Sieben OSI-Schichten sind oft zuviel : Rechnerkommunikation mittels Buskopplern, Teil 2

Autor(en): **Stein, Erich**

Objekttyp: **Article**

Zeitschrift: **Bulletin des Schweizerischen Elektrotechnischen Vereins, des Verbandes Schweizerischer Elektrizitätsunternehmen = Bulletin de l'Association Suisse des Electriciens, de l'Association des Entreprises électriques suisses**

Band (Jahr): **84 (1993)**

Heft 9

PDF erstellt am: **30.05.2024**

Persistenter Link: https://doi.org/10.5169/seals-902682

Nutzungsbedingungen

Die ETH-Bibliothek ist Anbieterin der digitalisierten Zeitschriften. Sie besitzt keine Urheberrechte an den Inhalten der Zeitschriften. Die Rechte liegen in der Regel bei den Herausgebern.
Die auf der Plattform e-periodica veröffentlichten Dokumente stehen für nicht-kommerzielle Zwecke in Lehre und Forschung sowie für die private Nutzung frei zur Verfügung. Einzelne Dateien oder Ausdrucke aus diesem Angebot können zusammen mit diesen Nutzungsbedingungen und den korrekten Herkunftsbezeichnungen weitergegeben werden.
Das Veröffentlichen von Bildern in Print- und Online-Publikationen ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. Die systematische Speicherung von Teilen des elektronischen Angebots auf anderen Servern bedarf ebenfalls des schriftlichen Einverständnisses der Rechteinhaber.

Haftungsausschluss

Alle Angaben erfolgen ohne Gewähr für Vollständigkeit oder Richtigkeit. Es wird keine Haftung übernommen für Schäden durch die Verwendung von Informationen aus diesem Online-Angebot oder durch das Fehlen von Informationen. Dies gilt auch für Inhalte Dritter, die über dieses Angebot zugänglich sind.

Ein Dienst der *ETH-Bibliothek*
ETH Zürich, Rämistrasse 101, 8092 Zürich, Schweiz, www.library.ethz.ch

http://www.e-periodica.ch

Mit der steigenden Zahl verschiedener Bussysteme steigt die Bedeutung von Buskopplern. Diese erlauben, gleiche und verschiedene Computerbusse hardwaremässig so zu koppeln, dass zwischen zwei oder mehreren Computern kommuniziert werden kann. Nachdem der Autor in einem ersten Teil (Heft 3/93) die Rechnerkommunikation von einer übergeordneten Sicht behandelt hat, geht er in diesem zweiten Teil auf den Aufbau, die Funktionen, die Software und die Anwendungen von Buskopplern ein, wobei als derzeit wichtigstes Kopplungsprinzip Shared Memory im Vordergrund der Betrachtung steht.

Sieben OSI-Schichten sind oft zuviel

Rechnerkommunikation mittels Buskopplern, Teil 2

■ Erich Stein

Mit der Lancierung und Einführung von immer neuen Bussystemen nimmt auch die Bedeutung der Busadapter zu [9; 10]. Zu den verbreiteten Industriebussen Multibus I und II und VME-Bus – der erstere wurde ursprünglich für die Intel 80x86-Prozessoren, der zweite ursprünglich für die Motorola-680X0-Prozessoren entwickelt [11] – sowie den PC-Bussen PC/AT-Bus (80x86, Industriestandard) und Nubus (680X0, Apple) sind in den letzten Jahren der Micro Channel von IBM [12] und der Eisa-Bus [13] hinzugekommen. In der DEC-Welt spielt der Q-Bus noch immer eine grosse Rolle, und über die Workstations sind in letzter Zeit Busse wie der S-Bus und der Turbochannel bekannt geworden. Daneben gibt es eine Vielzahl spezialisierter Parallel- und Seriell-Busse. In der Praxis sind alle Busse – durch den Markt bedingt, also nicht primär aus technischen Gründen – schwerpunktmässig an bestimmte Anwendungsbereiche gebunden. Daraus ergibt sich das Spektrum der für einen bestimmten Bus verfügbaren Funktionen (Boards). Beispielsweise stehen für PC-Busse eine grosse Vielfalt von Interfaces für Peripheriegeräte zur Verfügung, während Schnittstellen zu technischen Prozessen eher in der VME-Welt zu finden sind. Auch die Auswahl der für einen bestimmten Bus verfügbaren Betriebssysteme ist limitiert. Deshalb ist die Wahl eines Bussystems eine komplexe Aufgabenstellung. Bild 5 versucht eine grobe Klassifizierung der wichtigsten Bussysteme anhand der Kategorien Reifegrad und Komplexität. Der Marktanteil der einzelnen Bussysteme wird durch die Fläche der Kreisscheibe visualisiert.

