



Multicore im Embedded System – Wie geht das?

Renate Schultes

MicroConsult GmbH

Mario Cupelli

HighTec EDV-Systeme GmbH

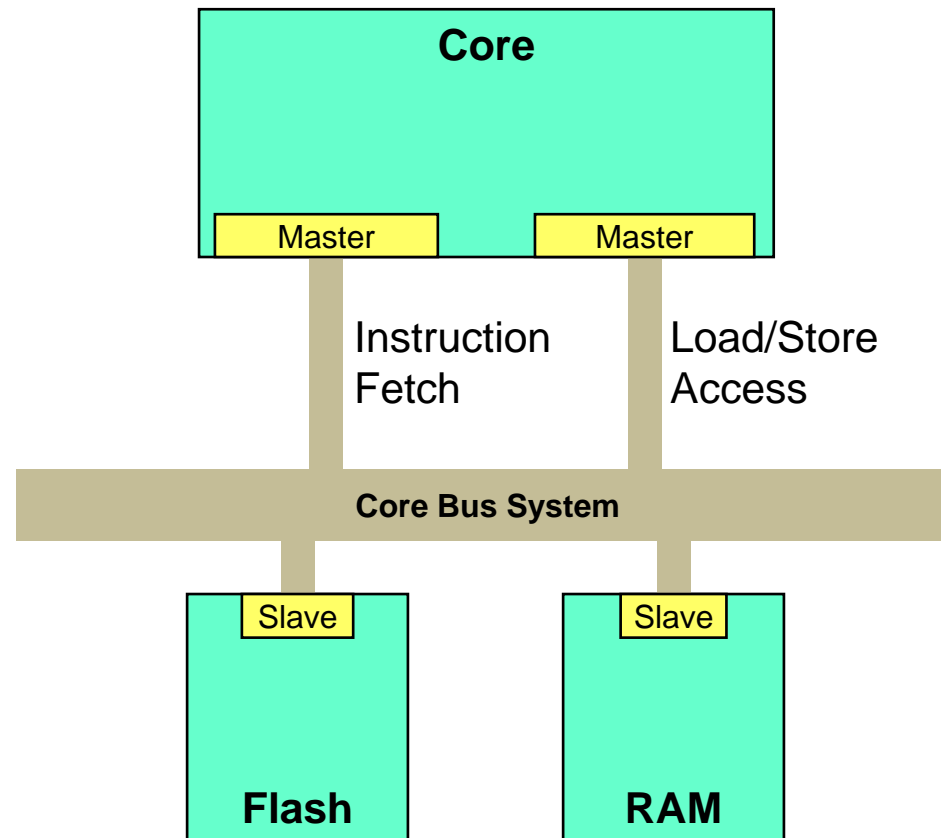
Agenda

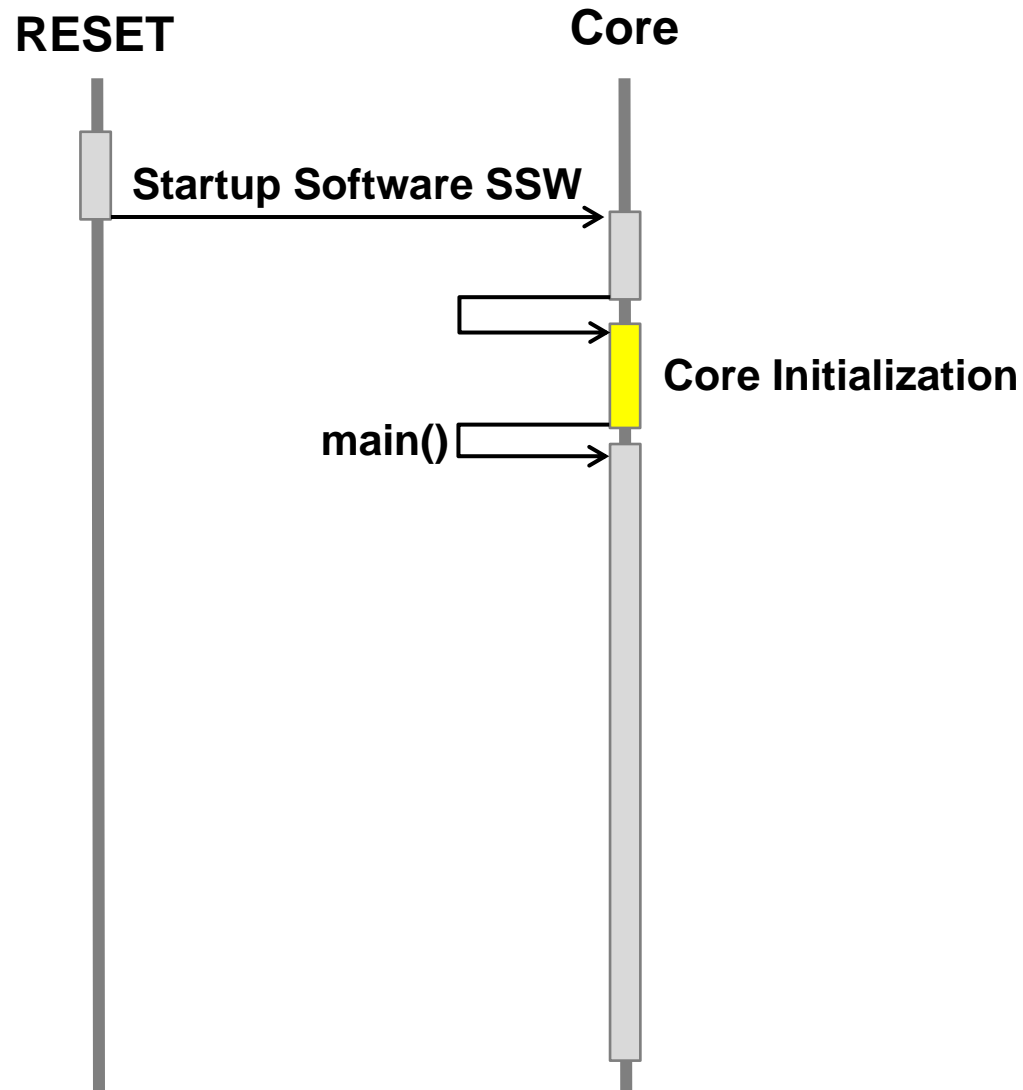
- **Grundlagen**
- **Aufwand und Nutzen**
- **Toolunterstützung**
- **Zusammenfassung**

