

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:	81 (1990)
Heft:	21
Artikel:	IBM Micro Channel Architecture : wieso, wozu : Teil 2 : Alternativen in Adapterbau
Autor:	Krähenbühl, Fred W.
DOI:	https://doi.org/10.5169/seals-903179

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: 14.02.2026

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

IBM Micro Channel Architecture – Wieso, Wozu

Teil 2: Alternativen in Adapterbau

Fred W. Krähenbühl

Im ersten Teil dieser Reihe wurden die grundsätzlichen Fragen nach Sinn und Zweck von Bussystemen, die Frage, weshalb ein neues Buskonzept im PC-Bereich überhaupt notwendig ist, sowie die wesentlichen Konzepte der IBM Micro Channel Architektur behandelt. Der vorliegende zweite Teil beschreibt Adapter-Konzeptionen für Micro Channel Maschinen, die sich hauptsächlich durch die Höhe der Adapter-Kompetenz und -Selbständigkeit voneinander unterscheiden.

Le premier article de cette série était consacré à des questions générales telles que la finalité des systèmes à bus, la nécessité d'un nouveau type de bus pour la micro-informatique ou encore les notions de base de l'architecture Micro Channel d'IBM. Le deuxième article, publié dans le présent numéro, traite des différents types d'adaptateurs utilisables sur des machines Micro Channel, que l'on peut différencier avant tout par des critères comme le niveau d'intelligence ou l'autonomie par rapport au processeur central.

Nachdem im ersten Teil die grundlegenden Eigenschaften des IBM Micro Channels aufgezeigt wurden, soll im zweiten Teil etwas näher auf den Adapterbau, d.h. auf das Design von Elektronik-Schaltungen, die dem Informationsaustausch zwischen dem System und der Außenwelt dienen, eingegangen werden. Dem Entwickler stehen im Adapterbau verschiedene Wege offen, je nach dem Mass, in dem er seinen Adapter der direkten Kontrolle durch den Systemprozessor unterstellen will.

Programmgesteuerte Ein-/Ausgabe

Der berühmte Spiel-Adapter des IBM PC zum Beispiel steht völlig unter Programmkontrolle; er tut nur dann etwas für den Benutzer, wenn das Programm ihn dazu auffordert. Das bedeutet, dass der Hauptprozessor immer wieder beim Adapter nachfragen muss, ob eine Meldung (z.B. die Bewegung des Steuerknüppels) vorliegt, die eine Aktion verlangt (Bild 15). Je höher die Anforderungen an die Multiprogramming- und Multitasking-Möglichkeiten unter Betriebssystemen wie IBM OS/2 sind, desto unabhängiger vom Betriebssystem müssen die Adapter arbeiten können. Der

Hauptprozessor in Bild 16 braucht den Adapter nicht dauernd abzufragen; er steht damit solange für andere Arbeiten zur Verfügung bis ein Interrupt von ihm eine Aktion verlangt, sei es weil irgendwo ein Schalter betätigt wurde, sei es weil ein voller Zwischen Speicher für Eingangsdaten nach dringender Bearbeitung ruft. Zu beachten bleibt jedoch, dass in diesem Beispiel die Abarbeitung der Interrupt-Service-Routine nach wie vor die Aufgabe des zentralen Prozessors ist, ohne den bei diesem Design keine Datenübertragung über den Kanal möglich ist.

DMA – Direct Memory Access

Der Direkte Speicherzugriff (Direct Memory Access, DMA, Bild 17) erlaubt, den Datentransfer von der Außenwelt über einen Bus zum Hauptspeicher und umgekehrt durch völliges Ausschalten des Prozessors wesentlich rascher zu gestalten. Der Ablauf ist der folgende: Initialisiert ein Applikationsprogramm einen Datentransfer, so erhält der betroffene Adapter einen entsprechenden Befehl. Der Adapter muss sich nun selbst um den Buszutritt bewerben. Hat er ihn gewonnen (er hat das ohne Hilfe eines anderen Systemteils herauszufinden), wird der DMA-

