Entwicklung eines Mac OS X Treibers für eine PCI-VME Interface Karte

Matthias Lange
Informatikstudent, TU-Dresden

27. September 2005

http://www.matze-lange.de

Warum entwickelt jemand einen Treiber für Mac OS X?

Darum!

- Ersetzen des bestehenden Mac/ MacVEE Aufbaus durch G4 Mac/ SIS1100 Systeme

- Umstieg vom klassischen Mac OS auf Mac OS X

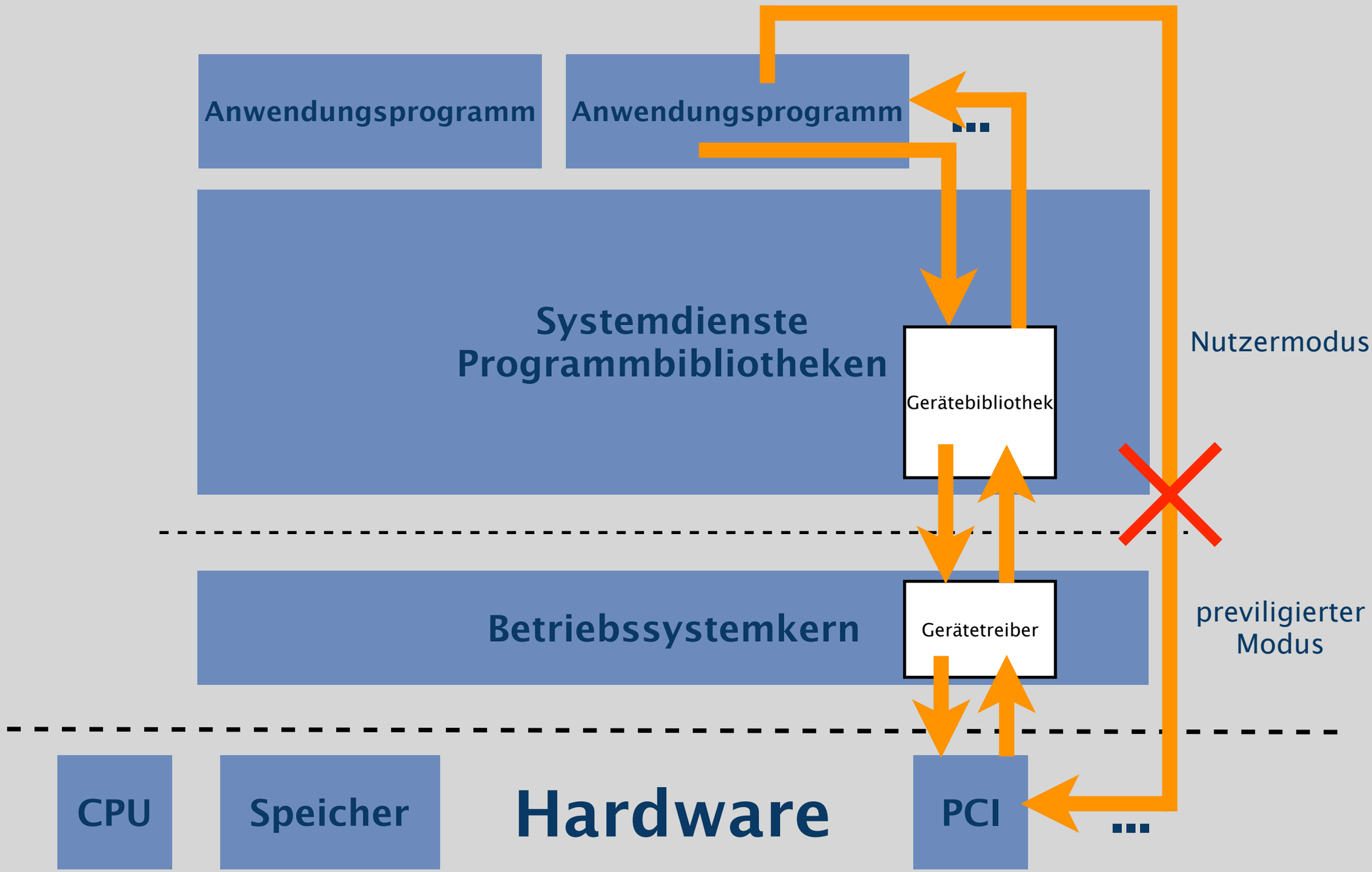
- erwartete höhere Leistungsfähigkeit für erweiterte Tests nutzen

- neue Hardware vorhanden, aber kein Treiber für Mac OS X

Einleitung

- Aufbau moderner Betriebssysteme
  - Struktur von Mac OS X
- SIS1100/SIS3100 Hardware
- Aufbau, Struktur und Funktionsumfang des Treibers
- Optimierungen
- Ergebnis und Ausblick

