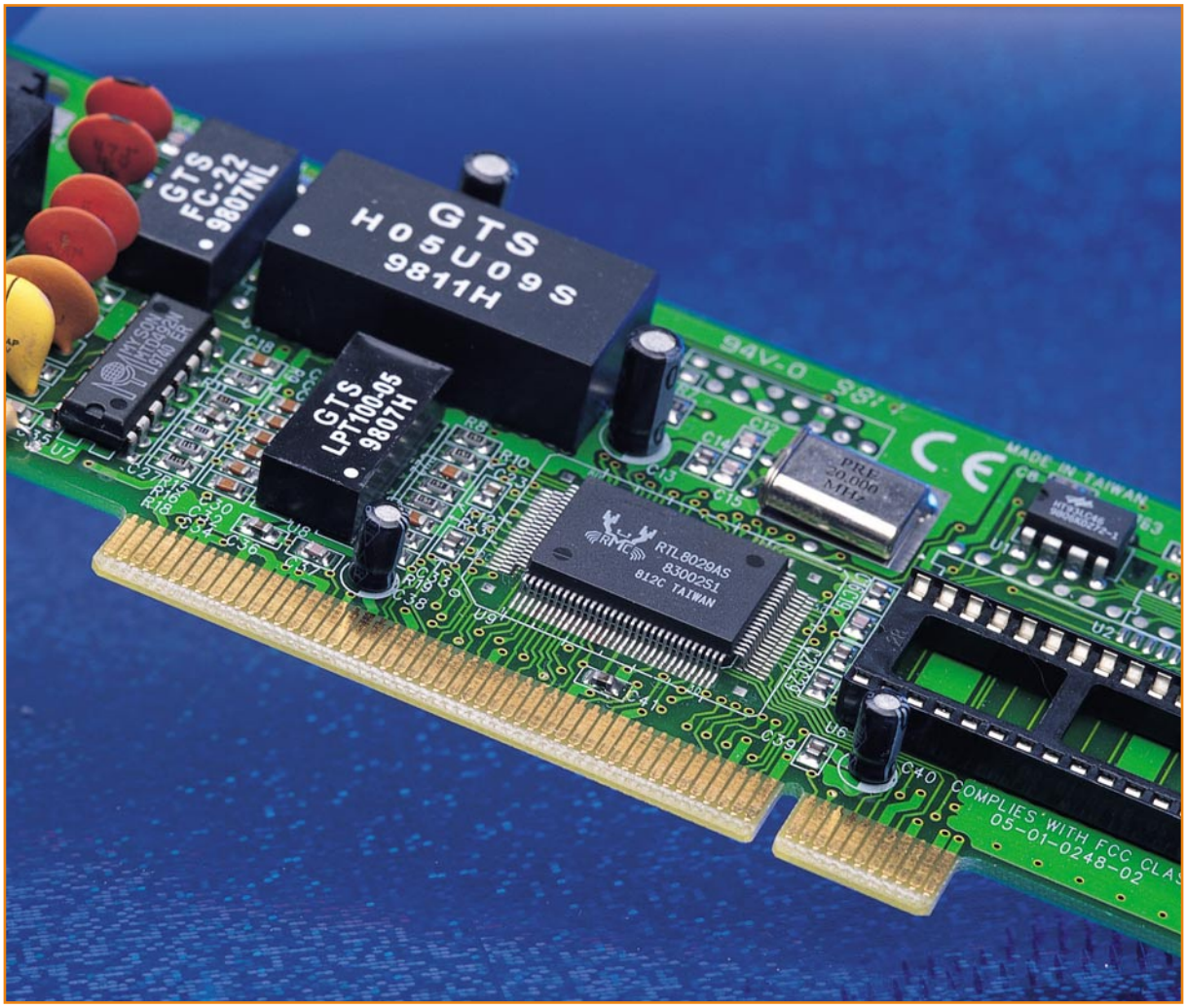

PCI-Grundlagen Teil 1

Moderne PCs verfügen über einen leistungsfähigen PCI-Bus, der die Verbindung zu den verschiedensten steckbaren Erweiterungskarten realisiert. Diese Artikelserie soll einen Überblick über die recht anspruchsvolle Thematik „PCI-Bus-Technologie" bis hin zur Einführung in die Entwicklung und Programmierung von PCI-Karten, unterstützt von weiterreichender Literatur und einem ELV-PCI-Entwicklungsboard, geben.