Bild 15
Programmgesteuerte Ein-/Ausgabe

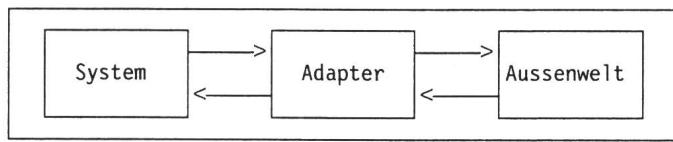
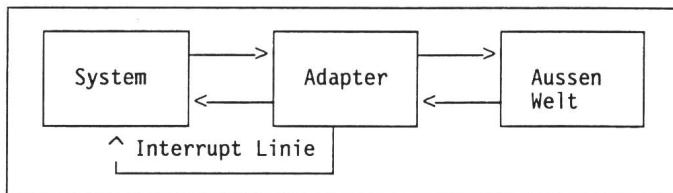


Bild 16
Programmgesteuerte Ein-/Ausgabe mit Interruptmöglichkeit



Adresse des Autors

Fred W. Krähenbühl, IBM Schweiz AG, General Guisan-Quai 26, 8022 Zürich

Controller des Systems den Transfer als Bus Master durchführen. Ein solcher DMA-Adapter bleibt also ein Sklave, da während des eigentlichen Transfers der DMA Controller des Systems als Bus Master arbeitet. Beim Adapterbau können auf diese Weise Kosten gespart werden, vorausgesetzt dass die erreichbare Geschwindigkeit für die Anwendung ausreicht.

Normalerweise werden ein oder zwei Bytes parallel übertragen. Bei einem Transfer vom Speicher in einen Ausgabe-Adapter werden die Daten in einer ersten Phase vom Speicher in den Controller übertragen; ein zweiter Transfer bringt die Daten vom Controller zum Ausgabe-Adapter. Man nennt dies «Serielle DMA». Während die IBM Micro Channel-Architektur eine parallele Datenübertragung von bis zu 64 Bit definiert, können bei den heute erhältlichen PS/2 über den DMA Controller nur 16 Bit gleichzeitig transferiert werden. Die Implementation in einzelnen Systemen ist für den Adapter wie auch für die unterstützende Software völlig transparent, weil über Protokolle die Transfer-Art zwischen den Partnern (hier Speicher – DMA Controller einerseits, DMA Controller – Ausgabe-Adapter andererseits) bei jeder Übertragung ausgetauscht wird.

Die möglichen Datenübertragungsarten wurden bereits im ersten Teil ausführlich behandelt.

Bus Master

Um den Datendurchsatz über den Kanal weiter zu beschleunigen, ohne die Taktgeschwindigkeit des Systems als ganzes, oder der Transferpartner allein erhöhen zu müssen, sind soweit wie irgendwie möglich organisatorische Operationen zu vermeiden (Bild 18). Als Beispiel mögen die Extra-Zyklen dienen, die bei DMA über den systemeigenen Controller nötig sind. Das kann erreicht werden, wenn der Adapter selbstständig alle nötigen Impulse für eine Kanaloperation generiert. Für den Partner muss völlig transparent sein, ob der Transfer durch den Prozessor, den DMA Controller oder einen Bus Master auf dem Kanal initialisiert wird.

Da der Transfer eines oder mehrerer (Burst-Mode) Datenworte ohne Hilfe von aussen stattfindet, haben die beiden Partner-Adapter unter sich den Übertragungsmodus auszuhandeln: Arbeiten beide Adapter mit derselben

Datenbreite? Der Bus Master mag mit 32 Bit arbeiten, der Partner kann aber vielleicht nur 16 Bit parallel akzeptieren. Kann der Empfänger mit der «Streaming Data Prozedur» arbeiten, also z.B. eine Übertragungsrate von 40 Mbyte/s akzeptieren? Ist einer der Partner langsamer als der andere? Kann ein Fehler während der Übertragung erkannt und gemeldet werden? Dies ist die Art und Weise, wie zwei Adapter während der Übertragung des ersten Zeichens das Vorgehen für den weiteren Verlauf des Transfers vereinbaren. Das sind Micro Channel-Definitionen, welche in allen PS/2-Maschinen (Modell 50 und höher) implementiert sind. Ausnahmen sind die Streaming Data-Prozedur und die Paritätsprüfung, welche erst bei den neueren Systemen zum Tragen kommen.