Einsatzfälle für Buskoppler

Aus dem oben gesagten wird klar, dass bei der Auswahl von Bussystemen Kompromisse eingegangen werden müssen. Ein alternativer Ansatz besteht darin, verschiedene Bussysteme in einer Anlage zu kombinieren und damit insgesamt die Anforderungen besser zu erfüllen, ohne den Aufwand in die Höhe zu treiben. Aus der Sicht eines Anwenders kleiner Systeme sind folgende Situationen denkbar: Er möchte Geräte, die er schon hat, die aber verschiedene Bussysteme verwenden, miteinander verbinden oder er möchte ein Bussystem A um eine Funktion ergänzen, die nur auf dem Bussystem B verfügbar ist. Oder er hat eine Workstation mit dem Bussystem X und 3 freien Steckplätzen, und möchte mit ihr 7 Karten des Bussystems Y steuern. In diesem Fall können X und Y durchaus dasselbe Bussystem darstellen.

Bei grossen Systemen, die von vorneherein auf mehrere Backplanes angewiesen sind, spricht man von Multi-Crate Systems. Im Grenzfall möchte man eine sogenannte Infinite Backplane, eine Backplane also, die beliebig erweitert werden kann. In all diesen Fällen ist der Einsatz von Buskopplern (Busadapter oder Busrepeater) eine nahelie-

Adresse des Autors
Erich Stein, Dipl. Ing., Käshaldenstrasse 39, 8052 Zürich.

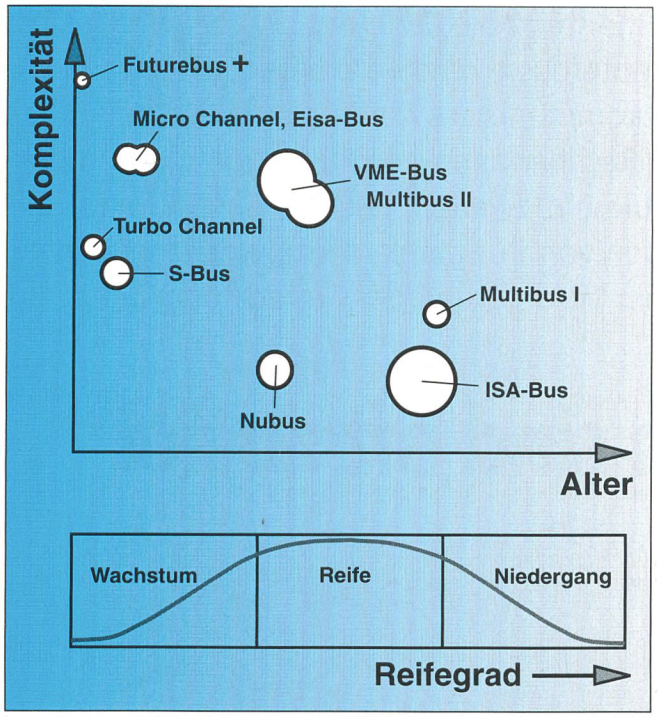


Bild 5 Bussysteme

gende Problemlösung, denn heute sind für die meisten Kombinationen von Bussystemen geeignete Buskoppler erhältlich [14; 15].

Aufbau und Funktionen von Buskopplern

Grundbegriffe

Normalerweise können beide Systeme, die über Buskoppler verbunden sind, Transfers auslösen und steuern. Das auslösende System wird als Quellensystem und das antwortende als Zielsystem bezeichnet. Das heisst, dass beide Systeme abwechselnd die Rolle des Quellen- und des Zielsystems spielen können. Ein System, das nur als Zielsystem funktioniert, wird als Empfängersystem bezeichnet. Analog dazu heissen Systeme, die sowohl Quellen- als auch Zielsystem sein können, Sendersysteme.

Zur Beschreibung der Abläufe in busgekoppelten Systemen muss auch die betrachtete Sicht genannt werden. Das lokale (Local) System, in dem man sich gedanklich befindet, kommuniziert mit dem entfernten (Remote) System. Konflikte können entstehen, wenn beide Systeme gleichzeitig als Quellensystem aktiv werden wollen. In diesem Fall ist eine Arbitrierung durch Software erforderlich, die wie folgt gelöst werden kann. System 2 sendet einen programmierten Interrupt (siehe unten), um seinen Zugriffswunsch mitzuteilen. System 1 verarbeitet diesen Interrupt durch eine unteilbare[1] Read-Modify-Write-Operation auf einem Flagbit. Dieses zeigt, wenn es gesetzt ist, dass das System 2 aktiv ist oder aktiv werden möchte. Sobald das Flagbit gesetzt ist, wird der Interrupt im System 2 zurückgesetzt; damit hat System 2 die Berechtigung zum Senden erhalten.

[1] Eine Operation, die auf keinen Fall durch einen anderen Buszugriff unterbrochen werden darf.

Aufbau von Buskopplern

Jeder Buskoppler besteht aus drei Teilen: je einer Adapterkarte für die beiden Bussysteme und einem Kabel, das die beiden Adapterkarten verbindet. Die Länge des Kabels kann bis zu 7,5 m (Herstellerangabe) betragen, so dass die Systeme entsprechend den räumlichen Anforderungen der Applikation aufgestellt werden können. Steckverbindungen für Busse können bis zu 200 Kontakte aufweisen. Da ein Kabel mit ebensovielen Adern nicht mehr zu handhaben ist, werden Steuersignale, Adressen und Daten sequentiell übertragen. Somit sind 50 Adern ausreichend. Für längere Distanzen bis zu 2 Kilometern gibt es auch Interfacekarten zu Busrepeatern, die den Anschluss einer optischen Übertragungsstrecke erlauben. Obwohl in diesem Fall alle Bits seriell übertragen werden, erlaubt die hohe Übertragungsgeschwindigkeit auf dem Faserkabel Datenraten von 20 MByte/s von Bus zu Bus.