Aufbau moderner Betriebssysteme

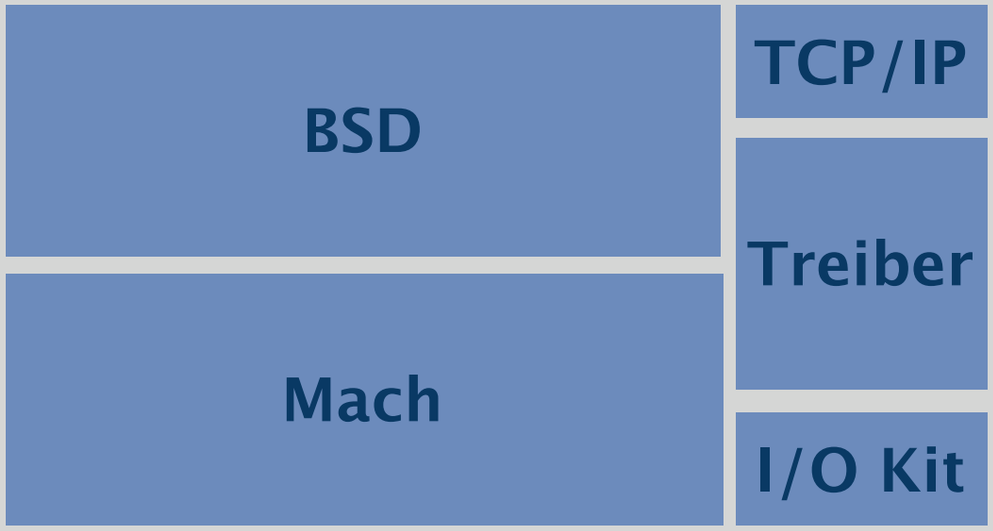


Betriebssystemkern

- exklusiver Zugriff auf die Computerhardware
- Mechanismen für die Konstruktion von Betriebssystemen (z.B. für Kommunikation zw. Programmen)
- verwaltet die Resourcen (CPU-Zeit, Speicher etc.)
- durch virtuellen Speicher Schutz von Anwendungsprogrammen vor gegenseitiger Beeinflussung
- Gerätetreiber

Der Wechsel in den Betriebssystemkern ist ein sehr teurer Vorgang!

