

**Zeitschrift:** Bulletin des Schweizerischen Elektrotechnischen Vereins, des Verbandes Schweizerischer Elektrizitätsunternehmen = Bulletin de l'Association suisse des électriciens, de l'Association des entreprises électriques suisses

**Herausgeber:** Schweizerischer Elektrotechnischer Verein ; Verband Schweizerischer Elektrizitätsunternehmen

**Band:** 79 (1988)

**Heft:** 1

**Artikel:** Multibus-2 : eine System- und Softwarearchitektur

**Autor:** Furrer, F. J.

**DOI:** <https://doi.org/10.5169/seals-903965>

### **Nutzungsbedingungen**

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften auf E-Periodica. Sie besitzt keine Urheberrechte an den Zeitschriften und ist nicht verantwortlich für deren Inhalte. Die Rechte liegen in der Regel bei den Herausgebern beziehungsweise den externen Rechteinhabern. Das Veröffentlichen von Bildern in Print- und Online-Publikationen sowie auf Social Media-Kanälen oder Webseiten ist nur mit vorheriger Genehmigung der Rechteinhaber erlaubt. [Mehr erfahren](#)

### **Conditions d'utilisation**

L'ETH Library est le fournisseur des revues numérisées. Elle ne détient aucun droit d'auteur sur les revues et n'est pas responsable de leur contenu. En règle générale, les droits sont détenus par les éditeurs ou les détenteurs de droits externes. La reproduction d'images dans des publications imprimées ou en ligne ainsi que sur des canaux de médias sociaux ou des sites web n'est autorisée qu'avec l'accord préalable des détenteurs des droits. [En savoir plus](#)

### **Terms of use**

The ETH Library is the provider of the digitised journals. It does not own any copyrights to the journals and is not responsible for their content. The rights usually lie with the publishers or the external rights holders. Publishing images in print and online publications, as well as on social media channels or websites, is only permitted with the prior consent of the rights holders. [Find out more](#)

**Download PDF:** 26.01.2026

**ETH-Bibliothek Zürich, E-Periodica, <https://www.e-periodica.ch>**

# Multibus-2: eine System- und Softwarearchitektur

F.J. Furrer

**Multibus-2 ist ein Standard für symmetrische Mehrprozessorsysteme. Seine wichtigste neue Eigenschaft gegenüber klassischen Bus-Systemen ist das Message-Passing; der Bus wird nicht mehr zur Speicher- und Ein-/Ausgabeadressierung benutzt, sondern nur noch zum Austausch von Meldungen (Messages) zwischen intelligenten Karten. Das Message-Passing begründet eine neue Systemarchitektur mit klarer, modularer Hard- und Softwarestruktur. Diese macht den Einsatz des Multibus-2 in leistungsfähigen oder wachstumsfähigen Systemen besonders geeignet und wirtschaftlich.**

**Le Multibus-2 est un standard pour systèmes de multiprocesseurs symétriques. Sa nouvelle et importante propriété en comparaison des systèmes de bus classiques est le message-passing. Le bus n'est plus utilisé pour l'adressage de mémoires, d'entrées et de sorties, mais uniquement pour l'échange de messages entre cartes intelligentes. Le message-passing forme la base d'une nouvelle architecture de système, avec une structure claire et modulaire du matériel et du logiciel. De ce fait, l'emploi du Multibus-2 est approprié et économique pour des systèmes puissants et capables d'évoluer.**

