

AKBPII - Hauptseminarvortrag

Thema: PCCard: Architektur und Treiberdesign

Michael Schmidt

(swmeschm@cip.informatik.uni-erlangen.de)



Inhalt

- Zielsetzung
- PCI-Bus
- PCMCIA
- Treiberdesign
- Zusammenfassung

Zielsetzung

- PCMCIA in JX
- Hotplugfähigkeit
- PC-Cards und CardBus Unterstützung

PCI-Bus

- PCI: Peripheral Component Interconnect
- CPU unabhängig, hochperformant
- PCI-Adresse --> Configuration Space
 - CONFIG_ADDRESS (0cf8h)
 - CONFIG_DATA (0xcfch)

PCMCIA

- Entstehung
 - PCMCIA: Personal Computer Memory Card Interface Association
 - Gründung 1989
 - Ziel: Standardisierung von Memory Cards / I/O Cards
- PCMCIA-Socket ist PCI-Device

PCMCIA

- Zugriffsmöglichkeiten auf Karten
 - Memory Mapping (ursprünglich)
 - I/O Mapping (ab Version 2.0)

Configuration Space zum T11520

Table 4-1. PCI Configuration Registers (Functions 0 and 1)

REGISTER NAME		OFFSET
Device ID	Vendor ID	00h
Status	Command	04h
Class code		08h
BIOS	Header type	Revision ID
	Latency timer	Cache line size
CardBus socket/ExCA base address		0Ch
Secondary status		10h
CardBus latency timer	Reserved	Capability pointer
	CardBus bus number	PCI bus number
CardBus Memory base register 0		18h
	CardBus Memory limit register 0	1Ch
	CardBus Memory limit register 1	20h
	CardBus Memory limit register 2	24h
	CardBus Memory limit register 3	28h
	CardBus I/O base register 0	2Ch
	CardBus I/O limit register 0	30h
	CardBus I/O base register 1	34h
	CardBus I/O limit register 1	38h
Bridge control	Interrupt pin	Interrupt line
Subsystem ID	Subsystem vendor ID	40h
PC Card 16-bit I/F legacy-mode base address		44h
Reserved		48h-7Ch
System control		80h
Reserved		84h-88h
Multifunction routing		8Ch
Diagnostic	Device control	Card control
		Retry status
Reserved		90h
Power-management capabilities		94h-9Ch
	Next-item pointer	Capability ID
Power-management data	Power-management control/status support extensions	Power-management control/status
	General-purpose event enable	General-purpose event status
	General-purpose output	General-purpose input
Serial bus control/status	Serial bus slave address	Serial bus index
	Serial bus data	Serial bus data
Reserved		B0h
		B4h-FC h

PCMCIA

■ PCCard / CardBus

Unterschiede zwischen PCMCIA und Card-Bus

Parameter	PCMCIA	Card-Bus
Schnittstelle	68 Pins	68 Pins
Takt (max.)	asynchron	33 MHz
Bandbreite	8-20 MByte/s	132 MBytes/s
Datenbusbreite	16 Bit	32 Bit
Adressierung	26 Bit	32 Bit
Spannung	5V, 3,3V	3,3V
Interruptkanäle	1	1
Konfiguration	Attributspeicher	Configuration Space
Karteninformationen	CIS	Configuration Space