Betriebssystemkern von Mac OS X

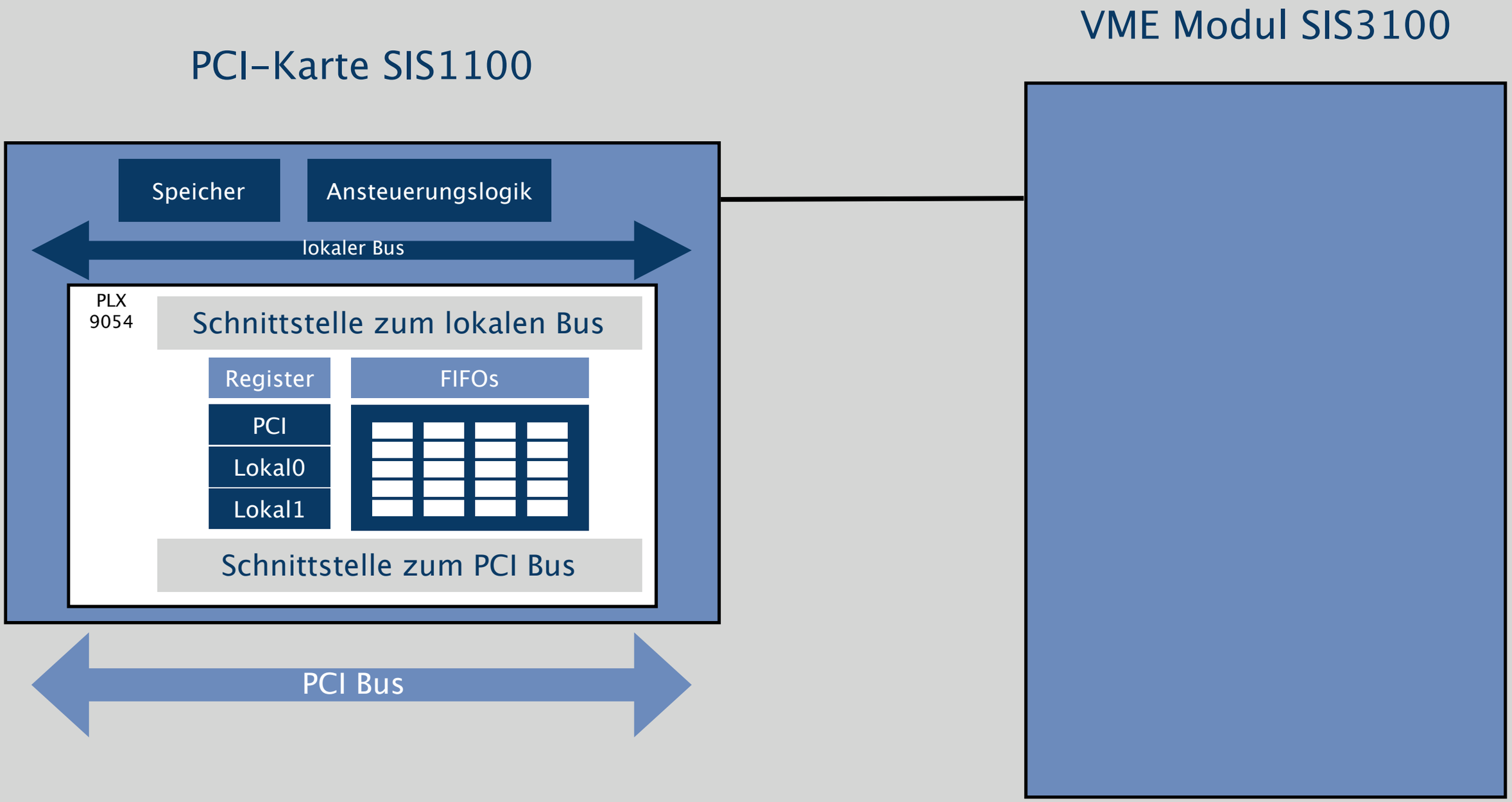


- Mach: Kommunikation, Scheduling, virtueller Speicher
- BSD: Dateisysteme, UNIX Schnittstelle POSIX, UNIX Sicherheitskonzept
- I/O Kit: abstrahiert Hardware und bietet allgemeine Schnittstellen für typische Geräteklassen (z.B. USB, PCI) an

SIS1100/SIS3100 Hardware

- **SIS1100**: PCI-Karte mit PLX9054 als Brückenchip und Controller (VME-Zugriffsprotokoll)
- **SIS3100**: VME Sequencer, VME Master/Slave
- Verbindung über Glasfaser, Kommunikation mittels Gigalink-Protokoll

Aufbau der SIS1100



Entwicklungsschritte

- Zugriff innerhalb des Treibers auf die Register der SIS1100 um Leuchtdioden an- und auszuschalten
- Implementierung des Einzelzugriffs und Zugriffsmöglichkeit für Anwendungen
- Implementierung des Blockzugriffs
- Test in Produktivumgebung, Test der Integration in LabView, Optimierungen
- parallel dazu Entwicklung von Testwerkzeugen

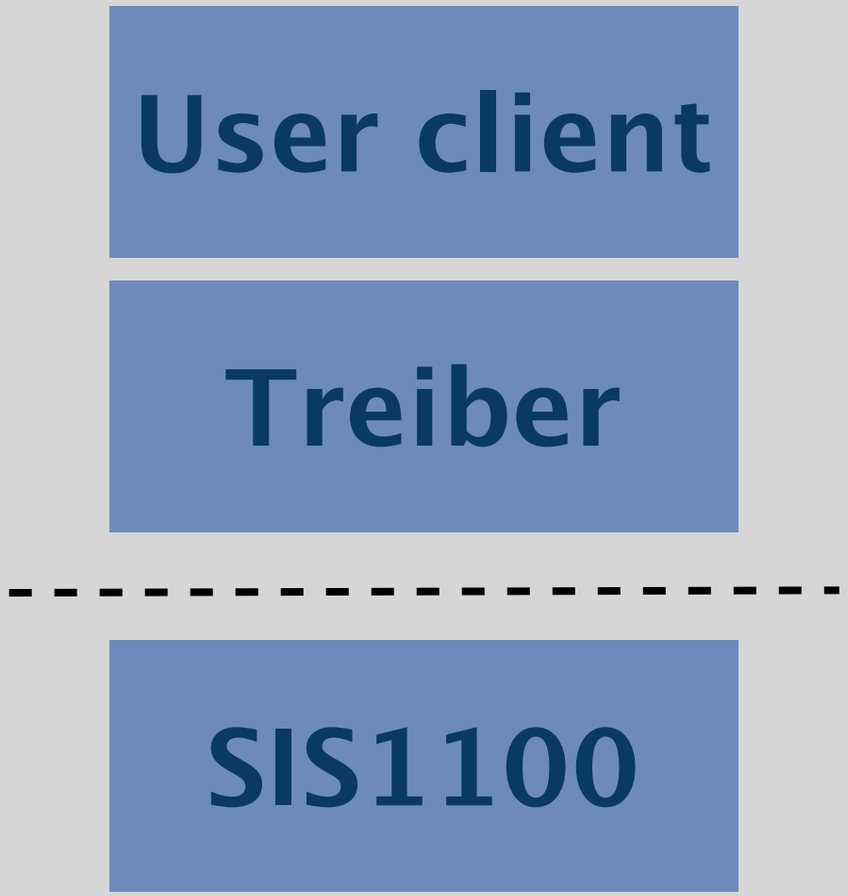
Entwicklungsschritte (Fortsetzung)

- nach WWDC Besuch Integration neuer Techniken um Daten zwischen Programm und Treiber auszutauschen (geteilter Speicher)
  - Kompatibilität mit Mac OS X 10.3 und 10.4
- Implementierung des direkten VME-Bus-Zugriffs durch Speichermapping auf der SIS1100
- Optimierung
- Dokumentation der Quelltexte