Agenda

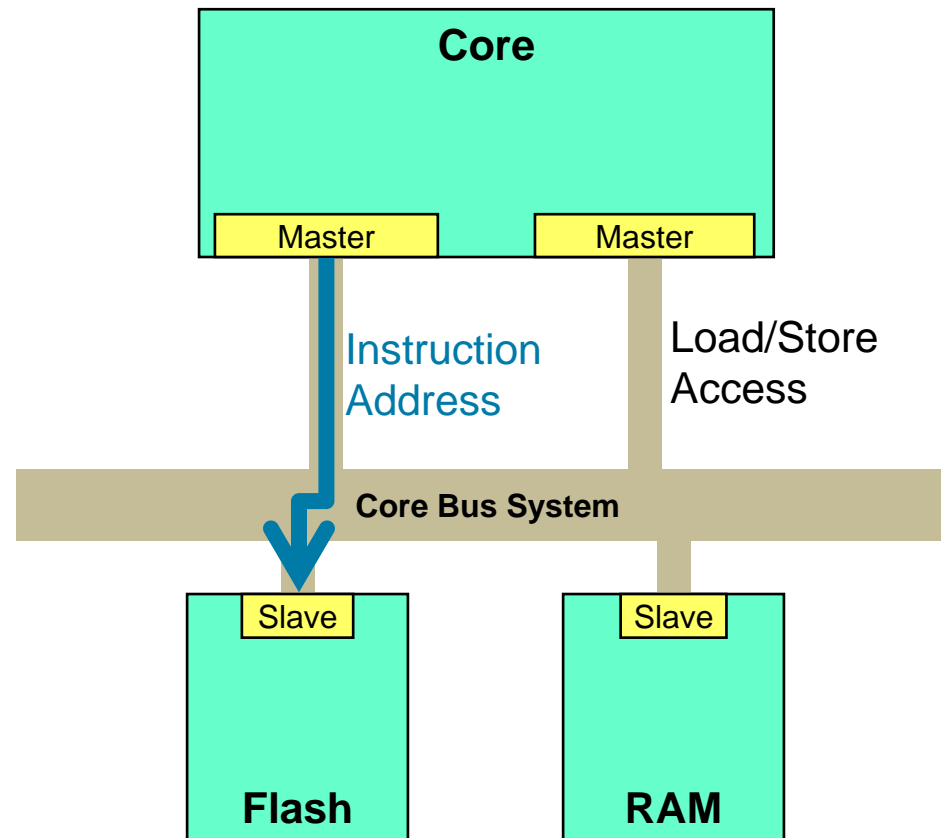
- **Grundlagen**
- Aufwand und Nutzen
- Toolunterstützung
- Zusammenfassung

Singlecore System mit Programm- und Datenspeicher

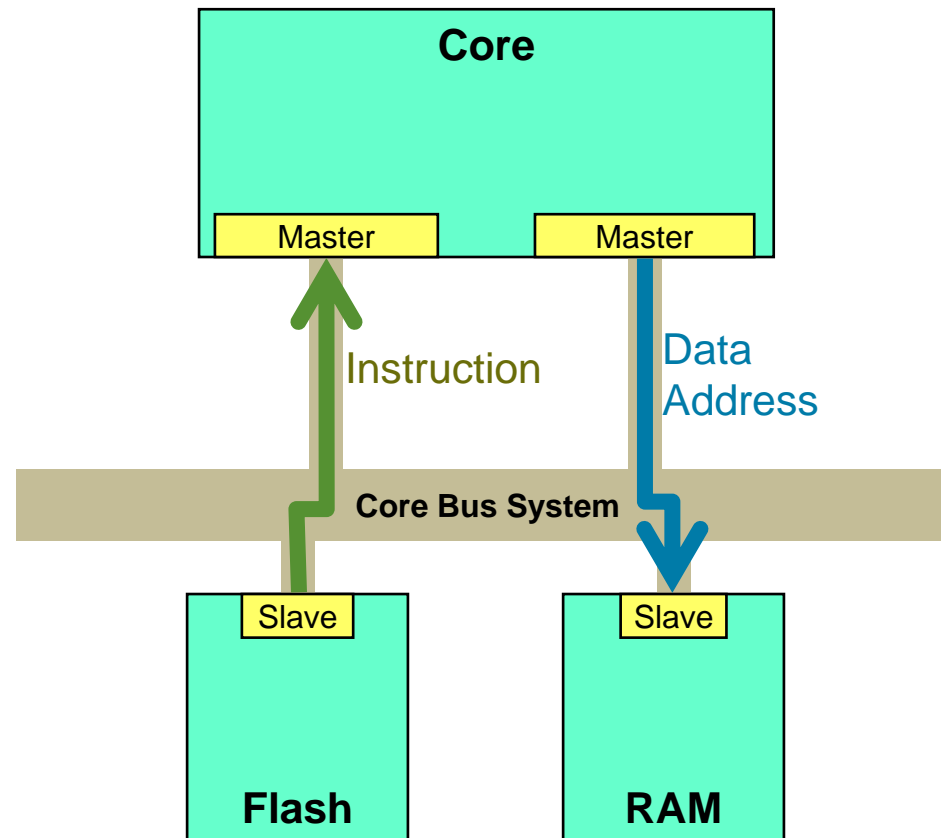




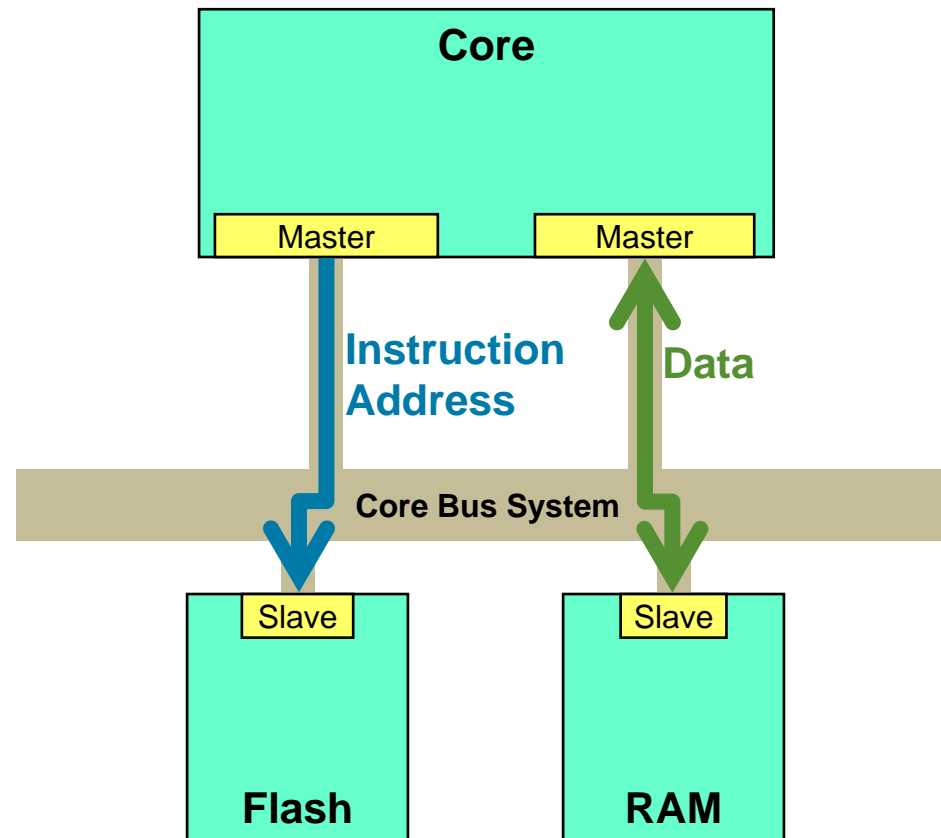
Singlecore System mit Programm- und Datenspeicher



