

**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:** 82 (1991)

**Heft:** 1

**Artikel:** IBM Micro Channel Architecture : wieso, wozu : Teil 3 : Entwicklung eines Experimentier-Adapters

**Autor:** Krähenbühl, Fred W.

**DOI:** <https://doi.org/10.5169/seals-902915>

### **Nutzungsbedingungen**

Die ETH-Bibliothek ist die Anbieterin der digitalisierten Zeitschriften. 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. [Siehe Rechtliche Hinweise.](#)

### **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. [Voir Informations légales.](#)

### **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. [See Legal notice.](#)

**Download PDF:** 31.03.2025

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

# IBM Micro Channel Architecture – Wieso, Wozu

## Teil 3: Entwicklung eines Experimentier-Adapters

Fred W. Krähenbühl

**Nach den grundlegenden Darstellungen der IBM Mikro Channel Architektur in den beiden ersten Teilen dieser Reihe befasst sich das dritte Kapitel sehr ausführlich mit der Hard- und Software-Realisierung eines experimentellen Adapters. Es wird gezeigt, dass die Entwicklung eigener Adapter für die Micro Channel Maschinen (IBM PS/2 Modelle 50 bis 95, IBM RISC System/6000 und andere) durchaus möglich ist, vorausgesetzt, dass man sich an die Konventionen hält.**

**Après la présentation des fondements de l'architecture Micro Channel d'IBM dans les deux premiers articles de cette série, le troisième chapitre traite très en détail de la réalisation du matériel et du logiciel d'un adaptateur expérimental. Il montre que le développement de ses propres adaptateurs pour les machines Micro Channel (IBM PS/2, modèles 50 à 95, IBM RISC System/6000 et d'autres systèmes) est certainement possible à condition de respecter les conventions.**

### Adresse des Autors

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

Nach dem im Teil 2 (Heft 21/90) bereits die Randbedingungen besprochen wurden, die der geplante Experimentier-Adapter zu erfüllen hat, soll nun intensiv auf die hard- und softwaremässige Realisierung des Adapters eingegangen werden. Das technische PS/2-Handbuch gibt Anweisungen, welche Grundfähigkeiten ein Adapter aufweisen muss, damit er in einer Maschine mit Micro Channel eingesetzt werden kann. Der hier vorgestellte Adapter ist auf PS/2-Maschinen der Modelle 50, 60 und 80 getestet worden. Es wird aber betont, dass diese Schaltung weder eine optimierte Entwicklung noch die einzig mögliche Implementation darstellt. Der Entwickler kann durchaus andere Lösungen wählen, sofern er die folgenden für den Setup geltenden Regeln berücksichtigt:

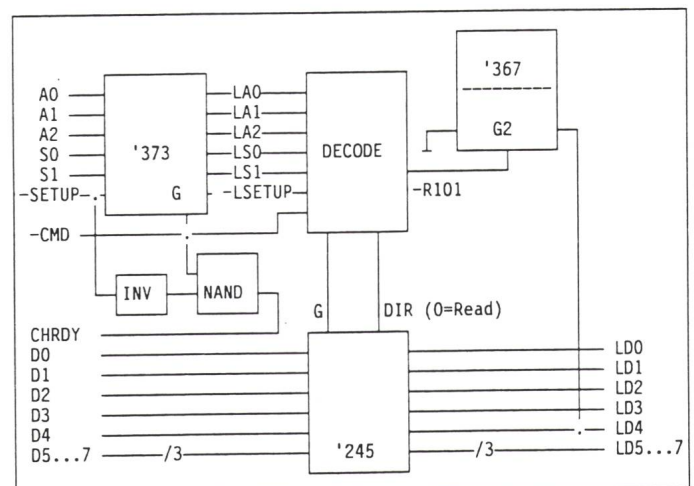
- Jeder Adapter muss Schalter zur Einstellung von Parametern durch Register ersetzen und die Logik für POS implementieren.
- Jeder Adapter muss eine Identifikation enthalten, die auf Anforderung gelesen werden kann.