Schwierigkeiten

- Schlechte Dokumentation!
- Widersprüchliche Dokumentation! (Big-/Little Endian)
- Fehlende Dokumentation!
- ABER: starke Unterstützung durch Matthias Kirsch bei der Beantwortung meiner Fragen

Aufbau des Treibers



- Treiber
  - programmiert die Register der SIS1100
  - steuert VME Transfers
- User client
  - Schnittstelle für Anwendungsprogramme
  - ruft Funktionen im Treiber auf

Funktionalität des Treibers

- alle Zugriffe lesend oder schreibend und funktionieren mit 8, 16 oder 32bit Werten
- optional mit address modifier
- Einzelzugriff: Zugriff auf eine VME-Adresse
- Blockzugriff: beginnend bei einer VME-Adresse sequentieller Zugriff auf eine bestimmte Anzahl von Werten

Funktionalität des Treibers (Fortsetzung)

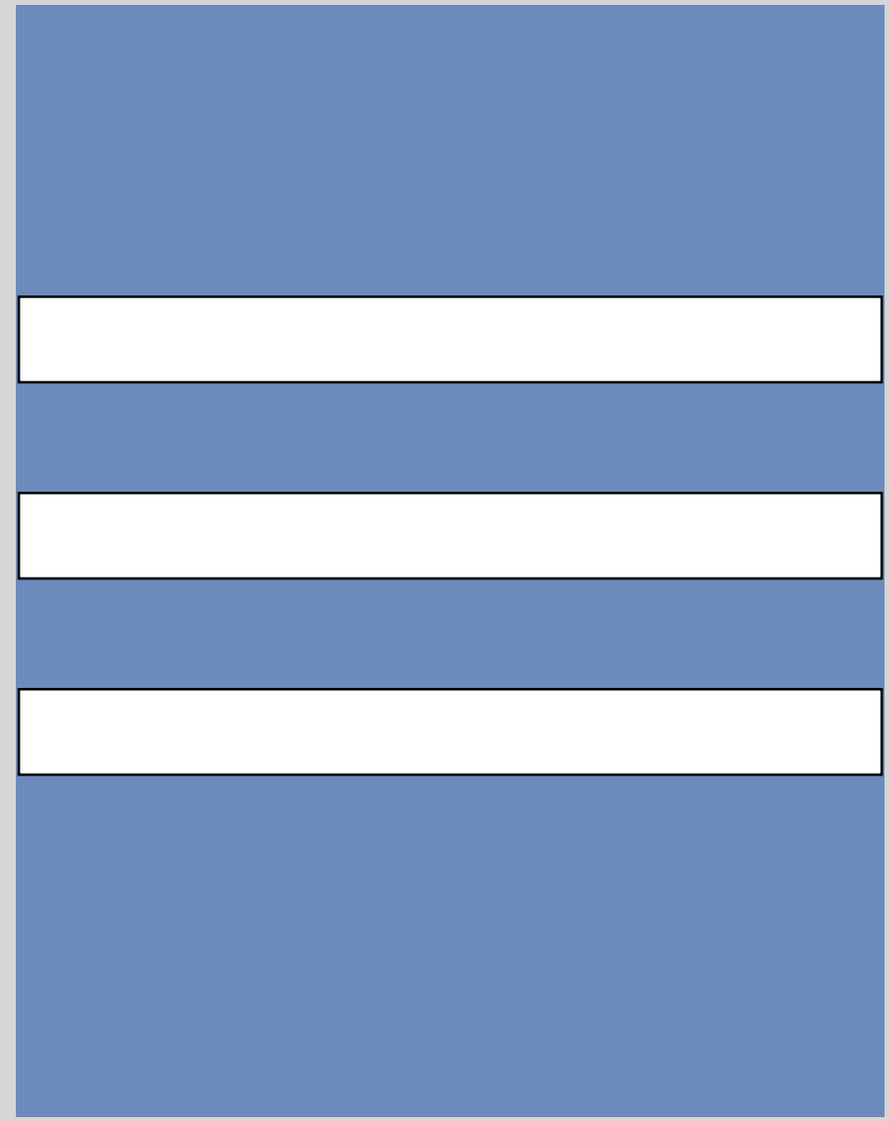
- Listenzugriff: Bündelung von Einzelzugriffen für wahlfreien VME-Zugriff
- direkter Buszugriff: direkter Zugriff auf VME-Adressen über auf der SIS1100 gemappten VME-Speicher (Begrenzung auf 64kB und 64 Mappings)
- Mappings werden separat angelegt und können wiederverwendet werden

Optimierung

- Systemauslastung senken
- geringeren Speicherverbrauch
- Kompromiss zwischen Geschwindigkeit und Speicherverbrauch
- nur sinnvoll bei häufig verwendeten Funktionen
- höhere Geschwindigkeiten beim VME-Zugriff