Singlecore System mit Programm- und Datenspeicher



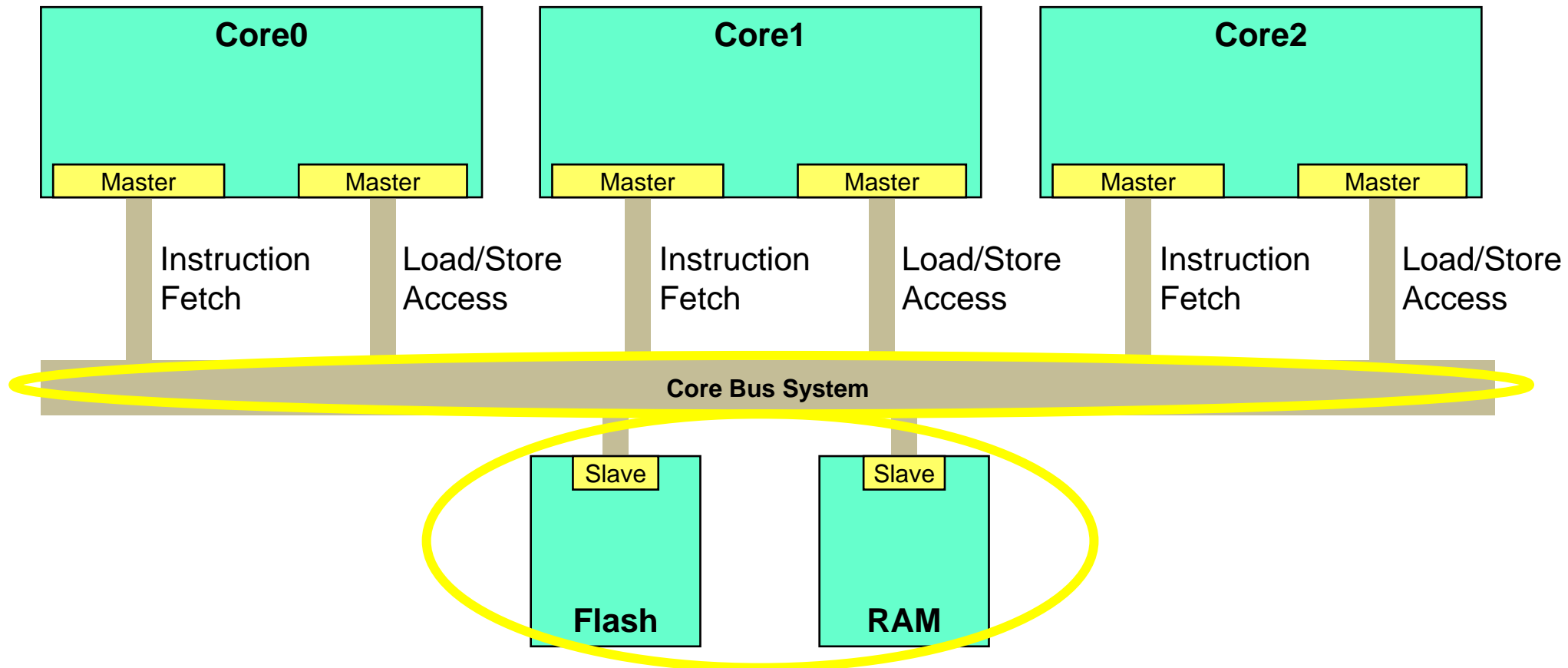
Singlecore System mit Programm- und Datenspeicher

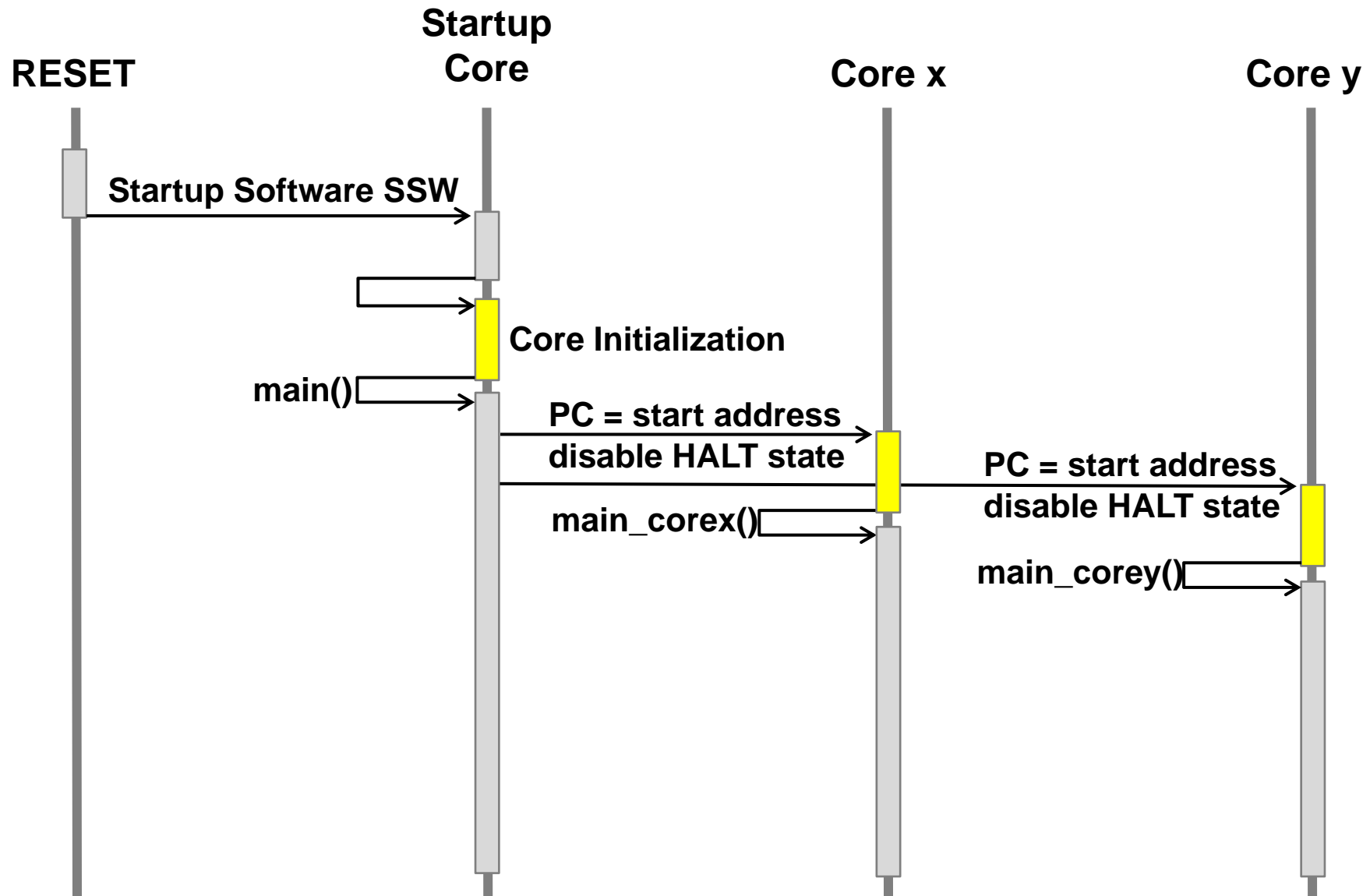


Wie kann die Performanz einer Embedded Applikation gesteigert werden?