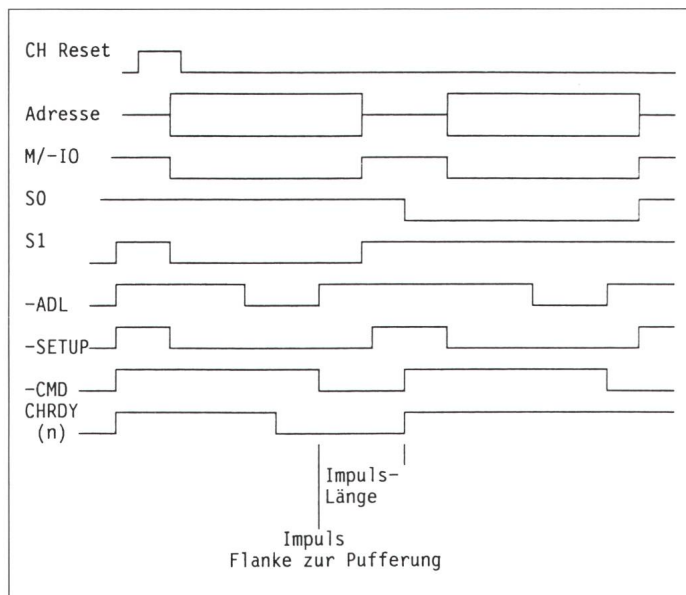
- Jeder Adapter, der sich in Betrieb befindet (ENABLE = 1) muss ein Card-Selected-Feedback-Signal abgeben, sobald die Karte adressiert wird. Das Signal darf nicht generiert werden, solange sich die Karte im Setup-Modus befindet.
- Die Karte muss das Enable-Bit während eines Resets in den Zustand 0 bringen (-SLEEP-Modus).
- Der POS-Zyklus ist mindestens 300 ns lang und kann über einen synchronen Waitstate darüber hinaus verlängert werden.

### Lesen der Identifikation

Sehen wir uns nun die Implementation an (Bild 23): Nach dem allgemeinen Reset-Impuls verlangt das System während der Initialisierungsphase (Setup) als erstes die Identifikation. Dies geschieht durch das Prozedere, das im Abschnitt über POS (erster Teil) ausführlich dargelegt worden ist. Das System wird dabei nacheinander jedem Adapter-Steckplatz den -SETUP-Impuls senden. Dieses Signal enthält zwei wichtige Informationen. Es signalisiert erstens eine I/O-Opera-

**Bild 23**  
Lesen der Adapter-Identifikation





**Bild 24**  
Auszug aus dem  
Basic Transfer  
Cycle

tion. Das Signal M/-IO ist also bereits darin enthalten und braucht nicht mehr speziell decodiert zu werden. Die zweite Information gibt an, dass es sich um die I/O-Adressen 100H bis 107H handelt. Deshalb brauchen nur die 3 niederwertigen Bits der Adresse decodiert zu werden (A0 bis A2). Auf Adresse 100H finden wir das niederwertige Byte der Identifikation, in 101H das höherwertige. Die Adresse 102H zeigt auf das POS-Register 2, das erste Byte mit Setup-Information. Zur Erinnerung: Das Bit 0 enthält das vorgeschriebene ENABLE/-SLEEP-Bit; ist dieses Bit gleich 0, muss der gesamte Adapter inaktiv sein.

Zurück zu unserer Schaltung: Der Baustein 74LS367 ist ein doppelter Linien-Treiber mit 3-State-Ausgängen. Ist dieser offen, wird das an seinen Eingängen anliegende Potential an seine Ausgänge übertragen. Das Signal -R101 öffnet die Schaltung und transferiert so das GND-Niveau vom Eingang an die Leitung LD4 (Latched Data Linie 4). Nur die Null-Niveaus sind zu treiben, da die Daten-Leitungen durch Widerstände auf +5 Volt gehalten werden. Damit können Treiber eingespart werden. Das ist auch der Grund dafür, dass keine Schaltung zur Generierung eines Signals -R100 vorhanden ist. Wenn das System das erste Byte der Identifikation abfragt, erfolgt keine Reaktion und das Resultat ist 0FFH. Das ist genau die gewünschte Antwort.

Die nächste Frage betrifft die Erzeugung des Lese-Impulses. Der Micro Channel liefert alle nötigen Impulsketten. Zuerst sind die drei gepuff-

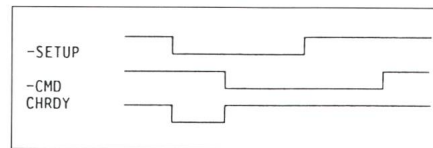
erten Adressbits zu decodieren (nur während der Setup-Phase sind die weiteren Adressbits nicht zu berücksichtigen). Gepuffert werden auch S0 und S1, welche festhalten, ob ein Lese- oder ein Schreibbefehl vorliegt. Unter Verwendung des Zeit-Impulses -CMD (Command) wird der Leseimpuls hergestellt. Wann? Ein Auszug aus dem Zeitablauf (Bild 24) gibt darüber Auskunft. Wir können sehen, dass sämtliche Adress- und Kontrollsignale zur Zeit der führenden Flanke (1 zu 0) von -CMD vorhanden sind. Wird aus irgendeinem Grund eine invertierte Flanke bevorzugt, kann auch die zweite Flanke (0 zu 1) von -ADL benützt werden. Mit diesen Flanken können alle zyklusbestimmenden Elemente erfasst und gespeichert werden. -CMD ist der benötigte Impuls für die Ausführung der Lese- und Schreibbefehle. Mit diesen Angaben können Formeln (Bild 25) zur Decodierung der Signale mittels eines Programmable Array Logic-Bausteins (PAL) angegeben werden (siehe Baustein DECODE in Bild 23).