Einleitung

Der PCI-Bus bildet den derzeitigen Industriestandard für den internen Anschluß von Peripheriebaugruppen, landläufig als Steckkarten bezeichnet, auf PC-Motherboards. Er löst den veralteten ISA-Bus ab, der lange Jahre den Industriestandard darstellte.

Die hiermit beginnende Artikelserie über den PCI-Bus soll zunächst einen Überblick über das PCI-Bussystem an sich geben, bevor sie tiefer auf die Spezifikationen der Architektur, des Controllings und der Programmierung eingeht.

Dabei werden folgende Themen behandelt:

- Der PCI-Bus als Erweiterungs-Bussystem der Zukunft
- Der PCI-Controller
- Die Software
- Der letzte Abschnitt beschäftigt sich mit einem PCI-Entwicklungsboard, das zum einen als Basis für eigene Anwendungen dienen kann, aber auch einfach als digitale I/O-Baugruppe einsetzbar ist.

Das komplette Thema PCI und Treiberentwicklung ist selbst für eine solche Artikelserie zu umfangreich. Aus diesem Grund wird ELV weiterführend das Buch „Windows-Hardware-Programmierung" zur detaillierten Beschreibung der PCI-Technologie und deren Programmierung anbieten. Dieses Buch ist auf das ELV-PCI-Entwicklungsboard abgestimmt und bildet damit eine sinnvolle Ergänzung.

Der PCI-Bus als Erweiterungs-Bussystem der Zukunft

Von vielen Anwendern praktisch unbemerkt, entwickelt sich der PC zu einem System für jedermann, bei dem es immer weniger Probleme bei der Installation von Zusatzhardware gibt. Natürlich geht diese Entwicklung nicht ohne Auswirkungen auf das Aussehen dieser Hardware vonstatten.

Jeder, der in der letzten Zeit ein modernes Motherboard für einen Standard-PC gesehen hat, wird feststellen, daß sich z. B. die Anzahl der ISA-Bus-Slots erheblich gegenüber älteren Motherboards reduziert

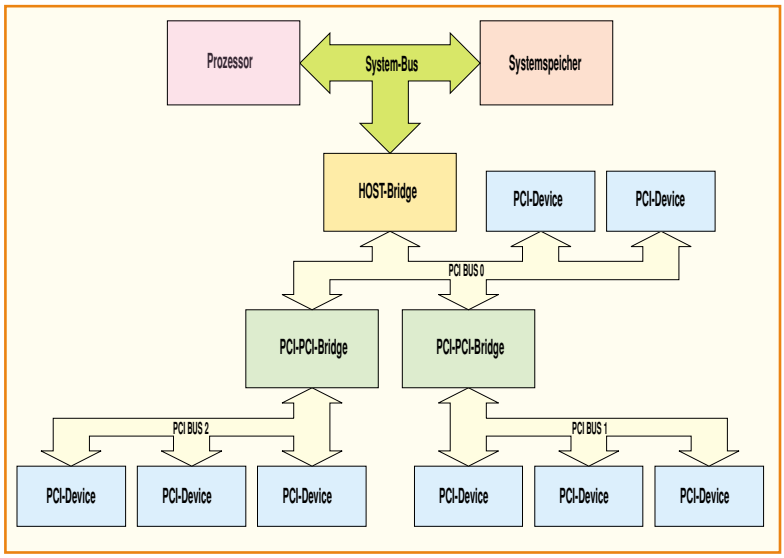

Bild 1: PC-System mit PCI-Bus

hat. In der nächsten Generation, die ab Mitte 1999 im Handel sein wird, werden sie sogar komplett verschwunden sein! Dies geschieht nicht zuletzt im Zuge der Standardisierung in der Computerwelt. Als Beispiel hierfür wäre die Spezifikation „PC-99" zu nennen.

Das Verschwinden des ISA-Busses hat einen guten Grund. Ein PC für jedermann soll stets auf Eingaben des Benutzers reagieren können und einfach zu konfigurieren sein. Diese Forderung ist mit dem ISA-Bus nicht mehr zu erfüllen, da hier eigentlich kein wirkliches „Plug and Play" möglich ist. Die heute noch im ISA-Bussystem eingesetzten „Plug and Play"-Controller führen in der Praxis immer wieder zu Konfigurationsproblemen. Eine Einstellung der Hardware über „Jumper" ist einem „Normaluser" nicht zuzumuten. Der Anwender möchte seinen PC kaufen, zu Hause aufstellen, anschalten und dann soll alles (und das möglichst für immer) funktionieren - daß dies in der Praxis anders aussieht, weiß wohl jeder von uns ...