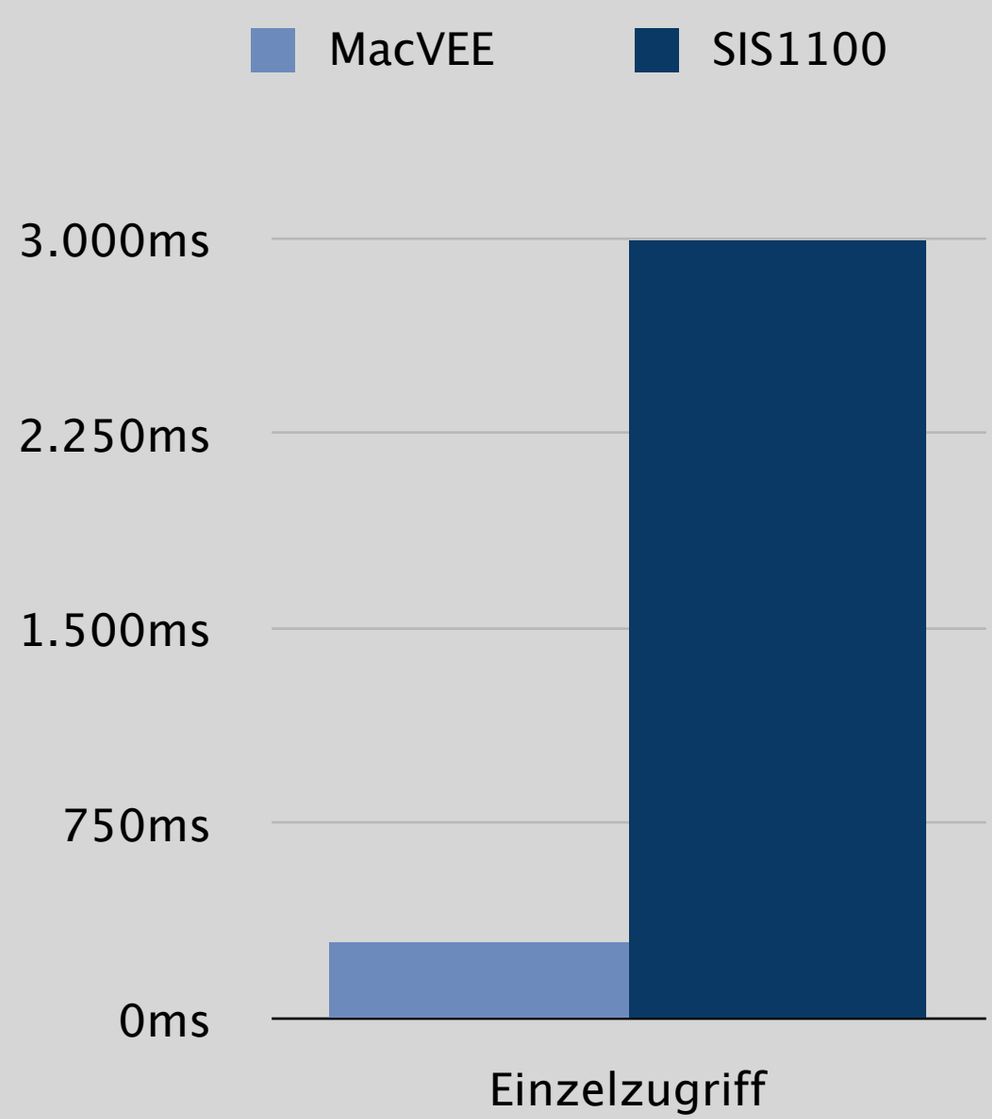
Beispiel: Programmladen beim OnSiROC

- sequentieller Zugriff auf drei Adressen
  - Zeilennummer
  - 16bit Code
  - 16bit Code
- Laden einer Programmzeile, insgesamt mehrere 10.000 mal