Einen Adapter zu entwerfen, der alle diese Aufgaben erfüllt, ist eine ausgesprochen komplexe Aufgabe. Es ist denn auch nicht weiter verwunderlich, dass in den ersten Jahren der

Software-Aspekte

Die Art und Weise, wie Adapter im Ein-/Ausgabebereich anzusprechen sind, variiert mit dem Adaptertyp und – mehr noch – mit dem verwendeten Betriebssystem. IBM DOS ist, wie weiter oben dargelegt, ein Single User-Single Tasking-Betriebssystem. Das bedeutet, dass im Normalfall nur die eine, den Adapter benützende Applikation im Hauptspeicher resident ist. Das bedeutet auch, dass das Anwendungsprogramm selbst auf den Adapter direkt zugreifen darf und kann. Das kann mit einfachen IN- und OUT-Befehlen geschehen.

Im Bereich von Multitasking sieht die Aufgabenstellung völlig anders aus. Unser Adapter – als Beispiel diene ein Adapter, der eine Harddisk bedient – steht einer Vielzahl von Programmen abwechselnd zur Verfügung. Jede Anwendung wird den Zugriff zu ihren eigenen Daten über diesen einen Adapter verlangen und auch erhalten.

Bild 17
DMA – Direct
Memory Access
(Direkter
Speicher-Zugriff)

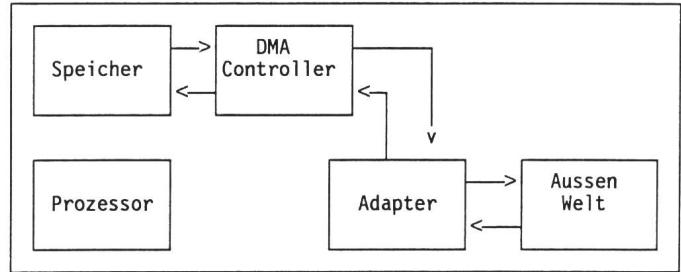
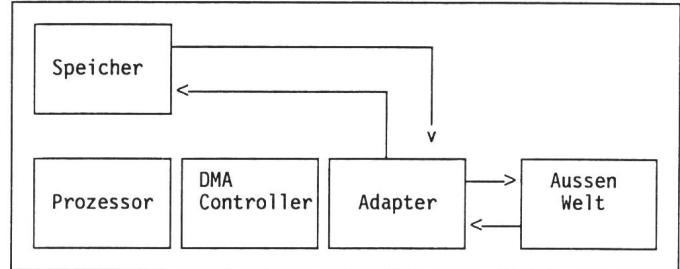


Bild 18
Bus Master



PS/2-Systeme nur wenige Bus Master auf den Markt gekommen sind. Nun haben zwei Firmen (Intel und Chips and Technologies) integrierte Schaltungen angekündigt, welche alle diese Funktionen beinhalten. Das erlaubt auch manch kleinerem Hersteller, an den Bau von Bus Master-Adaptoren zu denken.

Für die Datenübertragung über den Kanal eines Multitasking-Systems kommen vor allem die Typen DMA- und Bus Master-Adapter in Frage, da nur sie den Hauptprozessor während der Dauer der Datenübertragung freistellen und damit für andere Aufgaben verfügbar machen. In beiden Fällen erhält der Adapter, das Subsystem