Funktionen von Buskopplern

- wahlfreie Adressierung des anderen Bussystems
- Schreib- und Lesezugriffe auf einzelnen Adressen
- Übertragung von Datenblöcken (DMA)
- Übertragung von Interrupts
- Dual Port RAM als Pufferspeicher
- Überwachungsfunktionen
  Paritätsprüfungen
  Busfehler
  Interface Timeout
- Datenumformatierung
  Byte Swapping
- System Controller für den sekundären Bus

Tabelle III

Funktionen – Übersicht

Die wichtigsten Merkmale der Buskoppler sind die angebotenen Funktionen. Die Tabelle III gibt eine Übersicht über die wesentlichen Funktionen. Selbstverständlich hängt der verfügbare Funktionsvorrat von der betrachteten Kombination zweier Bussysteme ab. Im folgenden werden, um nicht im Allgemeinen hängen zu bleiben, konkrete Beispiele herangezogen. Bild 6 zeigt schematisch die drei wichtigsten Abläufe in einem System, das zwei Bussysteme durch einen Buskoppler verbindet. Solange auf beiden Bussystemen nur lokale Adressen auftreten, arbeiten beide unabhängig voneinander. Bei Ansprechen einer entfernten (busüberschreitenden) Adresse sind beide Bussysteme für die Dauer der Transaktion miteinander verbunden, das heisst sie haben dieselbe Adressinformation und äquivalente Steuersignale. Falls auf einem Buskoppler ein Dual Port RAM (siehe unten) eingesetzt wird, können das lokale und das entfernte System quasi gleichzeitig auf dieses zugreifen.

Adressierung, Adressfenster

Die Adressierung des entfernten Bussystems geschieht über Adressfenster (Bild 7) [16]. Dazu wird im Adressraum des Quellensystems ein Adressbereich – das sogenannte Adressfenster – definiert, und Zugriffe auf dieses werden auf den Adressraum des Zielsystems abgebildet. Bild 7c zeigt ein Beispiel für eine Kopplung von PC/AT-Bus und VME-Bus. Das bedeutet, dass der im Adressfenster liegende Speicherbereich nicht mehr als lokaler Speicher des Quellensystems verfügbar ist. Auf der Seite des Zielsystems kann das Adressfenster an eine andere Stelle des Adressbereichs gelegt werden. Dies verhindert Konflikte bei der Belegung der Adressräume der beiden Systeme. Für die Gestaltung der Adressabbildung gibt es zwei Möglichkeiten. Beim sogenannten Direct Mode ist die Grösse des Adressfensters gleich der Grösse des adressierbaren Speicherbereichs im Zielsystem und die Adressabbildung wird durch Steckbrücken oder DIP-Schalter festgelegt (Bild 7a). Die Grösse des Adressfensters kann typisch zwischen 128 K und einigen M liegen. Die zweite Möglichkeit ist der Page Mode (Bild 7b), bei dem die Grösse des Fensters konstant ist und

typisch 64 K Adressen entsprechend einer Adresswortlänge von 16 Bit, umfasst. Diese 16 Bit bilden den niederwertigen Teil der Adresse im Zielsystem. Der höherwertige Teil wird aus einem Register entnommen, das vom Quellensystem geladen wird. Auf diese Weise kann auf den gesamten Adressraum des Zielsystems zugegriffen werden.

Für Bussysteme, die getrennte Adressräume für den Hauptspeicher und die Ein- und Ausgabe haben, ist für die Abbildung der Ein- und Ausgabeadressen ein eigenes Adressfenster vorzusehen.

Steuersignale

Wenn nun der Adapter noch die Steuersignale der beiden Busse korrekt und mit dem

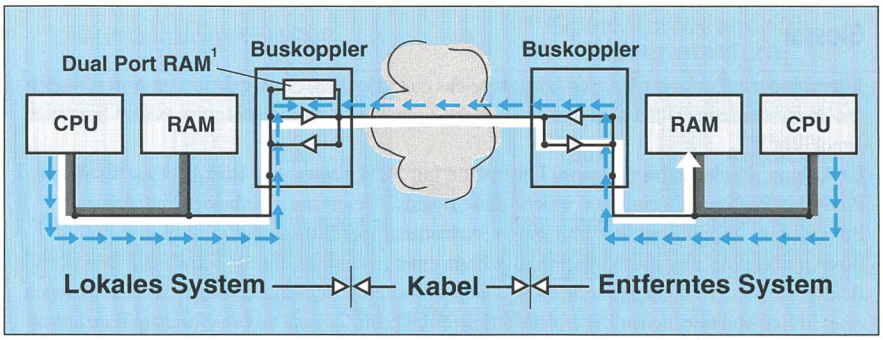


Bild 6 Wesentliche Abläufe in einem mittels Buskopplern verbundenen System

grau: lokale Speicherzugriffe können auf beiden Bussystemen gleichzeitig stattfinden.
weiss: entfernte Speicherzugriffe, die beiden Bussysteme sind für die Dauer des Zugriffs synchron.
Pfeile: Zugriffe auf das Dual Port RAM können vom lokalen und vom entfernten System aus quasi gleichzeitig stattfinden.
[1] ein Dual Port RAM darf nur auf einer Seite des Kabels installiert sein