Der PCI-Bus, der den ISA-Bus als internen Erweiterungs-Bus ablösen wird, ist von Anfang an darauf ausgelegt, daß eine vollständige Konfiguration ohne Eingriff des Benutzers möglich ist, d. h., Konfigurations-Handgriffe auf der Karte selbst sollen entfallen, und der Bus muß die Möglichkeiten bieten, die Hardware für die automatische Systemeinstellung vollständig zu erkennen.

Gleichzeitig ist der Datendurchsatz sehr viel höher als beim ISA-Bus. In einem heute üblichen PC beträgt die maximale Durchsatzrate ca. 133 MB/s. Beim ISA-Bus lag sie bei ca. 16 MB/s.

Der PCI-Bus ist ursprünglich eine Entwicklung der Firma Intel, wird aber inzwischen von einem Zusammenschluß aus mehreren großen Firmen, der sog. „PCI Special Interest Group" (kurz PCI-SIG) gepflegt und an neue Bedürfnisse der PC-Industrie angepaßt.

Alle Neuerungen dieser Arbeitsgruppe werden auf der PCI-SIG Internet-Hompage (www.pcisig.com) verbreitet. Auch die aktuellen Spezifikationen sind hier bestellbar.

Die aktuelle Version der PCI-Spezifikation ist die Version 2.1, die aber in Kürze eine Ablösung durch die Version 2.2 erfahren wird. Die neue Version 2.2 enthält einige wichtige Änderungen gerade im Bereich des „Powermanagements", das eine bessere Verfügbarkeit des PCs gewährleistet, was u. a. bedeutet, daß der Rechner immer dann und sofort nutzbar sein soll, wenn der Anwender ihn benötigt (ohne langwieriges Booten oder ähnliches).

PCI als Basis für die Systemarchitektur

Der PCI-Bus ist aber nicht nur ein Bus, der einfach in Form eines Steckers für Hardware-Systemerweiterungen zur Verfügung steht. Er ist vielmehr ein Bus, der nahezu alle Komponenten des Computersystems miteinander verbindet.

Kern dabei sind sog. PCI-Bridges, die unterschiedliche Funktionen auf dem PCI-Bus abbilden. Eine PCI-Bridge stellt z. B. den USB-Bus, den ISA-Bus (wenn noch vorhanden) und den SCSI-Bus dem Prozessorsystem zur Verfügung. Auch der eigentliche Prozessor ist über die sog. „Host-Bridge" an das PCI-Bussystem angeschlossen. Abbildung 1 zeigt die Funktion des PCI-Bus im System.

Wie in Abbildung 1 zu sehen, ist es durchaus möglich, daß mehrere PCI-Busse, die über sog. PCI to PCI Bridges verbunden sind, existieren können. Diese werden durch sog. „Busnummern" gekennzeichnet, über die der Host (Prozessor) feststellen kann, an welcher Stelle sich ein PCI-Device in einem System befindet.

Ein PCI-Device ist dabei eigentlich mit einem physikalischen Gerät, das an den PCI-Bus angeschlossen ist, vergleichbar. Dieses kann aber auch mehrere unterschiedliche, funktional getrennte Einheiten beinhalten (z. B. einen seriellen Schnittstellen-Controller und einen parallelen Schnittstellen-Controller).

Der PC verwaltet diese Einheiten unter unterschiedlichen „Functions", die mittels „Function Numbers" gekennzeichnet sind. Jedes Device hat darüber hinaus eine „Device Number", die durch die Position des Devices am Bus definiert wird. Diese wird in der Regel durch die PCI-Bridge vergeben.

Eine PCI-Funktion innerhalb eines PCs wird also über die Busnummer, die „Device-Number" und die „Function-Number" eindeutig ausgewählt und damit definiert, wo sie sich im System befindet.

Plug and Play

Das „Plug and Play", also die automatische Konfiguration, ist eines der großen Stärken des PCI-Busses.