Jeder Setup-Zyklus ist mindestens 300 ns lang, damit Verzögerungen durch RC-Glieder, die während des Setup aktiv sind, abgefangen werden können. Mittels eines synchronen Waitstates kann diese Zeit noch verlängert werden. Der Micro Channel

$$\begin{aligned} -R101 &= (-LA2) \wedge (-LA1) \wedge (LA0) \wedge (-LS1) \wedge (LS0) \wedge (-LSETUP) \wedge (-CMD) \\ -G(245) &= -R101 \\ DIR(245) &= (LS1) \wedge (-LS0) \end{aligned}$$

**Bild 25** Generierung der Leseimpulse

lässt dies auf einfache Art zu (Bild 26). Ein Inverter und ein Nand-Gatter generieren bei Bedarf den Impuls.



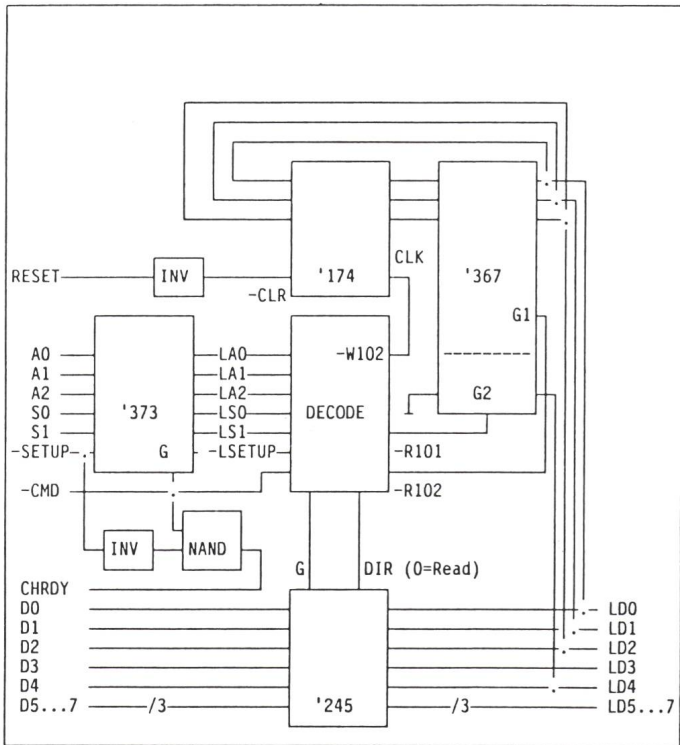
**Bild 26** Synchroner Warte-Zyklus

## Schreiben der POS-Information

Die Identifikation ist nun gelesen. Der nächste Schritt ist die Ausgabe der POS-Information (Bild 27) an den Adapter (elektronisches Setzen der Schalter). Die POS-Information wurde bereits bei der Installation der Karte in die CMOS-Speicher des Systems geladen; sie muss bei jedem Aufstarten von neuem in den Adapter geschrieben werden.

Wie bereits erwähnt, wurden für den Experimentier-Adapter nur das Byte 2 und davon wiederum nur 3 Bits implementiert. Das Speichermedium ist ein Baustein 74LS174, ein 6-Bit-D-Latch mit flankengesteuerter Datenübernahme und niveausensitiven -CLR-Eingängen. In Funktion des RESET-Impulses wird das Enable-Bit (Bit 0, POS 2) auf 0 gesetzt. Das kann auf einfache Art mit Hilfe des invertierten System-Reset-Impulses geschehen. Der gleiche Impuls wird selbstverständlich auch IOA und INTL auf ihren jeweilig niedrigen Zustand bringen (siehe Bild 20). Der Schreib-Impuls -W102, durch welchen die POS-Daten ins Register geschrieben werden, wird in sehr ähnlicher Weise wie der vorher besprochene Leseimpuls generiert. Die Adress-Bits, sowie LS0 und LS1 sind anzupassen. Das gleiche gilt für die Level-Kombinationen für -R102, den Lese-Impuls für POS 2. -G(245) wird nachgeführt. Somit ergeben sich die in Bild 28 dargestellten zusätzlichen Formeln des DECODE.