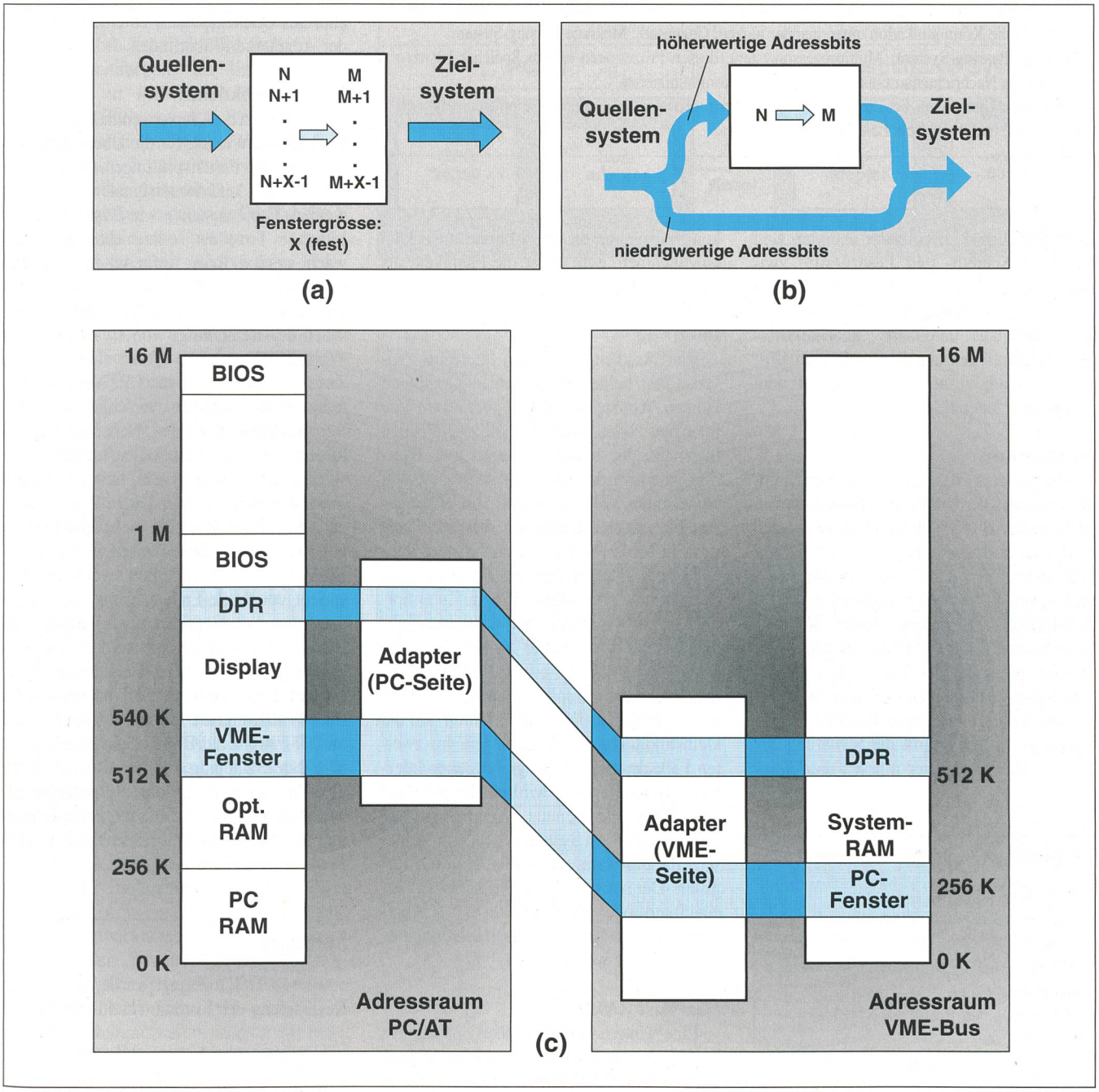


Bild 7 Adressabbildung bei Buskopplern

Glossar

Adressfenster: Derjenige Teil des Adressbereichs eines Quellensystems, der in den Adressbereich des Zielsystems abgebildet wird und dadurch die Kommunikation zwischen den beiden Systemen ermöglicht.
Backplane: Rückwandverdrahtung, Leiterplatte mit Steckplätzen, die Boards eines bestimmten Bussystems sowie die Leitungen, welche diese Boards untereinander verbinden, aufnehmen.
Busadapter: Anordnung zur Verbindung verschiedenartiger Bussysteme.
Busextender: Mechanische Vorrichtung in Form einer Leiterplatte für das betrachtete Bussystem. Beidseitige Steckverbindungen, die direkt miteinander verbunden sind, erlauben das Herausziehen einer in Betrieb befindlichen Leiterplatte aus dem Rack, um Zugang zu den einzelnen Bauelementen zu erhalten.
Busrepeater: Anordnung zur Verbindung gleichartiger Bussysteme.
Infinite Backplane: Backplane, die beliebig erweitert werden kann.
Multimasterbus: Bussystem, auf dem verschiedene Busteilnehmer abwechselnd die Steuerung übernehmen können.
Parallelbus: Bussystem, bei dem Adressen und Daten parallel übertragen werden. Typischerweise sind zwischen 36 und 200 parallele Leiter und Steckkontakte vorhanden.
Shared Memory: Multiprozessorsystem, dessen Prozessoren einen gemeinsamen Speicher(bereich) für die Kommunikation untereinander nutzen. Gegensatz: Message Passing-System.
Message Passing-System: Multiprozessorsystem, dessen Prozessoren eigene Speicher besitzen und mittels Nachrichtenaustauschs untereinander kommunizieren.
Seriellbus: Bussystem, bei dem Daten seriell übertragen werden. Seriellbusse erfordern typisch zwischen 2 und 4 parallele Leiter und Steckkontakte.

richtigen Timing aufeinander abbildet, sind wahlfreie Schreib- und Lesezugriffe zwischen den Hauptspeichern der beiden Systeme möglich. Spezielle Bustransaktionen, zum Beispiel unteilbare Read-Modify-Write-Operationen auf dem entfernten Bus, werden durch zusätzliche Logik auf dem Busadapter unterstützt.

Arbitrierung

Ein weiteres Problem gibt es, wenn mindestens eines der beteiligten Bussysteme ein Multimaster-Bussystem ist. Dann ist auf den Multimaster-Bussystemen eine Arbitrierung erforderlich, die durch einen in Hardware realisierten Arbiter vorgenommen wird und in folgenden drei Schritten abläuft: Die verschiedenen Busmaster bewerben sich zuerst um die Steuerung des Busses. Ein Arbiter wählt einen davon gemäss einem bestimmten Algorithmus aus. Dieser Busmaster übernimmt im dritten Schritt die Steuerung des Busses. Die Busadapter müssen die Fähigkeit besitzen, an diesen Arbitrierungszyklen teilzunehmen, evtl. sogar die Fähigkeit, als Arbiter zu fungieren.

