2.2.2 Testgeräte	7
2.3 ATM-Architektur	8
2.3.1 Zellaufbau	8
2.3.2 B-ISDN-Schichtenmodell	8
2.4 Probleme mit dem Einsatz von ATM	9
3 Funktionale Ziele des ATM-Testgenerator/Monitors	10
3.1 Anforderungen	10
3.1.1 Monitor-Mode	10
3.1.2 Generator-Mode	11
3.1.3 Funktionsumfang des ATM-Testgenerator/Monitors	12
3.2 Einsatzmöglichkeiten	13
4 ATM-Testgenerator/Monitor-Hardware	13
4.1 Implementierungs-Problematik	13
4.2 ATM-Controller	15
4.2.1 Übersicht	15
4.2.2 Adaption Layer Controller (ALC)	15
4.2.3 Dual Port RAM	20
4.2.4 Arbiter	20
4.3 Cellpack-Modul (COSY)	21
4.4 PowerPC embedded Controller	21

4.5 FPGA-Schnittstelle	22
5 ATM-Testgenerator/Monitor-Software	22
5.1 Ergebnisse	22
5.1.1 Monitor-Umgebung (eingeschränkter Funktionsumfang)	23
5.1.2 Real Time Monitor für PPC embedded Controller	23
5.1.3 Funktionsfähige noch nicht integrierte Software	23
5.2 Benutzerschnittstelle	25
5.3 Transportschnittstelle	26
6 Unterstützende Arbeiten	27
6.1 Analyse der möglichen Verarbeitungsraten	27
6.2 Analyse der Verarbeitungsraten unter ATM-Bedingungen	28
6.3 ATM Socket-Schnittstelle	29
7 Literatur	30

Abstrakt

Das Modell von B-ISDN beschreibt ein digitales Allzweck-Netzwerk, das die Übertragung von Daten und Sprache in einem gemeinsamen Netzwerk gleichermaßen unterstützt. Als Transportprotokoll für B-ISDN wurde ATM (Asynchronous Transfer Mode) von der ITU (International Telecommunication Union) festgelegt. Da es sich bei ATM um verbindungsorientierte Hochgeschwindigkeits-Netzwerke mit einer Übertragungsrate ab 155MBit/s handelt, unterscheiden sich die Methoden zur Fehlerverifikation wesentlich von denen in herkömmlichen Netzen.

In der vorliegenden Arbeit wird ein ATM-Testgenerator/Monitor vorgestellt, der auf der Basis einer spezifischen Entwicklungsumgebung für eingebettete Systeme des FZI Karlsruhe an der Universität Leipzig entwickelt wird. Ausgehend von den momentan auf dem Markt befindlichen ATM-Testgeräten und Teststrategien wird die Hard- und Software- Architektur des ATM-Testgenerator/Monitors erläutert und die auftretenden Probleme diskutiert.

Um den ATM-Testgenerator/Monitor auch von entfernten Lokationen anzusprechen, soll eine Benutzerschnittstelle über ein geeignetes Protokoll mit dem Netscape Browser kommunizieren. Mit Hilfe eines CORBA ORBs werden dem Nutzer unterschiedliche Anwender-Programme zur Verfügung gestellt.

1 Überblick

Mit der wachsenden Anzahl der Multimedia-Anwendungen wurden in den letzten Jahren die Grenzen von ISDN (Bandbreite eines ISDN-Kanals 64 KBit/s) deutlich. Bei der Übertragung verschiedener Medien wie Audio, Video, Text und Bild über ein Netz sind hohe Anforderungen an die Bandbreite und die Übertragungsqualität erforderlich. Bitraten von z.T. über 100 MBit/s bei Videoanwendungen überfordert die Möglichkeiten von ISDN.

Während bei der Übertragung alphanumerischer Zeichen (z.B. Bildschirminhalt 40*80*8 Bit) maximal eine Bandbreite von 25,6 KBit/s notwendig ist, liegt der Bedarf bei Multimedia-Anwendungen wesentlich höher. Dieser kann aber mit geeigneten Optimierungs- und Kompressionsverfahren erheblich reduziert werden. Trotzdem sind noch Übertragungsbandbreiten zwischen 1 MBit/s (Bildausschnitt: Sprechende Köpfe) und 10 – 100 MBit/s (großer Bildausschnitt, hohe Qualität) notwendig.

Mit der Entwicklung von B-ISDN steht ein Breitband-Netzwerk zur Verfügung, das für die Übertragung verschiedener Dienste mit hohen Bitraten (z.B. Multimedia-Daten) geeignet ist. Als Übertragungstechnik für B-ISDN bietet ATM hohe Übertragungsgeschwindigkeiten bis in den Bereich von mehreren Gigabits (STM-16: 2,488 GBit/s) und ist darüberhinaus in der Lage, bestimmte Dienstgüten für die Übertragung zu garantieren.

Infolge der hohen Komplexität von ATM-Netzwerken spielen Performance- und Fault-Management eine zunehmende Rolle. Neben der Verifikation von systematischen Fehlern konzentriert sich die Fehlersuche hauptsächlich auf Fehlerquellen, die nur in spezifischen Lastsituationen oder Zellmustern auftreten und deshalb nur mit Hilfe von kostenintensiven Diagnosegeräten zu lokalisieren sind.

In den folgenden Kapiteln wird das Konzept zum Aufbau eines ATM-Testgenerator/Monitors vorgestellt, das auf der Basis einer Entwicklungsumgebung für eingebettete Systeme (FZI Karlsruhe) an der Universität Leipzig entwickelt und getestet wird.

2 Problemstellung

2.1 ATM-Übersicht

ATM (Asynchronous Transfer Mode) ist eine verbindungsorientierte Switching-Technologie, welche mit Zellen an Stelle der üblichen Rahmen arbeitet. Zellen sind Dateneinheiten mit fester Länge, während Rahmen Dateneinheiten mit variabler Länge darstellen [Pry94].

Bevor die Datenübertragung beginnen kann, muß eine virtuelle Verbindung VC (Virtual Channel) zwischen Sender und Empfänger aufgebaut werden. Dieser Aufbau wird erreicht durch Anmelden einer permanenten Verbindung oder durch Signalisierung. Auf dem physikalischen Übertragungsmedium können gleichzeitig mehrere virtuelle Verbindungen (Kanäle) bestehen. Die Adreßinformation im Zellkopf

ordnet die Zelle einer bestimmten virtuellen Verbindung zu. Bestehende virtuelle Verbindungen unterliegen dem asynchronen Zeitmultiplexing.

Ein virtueller Pfad VP (Virtual Path) ist ein gemeinsamer Transportweg für ein Bündel virtueller Kanäle zwischen zwei oder mehreren Netzelementen. Die Informationen über den virtuellen Kanal (VCI) und den virtuellen Pfad (VPI) sind im Header der ATM-Zelle enthalten. Der eigentliche Vermittlungsvorgang erfolgt mittels Tabellen, um die Eingangs-VCI und -VPI den zugehörigen Ausgangs-VCI und -VPI zuzuordnen. Die Einrichtung und Aktualisierung dieser Vermittlungstabellen wird vom Netzmanagement oder von einer Metasignalisierung vorgenommen.

ATM bildet den Transportmechanismus für B-ISDN-Netzwerke. Die logische B-ISDN-Netzwerkarchitektur wurde in Anlehnung an das OSI-Schichtenmodell in vier voneinander unabhängige Schichten eingeteilt. Die Signalisierung in B-ISDN-Netzen unterscheidet sich von bisher üblichen Formen des Nachrichtenaustausches. Die Einordnung der Schichten und Teilschichten des B-ISDN-Referenzmodells in das OSI-Schichtenmodell erweist sich als kompliziert, da das B-ISDN-Referenzmodell zwar angelehnt ist, viele Eigenschaften aber auch aus dem ISDN-Referenzmodell übernommen wurden. Bis heute existiert keine eindeutige, anerkannte Beschreibung der Beziehung zwischen beiden.

Die B-ISDN-Signalisierung enthält eine Vielzahl von Signal-Protokollen, die zum Teil eine signifikante Bedeutung für das Monitoring besitzen (z.B. Q.2110 der AAL, Q.2931 und UNI 3.1 der Higher Layer). Eine Beschreibung der Signalisierung einschließlich der relevanten Signalisierungs-Protokolle ist in [Tit 95] eingehend beschrieben.

2.2 Test von ATM-Implementierungen

Die Vielzahl der vorhandenen ATM-Hardware und Umsetzung von Empfehlungen bzw. Implementierungs-Vereinbarungen hat zu einer rasanten Entwicklung von Teststrategien und Testgeräten für die Evaluierung dieser Technologie geführt. Bei Teststrategien handelt es sich um einen theoretischen Ansatz zur Definition von wesentlichen Merkmalen von ATM, entwickelt von den Mitgliedern der ITU und des ATM-Forums.

Existierende Testgeräte sind in der Lage, Teile dieses theoretischen Ansatzes in die Praxis umzusetzen und so je nach Art eine Aussage über das Verhalten der zu untersuchenden Implementierung zu liefern.

2.2.1 Teststrategien

Bei allen vorhandenen Teststrategien ist ein Bottom-Up-Ansatz zu erkennen. Die am besten definierten Testszenarien existieren für die unterste Schicht von B-ISDN (physikalische Schicht PLCP). Für die ATM-Schicht existieren Vorlagen, die detaillierte Test-Beschreibungen darstellen. Höhere Schichten sind nur unvollständig beschrieben und liegen fast ausschließlich für Vermittlungseinheiten (Intermediate Systems) aber nicht für Endsysteme vor.

Die existierenden Teststrategien können hinsichtlich ihres Ziels in Interoperabilitäts- und Konformitätstests unterteilt werden. Interoperabilitätstests garantieren das Zusammenwirken unterschiedlicher Geräte, jedoch nicht unbedingt eine Übereinstimmung mit den zugrundeliegenden Standards. Konformitätstests beantworten die Frage nach Übereinstimmung mit dem Standard, lassen aber keine sichere Aussage über die Verbindung zweier Geräte zu.

2.2.2 Testgeräte

Die Anzahl von ATM-Testgeräten, die auf dem Markt erscheinen, hat in den letzten Jahren stark zugenommen. Es wächst sowohl die Zahl der Anbieter als auch der Funktionsumfang der Testgeräte [Str96].

Der Leistungsumfang umfaßt:

- Unterstützung unterschiedlicher physikalischer Schnittstellen (SONET/SDH, TAXI,...), deren Überwachung und Test (z.B. durch Erzeugung fehlerhafter Transportrahmen)
- Generierung und Untersuchung von ATM-Zellströmen, Erzeugung von Verkehrsmustern
- Simulation, Emulation und Dekodierung von Protokollen (z.B. verschiedene AAL-Typen, Signalisierung)
- Datenanalyse in Echtzeit oder Speicherung der Meßdaten und Offline-Auswertung
- Bedienung des Gerätes über eine GUI

ATM-Testgeräte lassen sich in bezug auf ihre Funktionalität in zwei Gruppen einteilen: Protokoll-Analysatoren und Geräte zum Messen von Leistungsdaten [Bee 96]. Obwohl es Überschneidungen im Leistungsumfang der existierenden Testgeräte gibt, werden beide Aufgaben von einem modernen Gerät nicht zufriedenstellend erfüllt. Hinzu kommt noch der Umstand, daß ein Protokoll-Analysator nicht die gleichzeitige Analyse von verschiedenen Schichten erlaubt. Eine isolierte Betrachtung einer Protokollschicht führt aber zu einer Transparenz von Einflüssen der unteren Schichten, so daß dadurch spezifische Fehlerursachen nicht erkannt werden können.