Der zur Speicherung eingesetzte D-Latch '174 übernimmt die an seinen Eingängen anliegenden Daten durch einen positiven Spannungssprung am



**Bild 27**  
Lesen und Schreiben der POS-Information

denen der Adapter auf Grund seiner Elektronikausstattung arbeiten kann (jede Kombination zwischen Interrupt-Benützung und Adressbereich). Jede dieser Möglichkeiten führt zum Setzen der Bits 1 und 2 im POS-Register 2 (0102H). Diese Bits entsprechen den bei der Planung des Adapters vorgesehenen Kombinationen (s. oben). Es ist die Aufgabe der Adapter-Elektronik, diese Informationen auszuwerten und während der Betriebsphase zu nutzen.

Einige Anmerkungen dazu: Wo ein X im POS-Byte eingesetzt ist, wird während des Aufstartens keine Veränderung durchgeführt. Der Inhalt dieser Bits ist also nicht definiert. Dies trifft für das Enable-Bit (Bit 0) nur zum Teil zu: Dieses Bit wird durch RESET auf 0 gesetzt. Kann das System anschliessend die Konfiguration korrekt durchführen, wird die Information 1 eingeschrieben, was den Adapter in Betriebsbereitschaft setzt.

Die Texte in der Datei werden durch den Hersteller formuliert. Prompt- und Hilfe-Texte können deshalb in jeder beliebigen (durch den vorhandenen Zeichensatz darstellbaren) Sprache eingegeben und dem Benutzer verfügbar gemacht werden.

CLK-Eingang. Dies bedeutet, dass die Übernahme der Daten mit dem Ende des Zyklus, bei der Endflanke des Impulses -CMD, erfolgt. Beim Lesen sind die Datenleitungen während der ganzen Dauer von -CMD offen. Zu diesem Zeitpunkt liegen an den Adress- und Kontrolleitungen bereits wieder die Signale für den nächsten Zyklus an. Deshalb sind diese zu speichern. Die Regel:

- Puffern (Latching) der Kontroll- und Adressinformationen: Führen- de Flanke (1-0-Übergang) von -CMD, allenfalls die zweite Flanke (0-1-Übergang) von -ADL.
- Daten übertragen (Gating) mit dem Ende von -CMD.

Damit ist der Adapter für seine Arbeit konfiguriert, und das System wird den nächsten Adapter initialisieren. Nach Konfiguration des letzten Adapters wird der Setup-Modus verlassen, und die eigentliche Arbeitsphase beginnt.

### Adapter Definition File

Damit das System dem Adapter seine Konfiguration mitteilen kann, sind in seinem batteriegepufferten CMOS-Spezi- alspeicher die anlässlich der Installation ausgewählten Möglichkeiten festgehalten. Jeder Adapter wird von seinem Hersteller mit einer Adapter-Definitionsdatei geliefert, einer Da-

tei, welche sämtliche Konfigurationsmöglichkeiten des Adapters enthält. Der Inhalt dieser Datei für den Experiment-Adapter ist in Bild 29 wieder-

$$\begin{aligned} -R102 &= (-LA2) \wedge (LA1) \wedge (-LA0) \wedge (-LS1) \wedge (LS0) \wedge (-LSETUP) \wedge (-CMD) \\ -W102 &= (-LA2) \wedge (LA1) \wedge (-LA0) \wedge (LS1) \wedge (-LS0) \wedge (-LSETUP) \wedge (-CMD) \\ -G(245) &= (-R101) \vee (-3102) \vee (-W102) \end{aligned}$$

**Bild 28** Generierung der Leseimpulse

gegeben. Er zeigt einige Möglichkeiten auf. Man beachte die vier CHOI-CE (Auswahl)-Zeilen. Sie geben die total vier Möglichkeiten wieder, mit

Weitergehende Informationen über die Syntax finden sich in den technischen Handbüchern der verschiedenen Systeme.