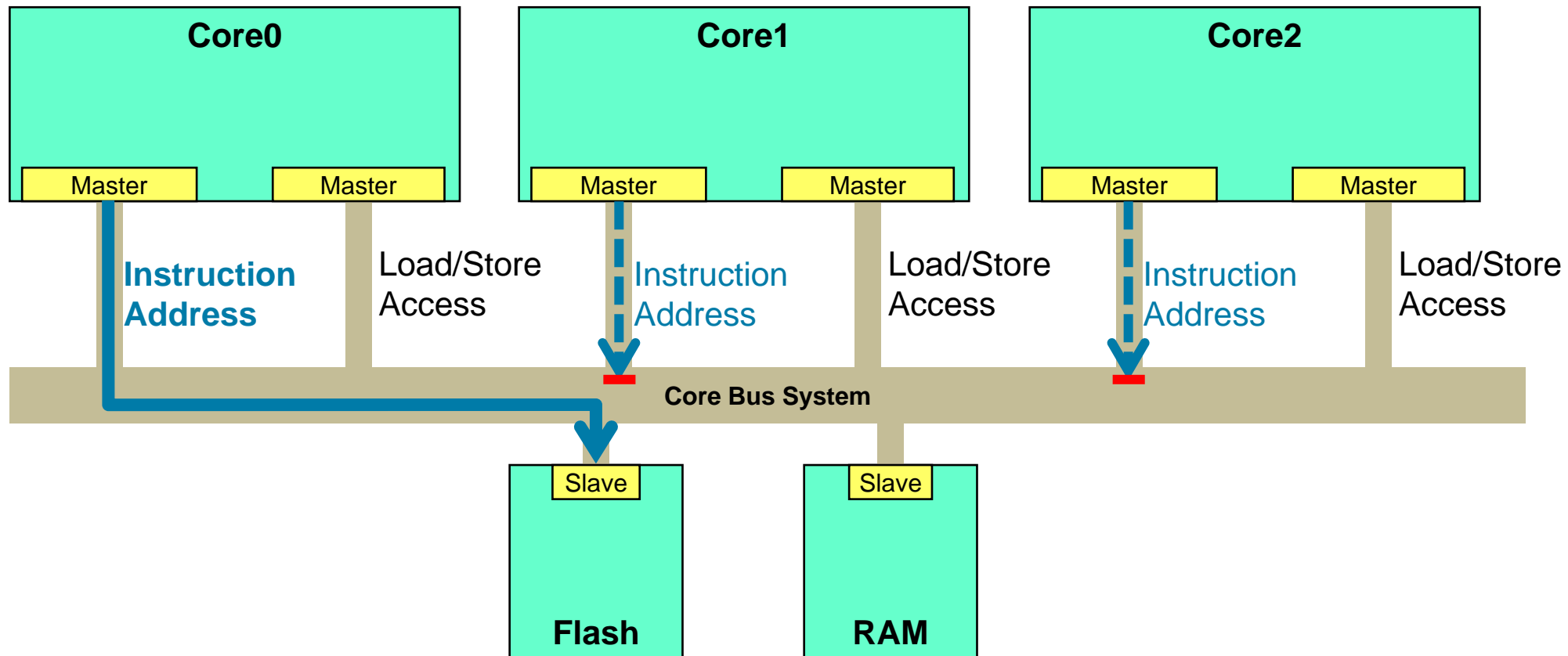
- **Erhöhen der Taktfrequenz** für die Befehlsabarbeitung:
 - Kostet mehr Strom!
 - Ist im Mikrocontroller nur begrenzt möglich!
- Eine **Verdopplung der Core-Anzahl** kann bei parallel ausgeführter Software zu einer Geschwindigkeitssteigerung um nahezu Faktor 2 führen.
 - Bei niedrigerem Takt beinahe Halbierung der Laufzeit!