Die Erweiterbarkeit bzw. Anpassungsfähigkeit der ATM-Testgeräte an nutzerspezifische Erfordernisse ist im allgemeinen nicht möglich. Ist eine Programmierschnittstelle vorhanden, bietet diese nur unzureichende Bibliotheksfunktionen und ist außerdem stark geräteabhängig. Die Anpassung der Software durch den Nutzer an spezifische Testfälle ist daher umständlich und zeitaufwendig sowie nicht für andere Testgeräte verwendbar.

2.3 ATM-Architektur

2.3.1 Zellaufbau

Das Prinzip der Datenübertragung des asynchronen Transfermodus beruht auf der Vermittlung von Datenpaketen. Die Datenpakete besitzen eine feste Länge und werden als Zellen bezeichnet. Die ATM-Zelle (Abbildung 1) besteht aus 53 Byte, davon nehmen die eigentlichen Nutzinformationen (ATM-Payload) 48 Byte ein. Der Zellkopf (ATM-Header) wird von Adreßinformationen (GFC, VPI, VCI), Typ- und Prioritätsinformationen (PT, CLP) sowie Informationen zur Fehlerkorrektur (HEC) gebildet und nimmt insgesamt 5 Byte in Anspruch.

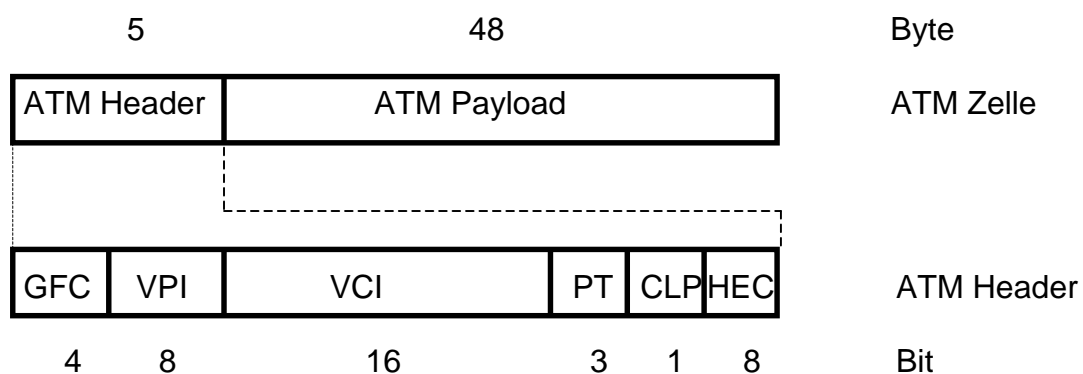


Abbildung 1: ATM-Zellaufbau

2.3.2 B-ISDN-Schichtenmodell

Die logische B-ISDN-Netzwerkarchitektur wurde in Anlehnung an das OSI-Schichtenmodell (ISO1) in vier voneinander unabhängige Schichten eingeteilt (Abbildung 2). Eine Besonderheit ist deren Unterteilung in drei verschiedene Ebenen. Die Benutzerebene (User Plane) überdeckt alle Schichten des Protokollmodells. Die Steuerebene (Control Plane) ist für den Auf- und Abbau sowie die Überwachung der Verbindung zuständig. Die Aufgaben der Verwaltungsebene (Management Plane) liegen in der Koordination der verschiedenen Ebenen, des Aufbaus des Signalisierungskanals und in der Erhaltung der Netzwerkleistungen.

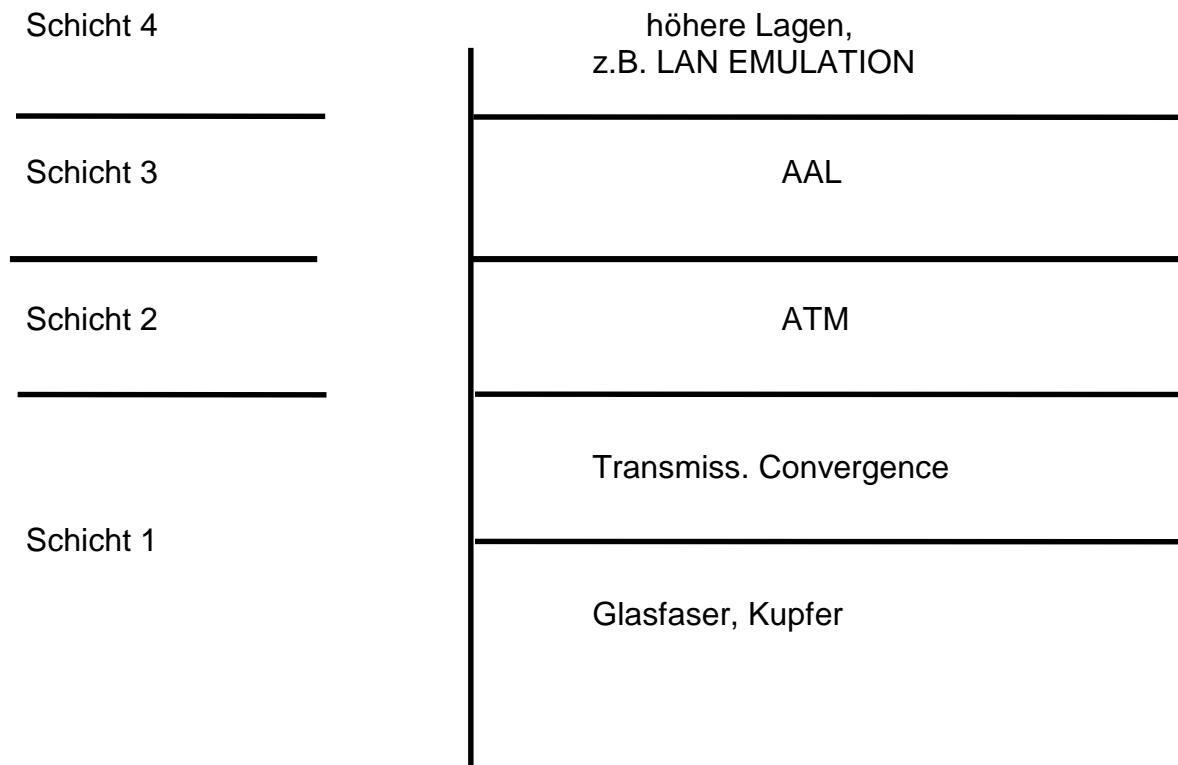


Abbildung 2: B-ISDN-Schichtenmodell

2.4 Probleme mit dem Einsatz von ATM

ATM-Hochgeschwindigkeitsnetze können im 'Wide Area Network'-Anwendungen alternativ als permanente virtuelle Verbindungen (Permanent Virtual Circuits, PVC) oder als geschaltete virtuelle Verbindungen (Switched Virtual Circuits, SVC) eingesetzt werden. Während PVC-Verbindungen heute mehr oder weniger problemlos eingesetzt werden können, bestehen bei SVC-Verbindungen noch erhebliche Schwierigkeiten, weil die vorhandenen verabschiedeten Standards von den Herstellern unterschiedlich interpretiert werden.

Die derzeitige LAN-Emulation Version 1.0 enthält noch eine Reihe von Interpretierungsschwächen. Dies führt dazu, daß Komponenten des gleichen Herstellers untereinander in der Regel relativ problemlos arbeiten. Beim Zusammenspiel von Komponenten unterschiedlicher Hersteller treten dagegen häufig Probleme auf, die sich in der Regel nur schwierig analysieren und eingrenzen lassen. Hierfür existieren spezielle ATM-Analysegeräte, die jedoch unverhältnismäßig teuer sind, was ihren Einsatz in einem mittelständischen Unternehmen bei der Entwicklung von ATM-Komponenten weitgehend ausschließt.

3 Funktionale Ziele des ATM-Testgenerator/Monitors

3.1 Anforderungen

Um Netzwerke oder ATM-Geräte testen zu können, wird ein ATM-Diagnosegerät benötigt, das zwei Funktionsblöcke enthält: Monitor- und Generator-Funktionsblock.

Diese Funktionsblöcke können unabhängig voneinander arbeiten. Dabei muß vorausgesetzt werden, daß das Diagnosegerät in der Lage ist, Übertragungsraten von mindestens 155MBit/s zu verarbeiten. Diese Bandbreite entspricht dem momentanen Stand der Technik.

Weitere Anforderungen an das Diagnosegerät sind:

- Daten-Simulationsrate von 50 MBit/s
- Bidirectionale Überwachung des Datenstroms
- Bidirektionale Simulation
- Gleichzeitige Echtzeitverarbeitung aller Monitoring-, Simulations-, Statistik- und Analyse-Prozesse
- Injektion und Entdeckung von physikalischen und Data Link Fehlern
- Echtzeit-Filter- und -Trigger- Fähigkeiten
- Betriebssystem-Unterstützung durch WindowsNT
- Agent für Anbindung an das Tivoli System Management
- CORBA-Implementierung ermöglicht Benutzung durch entfernte PC's unter Verwendung des Netscape Browsers

3.1.1 Monitor-Mode

Unter Monitoring verstehen wir das passive Capturing von ATM Zellen, die zwischen zwei Kommunikationseinheiten fließen. Dieser Vorgang ist transparent, die Daten auf der überwachten Leitung bleiben unverändert.

Capturing ermöglicht das Kopieren von Daten von der Leitung und ihre Abspeicherung in einem RAM Speicher unter vorgegebenen Trigger- und Filter-Bedingungen.

Im Monitor-Mode empfängt das Diagnosegerät die ATM-Zellen der zu testenden virtuellen Kanäle. Nach bestimmten vom Benutzer definierbaren Empfangsfehlern (z.B. falsche Paketlänge oder CRC-Fehler im Header) können die Zelldaten ausgewertet und gespeichert werden. Zur Auswertung der aufgetretenen Fehler dient eine Statistik über die Anzahl der empfangenen Datenpakete in Abhängigkeit

von den beim Empfang erkannten Fehlerzuständen. Auch über die Art und Häufigkeit der verifizierten Fehler kann eine Statistik geführt werden. Die Fehler-Diagnose kann sowohl auf Zellebene als auch auf der AAL-Schicht erfolgen. In der ersten Implementierung wird nur die AAL5-Schicht berücksichtigt.

Ein spezifisches Problem ist die Beobachtung des Leistungsverhaltens (z.B. verworfene Zellen) über einen gegebenen Zeitabschnitt.

3.1.2 Generator-Mode

Im Generator-Mode werden ATM-Zellen vom Diagnosegerät generiert. Hierbei können Zellparameter wie z.B. VCI, VPI oder Zelldaten vom Benutzer gewählt werden, um so bestimmte Zellmuster und Lastbedingungen zu erzeugen.

Zum Testen von Netzwerken oder ATM-Geräten werden vom Generator Zellen mit sich ändernden Dateninhalten und Parametern erzeugt. Zusätzlich ist durch eine definierte Fehler-Korruption der ATM-Zellen die Zuordnung von Fehler-Generierung und -Erkennung möglich.

Beim Testen von ATM-Geräten ist es in der Regel möglich, die vom Generator erzeugten Zellen über den Ausgang des zu testenden Gerätes dem Monitor zuzuführen.

Für den Aufbau des ATM-Testgenerator/Monitors bieten sich drei Alternativen an:

- Reine Software-Lösung
- FPGA-Lösung in Verbindung mit TTL-Bausteinen
- ASIC-Lösung

Eine reine Software-Lösung scheidet aufgrund der erforderlichen Übertragungsrate von 155 MBit/s aus. Bei der zweiten Alternative werden die TTL-Bausteine für den Aufbau der Hochgeschwindigkeits-Funktionsblöcke eingesetzt, während im FPGA Funktionen implementiert werden, die größere Ausführungszeiten erlauben. Diese Variante wurde aber auch verworfen, weil sie nur bei größeren Stückzahlen einen kostengünstigen Lösungsansatz darstellt.