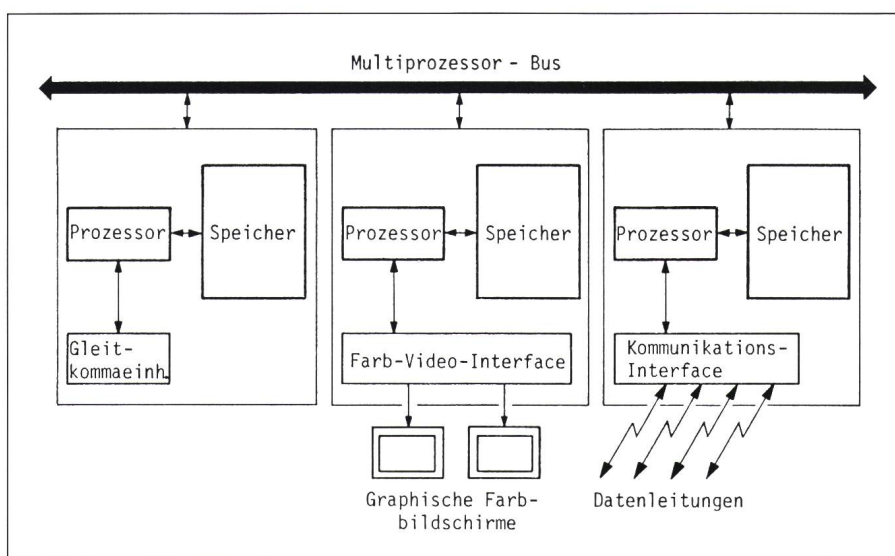
Ein Computerbus ist ein Bündel von Leitungen, welches (meist in der Form einer Backplane) der elektrischen Verbindung von Elektronikarten dient. Bei klassischen Bus-Strukturen (Q-, VME-, Uni-, ECB-Bus usw.) handelt es sich dabei um CPU-Karten, Speicherkarten, E/A-Karten usw. Der Bus wird als Datenübertragungspfad zwischen CPU und Speicher oder CPU und E/A-Registern benutzt. Der Multibus-2 geht von einer ganz anderen Voraussetzung aus: Alle Multibus-2-Karten sind intelligent, d.h. sie verfügen über einen eigenen Prozessor, einen eigenen (Instruktions- und Daten-)Speicher und sogar über eigene Eingabe-/Ausgabemöglichkeiten [1].

Jede dieser intelligenten Karten führt selbständig einen gewissen Satz von Funktionen aus: der Multibus-2 wird «nur» noch zum Austausch von Meldungen zwischen den intelligenten Karten benutzt. Er benimmt sich in diesem Sinne wie ein sehr schnelles,

paralleles *Lokalnetzwerk* (LAN); die einzelnen intelligenten Karten haben die Funktion von Stationen im Netzwerk (Fig. 1) [2].

## 1. Software-Architektur des Multibus-2

Die LAN-Struktur eines Multibus-2-Systemes hat gewaltige Konsequenzen auf die Software-Architektur; alle intelligenten Karten sind gleichberechtigt, führen ihre eigene Software (lokal) aus und kommunizieren miteinander über den Austausch von Meldungen (Message-Passing). Damit wird eine weitgehende Entkopplung der Softwareteile auf den einzelnen Karten erreicht, welches die Wiederverwertbarkeit und Modularität der Software leicht macht. Allerdings sind strenge Standards für die Prozeduren und Formate beim Meldungs-austausch notwendig; beim Multibus-2



**Figur 1 Hardwarestruktur des Multibus-2 (iPSB)**

Der Multibus-2 (iPSB) zeichnet sich durch seine lokalnetzwerkähnliche, symmetrische Kommunikationsstruktur aus, die Informationen zwischen intelligenten Mikroprozessorkarten vermittelt.

### Adresse des Autors

Frank J. Furrer, Dr.sc.nat., Syslogic Datentechnik AG, 8953 Dietikon.

wurde konsequent die *ISO-Struktur* (7-Schichten-Modell) angewandt [3].

## 2. Hardware-Grundlage des Multibus-2

Der Anwender sieht den Multibus-2 eigentlich nicht. Dieser ist für ihn ein zuverlässiges, schnelles und symmetrisches Transportmittel zwischen den Software-Prozessen. Trotzdem seien hier seine technischen Daten aufgeführt. Der Multibus-2 besteht eigentlich aus zwei Teilen:

- dem parallelen Systembus (iPSB = **Parallel System Bus**),
- dem seriellen Systembus (iSSB = **Serial System Bus**).