Diese wird dadurch erreicht, daß jedes PCI-Device über einen genau festgelegten Speicherbereich verfügt, in dem alle notwendigen Informationen zur Konfiguration der PCI-Karte enthalten sind. Diese können vom Betriebssystem oder vom

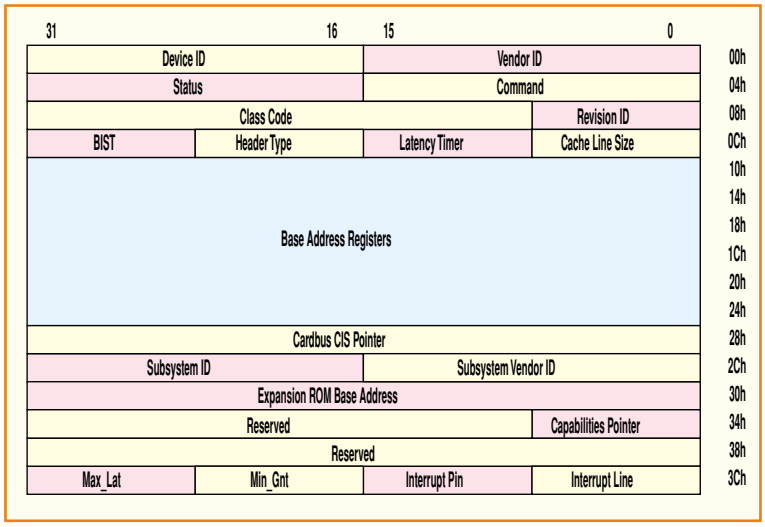

Bild 2: Configuration Space

BIOS gelesen und die vom System während des Bootens ermittelten Parameter - wie z. B. die Basisadressen von Speicherbereichen - zugewiesen werden. Der Speicherbereich wird „Configuration-Space" genannt (Abbildung 2). Er hat eine Größe von 256 Byte, wobei aber nur die unteren 64 Byte fest definiert sind. Zusätzlich sind definierte Erweiterungen - z. B. für das „Powermanagement" - in einer Art verketteten Liste in diesem Bereich abspeicherbar.

Zu einem gut funktionierenden „Plug and Play"-System gehört beim PCI-Bus immer auch Software, die mehr oder weniger gut die vorhandenen Rechner-Ressourcen verwaltet. In der Praxis treten aber kaum Probleme in diesem Bereich auf.
Die Aufgaben der Software definieren sich wie folgt:

- Erkennen neuer PCI-Devices durch Überprüfen aller möglichen Busse
- Erkennen von Änderungen (z. B. Einbau einer neuen Version oder einer völlig anderen Funktionalität an der gleichen Stelle im System)
- Verknüpfen von Treibern mit dieser Hardware.

Wichtig für das „Plug and Play" sind die IDs (Vendor, Device, Subsystem und Subsystem Vendor ID), über die das Gerät eindeutig definiert ist und jederzeit wiedererkannt werden kann. Anhand dieser Informationen stellt das System eine Verknüpfung aus dem Hardware-Typ und der Position der Hardware im System zu einem installierten Treiber her. Zusätzlich kann über das Feld „Class Code" eine Eingruppierung der Hardware in eine bestimmte Hardware-Gruppe erfolgen, die z. B. die Reihenfolge der Initialisierung festlegt.

Das Feld „Revision ID" ist eigentlich nicht direkt für das System, sondern eher für den Treiber gedacht, da anhand dieser Information bestimmte Eigenschaften der Hardware bei der konkreten Treiberkonfiguration berücksichtigt werden können.

Alle diese Felder sind vom Kartenhersteller frei bestimmbar. Da sie eine einzigartige Identifizierung der Hardware darstellen, müssen die entsprechenden Angaben der PCI-SIG mitgeteilt werden (gilt für Vendor und Device ID). Dies soll Doppelbelegungen von IDs verhindern. Der Wert für den „Class Code" ist durch die PCI-Spezifikation festgelegt. Der Hersteller muß daraus einen passenden Hardware-Typ auswählen.

Die von der Hardware repräsentierten Speicherbereiche im I/O- oder Systemspeicherbereich werden im Feld „Base Address Registers" angegeben. In diesem Bereich trägt das System die Basisadresse(n) der PCI-Karte ein. Eine funktionale Einheit eines Devices kann dabei max. 6 verschiedene Bereiche angeben. Welche Basisadresse für ein PCI-Device eingestellt ist, kann z. B. über die Windows-Systemsteuerung herausgefunden werden. Über diese Basisadressen ist die PCI-Karte ansprechbar.