Die ASIC-Lösung wurde ausgewählt, weil der Markt ASICs anbietet, die auf der AAL-Ebene arbeiten, so daß die Funktionen der AAL- und ATM-Schicht nicht implementiert werden müssen. Die Auswahl fiel auf einen ASIC der Firma Fujitsu. Dieser ASIC (ALC, Adaption Layer Controller) arbeitet auf der AAL-Ebene und bietet vielfältige Möglichkeiten zur Wahl von Zellparametern und Lastbedingungen für das Hochgeschwindigkeitsnetz.

Zur anwenderspezifischen Hardware gehören außer dem ALC noch ein Shared Memory, bestehend aus einem SAR (Segmentation And Reassembly)-Speicher und Arbitrer, und einem Cellpack-Modul der Firma HILAN Karlsruhe.

3.1.3 Funktionsumfang des ATM-Testgenerator/Monitors

Zum Funktionsumfang des ATM Testgenerator/Monitors gehören folgende geplante Implementierungen:

ATM-Simulation

Unter Simulation verstehen wir die voll Duplex-Kommunikation zwischen dem ATM-Testgenerator/Monitor und einem zu prüfenden Gerät. Die ATM-Simulation ermöglicht die Übertragung eines simulierten ATM-Nachrichtenverkehrs über definierte VPCI's.

Protokoll-Analyse

Die Protokoll-Analyse ist ein Anwendungsprogramm, das spezifische Hilfsmittel für die Auswertung eines vorher empfangenen ATM-Zellenstroms bereitstellt.

Statistik-Analyse

Die Statistik Analyse beinhaltet ein Anwendungsprogramm, das Hilfsmittel für die Auswertung von vorher aufgezeichneten ATM-Statistikdaten zur Verfügung stellt.

Configuration-Management

Configuration-Management deckt sowohl das Netzwerk mit seinen Switches als auch den Datenverkehr ab, der durch das Netzwerk fließt. Beim letzteren müssen spezifisch PVC's und SVC's aufgesetzt und geschlossen werden.

Die Hersteller von Switches rüsten diese mit SNMP und (heutzutage meistens miteinander inkompatiblen) MIBs (Management Information Bases) aus. Eine SNMP Komponente des ATM-Testgenerator/Monitors ermöglicht hierauf einen Zugriff. Geplant ist die Unterstützung der RFC 1695 MIB sowie der Sonet MIB.

Accounting-Management

Für die finanzielle Belastung von Anwendern sind Statistiken über die Nutzung von Ports und VC's erforderlich. Spezifisch gilt dies für Parameter wie die Lebensdauer eines VC, geographische Distanz oder spezielle Raten, die an die Tages- oder Wochenzeit gebunden sind.

Benutzerschnittstelle

Über eine CORBA/Java-Schnittstelle ist es möglich, den ATM-Testgenerator/Monitor von entfernten Lokationen aus zu benutzen, z.B. von einem Administrator während des Wochenendes mit Hilfe seines häuslichen PCs über das Internet.

3.2 Einsatzmöglichkeiten

Der ATM-Testgenerator/Monitor adressiert die Bedürfnisse der folgenden Benutzergruppen:

1. Netzwerk-Administratoren sind in der Lage, die statistischen und analytischen Daten-Einrichtungen zur Überwachung des Netzes zu verwenden.
2. Außendienst-Techniker können den ATM-Testgenerator/Monitor zur Fehlereingrenzung einsetzen.
3. Entwickler können die Kompatibilität mit existierenden ATM-Produkten austesten.

4 ATM-Testgenerator/Monitor-Hardware

4.1 Implementierungs-Problematik

Im Monitor-Mode erfolgt ein passives Capturing der ATM-Zellen, die zwischen zwei Kommunikationseinheiten fließen. Dieser Vorgang ist transparent und umfaßt die zeitlich parallele Überwachung von 4 K Virtual Pathes (VP0...VP4095) mit je 64 K Virtual Channels (VC0...VC65535). Beim Eintreffen fehlerhafter ATM-Zellen (Monitor) ist einerseits der betreffende Übertragungskanal zu identifizieren und andererseits die Fehlerart innerhalb der Zelle (z.B. CRC- oder Packet Length Error) zu bestimmen. Beide Fehlerklassen entsprechen jeweils unterschiedlichen Trigger-Funktionen, die vom ALC über Exceptions im PowerPC-Controller (PPC403GCX) der Entwicklungsumgebung generiert werden.

Unter Trigger-Funktion verstehen wir die Exception Handling Routine, die beim Eintreffen eines Events (Exception) vom ALC (z.B. Register 18) im PPC403GCX angestoßen wird. Die Exceptions dieses Controllers gliedern sich in drei Basis-Klassen: Asynchrone nichtpräzise, synchrone präzise und asynchrone präzise Exceptions. Letztere schließen u.a. externe Interrupts ein.

In [WKKR98] wird ein abstraktes Modell entwickelt, in dem diese Forderungen in ein reales eingebettetes System abgebildet werden. Dieses besteht aus dem Micro-Controller IBM-PPC403GCX (80 MHz), dem ATM-Board mit integrierten Adaption Layer Controller (ALC) und einem Shared Memory (SAR) sowie einem komplexen FPGA (Abbildung 3).

Um die Erkennung von Fehlern in einem ATM-Zellenstrom mit Hilfe des ATM-Testgenerator/Monitors zu realisieren, ist der Einsatz von Zusatz-Hardware notwendig. Eine Software-Lösung scheidet deshalb aus, weil die Ausführungszeit der zugehörigen Exception-Software-Routine, die nach einem Event (z.B. CRC-Error) abgearbeitet wird, zu viel Zeit in Anspruch nimmt (Verarbeitung einer ATM-Zelle bei einer Bandbreite von 155 MBit/s: 2,74µs). Geht man von ca. 250 Maschinenbefehlen in einer Exception-Behandlungs-Routine aus, so würde allein deren Ausführungszeit 3µs durch den PPC403GCX (80 MHz) in Anspruch nehmen. Dazu kommen noch Load/Store-Befehle (Statusregister), die mehr als einen Taktzyklus benötigen. Daraus ergibt sich die Schlußfolgerung, daß Fehler in einem ATM-Zellenstrom durch eine spezifische Exception-Behandlungsroutine (Software) nicht erfaßt werden können.

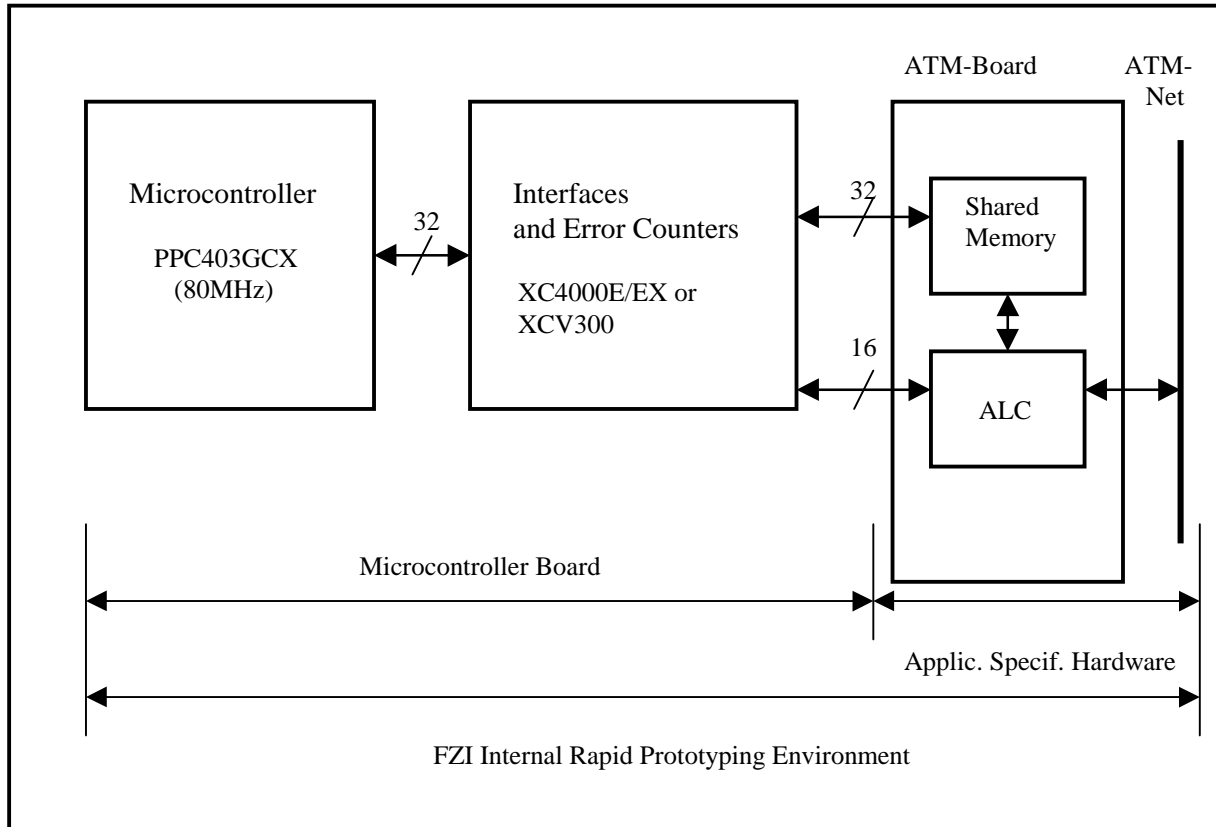


Abbildung 3: Abstraktes Modell für ein eingebettetes System „ATM Testmonitor“

Aus diesen Überlegungen heraus ergibt sich die Implementierung der Fehlerbehandlung in Hardware. Diese Fehlerbehandlung stellt eine bestimmte Menge von dedizierten Error Countern dar, wobei theoretisch jedem VC ein Error Counter (32 Bit) zugeordnet werden muß. Infolge der Anzahl der VCs kann nicht jeder auf ATM-Zellen-Fehler überwacht werden. Die Menge der implementierten Error Counter ist abhängig von den verwendeten Implementierungs-Methoden, die in [WKKR98] untersucht wurden. Daraus geht hervor, daß aus der Sicht des Nutzers die maximale Zahl der in einem FPGA implementierbaren Error Counter die auszuwählende Methode bestimmt. Es stehen zwei unterschiedliche Rekonfigurations-Methoden zur Auswahl: Compile Time Reconfiguration (CTR) und Global Run Time Reconfiguration (RTR).

Als Aufgabe ergibt sich die Lösung folgender Probleme:

1. Implementierung der CTR- und RTR-Methode auf der Baustein-Familie XCV. Diese Bausteine verfügen über mindestens 100.000 Gatteräquivalente. Dadurch erhöht sich die Zahl der auf dem FPGA implementierbaren Error Counter mindestens um den Faktor 10 im Verhältnis zur XC4000E/EX (XC4028EX)-Familie. Durch die segmentierten Routing-Ressourcen der Virtex-Serie ergeben sich möglicherweise günstigere Verdrahtungsmöglichkeiten, die sich bei der Implementierung der Error Counter on chip als limitierender Faktor erweisen.