Der *parallele Systembus* (iPSB) besteht aus 96 Leitungen, welche auf einem einzigen DIN-Stecker untergebracht sind. Der iPSB ist ein echter 32-Bit-Bus (Daten und Adressen). Er ist synchron mit 10 MHz getaktet, was eine maximale Datenrate von 32 Mbyte ergibt. Als Zugriffsprotokoll für den Bus wird eine faire Arbitrierung<sup>1</sup> verwendet, und jeder Buszyklus wird arbitriert! Jeweils 8 bit (Daten-, Adress- und Kontrolleleitungen) sind mit Parität versehen, und alle Buszyklen sind mit Time-Outs überwacht. Der PSB verfügt dadurch über eine ausserordentlich hohe Betriebszuverlässigkeit, was ihn speziell für industrielle Anwendungen geeignet macht. Der iPSB kann bis zu 20 Karten-Steckplätze aufweisen. Multibus-2-Karten entsprechen dem Doppelseuropakartenformat (240 mm × 220 mm) oder Einfacheuropakartenformat (100 mm × 220 mm). Speziell das Einfacheuropakartenformat wird in der Zukunft die Implementation von sehr leistungsfähigen, kompakten 32-bit-Systemen ermöglichen.

Der *serielle Systembus* (iSSB) besteht aus 2 Leitungen; er erlaubt die Fortsetzung des Message-Passing über mehrere Systeme hinweg. Er stellt kein Lokalnezwerk dar, sondern nur eine standardisierte Möglichkeit, mehrere (maximal 32) Multibus-2-Systeme zu verbinden.

<sup>1</sup> Alle Karten können auf den iPSB zugreifen. Um Kollisionen zu vermeiden, muss ein Schiedsrichtermechanismus (Arbitrierung) vorhanden sein, welcher die Zugriffsreihenfolge regelt. Eine faire Arbitrierung garantiert jeder Karte einen Zugriff innerhalb einer bestimmten Zeit.

## 3. Message-Passing als Grundlage einer verteilten Rechnerarchitektur

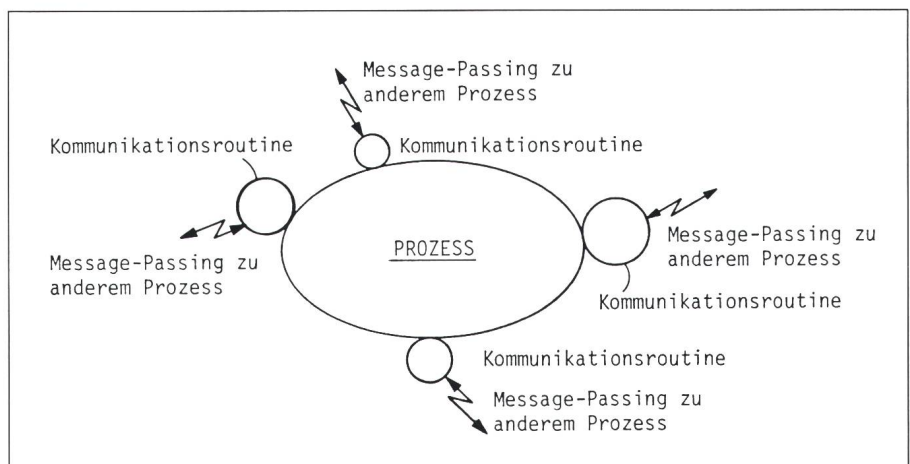
### Software-Prozesse

In der modernen Softwaretechnologie wird der Begriff Prozess verwendet. Ein Prozess ist ein selbständiges Stück Software, welches eine genau definierte Funktion ausführt. Prozesse müssen miteinander kommunizieren und untereinander synchronisiert werden. Zu diesem Zwecke ist eine Kommunikation zwischen den einzelnen Prozessen notwendig. Eine (durch den Multibus-2 standardisierte) Möglichkeit einer solchen Kommunikation ist das Message-Passing. Die Meldungen können dem Datenaustausch (Data