Alle weiteren Bereiche des „Configuration Space" sind für die Basisfunktionalität weitgehend uninteressant. Wer jedoch mehr hierüber wissen möchte, sollte sich die PCI-Spezifikation ansehen.

Powermanagement

Das Powermanagement ist eine zweite, sehr wichtige Eigenschaft des PCI-Busses, auch wenn sie in der PCI-Spezifikation erst in der Version 2.2 richtig zur Geltung kommt.

Unter „Powermanagement" versteht man ein System, das für den Anwender „scheinbar" immer verfügbar ist. Das heißt: keine lange Bootzeiten und die gestarteten Applikationen bleiben scheinbar aktiv, auch wenn das System ausgeschaltet wird. Daneben ist es möglich, daß bestimmte „Devices" das System „aufwecken können" (z. B. ein internes Modem als PCI-Karte für den Empfang von Faxen).

Alle diese Anstrengungen werden unter dem Schlagwort „OnNow"-Initiative zusammengefaßt, die sich sowohl auf die Hardware als auch auf die Software (BIOS und Betriebssysteme) bezieht.

Für ein PCI-Device bedeutet das, daß bestimmte Anforderungen an den PCI-Controller und an die Stromversorgung gestellt werden. Alle diese Anforderungen werden in der „PCI-Bus-Power-Management-Specification" des PCI-SIG beschrieben.

Der Kern der Spezifikation ist die Festlegung einer Erweiterung des „Configuration Space", um Powerzustände einzustellen und mitzuteilen. Daneben legt die Spezifikation die Stromaufnahme und die zur Verfügung stehenden Versorgungsspannungen fest, die in einzelnen Powerzuständen vorhanden sind oder nicht. Diese werden als „Device States" oder kurz „D-States" bezeichnet. Tabelle 1 enthält die durch die PCI-Spezifikation festgelegten Zustände D 3 bis D 0.

Wichtigste Neuerung bei der Stromversorgung des PCI-Bus ist die, daß eine Standby-Versorgungsspannung am PCI-Bus (Vaux) zur Verfügung zu stellen ist. Diese

Tabelle 1: Tabelle D-States

D-State	Beschreibung
D0	In diesem Zustand ist das PCI-Device voll betriebsfähig und kann ohne Verzögerungszeit von der Software angesprochen werden. In der Regel ist das auch der Zustand, bei dem die Leistungsaufnahme des PCI-Devices am größten ist.
D1	Die Implementierung dieses Zustandes ist optional. Er wird auch als „Light Sleep State" bezeichnet. Einige Funktionen des Devices werden dabei abgeschaltet, um die Leistungsaufnahme zu reduzieren.
D2	Dieser Zustand entspricht weitgehend dem Zustand D 1; jedoch soll hier eine signifikante Reduzierung der Leistungsaufnahme erkennbar sein. Die PCI-Spezifikation schweigt sich leider über genaue Werte aus. Die Steuerung ist aber auch mehr Sache des Betriebssystems (wie z. B. Windows 98 oder 2000). Ebenso wie im Zustand D1 muß der „Configuration Space" ansprechbar sein. Befindet sich ein PCI-Device in diesem Zustand, so ist es nicht erlaubt, direkt in den D1-Zustand zurückzuwechseln. Es ist stets zunächst der D0 Zustand einzunehmen. In der anderen Richtung, also von D1 nach D2, kann dieser Schritt ausgelassen werden.
D3hot	Der D3-Zustand ist eigentlich der „Aus"-Zustand des PCI-Devices. In jedem Fall muß das gesamte PCI-Device nach dem Übergang in den D0-Zustand neu initialisiert werden. D3-„Hot" kennzeichnet dabei einen Zustand, in dem das PCI-Device nicht betriebsbereit, jedoch der PCI-Bus aktiv ist. In diesem Zustand muß das PCI-Device auf Zugriffe auf den „Configuration Space" reagieren, wenn die Versorgungsspannung und der PCI-Takt eingeschaltet sind. Das PCI-Device kann also durch Programmierung in den D0-Status zurückkehren. Eine Initialisierung durch das RST# (RESET) erfolgt nicht. Das PCI-Device führt vielmehr einen internen „Reset" durch („Soft Reset").
D3cold	In diesem Zustand ist wie beim "Hot"-Zustand eine vollständige Initialisierung des PCI-Device notwendig, sobald es wieder in den D0-Zustand wechselt. Der Unterschied besteht darin, daß die Versorgungsspannung komplett abgeschaltet ist. Das PCI-Device kann daher nicht durch einfache Programmierung des „Configuration Space" nach D 0 wechseln. Möchte das System das PCI-Device wieder in den D0-Zustand versetzen, so muß zunächst die Versorgungsspannung bereitgestellt und anschließend das Businterface mittels des RST# (RESET) in einen definierten Zustand versetzt werden.