Anwendungsbereiche für Buskoppler
Simulation
Prozesssteuerung und -überwachung
Automatische Testsysteme
Bildverarbeitung
Signalverarbeitung
Entwicklung von Real-Time-Systemen
Datenerfassung
Automatisierungstechnik

Tabelle IV

Interrupts

Ein Busadapter soll auch Interrupts zwischen den beiden Bussystemen übertragen können. Konzeptionell sind drei Arten von Interrupts zu unterscheiden: die zwei lokalen Interrupts der beiden Systeme und Interrupts, die über das Kabel übertragen werden (sogenannte Kabelinterrupts). Im allgemeinen Fall möchte man auch mehrere Interrupts in beide Richtungen unabhängig voneinander übertragen können; das Kabel sollte dann mehrere Adern für die Interruptübertragung anbieten. Für die Interruptübertragung gibt es zwei Varianten. In der ersten werden die auf einem Bus entstehenden Interrupts transparent auf den anderen Bus durchgeschaltet und erfahren so die kleinstmöglichen Verzögerungen. Im zweiten Fall, der hier als programmierter Interrupt bezeichnet wird, wird ein Register auf dem Buskoppler benutzt. Wenn System B einen Interrupt an System A senden möchte, setzt die Software von System B dieses Register. Der Ausgang des Registers ist auf eine Interruptleitung des Systems A geschaltet, so dass im System A ein Hardwareinterrupt ausgelöst werden kann.

Dual Port RAM

Eine weitere Funktion eines Busadapters betrifft den Pufferspeicher, der vorteilhaft als Dual Port RAM (DPR) ausgeführt ist. In diesem Fall können die oben beschriebenen Zugriffskonflikte zwischen lokalem und entferntem System nicht auftreten, da sie durch das Dual Port RAM abgefangen werden. Das DPR kann auf einer der beiden Adapterkarten untergebracht sein[2]. Logisch gehört es zu beiden Systemen, wobei die Adressen auf den beiden Seiten nicht identisch sein müssen (Bild 7). Das DPR kann auch als Pufferspeicher für den Direct Memory Access (DMA) zwischen Quellen- und Zielsystem genutzt werden.

Direct Memory Access

Auf einigen Adaptern ist ein DMA-Controller vorhanden, der die Übertragung von Datenblöcken vom Speicher des Quellen- zum Speicher des Zielsystems steuert. Dadurch werden die CPUs entlastet. Alternativ kann die DMA-Übertragung auch vom Speicher des Quellensystems zu einem DPR auf der Adapterkarte stattfinden.