Multicore System mit globalem Programm- und Datenspeicher

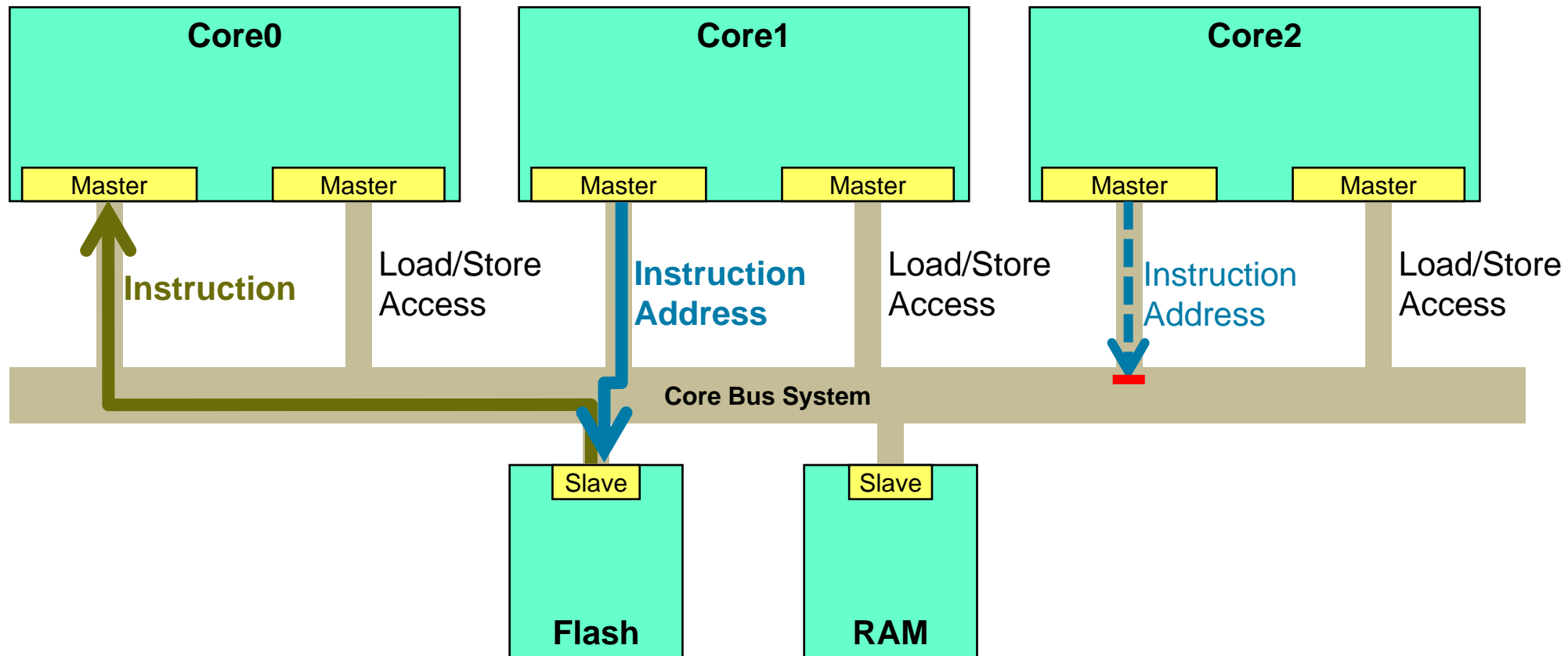




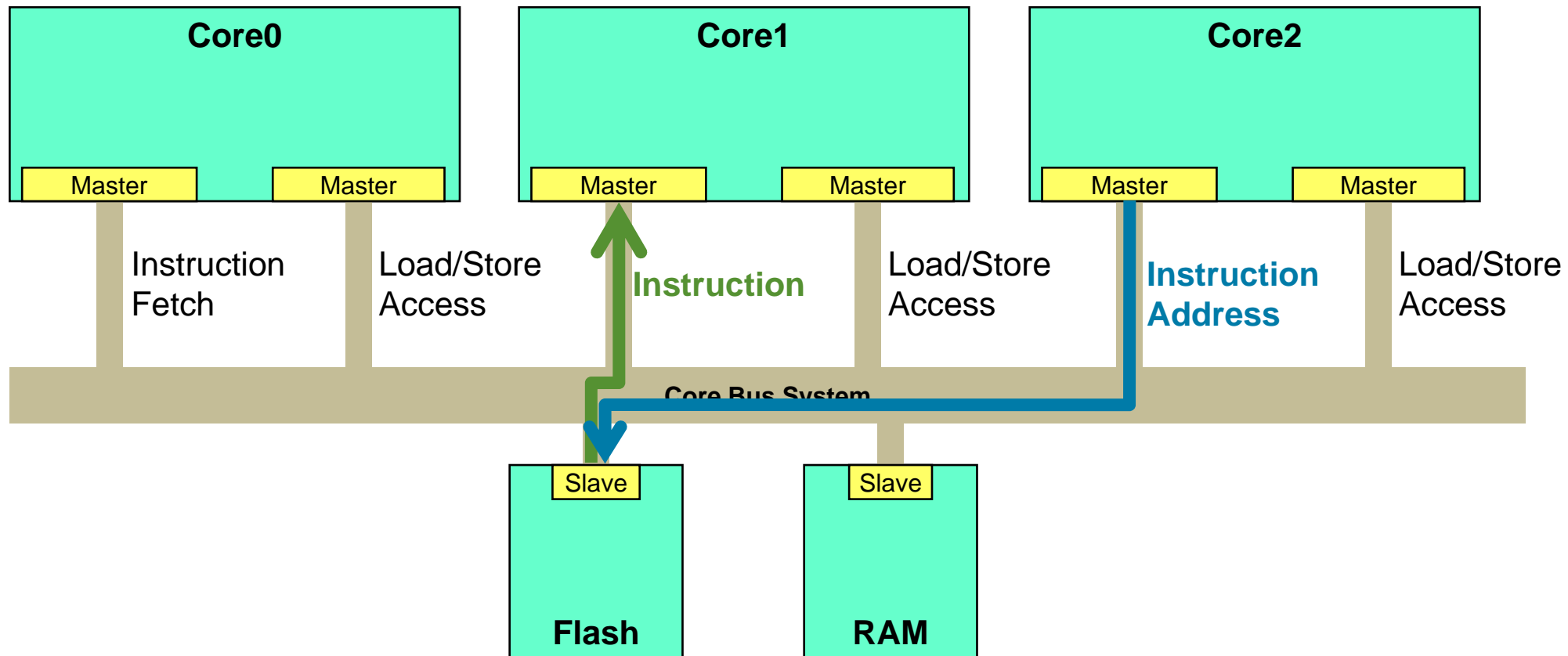
Multicore System mit globalem Programm- und Datenspeicher



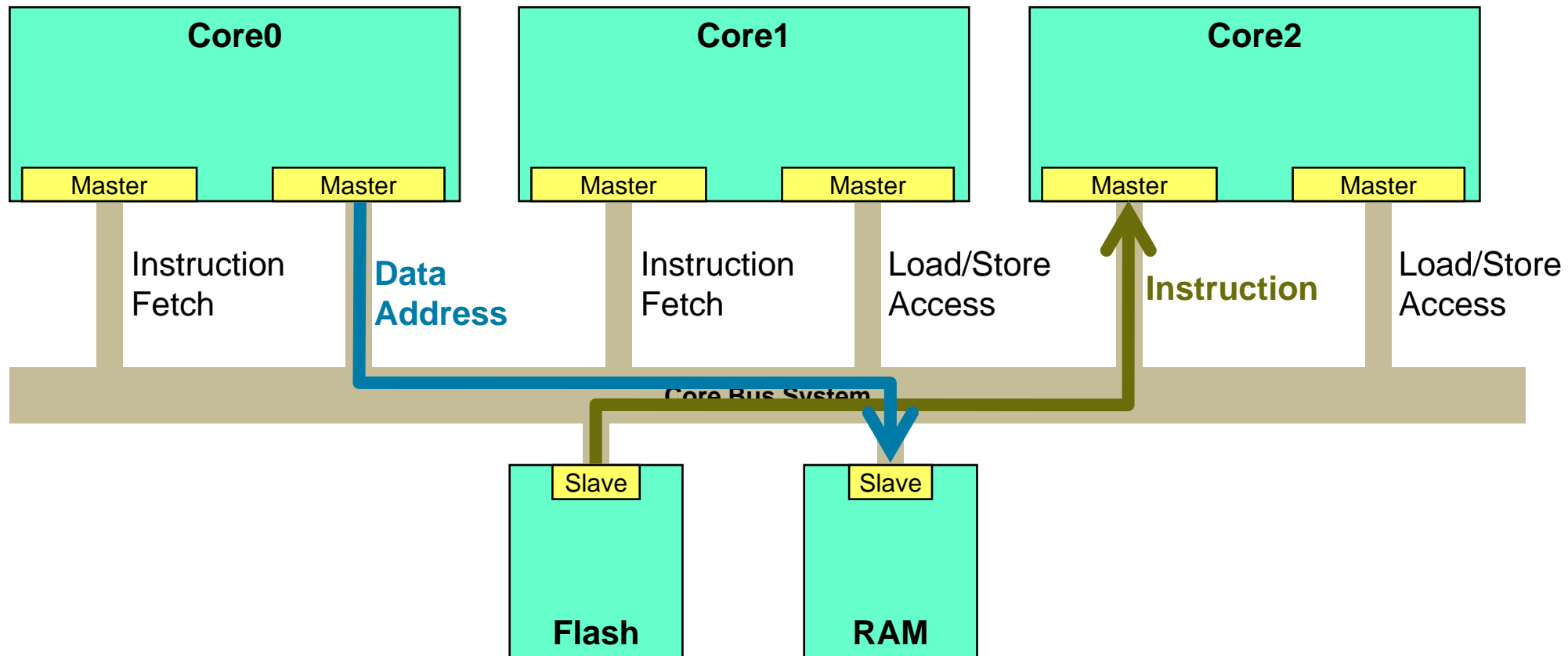
Multicore System mit globalem Programm- und Datenspeicher