2. Die verschiedenen Designs, die der Fehler-Überwachung auf den unterschiedlichen VCs entsprechen, müssen nach Auswahl durch den PPC403GCX während der Laufzeit einer Applikation ausgewählt werden. Dieses Verfahren setzt

voraus, daß die Designs in einem Filesystem untergebracht sind. Der Controller verwaltet dieses Filesystem über ein Echtzeitbetriebssystem (z.B. VxWorks, OS/Open) und lädt nach Anforderung ein bestimmtes in den FPGA. Diese Methode gestattet bei zwei verschiedenen Designs, die nicht disjunkt sind, eine partielle Rekonfiguration, wenn von einem Design auf das andere umkonfiguriert werden muss, d.h. es wird in diesem Fall nicht das gesamte Design neu konfiguriert sondern nur der Teil, der sich vom vorhergehenden unterscheidet.

Weitere Gründe für den Einsatz von Hardware anstatt Software ergeben sich aus Kostenabschätzungen. Die Entwicklung geeigneter Software im Fall der ATM-Fehler-Behandlung kann mit wesentlich höheren Kosten angesetzt werden als die Implementierung der Error Counter in FPGAs der XC4000E/EX-Familie, deren Stückpreis mit der Einführung der XCVirtex-FPGAs auf unter 10,- DM sinken wird. Zusätzlich zu diesem Kostenfaktor wird sich die Funktionalität dieser Bausteine noch erhöhen, so daß sich die Effizienz der eingesetzten Hardware multipliziert.

4.2 ATM-Controller

4.2.1 Übersicht

Für den Adaption Layer Controller (ALC) wird ein spezieller ASIC (MB86687A der Firma Fujitsu, 208 Pin QFP) eingesetzt. Dieses Chip gestattet wahlweise einen AAL3/4 und einen AAL5 Betrieb bei einer Datenrate von 155 MBit/s. Im vorliegenden Fall wurde AAL5 implementiert.

Über ein Interface hat der ALC Zugriff auf einen DPRAM (SAR Memory) von 512 KByte mit einer Zugriffszeit von 15 ns. Je ein Eingang wird vom ALC und vom Host PC benutzt. Der ATM-Controller enthält weiterhin einen Arbiter, der die Zugriffe auf das DPRAM regelt.

4.2.2 Adaption Layer Controller (ALC)

Der ALC führt selbstständig und parallel Segmentierung und Reassemblierung von Datenpaketen auf bis zu 1024 VCs aus. Die Datenpakete werden vom und zum Shared Memory übertragen. Dafür ist ein schneller DMA-Controller mit einer 32 Bit Datenbus-Breite verantwortlich. Die Daten werden im SAR-Speicher abgelegt. Letzterer kann entweder als DPRAM oder als Teil des System-Speichers implementiert sein.

Zu sendende Daten werden vom Host an das DP-Memory (SAR) übergeben. Der ALC führt daraufhin die notwendigen Funktionen der AAL- und ATM-Schicht zum Senden von Daten aus, d.h. die Anpassung der Datenstrukturen an die zellenorientierte Übertragungsweise, und übergibt die generierten Zellen Byte-weise an ein Cellpack-Modul (COSY), das die physikalische Schicht darstellt.

Die Abbildung 4 zeigt die möglichen System-Konfigurationen bei einer Implementierung des ALCs (mit und ohne Pfade in Graustufe) .

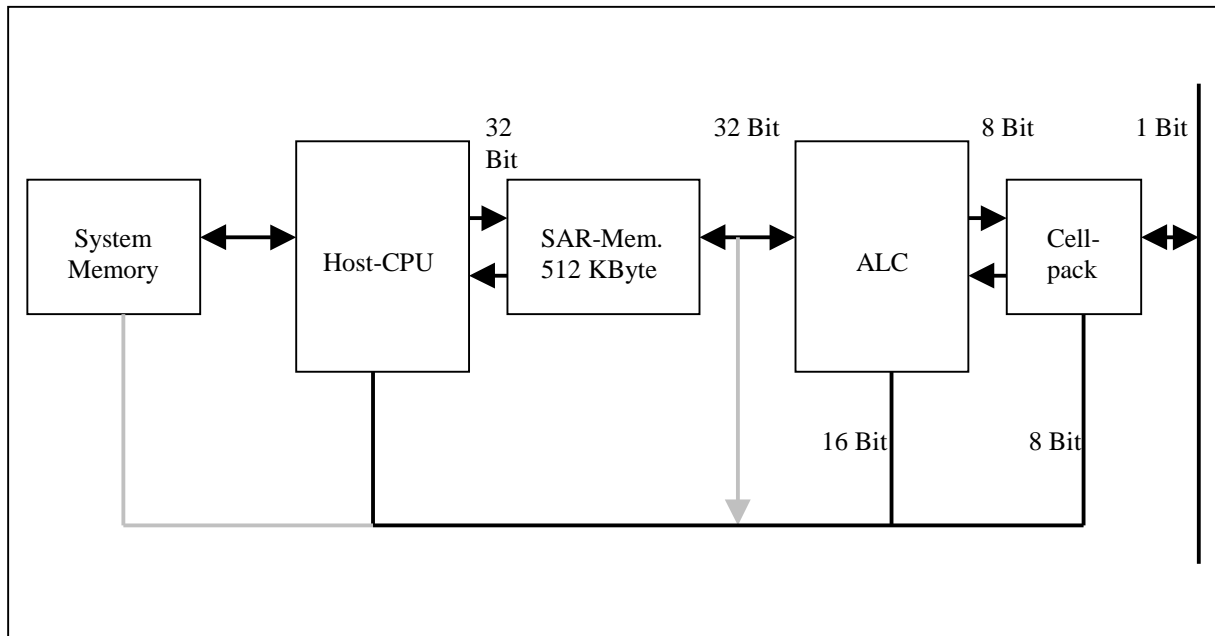


Abbildung 4: Mögliche Systemkonfigurationen

Im Cellpack-Modul findet die Anpassung der Zellenübertragung an das physikalische Netzwerk, das elektrischer (Koaxial- oder Twisted-Pair-Kabel) oder optischer (Mono- oder Multi-Mode Lichtwellenleiter) Natur sein kann, statt. Beim Empfang werden die Datenfelder der Zellen zu den ursprünglichen Datenstrukturen zusammengesetzt und an den Host übergeben.

Zum Betrieb des ALCs werden aber weitere Hardwarekomponenten sowie Möglichkeiten des Host für einen Zugriff auf die internen Register benötigt. Aus der Abbildung 4 geht hervor, daß die Implementierung des ALC die Benutzung des Host-Systemspeichers als Arbeitsspeicher zuläßt. Weiterhin kann das SAR Memory für den Datentransfer oder als Dual Port (DP)-RAM verwendet werden. Eine Kombination aus beiden ist zusätzlich möglich.

Die Abbildung 4 zeigt die Zugriffsmöglichkeit des ALC auf den Systempeicher des Host aufgrund eines Pfades zum Host-Systembus. Zur Datenübertragung über den Systembus stellt der ALC Intel- und Motorola-DMA-Modi zur Verfügung.

Für eine Anwendung als ATM-Diagnosegerät ist die Systemkonfiguration der Abbildung 4 (ohne Pfade in Graustufe) günstiger. Diese benutzt ein Shared Memory als Arbeitsspeicher und zum Datenaustausch, wodurch schnellere Speicherzugriffe erfolgen und keine Blockierungen des Host für die Dauer des Datentransfers entstehen. Zusätzlich bestehen Zugriffsmöglichkeiten des Host auf die internen Register des ALCs und des COSY-Moduls über einen 16 bzw. 8 Bit Bus.

In der Abbildung 5 sind die vom ALC verwendeten Funktionsblöcke zum Senden und Empfangen von ATM-Zellen dargestellt. Zum Senden dienen die Funktionsblöcke High Speed DMA-Controller, Traffic Management-Controller, Transmit Buffer, Segmentation & Convergence Sub-Layer-Controller und ATM Cell Transmitter. Nachdem der Host die zu sendenden Daten im SAR-Memory abgelegt hat, werden

diese über den High Speed DMA-Controller vom ALC bearbeitet. Der ALC benutzt diesen DMA-Controller, der mit einem Minimum an Host-Anweisungen arbeitet, für das Segmentieren und Zusammenfügen der Benutzer-Datenpakete.

Die Kommunikation mit dem Host-Prozessor findet über Datenstrukturen im SAR-Speicher statt.

Data Transmit

Für den Sendebetrieb ist es notwendig, daß der Host im SAR-Memory die folgenden Datenstrukturen aufbaut:

- Transmit Pending Queue
- Transmit Descriptor Table

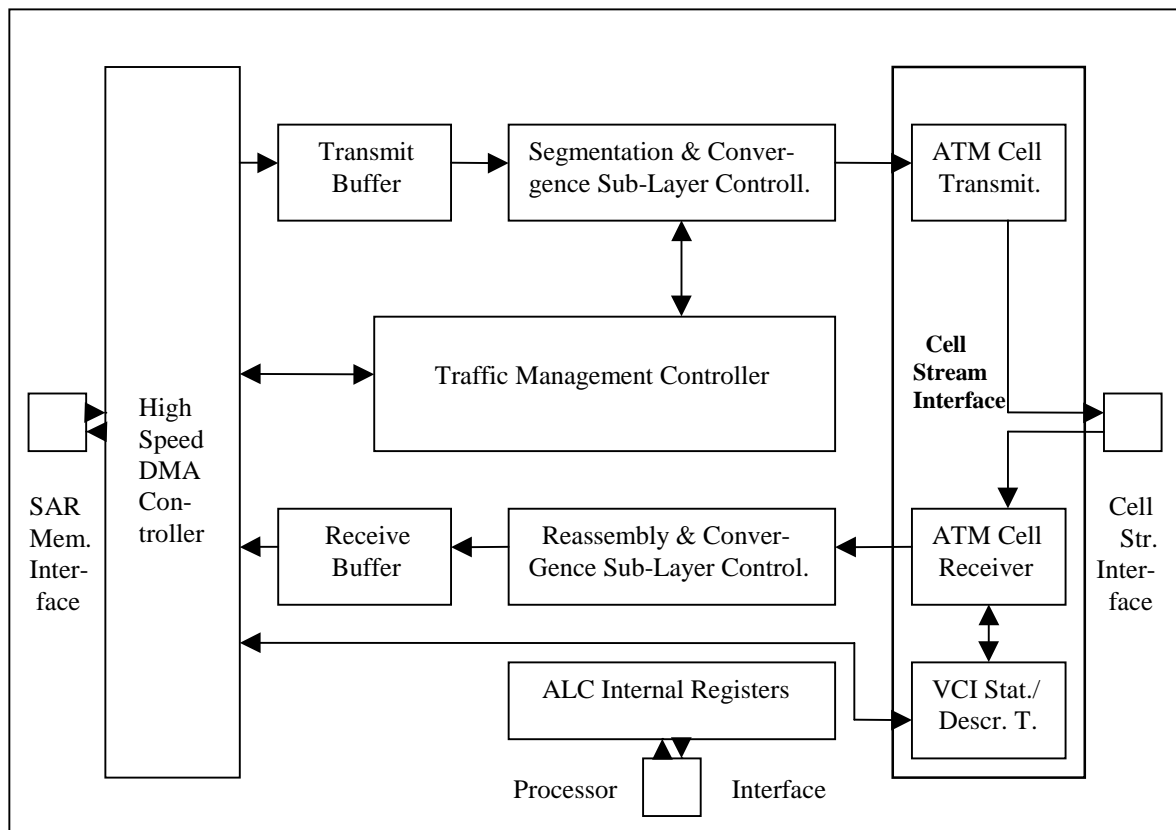


Abbildung 5: Funktionsblöcke des ALC

- Circuit Reference Table
- Transmit Buffer Release Queue

Transmit Pending Queue

Die Transmit Pending Queue (TPQ) wird vom Host benutzt, um den ALC anzuweisen, ein Datenpaket in eine Transmit Queue aufzunehmen. Die Felder eines Eintrags in der Queue enthalten einen 12 Bit Zeiger auf den entsprechenden Eintrag in der Transmit Descriptor Table, sowie einen TPQ-Code.