Hauptsystem

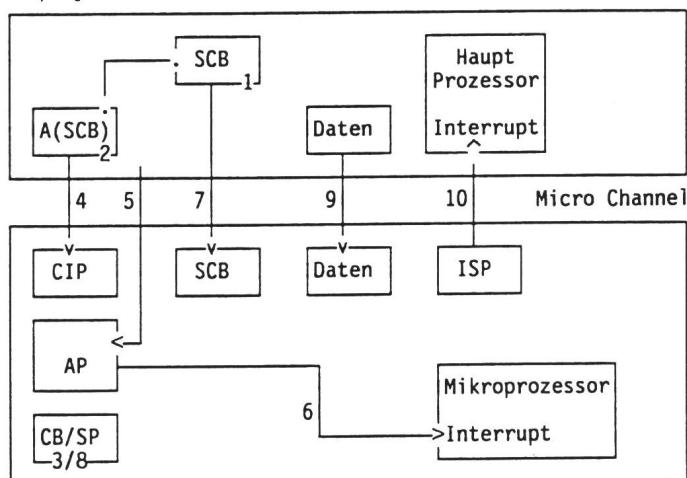


Bild 19
Steuerung einer
Übertragung unter
Subsystem Control
Block Architecture
(SCBA)

des zu übermittelnden Datenblockes und Operationscode gespeichert.

2. Die Adresse des SCB wird in einem bestimmten Bereich des Hauptspeichers festgehalten.

3. Der Device Driver erhält von der Anwendung den Befehl, die SCB-Adresse an den Command Interface Port (CIP) des Adapters zu übertragen. Um Kollisionen mit Transfers anderer (asynchron laufender) Tasks zu vermeiden, fragt er zuerst den Command Busy/Status Port (CB/SP) des Adapters nach dem Status ab, um herauszufinden, ob eine durch das Subsystem noch nicht verarbeitete Adresse im CIP pendent ist. Ist das nicht der Fall, wird im CB/SP ein Bit (Busy Bit) gesetzt, und damit der Zugriff für andere Tasks gesperrt.

4. Nun schreibt der Device Driver mittels einer normalen Ausgabe-Instruktion die SCB-Adresse in den Command Interface Port des Adapters.

5. Durch Ansprechen des Attention Ports (AP) mittels einer weiteren Ausgabe-Instruktion wird die Operation initialisiert. Von diesem Moment an hat der Hauptprozessor, was diesen Transfer betrifft, keine Aufgabe mehr. Er wendet sich anderen Tasks zu. Der Adapter erledigt den ganzen Transfer selbstständig.

6. Die Initialisierung hat zur Folge, dass der Mikroprozessor des Adapters einen Interrupt Request erhält. Sein Programm verzweigt in die Service-Routine, welche feststellt, dass der Grund für diesen Interrupt im Vorhandensein einer SCB-Adresse im CIP liegt, dass also ein Übertragungs-Auftrag vorliegt.

7. Der Adapter bewirbt sich jetzt um den Kanal. Hat er den Wettbewerb mittels seiner Priorität gewonnen, überträgt er selbst den SCB in einen bereitgestellten Speicherbereich auf dem Adapter. Dabei arbeitet das Interface – hier ein Intel 82325 – als Bus Master sowohl auf dem Micro Channel als auch gegenüber dem Mikroprozessor auf dem Adapter. Die Datenübertragung findet also ohne Intervention, weder durch den Haupt- noch den Adapter-Prozessor, statt.

8. In diesem Moment kann der CB/SP wieder freigegeben werden, da die im CIP vorhandene SCB-Adresse nicht mehr gebraucht wird und überschrieben werden darf.

9. Anhand des SCB-Inhaltes kann das Programm des Subsystems den verlangten Transfer erkennen, die Para-

Subsystem

SCB	Subsystem Control Block
SCBA	Subsystem Control Block Architecture
A(SCB)	Adresse des Subsystem Control Blocks im Hauptspeicher
CIP	Command Interface Port
AP	Attention Port
CB/SP	Command Busy/Status Port
ISP	Interrupt Status Port

also, eine Aufgabe, die er als Bus Master selbstständig im Multiprozessor-Modus oder als DMA-Sklave mit Hilfe des DMA Controllers durchführt und anschliessend über eine Interrupt-Leitung dem Hauptprozessor als beendet meldet.