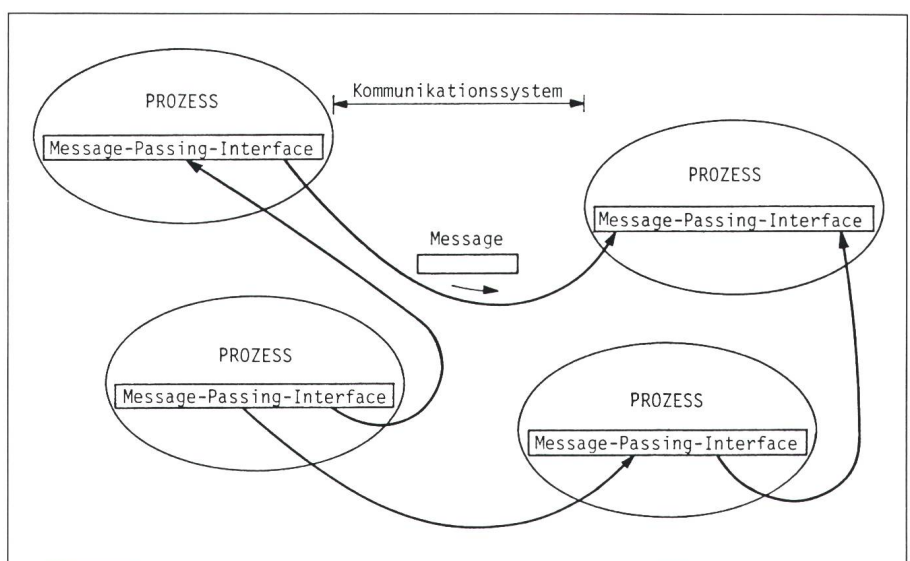
Message) oder der Synchronisation (Interrupt Message) dienen [4]. Ein Prozess kann damit symbolisch dargestellt werden (Fig. 2).

### Verteilte Rechnerarchitektur

Das Message-Passing ist eine Software-Architektur. Dem Anwendungsprogrammierer werden die Kommunikationsroutinen zur Verfügung gestellt, die Kommunikation selbst wird durch die Hardware – unsichtbar für den Programmierer und fehlerfrei durchgeführt. Verschiedene Prozesse können auf dem gleichen Prozessor (Multitasking) oder auf verschiedenen Prozessoren (Multiprocessing) beheimatet sein. Falls sich an der Art der Kommunikation in beiden Fällen



Figur 2 Ein Softwareprozess kommuniziert über Message-Passing mit anderen Prozessen



Figur 3 Verteilte Rechnerarchitektur

Die verschiedenen Softwareprozesse sind über Message-Passing gekoppelt. Sie können auf dem gleichen Prozessor (Multitasking) oder auf verschiedenen Prozessoren (Multiprocessing) ablaufen.



nichts ändert, so hat man ein echtes, ideales Message-Passing und eine weitgehend hardwareunabhängige Software-Architektur.

Das Message-Passing kann dabei über ganz verschiedene Kommunikationswege stattfinden: über den iPSB (mit 32 Mbyte/s), über eine serielle Datenleitung (mit 9600 bit/s), über ein LAN (mit 10 Mbit/s) usw. Die einzelnen Prozesse sind voneinander über das Message-Passing isoliert, und die Rechnerarchitektur wird zu einer *lose gekoppelten Struktur* (Fig. 3).