Transmit Descriptor Table

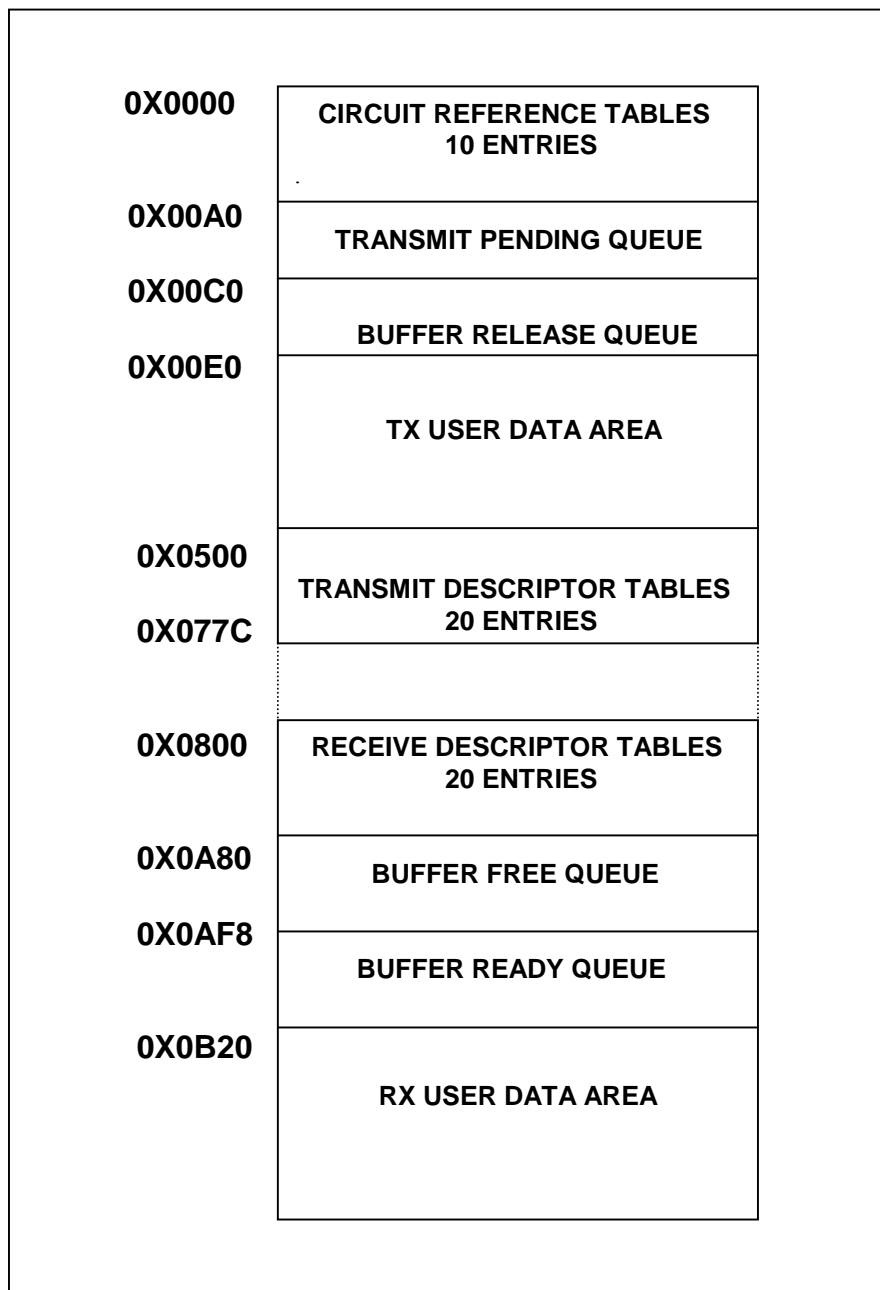


Abbildung 6: SAR Memory Map

Für die Übertragung eines Paketes (Zelle) ist die Generierung einer Transmit Descriptor (TD) Table notwendig. Diese kann aus bis zu 4096 Transmit Descriptor-

Einträgen bestehen. Jeder TD enthält einen Pointer auf den zu sendenden Datenbereich.

Die Transmit Descriptor (TD) Table ist als Speicherbereich im SAR-Memory implementiert, deren TDs im Streaming Mode vom Nutzer dort plaziert werden (20 TDs in der aktuellen Konfiguration). Die TDs der TPQ zeigen auf jeweils eine ATM-Zelle (12 Worte zu je 32 Bit), die sich gleichfalls im SAR-Memory befinden (User Data Area). Die physikalische Adresse der Einträge in der Transmit Descriptor Table umfaßt 24 Bit. Die Memory Map des SAR zeigt die Abbildung 6.

Circuit Reference Table

Die Circuit Reference Table stellt auch einen Speicherbereich im SAR-Memory mit physikalischen Adressen der Einträge von 24 Bit dar. Die Circuit Reference enthält den ATM-Zellkopf, sowie die Leaky Bucket Parameter des entsprechenden VCs. Für jeden aktiven VC existiert eine Circuit Reference.

Transmit Buffer Release Queue

Nach erfolgtem Senden übergibt der ALC den 12 Bit Zeiger des gesendeten Transmit Descriptors sowie einen TPQ-Code an die Transmit Buffer Release Queue.

Vor der Übertragung von ATM-Zellen durch den ALC ist die Initialisierung der Register erforderlich. Dabei werden Betriebsparameter und Startadressen der benötigten Tabellen festgelegt. Folgender Ablauf muß eingehalten werden:

- Eintrag in Circuit Reference Table für das zu sendende Paket
- Transmit Descriptor (TD) erstellen
- Eintrag in Transmit Pending Queue, der auf den zu verwendenden TD verweist
- Inkrementieren des Transmit Pending Queue Write Pointers

Das Senden eines Paketes wird vom ALC wie folgt signalisiert:

- Generieren eines Interrupts, falls das IOC Bit im TD gesetzt war
- Eintragen des TPQ-Codes und der Nummer des gesendeten TDs in der Transmit Buffer Release Queue durch den ALC
- Inkrementieren des Transmit Buffer Release Queue Read Pointers

Data Receive

Der Empfang der Daten erfolgt analog zur Sende-Prozedur. Er benötigt die Funktionsblöcke Cell Stream Interface, Reassembly & Convergence Sub-Layer-Controller, Receive Buffer und High Speed DMA-Controller.

4.2.3 Dual Port RAM

Das SAR-Memory dient zur Kommunikation der Teilsysteme PC, FPGA (PowerPC)-Bord, ALC und COSY. Infolge der hohen Übertragungsrate von 155MBit/s des ALC können nur SRAMs mit sehr geringen Zugriffszeiten verwendet werden. Um ausreichenden Speicherplatz für mindestens 100 aktive Kanäle bereitzustellen, implementiert das SAR-Memory eine Speicherkapazität von 512 KByte. Der Aufbau eines Dual Port RAMs dieser Kapazität wurde mit Hilfe von SRAM-Speicherbausteinen und einem Arbitrer realisiert. Letzterer koordiniert die asynchronen Zugriffe von beiden Seiten (Host-CPU, ALC).

Die Implementierung des Arbiters erfordert eine Steuerlogik, einen Multiplexer und Transceiver. Diese schalten PC- oder ALC-Daten und Adreßbusse auf das SAR-Memory durch. Die Steuer-/Arbitrerlogik regelt die Zugriffe und generiert die entsprechenden Signale für die Transceiver und das Memory. Zur Umschaltung der Byte Enable-Signale dient ein zusätzlicher Multiplexer.

Der Arbitrer (66 MHz) arbeitet mit doppeltem ALC-Takt (33 MHz), um möglichst schnell auf Speicheranforderungen reagieren zu können.

4.2.4 Arbitrer

Das Konzept der Zugriffsregelung für das SAR-Memory setzt voraus, daß vom ALC weitaus mehr Zugriffe vorgenommen werden als vom PC. Im Interesse einer hohen Datenrate wird dem ALC Priorität bezüglich der Speicherzugriffe eingeräumt (Betrieb

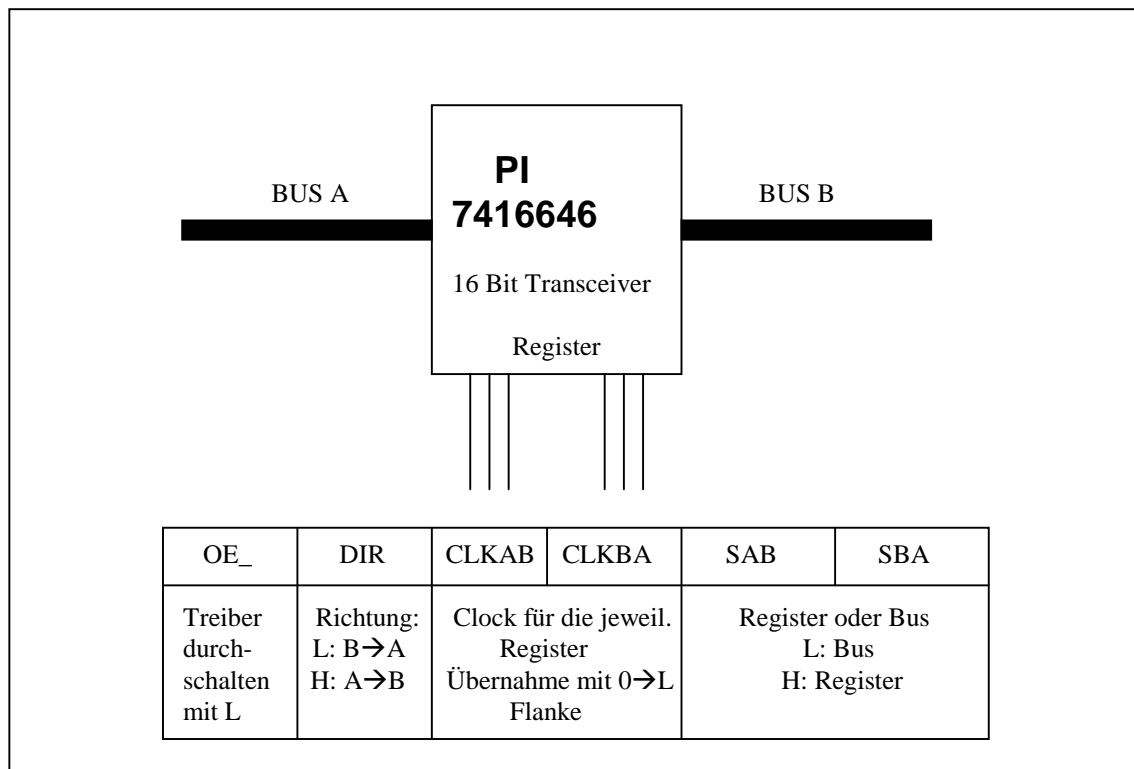


Abbildung 7: Transceiver mit Funktion der Steuersignale

ohne Wait-Zyklen). Der PC-Zugriff erfolgt in Abhängigkeit eines Acknowledge-Signals, das die Ausführung eines Speicherzugriffs quittiert.

Die Entwicklung der Arbitrator-Logik ist abhängig vom Timingdiagramm der Read- und Write-Zyklen. Für die Steuerung der Zugriffskontrolle auf das SAR-Memory werden vom ALC vier Start of Cycle Ausgangs-Signale (**CS_) zur Verfügung gestellt, die einen Takt vor einem Speicherzugriff generiert werden und danach sowohl einen lesenden als auch einen schreibenden Zugriff auf das SAR-Memory zulassen (TOCS_).