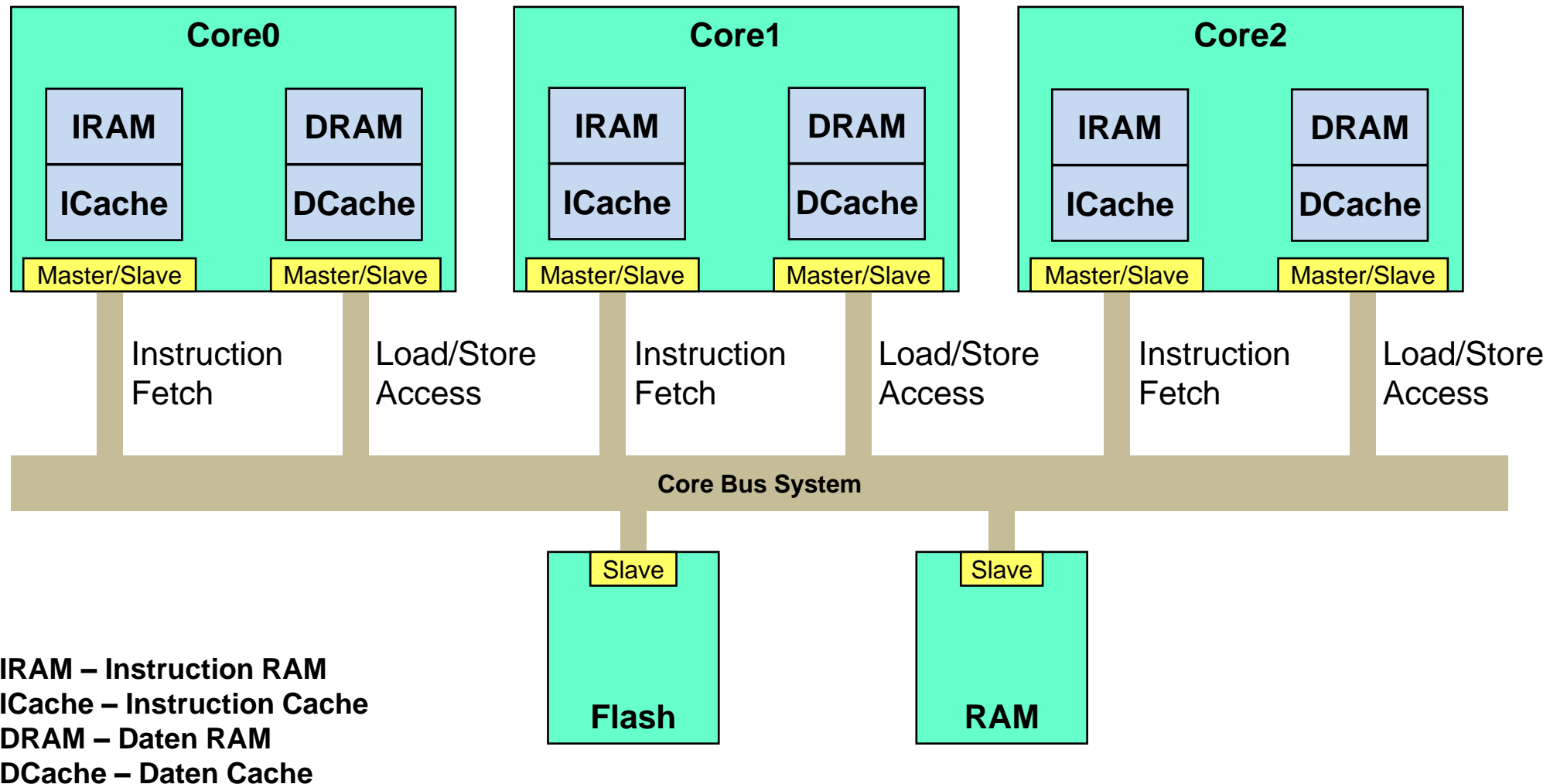
Multicore System mit globalem Programm- und Datenspeicher



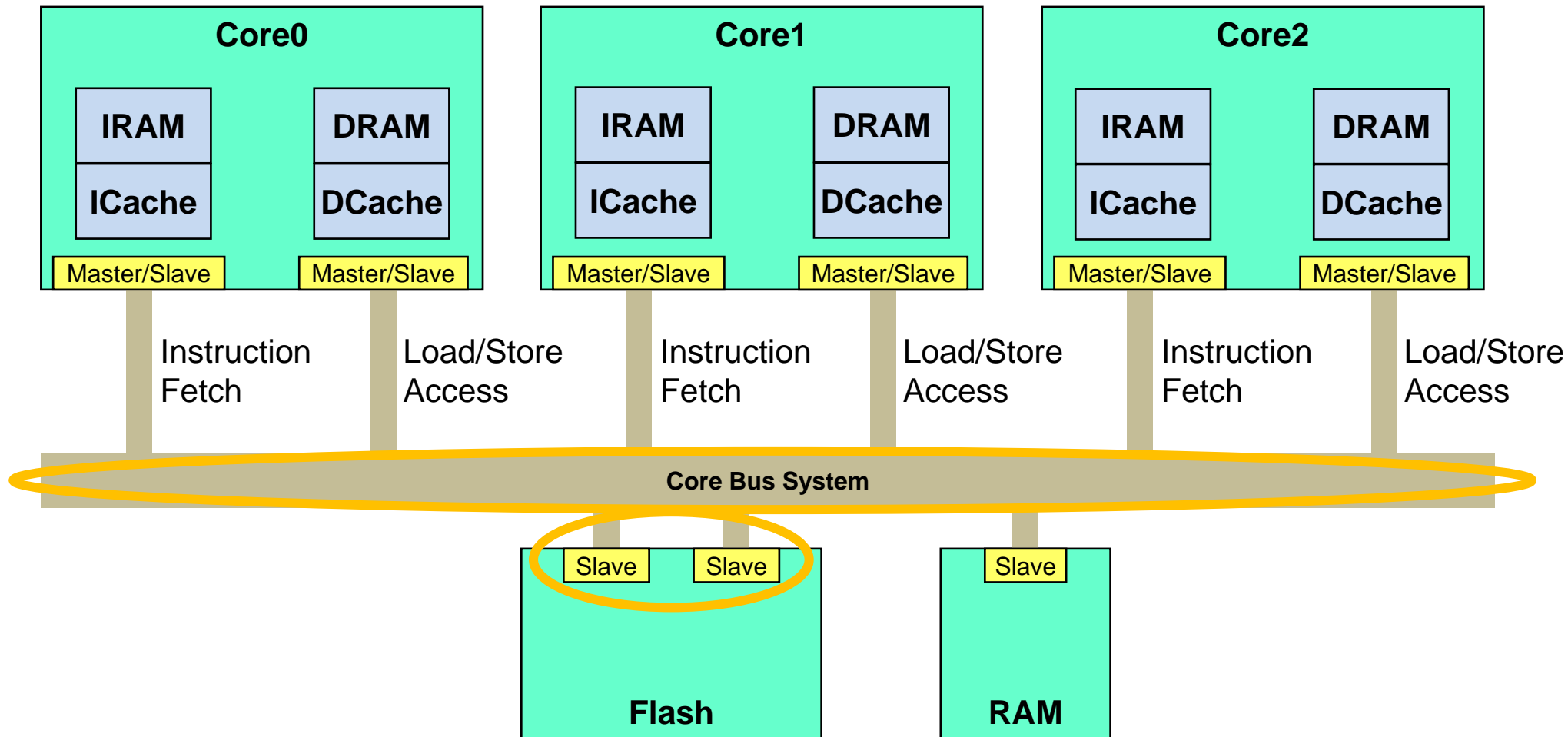
Multicore System mit globalem Programm- und Datenspeicher