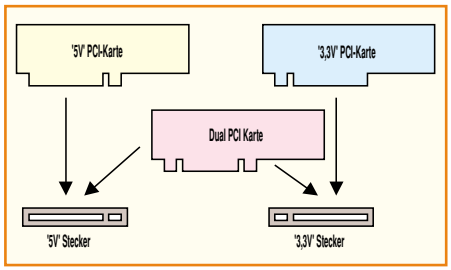

Bild 3: PCI-Vertauschungsschutz

3,3V-Versorgungsspannung dient im D3-cold-Zustand dazu, den Teil der PCI-Karte zu versorgen, der zur Erzeugung des PME#-Signals notwendig ist (z. B. eine Klingelzeichenerkennung bei einem Modem). Ein „Low"-Signal auf dieser Leitung führt zum „Aufwecken" des Systems.

Der PCI-Stecker

Da sich die meisten Interessenten an der PCI-Thematik vermutlich mit dem Design von Hardware für den Erweiterungsbus beschäftigen, sollen noch ein paar Worte über die mechanischen Anforderungen an eine PCI-Karte gesagt werden.

Sowohl die Lage, der Abstand, die Verschraubungen, das „Bracket", die Abmessungen der Platine als auch der Stecker selbst sind durch die PCI-Spezifikation detailliert festgelegt.

Der PCI-Bus existiert neben der heute in PCs üblichen Ausführung als 5V-Version auch in einer 3,3V- und in einer „dualen" Ausführung. Die Spannungsangaben beziehen sich dabei auf die Signalleitungspegel, nicht jedoch auf die Versorgungsspannungen. Um Vertauschungen dieser Karten mechanisch auszuschließen, sind im Stecker des PCI-Slots Stege (sog. Keys) vorhanden. Abbildung 3 zeigt das Prinzip dieses Vertauschungsschutzes.

Bei den sog. „dualen"-Karten sind zwei dieser Stege vorhanden, so daß sie in beide Stecker passen. Um dies elektrisch zu ermöglichen, ist ein separater Versorgungsspannungspin (VIO) vorhanden, mit dem die Bustreiber des PCI-Controllers versorgt werden.

Bustransaktionen

Es ist sicherlich unmöglich, in diesem Artikel alle möglichen PCI-Transaktionen darzustellen, deshalb sollen hier nur die wichtigsten Begriffe und Eigenschaften dieser Transaktionen beschrieben werden.

Grundsätzlich unterscheidet man beim PCI-Bus zwischen einem Target-Device und einem Master-Device. Das Master-Device ist dabei immer der Initiator einer Transaktion. Jedes PCI-Device kann sowohl Master als auch Target sein. Auch wenn die volle Master-Funktionalität nicht immer notwendig ist, ist es an einigen Stellen jedoch sinnvoll, diese Funktionen zu nutzen. Dies ist besonders beim DMA (Direct Memory Access) der Fall. Hier wird direkt vom PCI-Device in den Systemspeicher geschrieben bzw. davon gelesen. Diese Eigenschaft kommt im besonderen den Kartenherstellern zugute, deren Produkte in sehr kurzer Zeit sehr viele Daten übertragen müssen (z. B. Meßkarten).

Bevor ein PCI-Device eine Bustransaktion starten darf, muß dieses Device zunächst von einer Kontrollinstanz aufgefordert werden. Dies ist in der Regel die für diesen PCI-Bus zuständige PCI-Bridge. Um den Bus anzufordern, wird das Signal REQ# vom PCI-Device gesetzt. Die Kontrollinstanz setzt, wenn der Bus verfügbar ist, das Signal GNT#. Das PCI-Device muß nun innerhalb einer bestimmten Zeit die Transaktion starten.