Die Zugriffsregelung als State Machines muß folgende Funktionalität erfüllen:

- Der ALC erhält Zugriff auf den Speicher, wenn eines der **CS_- Signale aktiv ist.
- Der PC erhält Zugriff auf den Speicher, wenn das Anforderungs-Signal des PC aktiv ist und keine ALC-Anforderung besteht. Die Überprüfung der Anforderungen und der Start der Bearbeitung soll so erfolgen, daß der PC-Zugriff auch bei darauffolgender ALC-Anforderung komplett bearbeitet wird.
- Es muß sichergestellt werden, daß eine PC-Anforderung nicht mehrfach bearbeitet wird.

4.3 Cellpack-Modul (COSY)

Das Cellpack-Modul ist ein Standardbaustein der Fa. HILAN GmbH, Karlsruhe (Abbildung 8). Es packt/entpackt den ATM-Zellenstrom in die SDH/Sonet (Synchronous Digital Hierarchy)-Rahmen des STM1/OC-3-ATM-155 MBit/s-Standards. Hierdurch wird die effektive Datenrate von 155 auf 132 MBit/s reduziert. Der Ausgang verwendet Multimode Glasfaser. Die Standard-Schnittstelle des ALC gestattet wahlweise den Einsatz von Monomode-Glasfaser oder einer elektrischen Verbindung (Unshielded Twisted Pair, Mehrwertkodierung). Dies kann durch den Einsatz eines anderen Cellpack-Moduls erreicht werden.

4.4 PowerPC embedded Controller

Die Aufgabe des PowerPC403GCX besteht in der Verwaltung der verschiedenen Designs, die der Fehler-Überwachung auf den unterschiedlichen VCs entsprechen. Diese Aufgabe ist nur mit Hilfe eines geeigneten Echtzeitbetriebssystems (s. Abschnitt 4.1) möglich.

Die PPC Design-Umgebung besteht u.a. aus einem ISA-AT-Bus-Interface [HeWe96]. Dieses gestattet sowohl den Zugriff zu einem 2 KByte DPRAM vom PC als auch zu dem ISA-Bus-Controller (MACH210).

Die Bus Master-Steuerung des Controllers erlaubt eine Datenrate von 2 - 3 MByte/s.

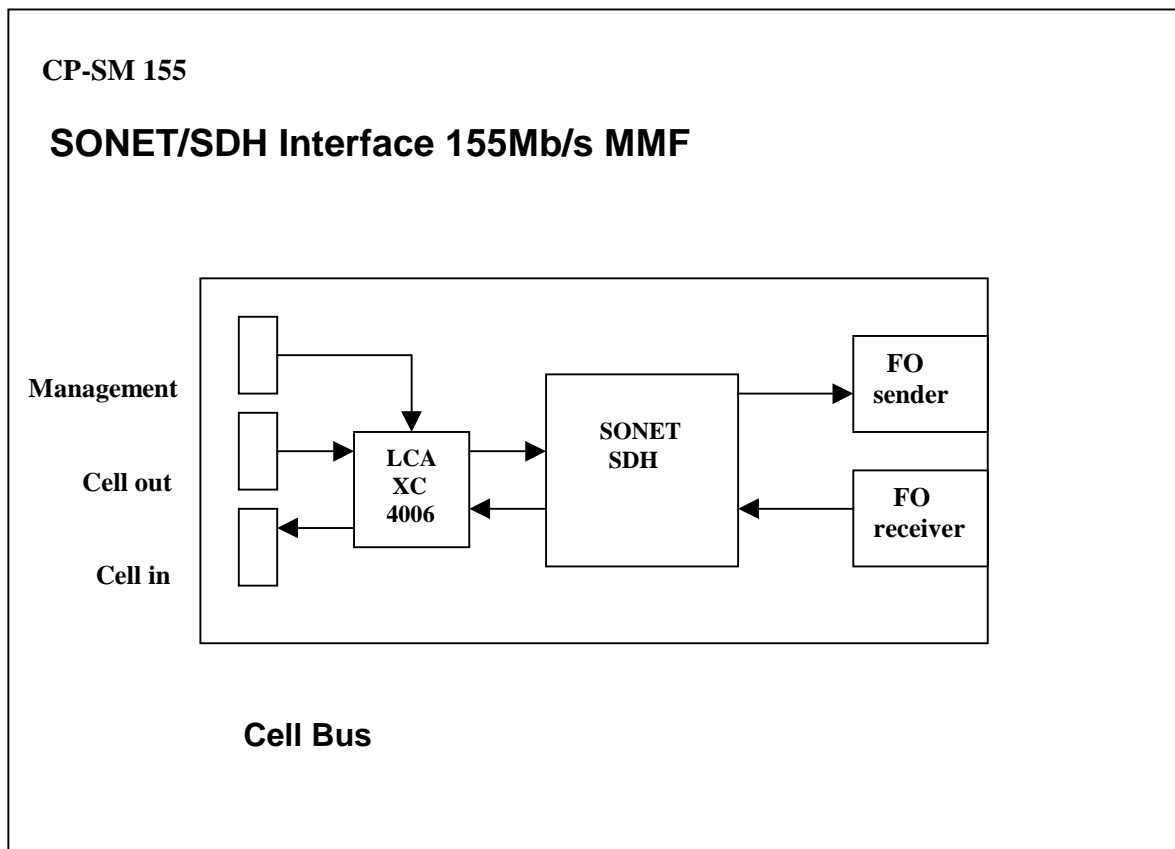


Abbildung 8: Cellpack Modul COSY

4.5 FPGA-Schnittstelle

Die FPGA Schnittstelle dient als Alternative zum Anschluß des Adaption Layer Controllers an den Host PC. In diesem Fall müssen die Daten vom/zum ALC-Controller direkt in den Hauptspeicher des Host PC's transportiert werden. Dies ist auf Grund der Pfadlänge des Host PCs nur mit reduzierter Datenrate möglich.

5 ATM-Testgenerator/Monitor-Software

5.1 Ergebnisse

Folgende Ergebnisse können momentan angegeben werden: Monitor-Umgebung mit eingeschränktem Funktionsumfang, lauffähiger Real Time Monitor für PPC Embedded Controller und funktionsfähige aber noch nicht integrierte Software.

5.1.1 Monitor-Umgebung (eingeschränkter Funktionsumfang)

Die Monitor-Umgebung setzt sich momentan aus dem FPGA-, ATM- und COSY-(Cellpack)-Board zusammen. Die Controller-Funktion wird vom PC übernommen (Datentransport PC \leftrightarrow ALC, PC \leftarrow SAR, SAR-Viewer, ALC-Initialisierung).

Eine ATM-Zelle wird durch den Sender über den ALC (Transmit-Modus) aus dem SAR gelesen und im COSY-Modul durch einen speziellen ASIC in einen Rahmen gepackt. Nach der Übertragung erfolgt dieser Vorgang im Empfänger in der umgekehrten Richtung über COSY-Modul, ALC (Receive Modus) und SAR. Der PC dient zur Implementierung der GUI.

5.1.2 Real Time Monitor für PPC embedded Controller

Das Ziel eines geplanten ATM-Testgenerator/Monitors ist die Integration der momentan aus drei Teilen bestehenden Hardware der Generator/Monitor-Umgebung (FPGA/PPC Controller-, ATM-, COSY CP-Board).

Die Aufgaben des PPC embedded Controllers (Kapitel 4.1) stellen harte Bedingungen an ein zu verwendendes Echtzeitbetriebssystem. Für den Test eines Real Time OS wurde der RTEMS (Real Time Executive for Military Systems) Kernel im PPC403GCX implementiert. Das RTEMS stellt keinen eigenständigen Kernel wie bei normalen Betriebssystemen dar sondern liegt als Library vor. Aus diesem Grund mußte ein Loader geschrieben werden, um den Code auf den PPC zu laden. Die Quelle für den Maschinencode bildet ein GNU-Crosscompiler, der außer auf einer Sun-Architektur auch auf anderen Hosts lauffähig ist. Damit stehen alle GNU-Werkzeuge (Assembler, Compiler, Linker usw.) zur Verfügung. Die erzeugten Object Files bzw. die gelinkten Binaries liegen im ELF-Format vor.

Neben dem Loader wurden noch weitere Programme erzeugt, d.h. ein Parser, der die Struktur eines ELF-Files anzeigt, und ein Viewer für den DPRAM (gemeinsamer RAM für den PC und den PPC Controller).

Beim Schreiben der Applikation werden die RTEMS-Funktionen verwendet. Letztere bieten u.a. Speichermanagement und Multithreading.

5.1.3 Funktionsfähige noch nicht integrierte Software

Analyse eines ATM-Zellenstroms

Unabhängig von der speziellen ATM-Testgenerator/Monitor-Hardware wurde ein ausschließlich in Software implementierter LAN-Emulation-Protokolldeko-der entwickelt. Die benötigte Hardware besteht aus einer regulären, auch anderweitig einsetzbaren Workstation, 2 ATM Adapter-Karten, sowie 2 Glasfaser-optischen Beam-Splitttern. Für diese Hardware ist in Leipzig ein LANEDUMP genanntes Software-Paket entwickelt worden, das eine Protokollanalyse ermöglicht [Moc97].

Hierzu werden die durch die SETUP- bzw. CONNECT-Kommandos generierten Signalisierungsnachrichten aus dem gesamten Zellenstrom herausgefiltert und bezüglich ihrer VCI/VPI-Werte analysiert. Diese Methode ermöglicht Aussagen über die Art der Verbindungen.

Die verfügbare Testumgebung bestand aus 2 ATM-Switches, einer ATM-Ethernet Brücke, sowie mehreren Workstations und PCs mit ATM-Adapterkarten unterschiedlicher Hersteller. Mit Hilfe von LANEDUMP war es möglich, dieses heterogene Umfeld lauffähig zu machen und die vorhandenen Probleme zu analysieren.

Spezifisch wurden Inkompatibilitäten des Flush-Protokolls untersucht. Weiterhin wurden einige Defekte gefunden, die zu einem unangemessen Leistungseinbruch führten.

Geplante Erweiterungsmöglichkeiten von LANEDUMP betreffen (neben einer SNMP-Schnittstelle) spezifisch die Implementierung des LAN-Emulation-Zustandsautomaten, welcher automatisierte Aussagen über die semantische Korrektheit des Protokollnachrichtenverkehrs ermöglichen würde.

Untersuchung des Signalisierungsprotokolls

Als bisher reine Software-Implementierung wurde ein Werkzeug "SigDump" erstellt, das beim Auftreten von Fehlern im ATM-Signalisierungsprotokoll diese benennen kann und damit ihre Beseitigung ermöglicht [Tit97].

Sigdump gestattet im Fehlerfall eine genaue Auflistung und Auswertung der Nachrichten zwischen zwei ATM-Teilnehmern. Es empfängt Daten einer Verbindung, entschlüsselt diese und stellt sie in einer lesbaren Form dar. Auf Differenzen zwischen den empfangenen Daten und dem im verabschiedeten ATM-Standard festgelegten Format wird hingewiesen.

Die mit diesem Testwerkzeug bisher erzielten Ergebnisse unterteilen sich in zwei Gruppen:

- Beschreibung wichtig zu testender Funktionalität für zukünftige Versionen von Software und Standards, Entwicklung von Testszenarien.
- Auswertung des Verzögerungsverhaltens der untersuchten Implementierungen

Erstellung eines SNMP-Agenten

Es wurde ein ATM SNMP-Signalisierungsagent "SigAgent" erstellt, der ebenfalls zunächst als reine Software-Lösung implementiert wurde [Str96]. Er ermöglicht, die Vorgänge auf einem Signalisierungskanal zu analysieren und darauf aufbauend eine Fehlersuche und Fehlerdiagnose durchzuführen. Die hierdurch aus der Signalisierung gewonnenen Informationen ermöglichen Aussagen darüber, welche Verbindungen in einem spezifischen Augenblick bestehen, oder welches Protokoll für den Transport von Nutzdaten auf einer virtuellen Verbindung eingesetzt wird.