**Bild 29**  
Adapter-Definitionsdatei für den Experimental-Adapter

```
AdapterId 0EFFFh
AdapterName "Experimental Adapter"
NumBytes 1
NamedItem
    Prompt "Choose Interrupt Level and I/O Address ranges:"
    choice "Int. 7, I/O Adr. 0300h" pos[0]=XXXXX00Xb
        io 0300h-0307h int 7
    choice "Int. 7, I/O Adr. 0308h" pos[0]=XXXXX01Xb
        io 0308h-030Fh int 7
    choice "Int. 12, I/O Adr. 0300h" pos[0]=XXXXX10Xb
        io 0300h-0307h int 12
    choice "Int. 12, I/O Adr. 0308h" pos[0]=XXXXX11Xb
        io 0308h-030Fh int 12
    Help
    "The Experimental Adapter can be assigned to use I/O-Addresses
    0300H and 0301H or 0308H and 0309H, as well as Interrupt Lines
    7 or 12. Use the F5=Previous and the F6=Next keys to change
    address and interrupt assignments if you are in the Change
    Configuration Screen. Conflicting assignments are marked with
    an asterisk and must be changed."
```

### Betrieb des Adapters im Arbeitsmodus

Für jeden Adapter gelten eine Reihe zusätzlicher Regeln, die ein korrektes Funktionieren im Arbeitsmodus garantieren sollen. Einen Auszug (die vollständige Beschreibung ist wiederum in den technischen Handbüchern der Systeme zu finden) stellt die folgende Liste dar:

- Jeder I/O-Adapter muss alle 16 I/O-Adress-Linien decodieren.
- Jeder Adapter, dessen Enable-Bit auf 1 (in Betrieb) steht, muss, sobald er adressiert worden ist, ein Signal «Card Selected Feedback» generieren.
- Jeder Adapter muss fähig sein, sämtliche Ausgänge zum Micro Channel, eingeschlossen Datenleitungen, Card Selected Feedback, Interrupts usw. in den hochohmigen Zustand zu setzen, wenn Bit 0 in POS 2 (Port 0102H, Enable) auf 0 steht. Dies erlaubt zweierlei. Erstens einen Adapter, dessen Konfiguration nicht im CMOS-Speicher festgehalten ist (also nicht korrekt installiert worden ist) und deshalb durch das Programm nicht korrekt angesprochen werden kann, zur Sicherung der Systemintegrität ausser Betrieb zu setzen und zweitens ein Konzept zu realisieren, bei dem in einer Maschine zwei redundante Adapter (z. B. für Kommunikation) eingesetzt sind. Während der eine Adapter normal arbeitet, kann der andere ausser Betrieb gesetzt sein. Taucht am arbeitenden Adapter ein Fehler auf, kann der neue Adapter programmgesteuert, eventuell sogar über ein Netzwerk, in Betrieb genommen und damit die Arbeit fortgesetzt werden.

- Die Interrupt Request-Linie muss beim Micro Channel über einen Treiber mit offenem Kollektor angesteuert werden, damit eine einzelne Interrupt-Leitung auf dem Micro Channel von mehreren Adaptern bedient werden kann (Interrupt Sharing). Daher muss in einem Register auf dem Adapter, das den Interrupt-Status festhält und über eine I/O-Instruktion lesbar ist, ein Bit für diese Aufgabe