Bei OS/2 ist – wie bei jedem Multitasking Betriebssystem – ein Device Driver das programmtechnische Verbindungsglied zwischen dem Anwendungsprogramm und dem Adapter. Im Idealfall liefert der Hersteller des Adapters den für sein Gerät entwickelten Device Driver gleich mit. Das von ihm entwickelte Protokoll regelt den Verkehr zwischen der Anwendung und diesem Device Driver. Jeder Hersteller muss dieses Protokoll von neuem entwickeln, es sei denn, eine Organisation nehme sich dieser Aufgabe an. Für die Bus-Master-Adapter hat IBM diese Arbeit geleistet. Die entsprechenden Definitionen sind in der «Subsystem Control Block Architecture – SCBA» enthalten. Sie regeln den Verkehr zwischen dem Device Driver unter OS/2 und den Bus Master-Adaptoren. Die Implementation erfolgt, indem man die neue Hardware entsprechend diesen Definitionen entwickelt. Auf einer solchen Basis ist das Programmieren des Device Drivers verhältnismässig einfach.

Die Chip Sets von Intel und Chips and Technologies, welche die Herstellung von Bus Master Adaptoren unterstützen, haben die nötige Hardware bereits integriert. Es sind Register, welche teils vom Trägersystem, teils vom Prozessor auf dem Adapter aus, teils von beiden her zugänglich sind. Damit ist erstens eine einfache Kommunikation zwischen dem Hauptsystem und dem Subsystem gewährleistet und kann zweitens das Ansprechen der Register direkt eine entsprechende Funktion des Adapters auslösen.

Anhand der Darstellung in Bild 19 kann der Ablauf einer Übertragung aufgezeigt werden. Dieser Darstellung liegt die Arbeitsweise des Chip Sets Intel 82325 zugrunde, einer Schaltung, die hauptsächlich auf Bus Master Adaptoren mit hoher eigener Intelligenz zu finden sein wird. Die Darstellung ist idealisiert (Fehlererkennung und -korrektur nicht berücksichtigt).

Der Ablauf erfolgt in folgenden Schritten, wobei alle erwähnten Register über normale Ein-/Ausgabe-Befehle ansprechbar sind:

1. Im Subsystem Control Block (SCB) des Hauptsystems, dessen Inhalt und Länge in der Subsystem Control Block Architecture (SCBA) definiert ist, werden die Parameter des durchzuführenden Transfers wie Basisadresse, Länge

Nicht belegt	Interrupt Level (INTL)	I/O Level (IOA)	ENABLE /-SLEEP
Bits 3 - 7	Bit 2	Bit 1	Bit 0

ENABLE/-SLEEP: Dieses Bit ist durch die Micro Channel-Architektur vorgeschrieben und muss in jedem Adapter an dieser Stelle implementiert werden. Seine Bedeutung: Bei 0 ist der Adapter ausser Betrieb, bei 1 ist er aktiviert.

I/O Level (IOA):0 bezeichnet den niedrigen Level, also die Benutzung der Adressen 300H/301H, 1 den hohen Level, also die Adressen 308H/309H.

Interrupt Level (INTL): 0 bezeichnet den niedrigen Level 7, 1 den höheren Level 12.

nicht belegt	Unterbrechg. pendent	nicht belegt	Schalter		Relais	
			Kanal 2	Kanal 1	Kanal 2	Kanal 1
Bit 7	Bit 6	Bit 5/4	Bit 3	Bit 2	Bit 1	Bit 0

Bit 6: Dieses Bit wird durch den Adapter selbständig gesetzt, sobald er auf Grund externer Einflüsse einen Interrupt verlangen muss.

Bits 3/2: Diese Bits repräsentieren jederzeit den Zustand der Schalter.

Bits 1/0: Diese Bits repräsentieren jederzeit den Zustand der Relais.

meter in den Registern des Intel 82325 speichern und damit den Transfer initialisieren. Die integrierte Schaltung wird den Auftrag nun in einem oder mehreren Transfers im Byte- oder im Burst-Modus durchführen. Der Intel 82325 ist in der Lage, jede Datenmenge bis zu einem Maximum von 16 MByte ohne Intervention eines der beiden Prozessoren selbständig zu übertragen.