Weitere Funktionen

Die meisten Adapter enthalten weitere Funktionen. Wichtig für die Übertragungssicherheit sind Paritätsprüfungen, die auf Daten-, Adress- und Steuersignale angewendet werden. Fehlerzustände wie Busfehler[3] oder Interface Timeouts sollten ebenfalls überwacht werden. Beim Auftreten eines Fehlers sollte der Busadapter einen Interrupt auslösen können. Byte Swapping – das Vertauschen der Reihenfolge von Bytes in einem Wort der Länge $n \cdot 8$ Bit ist eine nützliche Zusatzfunktion, wenn die CPUs auf den beteiligten Bussystemen verschiedene Byte-Reihenfolgen verwenden. Dabei geht es um folgendes: Oft müssen Daten kurzer Länge so zusammenzusetzen sein, dass ein Datum mit fester, vorgegebener Länge – ein Wort – entsteht [17]. 4 Bytes beispielsweise können auf zwei verschiedene Weisen zu einem 32-Bit-Wort zusammengesetzt werden: Man spricht von Little-Endian, wenn die einzelnen Bytes vom niedrigwertigen Ende des 32-Bit-Wortes aus aufgereiht werden, von Big-Endian, wenn man am höherwertigen Ende beginnt. Die Prozessoren 80x86 verwenden das Prinzip Little-Endian, während die 680X0-Familie auf Big-Endian ausgelegt ist. Mit Byte Swapping können also die Bytes eines Wortes in die richtige Reihenfolge gebracht werden, wenn CPUs mit Little-Endian und Big-Endian an den beiden Enden eines Buskopplers eingesetzt werden.

Software

Wie in Teil 1 gezeigt wurde, bedingt die Realisierung der Kommunikation in offenen

[2] Die entsprechenden Adressen im Hauptspeicher bleiben dann frei.

[3] Busfehler entstehen beispielsweise, wenn ein nicht vorhandener Speicher adressiert wird.

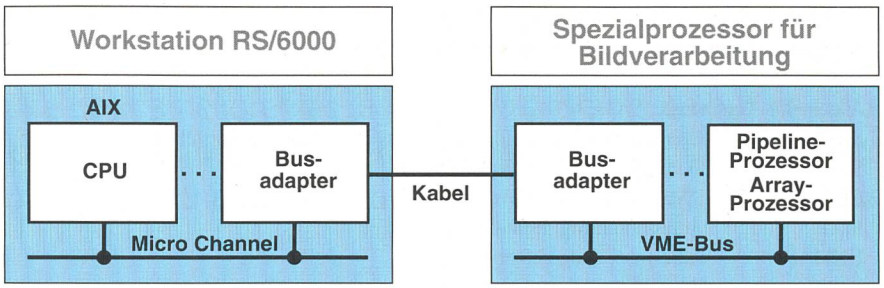
Systemen einen grossen Softwareaufwand. Die Busadapter sind in dieser Hinsicht viel bescheidener. Im Idealfall sind sie transparent und benötigen keinerlei Software. Allenfalls notwendige Einstellungen zur Konfiguration werden über Steckbrücken vorgenommen. Zur Erhöhung der Flexibilität ist es jedoch wünschbar, die Konfiguration per Softwaresteuerung verändern zu können. Wenn Funktionen mit einer gewissen Intelligenz, wie zum Beispiel DMA, Interrupt-Subsysteme oder Überwachungsfunktionen auf dem Buskoppler implementiert sind, ist ebenfalls Software zu deren Betrieb erforderlich. Diese Software wird in Form von systemspezifischen Treibern verfügbar gemacht, die von der eingesetzten CPU und dem (Real-Time-) Betriebssystem abhängig sind. Softwaregesteuerte Buskoppler verzichten soweit wie möglich auf eine manuelle Initialisierung durch Steckbrücken und DIP-Schalter.

Zur Vereinfachung der Integration der Buskoppler in verbreiteten Anwendungssystemen (z.B. Sparcstation, RS/6000-Workstation, Decstation 5000) werden ladbare Device Driver, Installationsscripts und Beispielprogramme zusammen mit der Buskoppler-Hardware angeboten. Die Tabelle V zeigt als Beispiel die Funktionen eines Device Drivers für einen Microchannel-VME-Busadapter. Er läuft auf einer RS/6000-Workstation unter dem Betriebssystem AIX.

Eine nützliche Funktion des Device Drivers ist ein Interrupt Handler für den entfernten Bus, der, müsste man ihn selbst programmieren, einigen Aufwand verursachen würde. Die Beispielprogramme sollen die Funktionen des Buskopplers aufzeigen und Debugging und Test bei der Systemintegration unterstützen.