Zugriffsdauer im Vergleich zum alten System

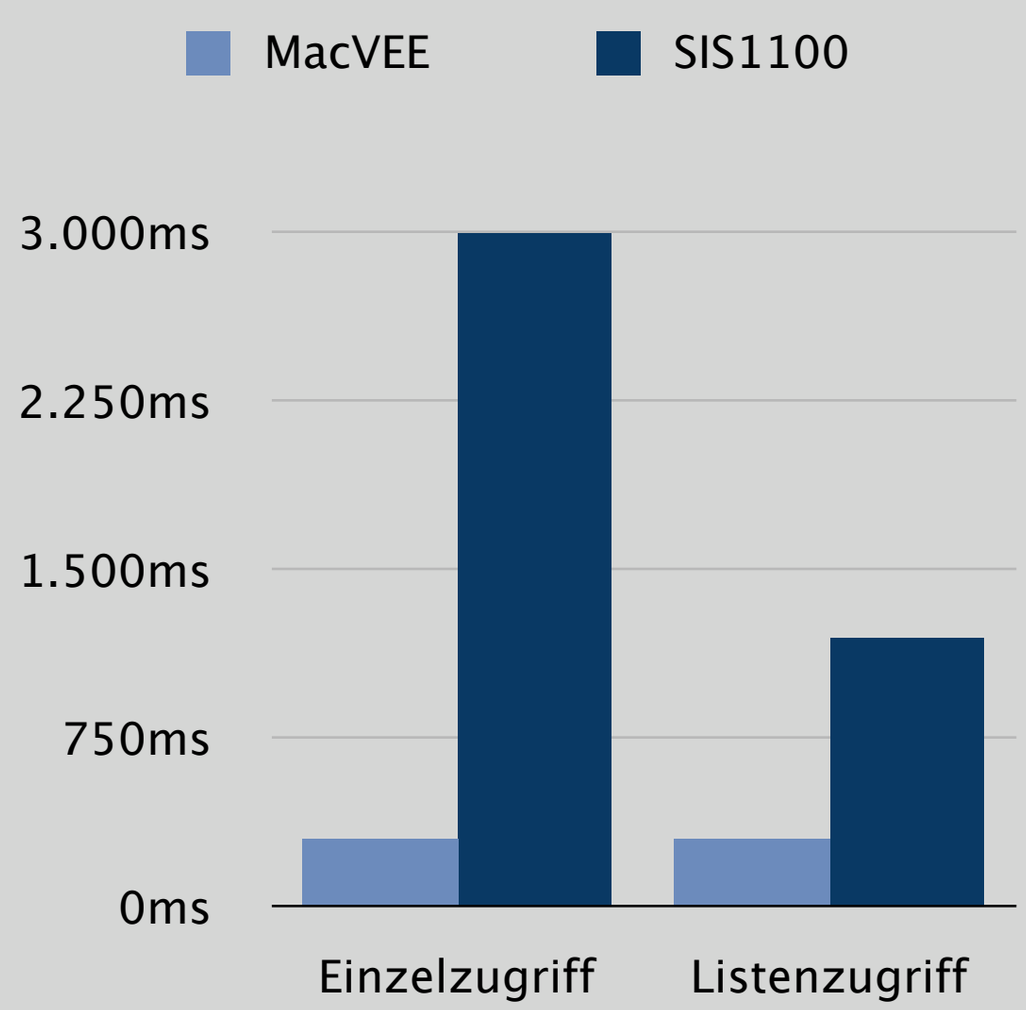
- Implementierung mit Einzelzugriffen (kein Blockzugriff da Adressen nicht sequentiell)
- für jeden Wert
  - ein Kerneintritt (pro Zeile Code drei Mal)
  - 4 Registerzugriffe



(Messung auf G4 Mac mit 400MHz)

Beschleunigung 1. Schritt

- Zusammenfassen der Einzelzugriffe in einer Liste um Kerneintritte zu sparen
- Implementierung der neuen Zugriffsart als Listenzugriff
  - im Treiber weiterhin Einzelzugriffe mit 4 Registerzugriffen je Wert



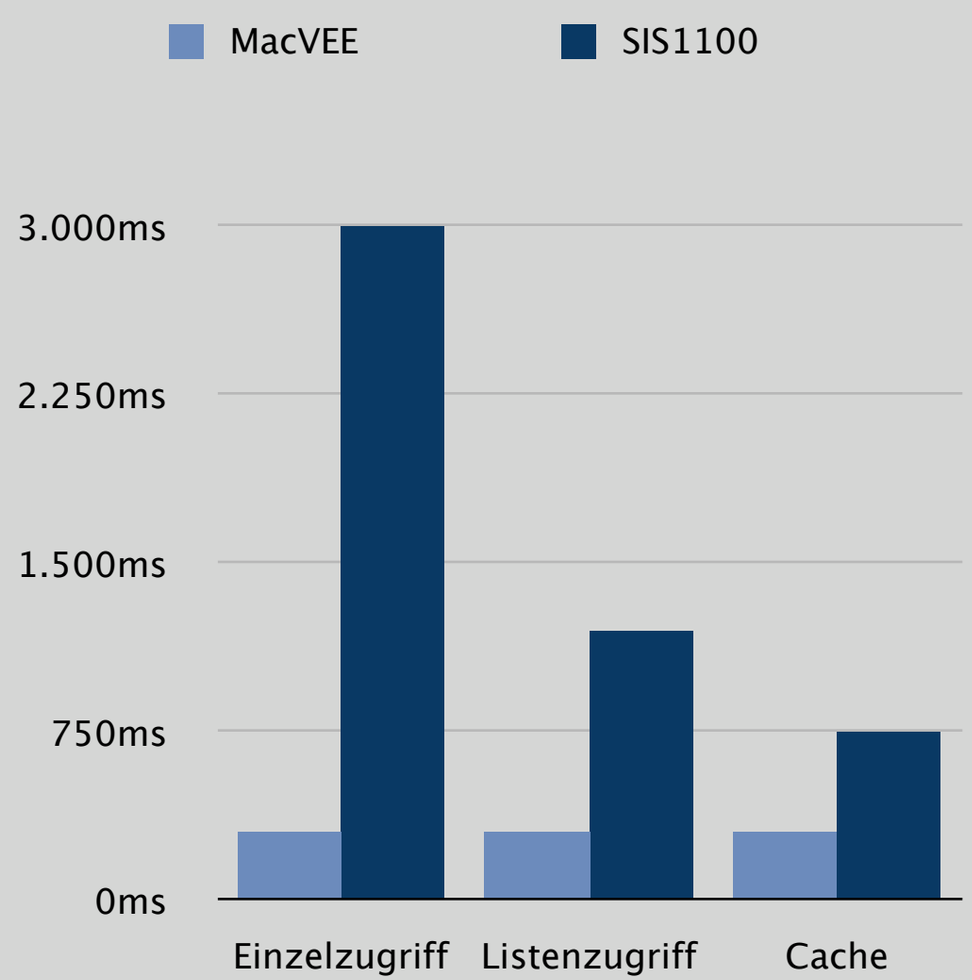
(Messung auf G4 Mac mit 400MHz)