10. Sobald die letzte Teil-Übertragung beendet ist, wird über den Micro Channel ein Interrupt signalisiert, damit der Device Driver dem Anwendungsprogramm das Ende der Übertragung melden kann.

Diese Definitionen einerseits und deren Implementation in Hardware andererseits erlauben, Bus Master-Adapter mit einem Minimum an Aufwand herauszubringen. Die Unterstützung dieser Adapter durch Software (Device Driver) und die Definition der Subsystem Control Block Architecture haben den Aufwand zusätzlich entscheidend verringert.

Ein konkreter Ein-/Ausgabe-Adapter für digitale Signale

Kurz nach der Ankündigung der PS/2 wurde von uns der Versuch unternommen, einen einfachen Adapter der Kategorie «Programmgesteuerter Adapter mit Interruptmöglichkeit» zu bauen. Er sollte den folgenden Anforderungen genügen:

- 4 digitale Ausgangsleitungen (2 Kanäle mit je 2 Relais-Ausgängen)
- 4 digitale Eingangsleitungen (2 Kanäle mit je 2 Schalter-Eingängen, durch Hardware entprellt)
- Der Adapter soll über Ein-/Ausgabe-Adressen und -Befehle ansprechbar sein.
- POS (Programmable Option Select) soll die folgenden Funktionen umfassen:
 - Interrupt Level 7 oder 12
 - Ein-/Ausgabe-Adressbereich wählbar zwischen den Adressen 300H/301H einerseits und 308H/309H andererseits.

Bit Aktion	Interrupt	Schalter	Relais
Gesetzt durch:	Extern ausgelöst	Extern ausgelöst	Programm kontrolle
Zurückgesetzt durch:	Reset oder Programm kontrolle	Extern ausgelöst	Programm Kontrolle

Bild 20
POS-Steuerung des Experimental-Adapters

- Es soll zudem ein Device Driver erstellt werden, der als Interface unter OS/2 zwischen dem Anwendungsprogramm und dem Adapter dient, wobei normale Ein- und Ausgabebefehle für Dateien der Programmiersprache C verwendet werden sollen.

Systementwurf

Programmable Option Select (POS)

Über die beiden Adressen 100H und 101H soll die Identifikation im Nur-Lese-Modus zugänglich sein. Dem zu besprechenden Adapter wurde durch die IBM Koordinations-Stelle die Identifikation 0FFFFH zugeteilt. Damit steht in Byte 100H der Hex-Wert FF (niederwertig) und in 101H der Wert EF (höherwertig). Für die POS-Steuerung des Adapters (elektronische Schalterstellung) sind 3 Bits im POS-Register 2 (102H) vorgesehen (Bild 20).

Anwendungs-Register

Das Anwendungs-Register wurde gemäss Bild 21 belegt. Die Darstellung in Bild 22 gibt den Überblick, wie ein Bit ein- und ausgeschaltet wird. Man erkennt, dass die Bits, welche die Schalterstellung repräsentieren, nur durch externe Auslöser beeinflusst werden. Die Information kann mit Hilfe des Programms über folgende Adressen gelesen werden.

- 300H: Kanal 0, niedriger Adressbereich
- 301H: Kanal 1, niedriger Adressbereich
- 308H: Kanal 0, höherer Adressbereich
- 309H: Kanal 1, höherer Adressbereich

Bei jeder Schalterbetätigung (Ein und Aus separat) soll ein Interrupt ausgelöst werden. Diese Vorgabe macht nötig, dass die Elektronik bei jeder Veränderung der Schalterstellung das Bit «Interrupt pendent» setzt. Die Rücksetzung erfolgt unter Programmkontrolle, sobald der Interrupt bearbeitet ist. Man beachte, dass alle Bits, mit Ausnahme des ENABLE-Bit, durch die Definition der Anwendung vorgegeben sind. Die Micro Channel-Architektur macht in dem verwendeten Bereich keine weiteren Vorschriften.

Im nächsten Teil wird gezeigt werden, wie dieser Systementwurf in die Realität umgesetzt wurde.

Bild 22
Ereignis-Auslösung

Teil 3 folgt in Heft 1/91