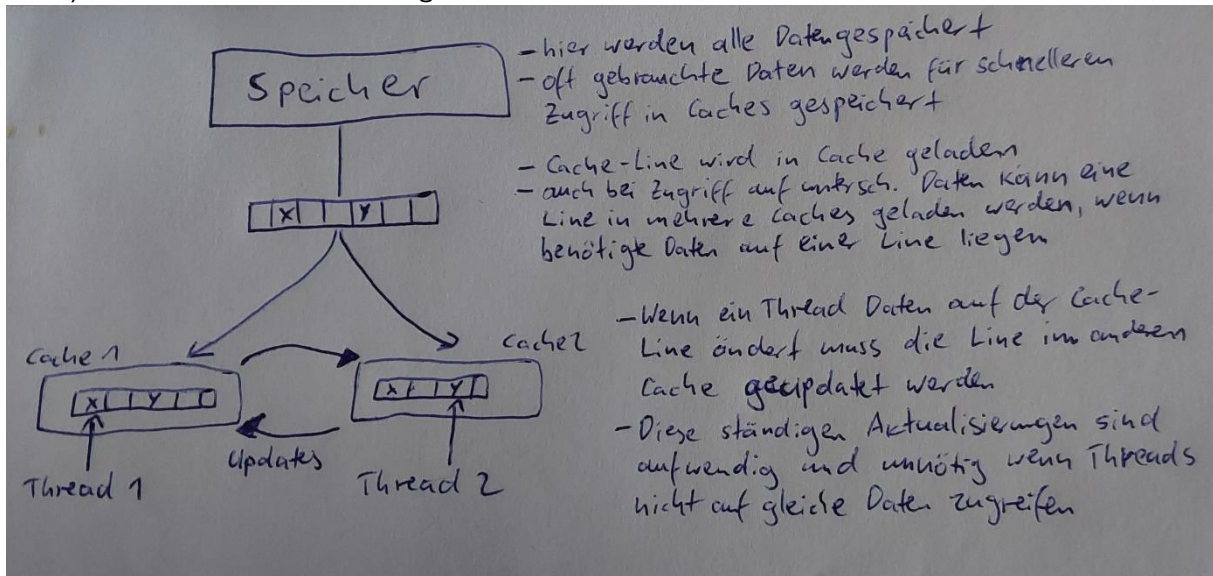
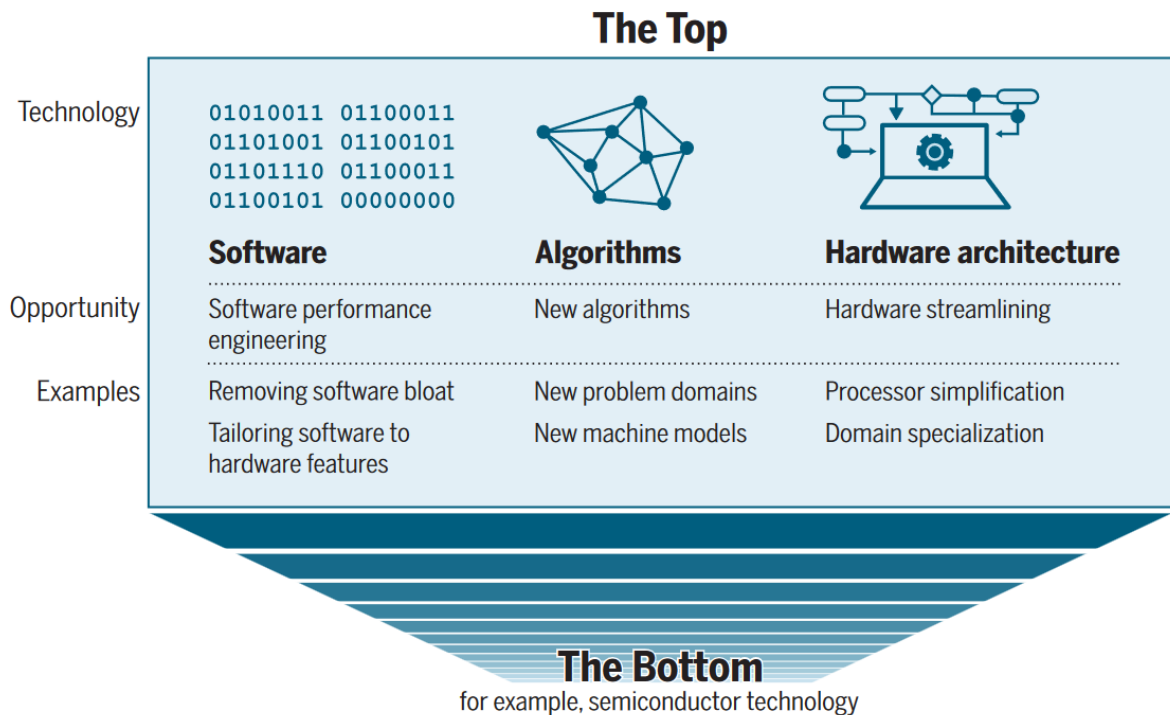