Beschleunigung 2. Schritt

- für jeden Einzelzugriff Schreiben von Protokollkopf, AM, Adresse und Wert

- Beobachtung:
  - Protokollkopf und AM ändern sich (fast) nie

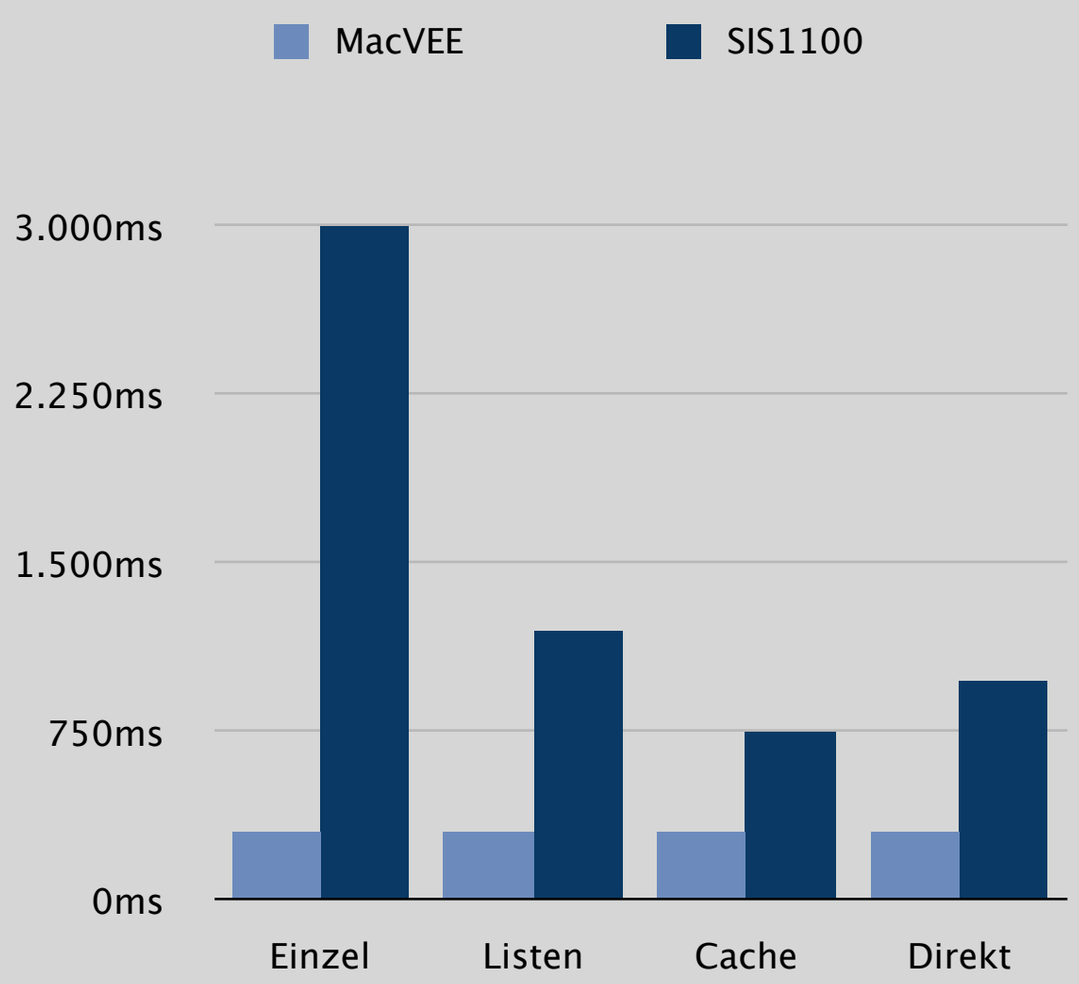
➡ daher nur bei Änderung neu schreiben



(Messung auf G4 Mac mit 400MHz)