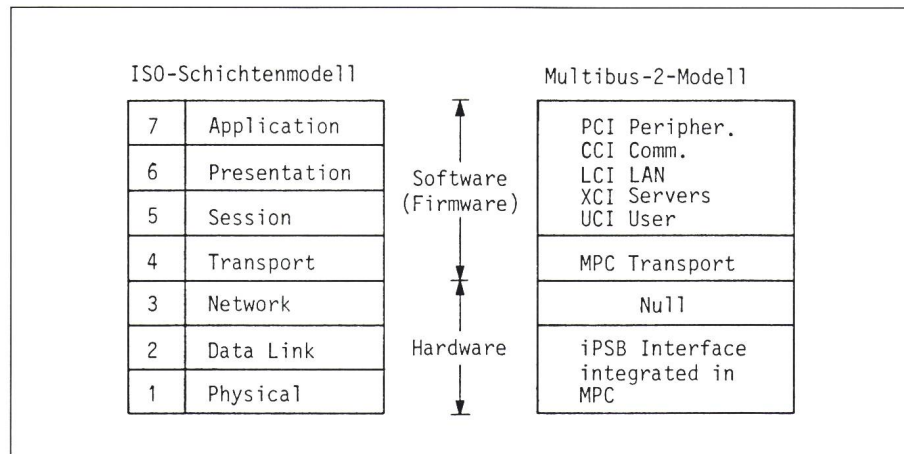
### ISO-Schichtenmodell

Falls sich eine solche Software-Architektur durchsetzen soll, so muss sie hochgradig standardisiert werden; alle Anwender müssen sich strikt an den definierten Standard halten, da sonst die Kommunikation zwischen Prozessen von verschiedenen Anwendern nicht möglich ist. Eine der grossen Leistungen der Multibus-2-Definition ist die Standardisierung des Message-Passing. Der Multibus-2-Standard stützt sich auf das 7-Lagen-ISO-Schichtenmodell (Fig. 4) ab. Die untersten drei Schichten sind beim iPSB in einer speziellen, integrierten Schaltung, dem Message-Passing Coprocessor (MPC) realisiert. Diese 77 000-Transistor-Schaltung stellt automatisch die Kompatibilität aller Multibus-2-Karten untereinander (auch von verschiedenen Herstellern) sicher. Der MPC-Transport ist ein eigener Standard, welcher exakt die Funktionen für die Kompatibilität von Multibus-2-Karten untereinander definiert. Die höheren Schichten (Session, Presentation, Application) werden in Abschnitt 5 beschrieben.

## 4. Systemstruktur von Multibus-2-Systemen

### Funktionsaufteilung

In klassischen Bus-Systemen wurden CPU-Karten, Speicherkarten, Ein-/Ausgabekarten usw. miteinander kombiniert, bis die gewünschte Hardware-Funktionalität und Leistungsfähigkeit erreicht war. Auf dem so zusammenkonfigurierten System wurde dann eine Software entwickelt! Beim Multibus-2 verläuft der Entwurfszyklus umgekehrt: Man geht von der gewünschten Funktionalität aus, definiert die dazu notwendigen Software-Prozesse und verteilt darauf die Software-Prozesse auf eine Anzahl Multi-



Figur 4 Message-Passing Softwarestruktur im Multibus-2

MPC Message-Passing Coprocessor  
iPSB Parallel System Bus  
Null Nicht vorhanden  
PCI Peripheral Communications Interface  
CCI Communications Interface

LCI Local Area Communications  
Interface  
XCI X-Interface (symbolisch für weitere  
Interfaces)  
UCI User Communications Interface

bus-2-Funktionsmodule. Ein Funktionsblock umfasst z.B. sämtliche Massenspeicher-Verwaltungsfunktionen; man benützt dazu einen Disk-Server (oder sogar File-Server, vgl. Abschnitt 5). Die Funktionen des Disk-Servers stehen nun allen anderen Prozessen zur Verfügung. Diese fordern Disk-Blöcke<sup>2</sup> über Message-Passing an und erhalten sie über Message-Passing zugeteilt. Mutierte oder neue Disk-Blöcke werden wieder über Message-Passing an den Disk-Server zur Speicherung zurückgegeben. Neben diesen Funktionen führt der Disk-Server selbstständig alle Save-Funktionen (Kopieren auf Streamer-Tape oder Mirror-Disk) aus.