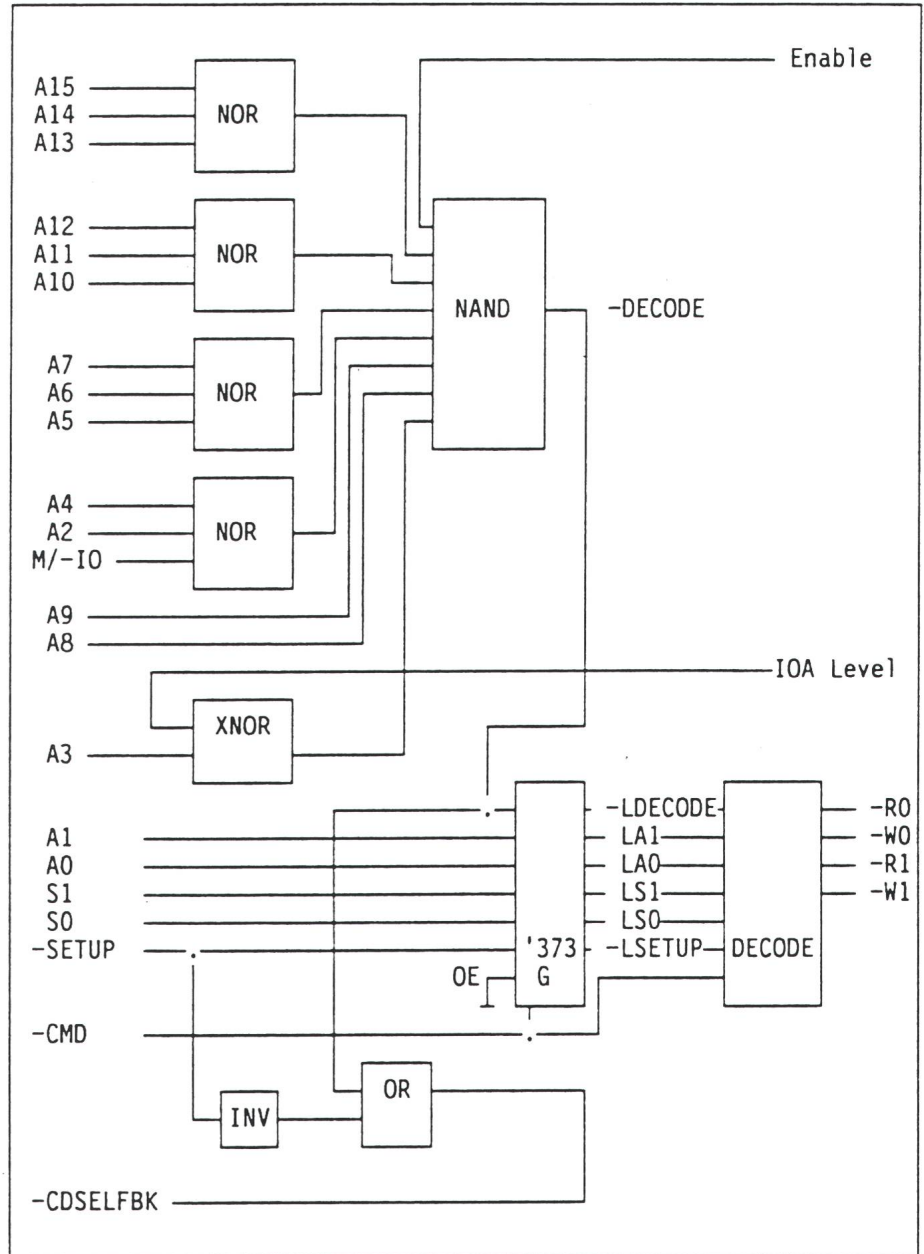


Bild 30 Lesen und Schreiben der Anwendungs-Register

reserviert werden. Der Adapter muss den Registerinhalt und die Interrupt Request-Linie aktiv halten, bis der Zustand durch Software (Interrupt Service Routine) zurückgesetzt wird. Das Anwendungs-Register (Bild 21) enthält diese Information. Eine mögliche Implementation zeigt der Schaltungsauszug von Bild 30. Die Decodierung der Adressen 300H, 301H,

308H und 309H kann gut anhand der Zeichnung nachvollzogen werden. Einige Punkte sind dennoch wert, speziell erwähnt zu werden:

- IOA, der I/O-Adress-Level aus POS, wird auf High (1) sein, wenn über POS die Benützung der hohen Adressen 308 und 309 eingestellt worden ist. Andernfalls wird das Signal auf Low sein. Das Adressbit 3 folgt der gleichen Logik. Daraus geht hervor, dass, wenn immer diese Signale gleich sind, Bit 3 als gültig betrachtet werden kann. Dieser Vergleich wird durch ein XNOR-Tor, dessen Ausgang bei Gleichheit 1 ist, realisiert.
- Das Signal M/-IO, dessen tiefes Niveau einen Ein-/Ausgabe-Zyklus si-

$$\begin{aligned}
 -RO &= (-LDECODE) \wedge (-LA1) \wedge (-LA0) \wedge (-LS1) \wedge (LS0) \wedge (LSETUP) \wedge (-CMD) \\
 -R1 &= (-LDECODE) \wedge (-LA1) \wedge (LA0) \wedge (-LS1) \wedge (LS0) \wedge (LSETUP) \wedge (-CMD) \\
 -WO &= (-LDECODE) \wedge (-LA1) \wedge (-LA0) \wedge (LS1) \wedge (-LS0) \wedge (LSETUP) \wedge (-CMD) \\
 -W1 &= (-LDECODE) \wedge (-LA1) \wedge (LA0) \wedge (LS1) \wedge (-LS0) \wedge (LSETUP) \wedge (-CMD)
 \end{aligned}$$

Bild 31 Erzeugung der Lese- und Schreibimpulse

gnalisiert, kann genau gleich, wie ein normales Adressbit behandelt werden.

– Das Signal -DECODE ist nur dann 0, wenn alle Eingänge des Nand-Gatters, auch ENABLE, gleich 1 sind. Es wird – zusammen mit den niederwertigen Adress- und den beiden Statusbits sowie dem Signal -SETUP – mittels der führenden Flanke von -CMD gespeichert.