■ PCCard - CIS

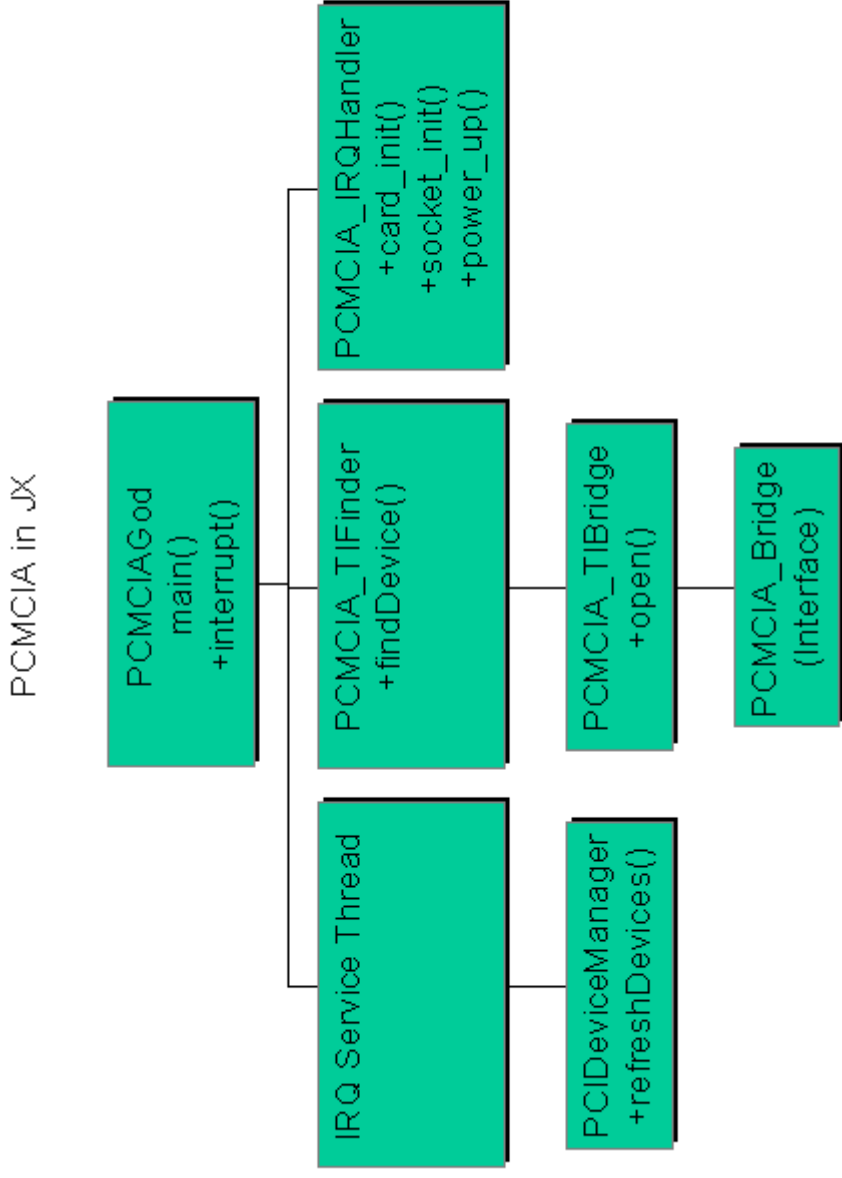
■ CardBus

- PCI-Geräten sehr ähnlich
- Configuration Space

PCMCIA

- TI 1520: kompatibel zu Intel 82365SL-DF
- ExCA Register
 - Konfiguration der Steckkarten
 - Statusinformationen
 - Steuerinformationen
- CardBus Register

Treiberdesign - Klassenhierarchie



PCMCIAGod

- Startpunkt
- IRQ-Handler
- startet Finder
- erzeugt PCMCIA_IRQHandler pro Socket
- erzeugt Service Thread

PCMCIA_TIFinder

- Finder-Klasse
- ermittelt PCMCIA-Bridges
- erzeugt PCMCIA_Bridge Objekte

PCMCIA_Bridge / PCMCIA_TIBridge

- PCMCIA_Bridge: Interface zwecks Kompatibilität
 - Methoden:
 - getPCIDevice(), getIRQ(), getRegister(), open()
- kapselt PCIDevice (Socket-Objekt)
- TIBridge
 - Implementierung für TI 1520 Bridge
 - open(): legt IRQ für Socket fest
 - IRQ: via Args, oder aus PCI Configuration des Sockets

PCMCIA_IRQHandler

- zentrales Verwaltungs-Objekt für Socket
- Methoden:
 - `card_init()`, `socket_init()`, `power_up()`, `cardtype_cb()`

IRQ Service Thread

- blockiert an AtomicVariable
- Übergabe-Parameter: PCIDevice zur Deblockierung
- Portal-Aufruf refreshDevices() an PCIDeviceManager --> Treiber nachladen bzw. entfernen
- Socket neu initialisieren

Zusammenfassung

- PCMCIA-Sockets
- CardBus-Karten
- Vorbereitung für PCCard-Karten

Vielen Dank für die Aufmerksamkeit!

Fragen??