Multibus-2-Systeme bestehen daher aus Funktionsmodulen wie Disk-Servern, Kommunikationsservern, LAN-Servern, Drucker-Servern usw. sowie Applikationsmodulen. Alle diese Funktionsmodule sind untereinander lose über das standardisierte Message-Passing gekoppelt. Falls z.B. vom Disk-Server mehr Leistung verlangt wird, so kann der Disk-Server gegen eine leistungsfähigere Version (schnellerer oder grösserer Speicher) ausgetauscht werden, ohne dass an den anderen Funktionsmodulen das geringste geändert werden muss. Multibus-2-Architektur wird in diesem Sinne zu einem modularen «Systembaukasten».

### Physische Struktur

Multibus-2-Karten (Funktionsmodule) sind Elektronikarten in Europaformat. Sie besitzen als Bus-Stecker einen 96poligen DIN-Stecker (P1), auf welchem alle 32 Daten- und Adressleitungen sowie alle notwendigen Kontroll- und Stromversorgungsleitungen untergebracht sind. Im Doppelpaformat (240×220 mm) steht der zweite DIN-Stecker (P2) für private Zwecke zur Verfügung. Die iPSB Backplane verbindet alle P2-Stecker der Multibus-2-Karten und kann bis zu 20 Steckplätze aufweisen.

Die Ein- und Ausgabe von Signalen zur Umwelt findet über die Frontplatte der einzelnen Karten statt. Bei Karten mit vielen Signalen zur Peripherie müssen hochdichte, zuverlässige Stecker verwendet werden, welche dann auf einem externen «Distribution-Panel» auf Standard-Stecker ausgefächert werden.

### Selbsttest und Selbstkonfiguration

Neben dem Message-Passing verfügt der Multibus-2 noch über zwei weitere, interessante Eigenschaften:

<sup>2</sup> Ein Disk-Block ist die Organisationseinheit der Daten auf dem Massenspeicher (z.B. 256 Bytes). Der Verkehr zwischen Massenspeicher und Prozessor findet in derartigen Disk-Blöcken statt.



- dem BIST (= Built-In Self-Test) und
- dem Interconnect.

**BIST** ist eine sehr weit getriebene Selbst-Testmöglichkeit von Multibus-2-Systemen. Bei jedem Hochfahren des Systemes führt jede Multibus-2-Karte den auf der Karte eingebauten Selbsttest (mit Fehlerdiagnose) aus. Die Resultate aller Selbsttests werden einem zentralen Funktionsmodul gemeldet, welches dann die fehlerfreie Betriebsbereitschaft des Gesamtsystemes (oder eben einen diagnostizierten Fehler) an den Benutzer ausgibt. Durch diese Eigenschaft werden Multibus-2-Systeme, welche sehr komplexe Gebilde sein können, ausserordentlich wartungsfreundlich. BIST und System-Confidence Test (SCT) lassen sich auch von ferne (Remote, über Modem) ausführen, was eine Fern-diagnose ermöglicht.

Dem **Interconnect** liegt die Idee zugrunde, sämtliche Konfigurationsprobleme aus der Hardware in die Software zu verlagern. Die Konfigurationssteckbrücken für Adressen, Optionen usw. sind von den Multibus-2-Karten verschwunden bzw. durch software-setzbare Register ersetzt worden. Die Konfigurationsinformation befindet sich in der Software eines ausgewählten Funktionsmodules und die Konfiguration wird beim Hochfahren des Systemes jedem einzelnen Funktionsmodul (d.h. jeder einzelnen Karte) übermittelt. Dies ist ein enormer Vorteil, insbesondere für die Systemwartung, können doch Karten aus der Verpackung direkt in das System gesteckt werden. Die Konfiguration der Karten wird von der Interconnect-Software ausgeführt.