Multicore System mit lokalem und globalem Programm- und Datenspeicher

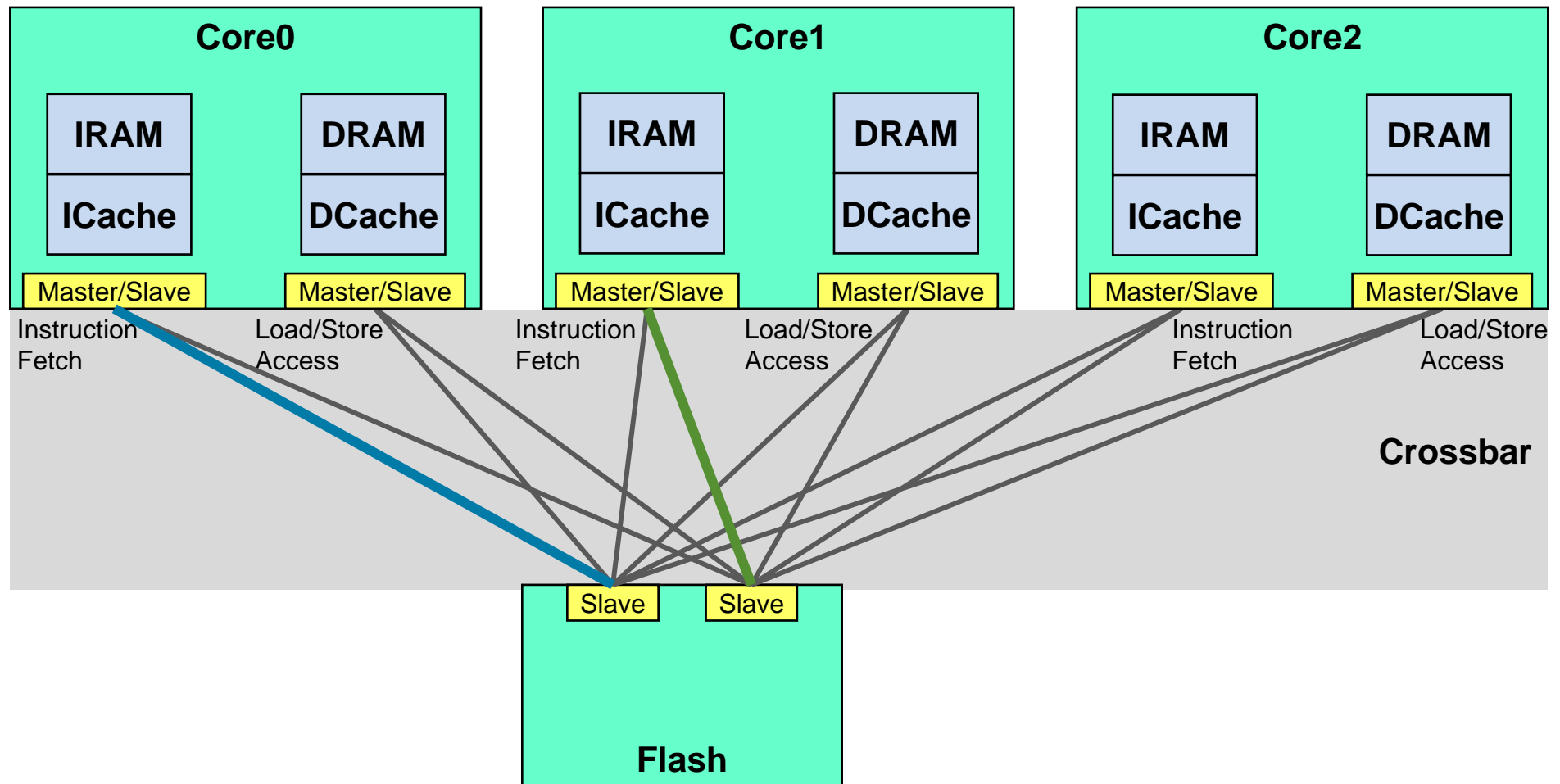


Multicore System mit lokalem und globalem Programm- und Datenspeicher: ➔ Globaler Programmspeicher mit mehreren Busschnittstellen



Multicore System mit lokalem und globalem Programmspeicher:

⇒ Crossbar als Core Bus System



Agenda

- Grundlagen
- **Aufwand und Nutzen**
- Toolunterstützung
- Zusammenfassung

- Eine **Verdopplung der Core-Anzahl** kann bei parallel ausgeführter Software zu einer Geschwindigkeitssteigerung um nahezu Faktor 2 führen.
- Um diesen Faktor zu erreichen muss die **Software skaliert** und auf die verschiedenen Cores aufgeteilt werden.
 - Wie bei einem Multitasking System werden die einzelnen Teilaufgaben auf verschiedene Cores verteilt.
- Werden die Ressourcen wie **lokale RAMs, Caches, Crossbar** und mehrere **Busschnittstellen** zu den globalen Speichern nicht richtig genutzt, kann das zum Flaschenhals des Systems werden.
 - Die Zuordnung von Code und Daten zum vorhandenen Speicher sollte möglichst spät erfolgen können.