Beschleunigung 3. Schritt

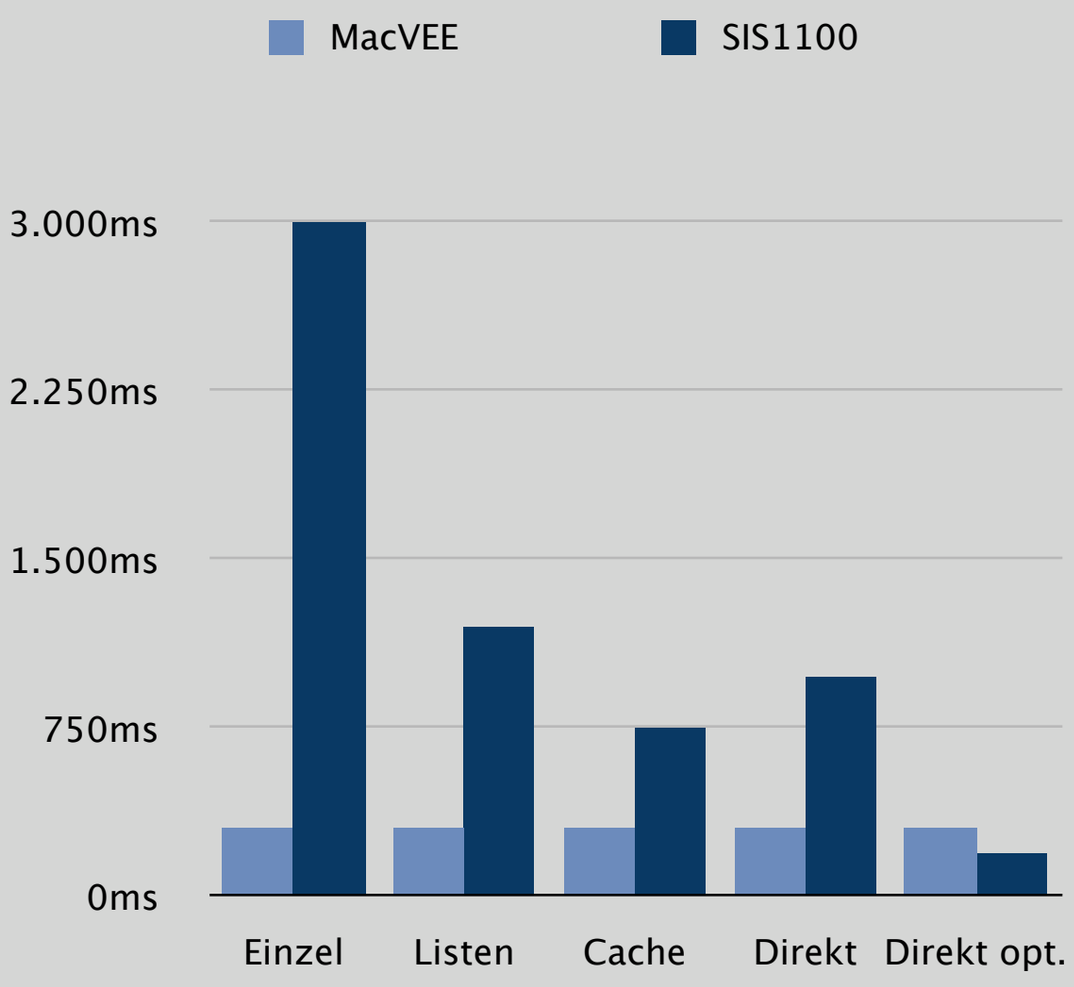
- direkter VME-Bus-Zugriff über Speichermapping
  - Anlegen des Mappings ist vernachlässigbar
- nur noch ein Registerzugriff pro Wert
- Implementierung als neue Zugriffsmöglichkeit



(Messung auf G4 Mac mit 400MHz)

Beschleunigung 4. Schritt

- Schreib- und Lesewarteschlange in PLX9054
- bei Überfüllung verzögert der PLX9054 die Freigabe des PCI-Bus
- Überfüllung durch einmaliges Lesen oder Schreiben verhindern



(Messung auf G4 Mac mit 400MHz)

Beschleunigung 4. Schritt

- Schreib- und Lesewarteschlange in PLX9054
- bei Überfüllung [illegible] der PLX905[illegible] des [illegible]
- [illegible]ung durch einmaliges Lesen oder Schreiben verhindern

(Messung auf G4 Mac mit 400MHz)

Beschleunigung um Faktor 16

1.5 mal schneller als MacVEE

Ergebnis

Dokumentation

SIS.framework
(Bibliothek)

SIS1100Driver
(kompatibel mit PPC und Intel)

Poke VI

Peek VI

SISMonitor

ZaradannX
(Vorschau)

SISPerformance

... there is one more thing

maccent

Ausblick

- Speichermapping von VME-Speicher in Anwendungsprogramme für direkten VME Zugriff
- Implementierung von DMA-Transfers (für die Übertragung großer sequentieller Datenblöcke)
- Wunsch: Äquivalenz zum Linux-Treiber
  - nur mit direkter Unterstützung von Struck realisierbar, weil das Wissen nicht vorhanden ist (über VME und SIS Hardware)

Fragen?

Quellen und Hinweise

http://developer.apple.com	Mac OS X, I/O Kit
http://www.struck.de	SIS1100/SIS3100 Hardware
http://www.maccent-software.de	SIS1100Driver, Dokumentation
http://www.matze-lange.de	private Homepage