## 5. Servers und Clients

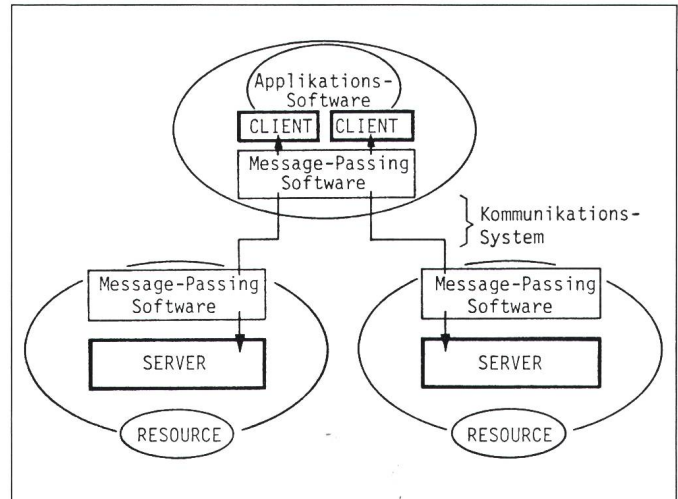
Multibus-2-Systeme benutzen die Client-Server-Software-Architektur: Das Message-Passing ermöglicht den Austausch von standardisierten Meldungen (Daten- und Synchronisationsmeldungen) zwischen den Softwareprozessen. Die Client-Server-Architektur beruht auf dem 7schichtigen ISO-Modell (Fig. 4) und beinhaltet die oberen 3 Schichten (Session, Representation und Applikation).

### Servers

Gewisse Multibus-2-Funktionsmodule arbeiten als Server für andere

**Figur 5**  
**Client-Server-Beziehungen in einem Multibus-2-System**

Clients greifen in standardisierter Form auf die Ressourcen (Disk, LAN, Kommunikation usw.) zu, welche von Servern verwaltet werden.



Funktionsmodule (Fig. 5). Der Server verwaltet eine (oder mehrere) Ressourcen und stellt seine Dienste mehreren Clients zur Verfügung. Das Protokoll zur Benützung dieser Dienste ist exakt definiert; mit jedem Server wird eine Betriebsanleitung für die genormten, höheren Protokollschichten mitgeliefert [5].

### Server-Funktionsmodule

Die Funktionen von typischen Servern sind im folgenden kurz beschrieben:

**Disk-Server** stellen den Clients den Betrieb und die Verwaltung von Massenspeichern (Hard-Disk, Optical Disk, Floppy-Disk, Streamer-Tape usw.) zur Verfügung. Sie führen selbstständig Daten-Rettung durch (Streamer-Tape oder Mirror-Disk).

**File-Server** enthalten ganze File-Systeme (betriebssystemunabhängig) und stellen den Clients Files oder Ausschnitte aus Files (inkl. Directory-Verwaltung) zur Verfügung.

**Kommunikations-Server** unterhalten selbstständig synchrone Kommunikationsverbindungen (X.25, SNA-3270, LU6.2 usw.) zu Grossrechnern und stellen den Clients logische Links zur Verfügung. Sie führen selbstständig die Fehlerbehandlung der Kommunikationsleitungen durch.

**Terminal-Server** betreiben Clusters von Terminalen mit verschiedenen Bildschirmemulationen. Sie stellen den Clients ein einheitliches, «virtuelles Terminal» in n-facher Ausführung zur Verfügung.

Server-Funktionsmodule für den Multibus-2 können heute als fertige

Subsysteme gekauft werden. Der Benutzer kauft damit nicht mehr Hardware und Software getrennt ein, sondern er kauft ein lauffähiges, integriertes Funktionsmodul mit einer vollständig standardisierten Hard- und Software-Schnittstelle.