Die Steuerung der eigentlichen Transaktion erfolgt im wesentlichen durch die Signalleitungen TRDY#, IRDY# und FRAME#:

- FRAME#: Wird vom Master-Device getrieben und zeigt den Beginn und Dauer einer Transaktion an.
- IRDY#: Wird ebenfalls vom Master-Device getrieben und zeigt an, daß eine Datenübertragung erfolgen kann.
- TRDY#: Wird vom Target-Device getrieben und zeigt an, daß Daten übertragen werden können.

Mit diesen drei Signalen realisiert man also eine Art „Handshake" zwischen „Master" und „Target".

Abbildung 4 zeigt als Beispiel eine Schreibtransaktion (Daten werden vom Master zum Target übertragen).

Alle Transaktionen laufen synchron zum PCI-Takt ab (CLK). Entscheidend sind jeweils die steigenden Flanken dieses Signals, die zur besseren Beschreibung numeriert sind.

Jede Transaktion beginnt zunächst mit einer Adreßphase, bei der die AD-Signalleitungen die angesprochene Adresse repräsentieren. Danach folgen eine oder mehrere Datenphasen, innerhalb derer diese Signalleitungen die zu übertragenden Daten führen. Folgen mehrere Datenphasen, so spricht man von einer „Burst-Transaktion". Die Adreßphase und damit der Beginn der Transaktion wird durch Setzen des FRAME#-Signals zum Zeitpunkt 2 angezeigt (Zustandswechsel zwischen Zeitpunkt 1 und 2).

Danach setzt das durch die Adresse angesprochene Device das DEVSEL#-Signal (im Beispiel zum Zeitpunkt 3). Die Datenübernahme zwischen dem Target- und dem Master-Device erfolgt immer dann, wenn IRDY# und TRDY# gesetzt sind (im Beispiel zum Zeitpunkt 3,4 und 8). Diese Signale verzögern die dritte Datenphase.

Das Ende der Transaktion wird durch Löschen des FRAME#-Signals während der letzten Datenphase angezeigt.

Die Signalleitungen C/BE# führen während der Adreßphase das sog. Buskommando, das festlegt, welcher Adreßbereich (Systemspeicher, I/O-Bereich oder „Configuration- Space") aktuell angesprochen ist.

Während der Datenphase geben die Zustände der Datenleitungen an, welche Bytes des 32 Bit breiten Datenbusses gültig sind und zum bzw. vom Target übertragen werden.

Das Beispiel stellt eine Schreibtransaktion dar, bei der die AD-Signalleitungen über die gesamte Dauer der Transaktion vom Master-Device getrieben werden. Die Lesetransaktion läuft ebenso, jedoch erfolgt hier das Treiben der AD-Signalleitungen während der Datenphase vom Target-Device.

Im zweiten Teil der Artikelserie befassen wir uns schwerpunktmäßig mit dem PCI-Controller. ELV

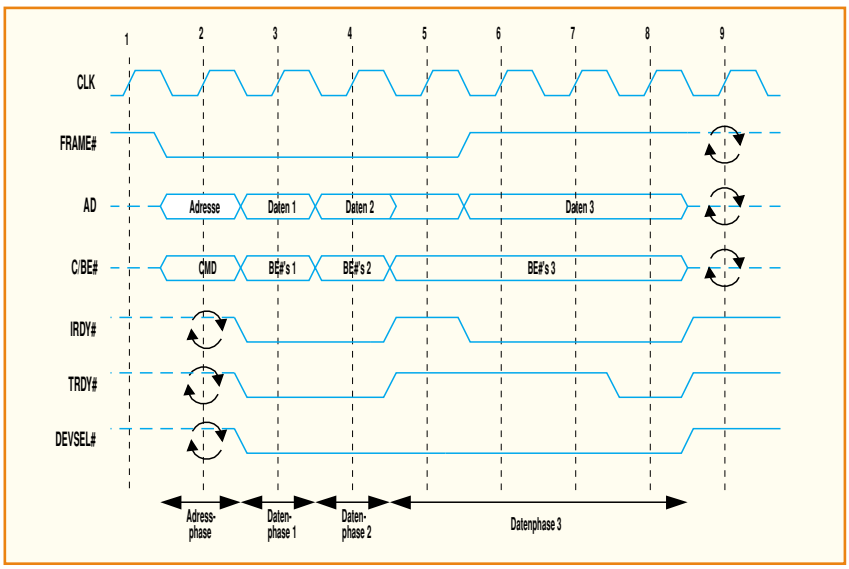

Bild 4: PCI-Schreibtransaktionen