Aufwand

▪ Skalieren der Software

- Ähnlich der Aufteilung der Aufgabe in einem Multitasking System

▪ Synchronisieren der Cores

- Über geeignete Hardware Mechanismen, wie zum Beispiel Software Posted Interrupts

▪ Geeignete Interfaces für Datenaustausch schaffen

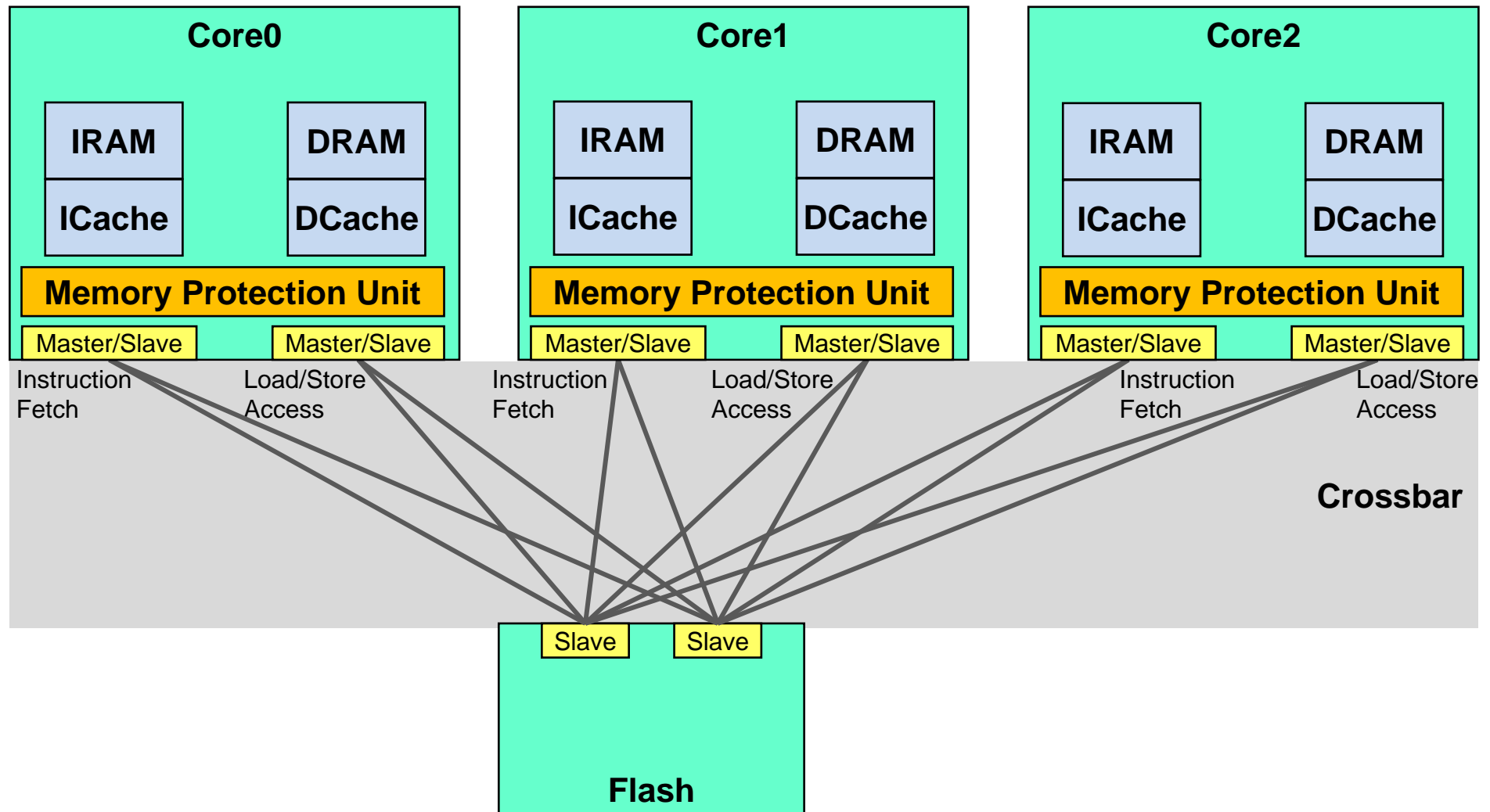
- Shared Variablen im lokalen oder globalen RAM

▪ Schutz der Ressourcen der einzelnen Cores

- Wenn vorhanden über Hardware Mechanismen, wie zum Beispiel **Memory Protection Unit MPU**

Multicore System mit lokalem und globalem Programmspeicher:

➡ Memory Protection Unit PMU



▪ **Austausch von Nachrichten**

- Über gemeinsam genutzte Speicherbereiche
- Globale – für mehrere/alle Cores sichtbare – Variablen

▪ **Schutz der Ressourcen** vor “falschen” Zugriffen

- Memory Protection Unit MPU
- Intelligente Busschnittstellen mit Überprüfung der Master-ID

▪ Ein **geeignetes Betriebssystem** übernimmt unter anderem die Konfiguration und Nutzung der MPU.

▪ Tools müssen die **Aufteilung der Speicherbereiche** für die einzelnen Cores und die Sichtbarkeit von Variablen steuern.

- Variable nur auf Core-Ebene sichtbar
- Variable für mehrere/alle Cores sichtbar

Agenda

- Grundlagen
- Aufwand und Nutzen
- **Toolunterstützung**
- Zusammenfassung



Agenda

- Grundlagen
- Toolunterstützung
- Aufwand und Nutzen
- **Zusammenfassung**

Zusammenfassung

- Aus Hardware-Sicht:
 - Einfach zu realisierende Steigerung der Performanz bei geringen Hardwarekosten
- Aus Software-Sicht:
 - Analyse und Design rücken in den Vordergrund
 - Ohne geeignetes OS mit extrem hohem Aufwand verbunden



Finden



**TECHNISCHES
TRAINING**

**MANAGEMENT-
TRAINING**

**COACHING &
CONSULTING**

SERVICE

UNTERNEHMEN

■ Software- & System-Engineering ■ Qualität & Test ■ Programmierung ■ Betriebssysteme ■ Mikrocontroller

TECHNISCHES TRAINING

Die optimale Ausrüstung
für jede Tiefe



**MICROCONSULT
BITTE MELDEN !**

Einfach Kontaktformular
ausfüllen und wir melden uns
bei Ihnen.



**Der neue Trend Guide
"EMBEDDED SAFETY &
SECURITY"**

Zur Leseprobe



KARRIERE

Mikrocontroller- &
Embedded-Experte (m/w)



**MANAGEMENT
TRAINING & COACHING**

Methoden & Impulse
für Führungskräfte



Renate Schultes

MicroConsult GmbH

Trainer & Coach Mikrocontroller Hardware & Software

r.schultes@microconsult.de

www.microconsult.de

Dip.-Ing. Mario Cupelli

HighTec EDV-Systeme GmbH

mario.cupelli@hightec-rt.com

www.hightec-rt.com