– Der Decoder ist wiederum als PAL realisiert, dessen Formeln in Bild 31 dargestellt sind. G (245) aus Bild 28 ist entsprechend nachzuführen.

– Das Signal CDSELEFBK (Card Selected Feedback) muss immer generiert werden, wenn die Karte nicht im Setup-Modus ist und durch einen Partner adressiert worden ist.

Dies wird durch die Kombination von -DECODE und -SETUP erreicht. Das Signal muss innerhalb 60 ns nach dem Erscheinen einer gültigen Adresse und der zugehörigen Kontrollsignale stabil sein. Der gezeigte Mechanismus benötigt in seinem längsten Pfad etwa 50 ns. Damit ist das Ziel erreicht.

Zum Schluss sei mit der Darstellung in Bild 32 das Interrupt-Konzept dargelegt. Wann immer ein Interrupt nötig ist, wird die externe Anforderung im Baustein '74 festgehalten und anlässlich eines Lesezyklus über den '244 auf die Datenleitung 6 gegeben. Die Kombination der aus dem POS-Byte 2 stammenden Information INTL (Interrupt 7 oder 12) und dem ENABLE-Bit aus dem gleichen POS-Byte ergibt das eigentliche Interruptsignal, das über die Treiber '05 (offener Kollektor) auf je eine Interrupt-Leitung des Micro Channels geleitet werden kann. Die Schaltungen dieses Kapitels wurden auf einem Prototyp-Adapter mittels «Vector Wiring» realisiert und erfolgreich getestet.

## Software

Unter DOS kann der Adapter normal über die Adressen 300H/301H oder 308/309H angesprochen werden. Hier muss sich das Anwendungsprogramm aber mit dem gesamten Verkehr zwischen dem System und dem Adapter auseinandersetzen. Unter DOS wird die gesamte Maschine gleichzeitig nur dieser einen Aufgabe zur Verfügung stehen können.

Unter OS/2, in einer Multitasking Umgebung, in der die Leistung der Maschine für viele Arbeiten gleichzei-

tig ausgenutzt werden kann, ist diese Art und Weise der Programmerstellung nicht mehr sinnvoll, weil eine «gleichzeitige» Verwendung eines Adapters durch verschiedene Programme (z. B. Harddisk-Zugriffe) nötig sein kann. Für den Adapter wird also ein Device Driver-Programm geschrieben, welches den Adapter einerseits und die darauf aufbauenden Anwendungen andererseits unterstützt.

In unserem Fall wird der Device Driver in seiner Initialisierungs-Routine, welche während der Aufstartphase des Systems abläuft, unter anderem herausfinden, mit welchen I/O-Adressen und mit welchem Interrupt-Level gearbeitet werden soll. Diese Informationen sind im CMOS-Speicher des Systems erhältlich. Für den Zugriff auf diese Speicher stellt BIOS die entsprechenden Routinen zur Verfügung. Der Device Driver wird eine Interrupt-Service-Routine zur Verfügung stellen, die auf der gleichen

Damit ist dokumentiert, dass das Gespann PS/2 – OS/2 für viele Echtzeit-Applikationen geeignet ist und auch preislich interessante Lösungen möglich macht.

## Im Nachhinein ist man immer klüger

Dieser Spruch hat sich auch bei der Entwicklung des Experimentier-Adapters bewahrheitet. Eine wichtige Konsequenz daraus ist zweifellos die folgende: Adapter, Device Driver und Anwendungsprogramm sind so aufeinander abzustimmen, dass das Multitasking-Konzept auf allen Ebenen korrekt und sauber implementiert wird. Dies ist speziell wichtig bei der Anlage der Register. Im vorliegenden Fall wäre es besser gewesen, für jeden benützbaren Ein- und Ausgabekanal je einen völlig unabhängigen Registersatz mit eigenen Adressen zu schaffen.