Die Struktur von SigAgent kann in 3 Teile gegliedert werden:

- Monitoring des Signalisierungskanals
- Informationsverarbeitung
- Kommunikationsschnittstelle

Durch eine Parametereinstellung während der Compilerungsphase ist die wahlweise Verwendung von ATM Adapter-Karten unterschiedlicher Hersteller, spezifisch von FORE und SUN, möglich.

Die Informationsverarbeitung erstellt eine Liste mit den auf dem Kommunikationskanal bestehenden Verbindungen. Es wurde ein Zustandsautomat implementiert, der an Hand der Typen der Signalisierungsnachrichten und der Call-Referenz den Auf- und Abbau von Verbindungen überwacht. Hieraus wird der Zustand der Verbindung berechnet.

Das verwendete Kommunikationsprotokoll ist SNMP. Die Basis hierfür ist eine Public Domain Implementierung der Carnegie Mellon Universität in Pittsburgh, Pa. Die Kommunikationsschnittstelle ermöglicht die Abfrage der ermittelten Werte, sowie die Steuerung und Meldung von außergewöhnlichen Ereignissen des SigAgent. Die Abfrage der ermittelten Werte erfolgt über einen SNMP ASAP (Application Service Access Point). Dieser wird dem Programm durch einen Kommandozeilen-Parameter mitgeteilt.

Steuerfunktionen beziehen sich auf das Löschen von Verbindungen aus der Verbindungsliste, sowie das Einstellen des Zeitpunktes, zu dem Informationen in der Verbindungsliste automatisch gelöscht werden. Die Meldung von außergewöhnlichen Ereignissen bezieht sich besonders auf den Aufbau einer Verbindung.

Geplant ist die Implementierung eines zentralen Agenten, der eine Verbindung zwischen den einzelnen Modulen herstellt.

5.2 Benutzerschnittstelle

Geplant ist eine einheitliche Benutzerschnittstelle für die vorgesehenen vier ATM-Testgenerator/Monitor-Anwendungen (Abbildung 6). Die Benutzer-Schnittstelle wird wie die Anwendungen in C++ implementiert. Es wird die Netscape Browser GUI eingesetzt.

Es ist vorteilhaft, wenn ein Administrator den ATM-Testgenerator/Monitor auch von einer entfernten Lokation ansprechen kann. Zu diesem Zweck kommuniziert die Benutzer-Schnittstelle über ein geeignetes Protokoll mit dem Netscape Browser. Diese Möglichkeit wurde im Rahmen einer Diplomarbeit für eine Tivoli-Anbindung unter Benutzung eines proprietären Protokolls implementiert [Fle98]. Das ermöglicht gleichzeitig den Einsatz des ATM-Testgenerator/Monitors im Rahmen eines System Management-Produktes wie Tivoli.

Effizienter ist der Einsatz eines Standard-Protokolls wie z.B. die Anbindung des Netscape Browsers über einen CORBA ORB. Java Applets können bei Bedarf in den

Browser geladen werden. Die C++ Programme werden über eine Wrapper-Funktion integriert. Dieser Ansatz wurde in einer weiteren Diplomarbeit unter Benutzung des Visigenic ORB's und des Netscape Browsers implementiert und verifiziert [Bes97]. Die gleiche Struktur wird verwendet, unabhängig ob der Browser lokal oder remote benutzt wird.

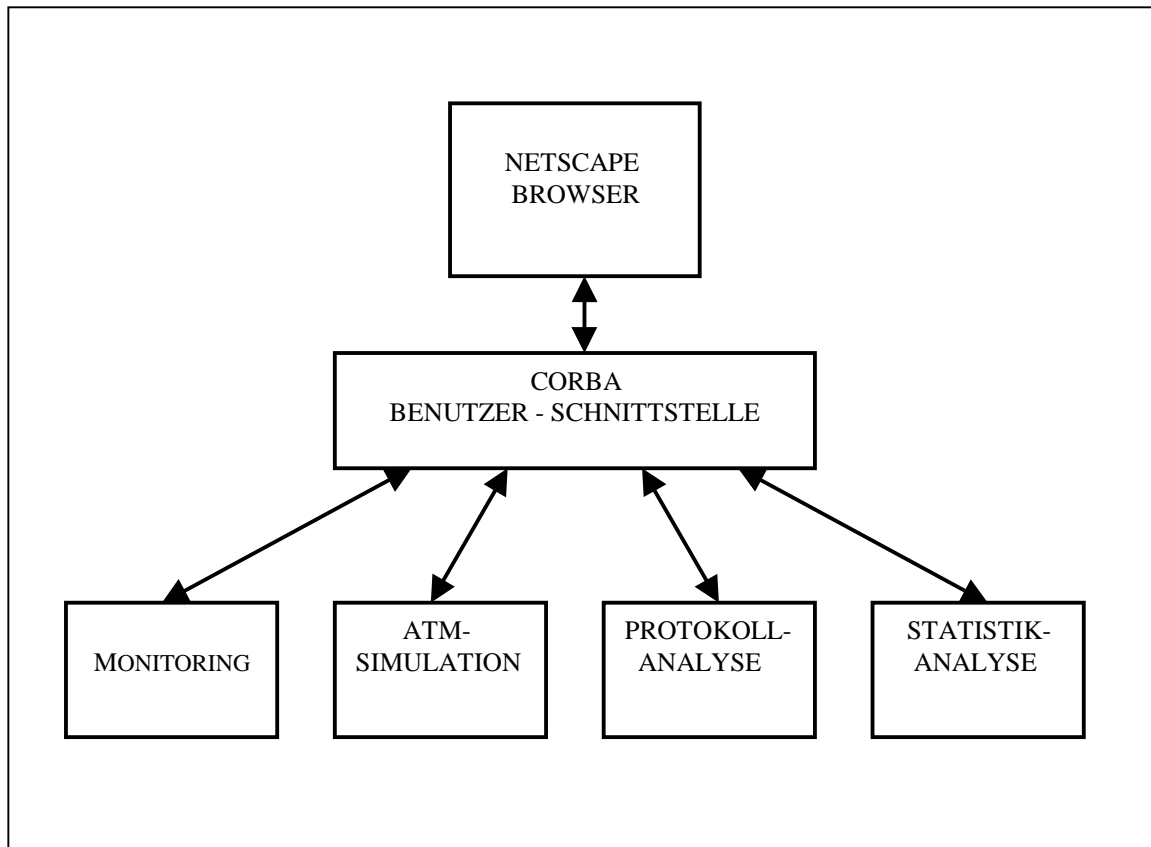


Abbildung 6: Einheitliche Benutzerschnittstelle für verschiedene ATM-Testgenerator/Monitor Anwendungen

CORBA verwendet das IIOP Protokoll, welches wiederum auf der Socket-Schnittstelle aufbaut. In Fortführung der in Abschnitt 7.3 beschriebenen Arbeit soll die derzeitig implementierte Schnittstelle hierzu auf Winsock umgestellt werden.

5.3 Transportschnittstelle

Die Socket-Schnittstelle entstand ursprünglich als Teil der Berkeley Software Distribution (BSD) der Universität von Californien in Berkeley. Dieses BSD Socket-Paradigma wurde von Microsoft als API für Windows 3.x, Windows 95 sowie Windows NT übernommen.

Neben der protokollunabhängigen Architektur besitzt die Version 2 des Winsock-API (Abbildung 7) weitere Neuerungen, unter anderem:

- Protokollunabhängige Name-Resolution

- Unterstützung von Overlapped-IO Operationen
- Implementierung von protokollunabhängigen Point-To-Multipoint- bzw. Multicast-Mechanismen
- Quality of Services

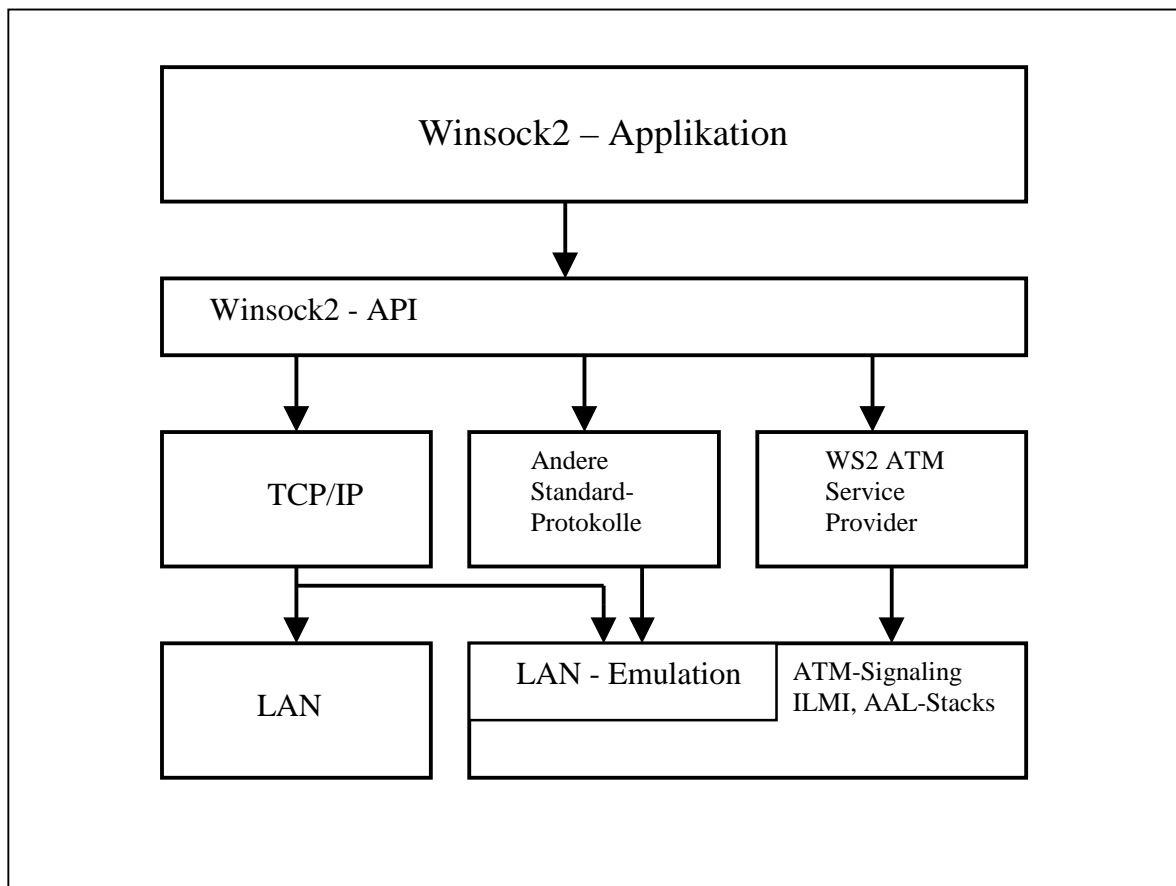


Abbildung 7: Winsock2-API

6 Unterstützende Arbeiten

6.1 Analyse der möglichen Verarbeitungsraten

Heutige Hochgeschwindigkeitsnetze (z.B. ATM) arbeiten mit Datenraten von 155 MBit/s oder mehr. Arbeitsplatzrechner, die über LAN-Emulation direkt an ein ATM-Netz angeschlossen sind, können diese Datenrate nicht mehr als zu 1/3 oder 1/2 ausnutzen. Es existieren Aussagen, daß die Protokollschichten 3 - 7 des OSI Modells auf heutigen Rechnern nur eine begrenzte Datenrate verarbeiten können.