### Clients

Die Clients führen die eigentlichen Applikationsfunktionen durch, sie beanspruchen die Dienste der Servers (z.B. Massenspeicherdaten, Kommunikationsverbindungen, Terminaldialoge usw.) und verarbeiten die Daten. Clients sind häufig spezielle Applikationsprozessoren.

## 6. Applikationsprozessoren

Die Applikationsprozessoren führen die eigentlichen Applikationsfunktionen durch. Die Applikationssoftware ist aufgeteilt in Applikationsprozesse; die einzelnen Applikationsprozesse beanspruchen die Dienste der Servers und benehmen sich als Clients. Die Servers werden fertig gekauft, und die Arbeit des Multibus-2-Programmierers konzentriert sich auf die Programmierung der Applikationsprozessoren (Clients) [6]. Weil die einzelnen Applikationsprozessoren durch das Message-Passing voneinander isoliert sind (sehr lose, standardisierte Kopplung), können auf den verschiedenen Applikationsprozessoren verschiedene Betriebssysteme benützt werden. Der Multibus-2 bietet damit dem Anwender eine echte Betriebssystem-Unabhängigkeit, d.h. für jeden Applikationsprozessor kann das bestgeeignete

Betriebssystem benutzt werden. Im gleichen Gehäuse (d.h. in der gleichen Backplane) können dadurch die verschiedensten Betriebssysteme (und auch Mikroprozessoren) friedlich zusammenarbeiten, wie z.B.:

- *Echtzeitbetriebssysteme* (z.B. INTEL iRMX, iRMK) für die Echtzeit- und Prozesssteuerungsanwendungen,
- *UNIX, XENIX, DOS* auf Applikationsprozessoren, welche von der grossen, bestehenden Softwarebasis Gebrauch machen wollen,
- *eigene Kernels* bei speziellen Aufgaben (SPS-Emulationen, MODULA-2 Executive usw.).

Die Servers (Disk-Server, Kommunikationsserver, LAN-Server) stellen ihre Dienste sämtlichen Applikationsprozessen (Clients) zur Verfügung. Die Verschiedenheiten der Betriebssysteme auf den Applikationsprozessoren

werden durch das Message-Passing-Konzept unsichtbar.

## 7. Ausblick

Die Multibus-2-Architektur erlaubt die Implementation moderner, modularer System- und Softwarearchitekturen, welche auf den Client/Server-Konzepten von symmetrischen Multiprozessorsystemen basieren. Die darauf beruhende Softwaretechnologie bringt enorme Vorteile in bezug auf Unabhängigkeit von Prozessoren und Betriebssystemen, Modularität und Wiederverwertbarkeit von Programnteilen und für die Systemwartung. Zudem wird die durch den Multibus-2 erzwungene Standardisierung der Hardware- und Softwareschnittstellen eine Vereinheitlichung der Arbeitsweise der Hersteller und Software-Lieferanten bewirken. In diesem Sinne

macht der Multibus-2 einen wichtigen Schritt hin zur Rationalisierung der Softwareerstellung.

## Literatur

- [1] Multibus-2 architecture specification handbook. Version C (order-code 146077C). Santa Clara/California, Intel-Corporation, 1985.
- [2] F. J. Furrer: Moderne Mikroprozessorbuse - der Multibus-2. Syslogik-Mitteilungen -(1986)3, S. 10...22.
- [3] Multibus-2 transport protocol specification. Version 1 (order-code 149247). Santa Clara/California, Intel-Corporation, 1986.
- [4] F. Huguenin: Multibus II - «Message-Passing» - Software. Elektroniker 26(1987)9, S. 93...98.
- [5] Multibus-2 peripheral communications architecture specification. Version 2.0 (order-code 135173). Santa Clara/California, Intel-Corporation, 1986.
- [6] F. J. Furrer: Struktur und Anwendung von Multibus-II-Systemen. Techn. Rdsch. 79(1987)36, S. 158...161.