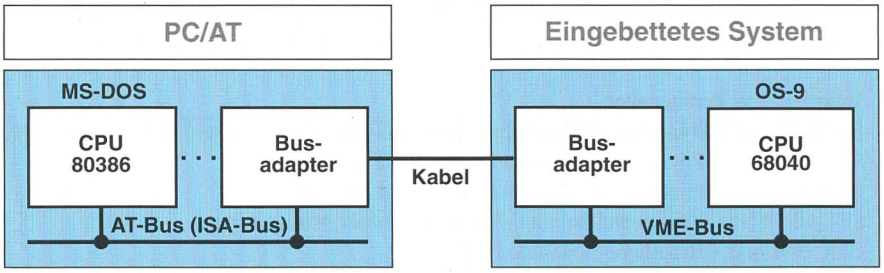
Anwendungsbeispiele

Aus der Sicht der Praxis bieten Buskoppler dem Systemintegrator eine einfache Möglichkeit, Systeme mit verschiedenen Bussystemen aufzubauen. Die Tabelle IV zeigt Bereiche, in denen Buskoppler erfahrungsgemäss häufig eingesetzt werden. Zur Veranschaulichung des Nutzens sind in Bild 8 drei Anwendungen beispielhaft dargestellt. Das erste Beispiel zeigt eine typische Anwendung in der Bildverarbeitung. Ein weiterer Anwendungsschwerpunkt besteht darin, dass man ein Entwicklungssystem mit einem zu entwickelnden System verbindet (Bild 8, Beispiel 2). Der systematische Einsatz von Buskopplern zum Aufbau von generell einsetzbaren Multiprozessorsystemen aus Standard-Rechnersystemen, zum Beispiel Workstations, wurde bisher offenbar nur wenig vorangetrieben. Das Beispiel 3 in Bild 8 zeigt



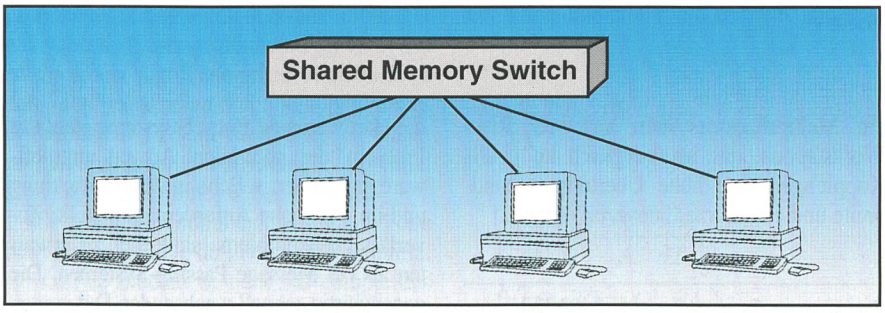
Funktionen	Nutzen der direkten Buskopplung	Funktionen
– Gewohnte Umgebung der RS/6000 Workstation und AIX (UNIX) – Steuerung des VME-Bussystems – Download der Programme zum VME-Bussystem	Einbindung spezieller Funktionen, die unter anderen nur in wenigen Varianten am Markt verfügbar sind, in eine Standard-Umgebung	Aufgabenspezifisch konfiguriertes VME-Bussystem (>1000 Board-Typen von etwa 100 Herstellern verfügbar)

Bild 8.1 Beispiel 1: Workstation mit Spezialprozessor für Bildverarbeitung



Funktionen	Nutzen der direkten Buskopplung	Funktionen
– Standardisierte PC-Umgebung – Case-Tools für SW-Entwicklung – Cross Compiler für Zielsystem auf VME-Bus – Download für Maschinencode des Zielsystems – Diagnose-SW für Zielsystem	Standard-PC als kostengünstiges Entwicklungs- und Diagnosesystem	Wie bei Beispiel 1

Bild 8.2 Beispiel 2: PC als Entwicklungs- und Diagnosesystem



Funktionen (Workstations)	Nutzen der direkten Buskopplung	Funktionen (Shared Memory Switch)
– Parallelverarbeitung von C- und Fortran-Programmen – Express: SW für Parallelprogrammierung in C und Fortran, inkl. Support für Shared Memory Switch – Micro-Channel-Adapter für Kommunikation mit Shared Memory Switch	– Clustering von Standard-Workstations für die Parallelverarbeitung – Latenzzeit 3,5 µs zwischen Workstation und Shared Memory Switch – Datenrate bis 14 MByte/s auf Ebene der AIX-Tasks	VME-Bussystem mit – 8 MByte RAM – Arbiter – AIX Device Driver – bis zu 4 Busadapter VME-Bus zu Micro Channel