Diese Problemstellung wurde näher untersucht [Bee97]. Die für die Untersuchungen benutzte Meßumgebung und die verwendeten Meßinstrumente für Classical IP bestanden aus zwei RS/6000 Rechnern, die über 100 MBit/s MC ATM-Adapterkarten

und einen IBM 8260 ATM-Switch miteinander verbunden waren. Als Meßwerkzeug wurde IBM Netview 6000 eingesetzt. Für LAN-Emulation wurden zwei SUN Sparc 10- Rechner verwendet, die über 155 MBit/s S-Bus-Adapterkarten miteinander verbunden waren. Die Classical IP-Messungen erfolgten in Leipzig; für die LAN-Emulation-Messungen stellte das FOKUS-Institut in Berlin freundlicherweise die erforderlichen Einrichtungen zur Verfügung. Als Analysewerkzeug wurde das HP Tool netperf eingesetzt.

Bei den Classical IP-Messungen wurde teilweise ein im Hintergrund laufender Lastprozeß gestartet. Erwartungsgemäß erlauben große MTU's (Maximum Transfer Unit) eine höhere Datenrate als kleine MTU's. Außerdem kann auf den MC-Adapterkarten die maximal zulässige Datenrate eingestellt werden. Die tatsächlich erreichte Datenrate wächst ebenfalls mit steigender maximaler Datenrate. Insgesamt sind jedoch nur maximal übertragbare Datenraten von etwa 20 MBit/s möglich; bei mitlaufendem Lastgenerator sind es deutlich weniger. Dies erlaubt die Schlußfolgerung, daß die Abarbeitung der Pfadlänge in OSI Schicht 3 und 4 der limitierende Faktor ist. Gleichzeitig liegt der Schluß nahe, daß für die derzeitig verfügbaren Arbeitsplatzrechner eine ATM-Anschlußmöglichkeit von 25 MBit/s in den meisten Anwendungen vollkommen ausreichend sein dürfte.

6.2 Analyse der Verarbeitungsraten unter ATM-Bedingungen

In Leipzig wurde die Leistungsfähigkeit des TCP/IP Stacks am Beispiel einer spezifischen Hardware-Plattform - SUN Sparcstation 20, 25 MHz, 40 MByte Hauptspeicher - und einer spezifischen Software-Implementierung (SunOS 4.1.3) untersucht [Wei97]. Die experimentellen Voraussetzungen (Hardware, Software, Instrumentierung) standen zum Zeitpunkt der Anfertigung der Arbeit in Leipzig nicht zur Verfügung.

Aus den Untersuchungen wurde ein Queuing abgeleitet und geeignete Meßpunkte innerhalb des Programmcodes instrumentiert. An diesen Punkten wurden kurze (12 Befehle) Zeitmessungsroutinen in den vorhandenen Treiber- und TCP/IP-Code eingefügt. Damit war es möglich, das Zeitverhalten an diesen Meßpunkten zu beobachten. Hiermit wurde eine Verteilung der Bearbeitungszeiten für die einzelnen TCP/IP-Komponenten gemessen. Dies geschah zunächst auf der Empfangsseite, und anschließend auf der Sendeseite.

Schließlich wurden konkrete Durchsatzmessungen zwischen den Anwendungsschichten von zwei Rechnern unter 155 MBit/s ATM und zum Vergleich unter 10 MBit/s Ethernet durchgeführt. Unter ATM konnte eine maximale Datenrate von 45 - 50 MBit/s erreicht werden. Erwartungsgemäß waren die Datenraten bei kleinen Paketgrößen wesentlich geringer.

Dabei ließen sich die Unregelmäßigkeiten in den Meßergebnissen sehr gut aus der Analyse der Einzelmessungen an den erwähnten Meßpunkten ableiten. Als Ergebnis sind nicht nur Durchsatzmeßdaten ermittelt worden, sondern es konnte auch geklärt werden, warum die Daten diese Charakteristik haben.

Die Netzwerk-Schnittstelle bietet die Möglichkeit, aus dem entsprechenden Agenten Informationen aus der Signalisierung zwischen ATM-Switches zu sammeln [Str97].

6.3 ATM Socket-Schnittstelle

Die Firma Fore stellt für ihre ATM-Adapter-Karten einen Protokoll Stack zur Verfügung, der die Winsock 2-Schnittstelle direkt auf AAL5 implementiert. Es wurde untersucht, welche Geschwindigkeitsverbesserung erzielt werden kann, wenn ein Winsock ATM Stack an Stelle eines TCP/IP und LANE Stacks eingesetzt wird. Die Ergebnisse sind in den Abbildungen 8 und 9 wiedergegeben.

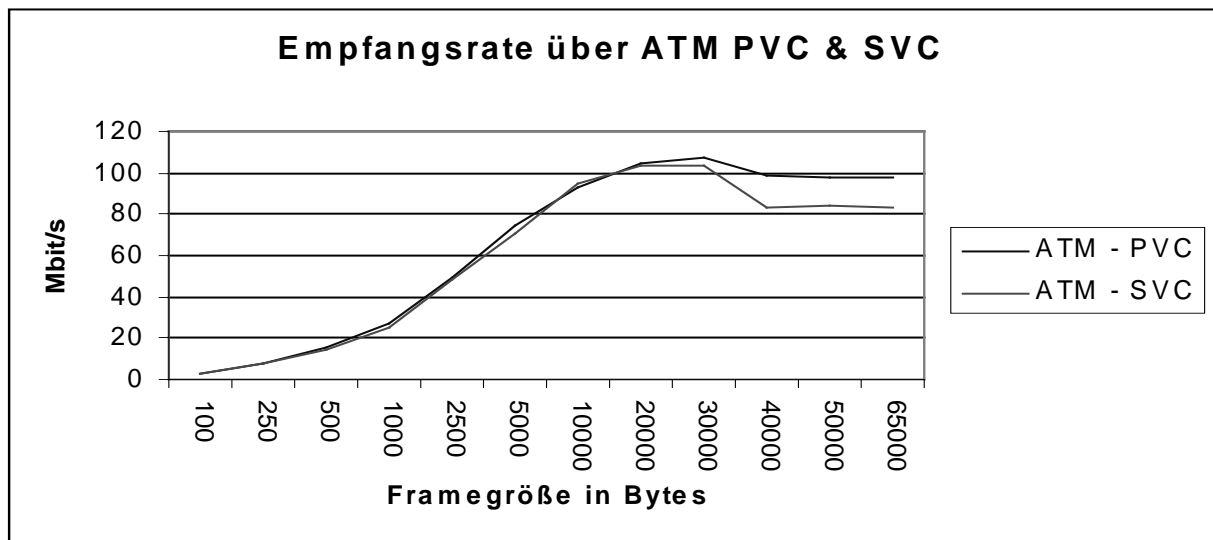


Abbildung 8: Vergleich der Empfangsrate (über ATM PVC & SVC) in Abhängigkeit von der Rahmengröße

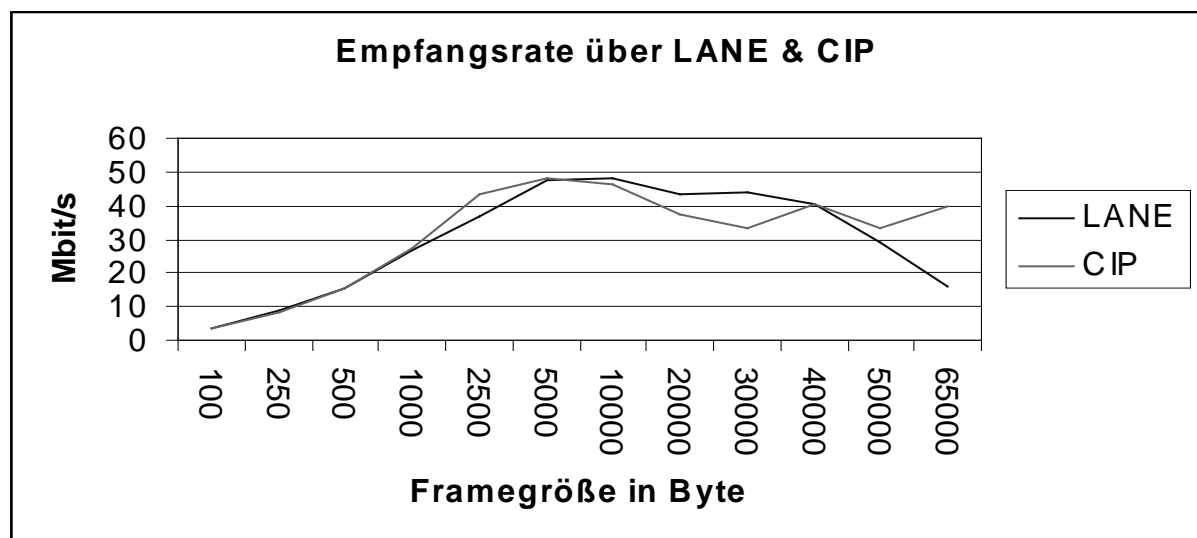


Abbildung 9: Empfangsrate (über LANE & CIP) in Abhängigkeit von der Rahmengröße

Während bei kleineren Rahmenlängen, z.B. 1000 Bytes, kein wesentlicher Unterschied festzustellen ist, wird z.B. bei einer Rahmengröße von 10 000 Bytes das Leistungsverhalten des ATM Stacks um einen Faktor 2 besser.

7 Literatur

- [Bee96] H. Beese: Performanceanalyse von ATM-Netzwerken in Form von Durchsatzmessungen unter „Classical IP over ATM“ und „LAN Emulation over ATM“, Dipl.-Arbeit, Univ. Leipzig, 1996
- [Bes97] K. Beschorner: Realisierung einer Client/Server-Anwendung mit CORBA und Java unter Berücksichtigung bestehender C++ Komponenten, Dipl.-Arbeit, Univ. Tübingen, 1995
- [Fle98] T. Fleischer: Netzwerkinventarisierung und –dokumentation, Design und Implementierung einer plattformunabhängigen Softwarelösung, Dipl.-Arbeit, Univ. Leipzig, 1998
- [HeWe96] A. Hergenhan, K. Weiß: User Manual Microcontroller-Design Environment, FZI Karlsruhe, 1996
- [Kie96] B. Kieling: Leistungsuntersuchungen von Multimedia-Anwendungen in ATM-Netzen, Dipl.-Arbeit, Univ. Leipzig, 1996
- [Moc97] K. Mochalski: Entwurf und Implementierung eines Protokoll-Dekoders für LAN-Emulation über ATM, Dipl.-Arbeit, Univ. Leipzig, 1996
- [Pry94] M. de Prycker: Asynchronous Transfer Mode, Prentice Hall, 1994
- [Str96] P. Sträuber: Entwicklung eines Konzeptes und Implementierung eines Software-Werkzeuges zur Analyse von ATM-Verbindungen, Dipl.-Arbeit, Univ. Leipzig, 1996
- [Tit95] C. Titel: Entwicklung und Implementierung eines Signalisierungs-Testtools zur Evaluierung von ATM-Komponenten, Dipl.-Arbeit, Univ. Leipzig, 1995
- [Wei97] D. Weigenand: Analyse des Zeitverhaltens und der Ressourcen-Nutzung von Protokollstacks am Beispiel der TCP/IP-Implementierung in SunOS 4.1.3, Dipl.-Arbeit, Univ. Leipzig, 1996