

Adressenverteilung der PCI-Basis.- und Offsetadressen

Stand: 06.2013 / von H.Kolter

Im Gegensatz zu ISA ist **Peripheral Component Interconnect (PCI)** von Anbeginn Plug&Play-fähig. Für alle PCI-Einheiten ist ein Konfigurationsbereich von 256 Byte vorgesehen, um kartenspezifische Eigenschaften dem System beim Booten mitzuteilen. Mit Hilfe dieses Bereiches und dem für PCI erweiterten BIOS (PCI-BIOS) wird eine automatische Konfigurierung der einzelnen PCI-Devices (PCI-Einheiten) erreicht. Zwingend ist jedoch min. ein 64 Byte großer Konfigurationsbereich (configuration space) der die wichtigsten Karten-Daten beinhaltet, damit das System eine erfolgreiche Inbetriebnahme des Device sicherstellen kann.

Die Aufteilung der Adressbereiche innerhalb der PCI-Dekodierung (Chip) ist linear, der Offset ist hingegen von der verwendeten Rechnerkonfiguration abhängig und unterscheidet bzw. verändert sich je nach Ausbaustufe der Zusatzkomponenten, da das PCI-Rechner-BIOS eine Zuteilung der I/O- und Memory-Ressourcen mit dem Boot-Vorgang zwischen dem System und den PCI-Dekodern selbstständig abstimmt. Um dennoch auf eine absolute physikalische I/O-Adresse zu programmieren bzw. zuzugreifen, bedient man sich diverser PCI-Tools, die den Konfigurationsheader des jeweiligen PCI-Dekoders im System ausliest (über INT) und der jeweiligen Anwendung entsprechend mitteilt, bevor dann auf dieses Device zugegriffen wird. Diese PCI-BIOS-Daten sind in jedem System dynamisch angeordnet und können nur über bestimmte Interrupt-Routinen ausgelesen werden, da jeder Motherboard-Hersteller ein anderes BIOS bzw. Memorymodell verwendet und somit keine absoluten Speicheradressen für die Konfigurationsdaten zur Verfügung gestellt werden können. Zu diesem Thema gibt es ausreichend Literatur, wozu wir auch jetzt nicht weiter auf diese Programmierweise eingehen möchten.

Die Klassifizierung einer PCI-Karte wird durch den im Konfigurationsheader enthaltenen Datensätzen wie Vendor-ID und Produkt-ID erreicht. Über diese Parameter lässt sich eine Karte erkennen und entsprechend einbinden. Jeder Hersteller hat WELTWEIT eine Vendor-ID, über die sich seine Produktserie darstellen lässt. Damit auch eine klare Zuweisung seiner Produkte im System möglich wird, besitzt jede PCI-Karte eine eigene Produkt-ID. Über weitere Datensätze, die in jedem PCI-Dekoder enthalten sind, lässt sich die jeweilige Karte noch tiefer klassifizieren.

