



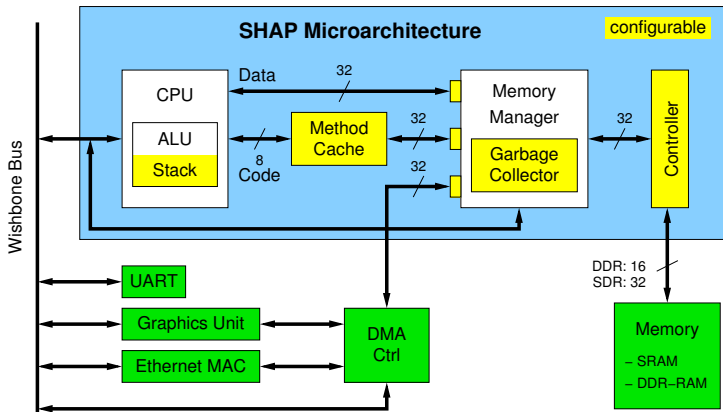
Multi-Port-Speichermanager für die Java-Plattform SHAP

DASS 2008

Martin Zabel, Peter Reichel, Rainer G. Spallek

Dresden, 16.05.2008

SHAP-Mikroarchitektur



Gliederung

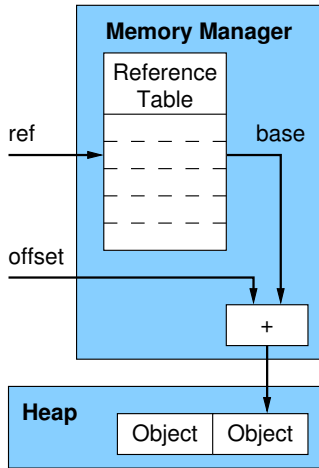
- Objektorientierter Speicherzugriff
- Single-Port-Speichermanager
- Multi-Port-Speichermanager
- Anwendung
- Zusammenfassung

Motivation

- Unterstützung objektorientierter Konzepte in Hardware.
- Speichermanager zur autonomen Verwaltung des Heaps.
- High-Level-Kommandos.
- Anbindung mehrerer Komponenten, wie Multi-Core, DMA.
- Beispiel: Java-Plattform SHAP.

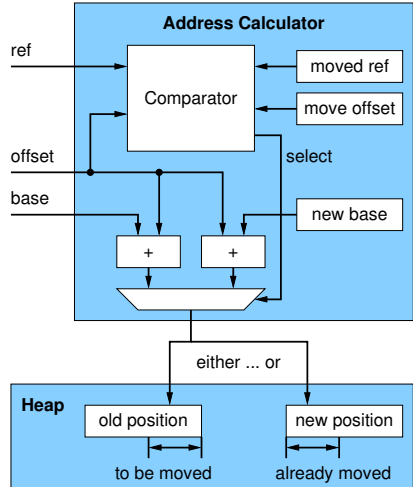
Objektorientierter Speicherzugriff

- Adressierung mittels Objektreferenz und Objektoffset.
- Direkt: $\text{base} = \text{ref}$.
- Indirekt: $\text{base} = \text{table}[\text{ref}]$.
 - Günstig für *Copying Garbage Collection*.
 - Einsatz in der SHAP-MA.
 - *Aktivierung einer Referenz*.
 - Laufzeit für Aktivierung kann „versteckt“ werden.



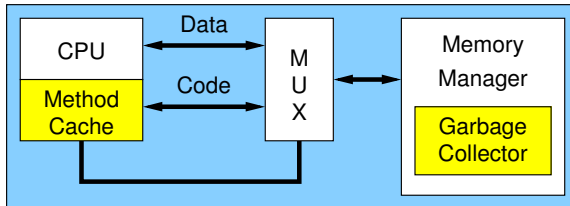
Einsatz von SHAP in Echtzeitsystemen

- Hardware-integrierter, nebenläufiger *Garbage Collector*.
- Nebenläufige Objektverschiebung.
- Automatische Umlenkung von Speicherzugriffen.



Single-Port-Speichermanager

- Multiplexer-Schaltung für die Anbindung von CPU und Methoden-Cache.
- Steuerung durch Methoden-Cache.
- **Aber:** Nur eine Referenz aktivierbar.



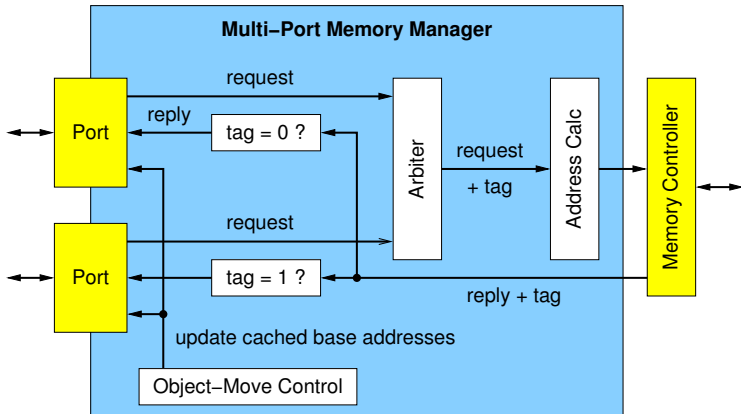
Nachteile des Multiplexer-Prinzips

- Atomare Operationen
bestehend aus Referenzaktivierung und Lese-/Schreibzugriff.
- Ständiger Wechsel der Referenz.

Lösung:

Ein eigenständiger Port je angeschlossener Komponente.

Multi-Port-Speichermanager

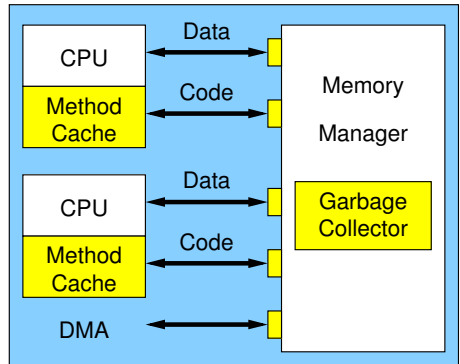


Multi-Port-Speichermanager

- **Port:**
 - Zerlegung der High- in Low-Level-Kommandos.
 - Caching der Basisadresse.
- Arbitrierung auf Ebene der Low-Level-Kommandos.
- Getrennte Kanäle für *Requests* und *Reply*.
 - Zuordnung mittels *Tags*.
 - Pipelining \Rightarrow Effiziente Abarbeitung.
- Weiterer Kanal für das Update der in den Ports zwischengespeicherten Basisadressen.

Anwendung

- Optimierung der Methoden-Cache-Anbindung.
- *Direct Memory Access* für *Graphics Processing Unit*.
- Zukünftig: *Multi-CPU* und *Multi-Core*.



Zusammenfassung

- **Problem:** Indirektionskonzept des in Hardware integrierten, nebenläufigen *Garbage Collectors*.
- **Lösung:** Speichermanager mit mehreren Ports:
 - Gleichzeitigen und unabhängigen Objektzugriff für verschiedene Komponenten.
 - Arbitrierung der Low-Level-Kommandos.
 - Pipelining mittels getrennter Kanäle für *Requests* und *Replies*.
- **Anwendung:** Methoden-Cache, DMA, Multi-CPU, Multi-Core, ...

<http://shap.inf.tu-dresden.de>