Bild 8.3 Beispiel 3: Cluster für Parallelverarbeitung auf Workstations

Bild 8 Konkrete Anwendungsbeispiele für Buskoppler

Softwarefunktionen für den Microchannel-VME-Buskoppler

Allgemeine Commands	
SETUP	Grundzustand beider Adapter und des Device Drivers herstellen
STATUS	Status des Device Drivers abfragen
CLR_STATUS	Status des Device Drivers löschen
MAP	Adressabbildung festlegen
UNMAP	Adressabbildung aufheben
IOREG	Register auf Adapter lesen / schreiben
RESET	Entfernten Bus zurücksetzen, Grundzustand des Adapters herstellen
Atomare Transaktionen	
TAS	Unteilbare Test-and-Set-Operation auf dem entfernten Bus
CAS	Unteilbare Compare-and-Swap-Operation auf dem entfernten Bus oder Dual Port RAM
Interrupt-Management	
REG_INTR	Aufrufende Task für Interrupt-Signal registrieren
AQR_INTR	Adapterzustand zum Zeitpunkt des Interrupts abfragen
CLEAN	Interrupt-Registrierung und -Datenblöcke einer Task freigeben
SND_INTR	Programmierten Interrupt zum entfernten Bus senden
Steuerung und Konfiguration	
IOCINFO	Systemspezifische Adapterinformation abfragen
SET_PRIV	Privilegüberprüfung im Device Driver ausschalten
CLR_PRIV	Privilegüberprüfung im Device Driver einschalten
SET_HNDSHK	Adapter und Device Driver auf Handshake Mode setzen
CLR_HNDSHK	Handshake Mode ausschalten
PARAM	Parameter des Device Driver setzen
DEV_ATTRIB	Parameter des Device Driver abfragen
LOG_ERROR	Interrupts wegen Statusfehler auf System Error Log aufzeichnen
NOLOG_ERROR	Nicht aufzeichnen
SET_DMA	DMA-Optionen setzen oder abfragen
Kernel-Funktionen	
REMBUS_INSTALL	Interrupt Handler der Kernel-Ebene mit Device Driver verbinden
REMBUS_REMOVE	Interrupt Handler abhängen
KMAP	Vom Device Driver benutzte Registerstruktur abfragen

Zur Vereinfachung wurde der allen Funktionen gemeinsame Prefix BIOC_ weggelassen.

Tabelle V

ein Multiprozessorsystem aus vier Risc-Workstations, das im Vergleich zu LAN-Kopplungen eine hohe Übertragungsbandbreite und eine kurze Latenzzeit erreicht.

Zusammenfassung und Ausblick

Die Rolle der direkten Buskopplung wurde im übergeordneten Rahmen der Rechnerkommunikation beleuchtet. Funktionen von Buskopplern, ihre Software und Anwendungen wurden beschrieben. Dem relativ unkomplizierten Einbinden der Buskoppler in die Hardware und Betriebssysteme steht das Problem der parallelen Anwendungssoftware gegenüber, welche in der Entwicklung und im Test hohe Anforderungen stellt; Zuverlässigkeitsprobleme sind eher zu erwarten als bei Message Passing-Systemen. Die gegenwärtig schnell wachsenden Datenraten auf Netzwerken (LAN, MAN und WAN) erlauben bereits Buskopplungen über grössere Distanzen. Wegen der grösseren Latenzzeit dürften allerdings geeignete Anwendungen eher selten sein. Message Passing-Systeme dürften in dieser Beziehung interessanter als Shared Memory-Systeme sein. Welche Abgrenzungen sich hier bei realen Systemen herausbilden werden, bleibt abzuwarten.

Literatur

[9] *G. Färber (Hrsg.):* Bussysteme. München: Oldenbourg, 3. Aufl., 1992.

[10] *J. di Giacomo (ed.):* Digital Bus Handbook. New York: McGraw-Hill, 1990.

[11] *W. Hilf:* Der VME-Bus – ein schnelles und modernes Bussystem. Bull. SEV/VSE 79(1988)1, S. 7–10.

[12] *F. Krähenbühl:* IBM Micro Channel Architecture – Wieso, Wozu? Bull. SEV/VSE 81(1990)17 und 21, S. 19 ff. und 29 ff., 82(1991)1 und 5, S. 37 ff. und 33 ff.

[13] *E. Stein:* Der Eisa-Bus – Ist er die Grundlage einer neuen PC-Generation? Bull. SEV/VSE 82(1991) 17, S. 15–20.

[14] Marktübersicht Buskoppler. VME-Bus 3(1989)5, S. 63.

[15] Which Buses and Systems? BIT3 Computer Corp.: Product Summary, Aug. 1992, S. 13.

[16] *P. Vukovic:* Bus-zu-Bus-Verbindungen. VME-Bus 3(1989)5, S. 62–66.

[17] *J. L. Hennessy, D. A. Patterson:* Computer Architecture – A Quantitative Approach. San Mateo: Morgan Kaufmann, 1990, S. 95.

Sept couches OSI sont souvent trop

Communications entre ordinateurs par coupleurs de bus, partie 2

L'importance des coupleurs de bus croît avec la croissance du nombre des différents systèmes de bus. Ces coupleurs permettent de coupler des ordinateurs pareils et différents côté matériel. Après que l'auteur, dans une première partie (cahier 3/93), avait traité de la communication entre ordinateurs dans une optique supérieure, il décrit dans cette deuxième partie la structure, les fonctions, le logiciel et les applications des coupleurs de bus, en mettant en avant de l'approche le principe de couplage Shared Memory, le plus important actuellement. L'auteur souligne que le coupleur matériel n'est pas le problème principal – pour la plupart des combinaisons de systèmes de bus il existe des coupleurs appropriés – mais le logiciel d'application parallèle.