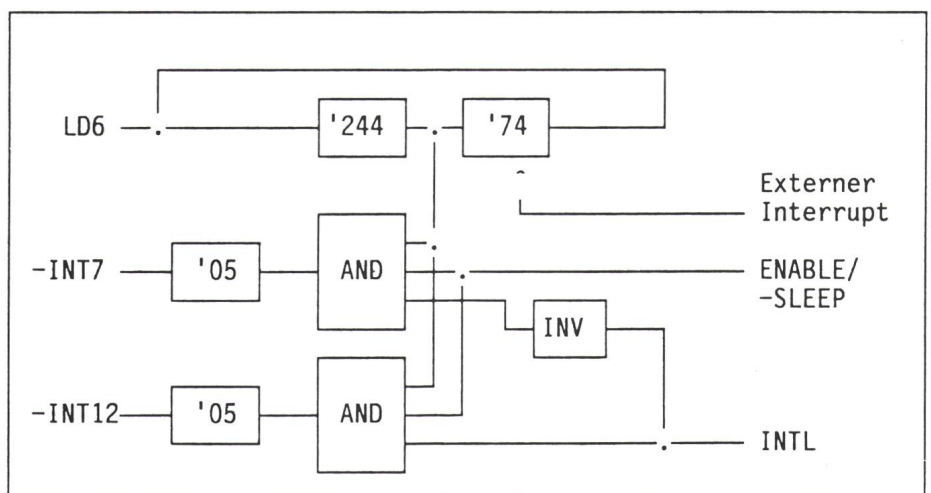


Bild 32 Interrupt

(höchsten) Prioritätsstufe arbeitet, wie der Kern des Betriebssystems selbst (Architektur OS/2). Damit ist sichergestellt, dass die Interrupts mit einem ausserordentlich hohen Servicegrad bearbeitet werden. Messungen an einem IBM PS/2 Modell 80-111 haben die folgenden Ergebnisse erbracht:

- Reaktion bei Interrupt aus dem Device Driver (was man ausser zum Rücksetzen des Interrupt Pending Bits normalerweise nicht tun soll): 135 µs.
- Reaktion bei Interrupt aus der Anwendung über den Device Driver (was man normalerweise tun sollte): 640 µs.

Das hätte die identische Aufteilung in einzelne Tasks von der Hardware über den Device Driver bis hin zum Anwendungsprogramm selbst bedeutet. Das ist ein anstrebenwertes Ziel.

Auf den relativ kleinen Adaptern ist viel Logik unterzubringen. Die Verwendung der kleinformigen Steckkarten beim PS/2 erlaubt zwar elegante Systeme zu konzipieren, zwingt aber andererseits zu einer Packungsdichte auf den Adaptern, die mit konventionellen TTL-Bausteinen nur schwer zu erreichen ist. PALs wie auch SMT-Bauteile können, wo nötig, wirkungsvolle Hilfe bringen.

Teil 4 folgt in Heft 5/91



# Schaltuhren

(und Stundenzähler)

sind unsere Spezialität

**e.o.bär**

3000 Bern 13

Postfach 11  
Wasserwerkstrasse 2  
Telefon 031/22 76 11

Inserieren Sie im

## Bulletin SEV/VSE

86% der Leser sind  
Elektroingenieure ETH/HTL

91% der Leser haben  
Einkaufsentscheide zu treffen

**Sie treffen ihr  
Zielpublikum**

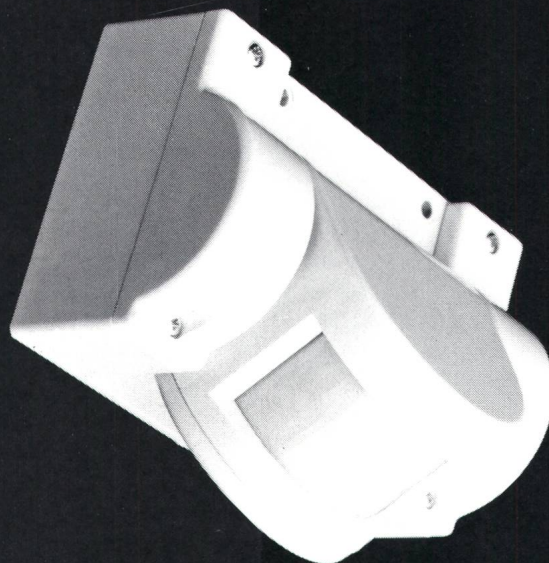
Wir beraten Sie kompetent  
Tel. 01/207 86 32

Damit Sie nicht im Dunkeln tappen . . .

## Sicherheitsschalter MZ-2000

Passiv-  
infrarot-  
Bewegungsmelder  
mit 180 Grad  
Erfassungsbereich.

- Schaltet Licht
- steuert Treppenhaus-  
automaten oder  
Relais
- überwacht Türen
- löst Alarm aus
- Übermittlung von  
Schaltsignalen  
mit Funk



- Überwachungsbereich  
180 Grad auf  
mehreren Ebenen
- Reichweite 15 m
- Integrierter  
Dämmerungssensor
- Integrierter Timer
- Schutzart IP 54
- SEV-geprüft

Weitere Modelle:  
Kurzimpuls Version  
9-Volt Gong Version  
Funk Version



**M. Züblin AG**

Bächlerstrasse 31 CH-8046 Zürich  
Tel. 01 372 00 05 Fax 01 371 61 96