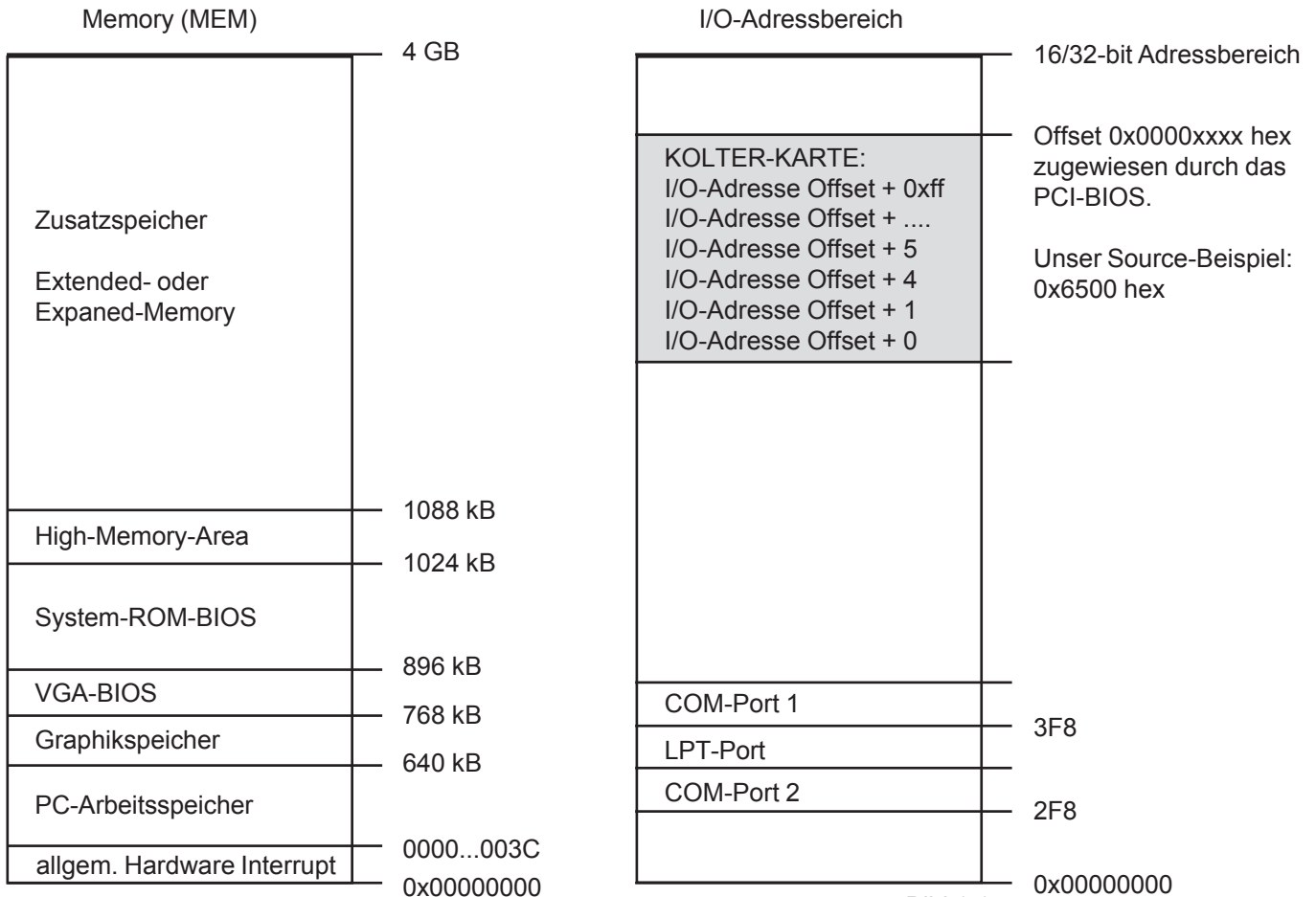


Bild 1.1

Adressenverteilung der PCI-Basis.- und Offsetadressen

Die interne Aufteilung des PCI-Dekoders hinsichtlich der linearen Adressverteilung ist wie folgt spezifiziert:

```
KOLTER-KARTE:  
I/O-Adresse Offset + 0xff  
I/O-Adresse Offset + ....  
I/O-Adresse Offset + 5  
I/O-Adresse Offset + 4  
I/O-Adresse Offset + 1  
I/O-Adresse Offset + 0
```

Der PCI-Dekoder besitzt eingangsseitig einen 32-bit PCI-Adress/Datenbus und am Ausgang einen 16-bit User-Bus, der Ähnlichkeiten mit einem normalen ISA- oder Z80-Standard-Bus aufweist. Neben der State-Machine für den Datenverkehr zwischen configuration-space, pci-interface und Steuerlogik besitzt unser PCI-Dekoder noch einen TCB-Block (timing-correction-block) der das User-Bus-Timing auf ein normales 80er-Niveau konvertiert und somit jeden Peripherie-Baustein am PCI-Bus zulässt.

Der Adressbereich des PCI-Dekoders umfasst immer eine feste Blockgröße von 256 Bytes (00...FFh). Da jedoch ausgangsseitig ein 16-bit-Datenbus verwendet wird, können nur insgesamt 128 Adressen ausdekodiert werden, was jedoch völlig ausreichend sein dürfte. Alle dazwischenliegenden I/O-Adressen werden ignoriert und können bzw. dürfen nicht verwendet werden. In der fortlaufenden Adressierung bilden sich somit Lücken von jeweils 2 Byte, nach jeweils 2 Byte, die Verwendung finden.

Beispiel:

```
var  offset      : word;  
     a           : byte;  
  
offset := 0x6500;           // nicht redundant, vom PCI-BIOS vergeben  
  
out(offset + 0,byte);      // schreibt auf die erste Adresse der PCI-Karte  
out(offset + 1,byte);      // schreibt auf die zweite Adresse der PCI-Karte  
out(offset + 2,byte);      // Fehler !!!  
out(offset + 3,byte);      // Fehler !!!  
  
a := inp(offset + 4,byte);  // liest auf der vierten Adresse der PCI-Karte  
a := inp(offset + 5,byte);  // liest auf der fünften Adresse der PCI-Karte  
a := inp(offset + 6,byte);  // Fehler !!!  
a := inp(offset + 7,byte);  // Fehler !!!
```

u.s.w.

Der Grund, warum nur ein 16-bit-Daten-Bus verwendet wurde, liegt in der begrenzten Anschlussmöglichkeit des LATTICE-Bausteins ispLSI1032 (72 I/O-Pins) sowie in der Gatterzahl von nur 6000 PLD-Gates bzw. 128 Makrozellen.

Da der cPLD-Baustein jedoch als PLCC-84-Gehäuse zur Verfügung steht, ist er ideal für schnelle und einfache PCI-Entwicklungen, die nur eine geringe Einarbeitung in das Thema PCI fordern. Eines der wichtigsten Kriterien ist die Lösung: "time-to-market", die dieser Baustein für sich reklamiert, da Prototyping und vorhandene Schaltpläne, Layouts und Know-How eine rasche Einführung neuer Produkte ermöglicht.

Bei Neuentwicklungen oder Re-Design von ISA-Karten auf PCI-Bus sind wir Ihnen gerne behilflich.

KOLTER ELECTRONIC
Steinstrasse 22
50374 Erftstadt
Tel. 02235-76707
Fax. 02235-72048
<http://www.pci-card.com>