Vorlesung 2 – Exam Assignments

1) What causes false sharing?



2) Performance gains after Moore's law ends



- Die Autoren unterscheiden in Computersystemen zwischen der „Spitze“ und dem „Boden“
- Mit dem Boden ist die Halbleitertechnologie gemeint
- Die vergangenen Leistungssteigerungen in der Computerentwicklung beruhten zu großen Teilen auf der Miniaturisierung der Halbleiter (Moore's-Gesetz: Verdopplung der Halbleiter auf einem Computerchip verdoppelt sich alle 2 Jahre)
- Aufgrund physikalischer Grenzen schreitet diese Miniaturisierung allerdings immer langsamer voran – Möglichkeiten für Leistungssteigerungen am Boden nehmen dramatisch ab
- Autoren sehen aber noch große Verbesserungsmöglichkeiten an der Spitze (höhere Ebene der Computertechnologie)

- Im Gegensatz zum Moor'schen Gesetz, dass die Leistungssteigerungen für alle gleich angehoben hat, wird die Verbesserung der Spitze nur opportunistische, ungleichmäßige und sporadische Gewinne bringen
- Großteil der vorhandenen Software nutzt die architektonischen Merkmale von Chips, wie z. B. Parallelprozessoren und Vektoreinheiten, nicht aus
- Einteilung der Spitze in 3 Schichten: Hardware-Architektur (programmierbare digitale Schaltungen, die Berechnungen durchführen) Software (Code, der die digitalen Schaltungen anweist, was zu berechnen ist) Algorithmen (effiziente Problemlösungsroutinen, die eine Berechnung organisieren)
- Umstrukturierung von Software (Software performance engineering) kann dazu beitragen, dass Anwendungen schneller laufen
 - entfernen aufgeblähter Programme (einfaches Programmieren spart Zeit, erzeugt aber langsame Programme)
 - Zuschneiden der Software auf bestimmte Merkmale der Hardware-Architektur zB. Parallelität oder Lokalität (Fähigkeit effizient auf Datenelemente zuzugreifen)
- Neue Algorithmen können Leistungssteigerungen bringen
 - Angriff auf neue Problembereiche (es treten neue Probleme auf oder werden wichtiger, die noch nicht algorithmisch untersucht wurden)
 - Berücksichtigung von Skalierbarkeitsproblemen
 - Algorithmen an moderne Hardware anpassen (Berücksichtigung von Eigenschaften wie Parallelität, Vektoreinheiten oder Caching)
- Hardware-Architektur anpassen
 - Verschlinkung der Hardware (Implementierung von Hardware-Funktionen mit weniger Transistoren und Siliziumfläche -> mehr Chipfläche für parallel arbeitende Schaltkreise) (On-Chip-Verbindungen können einfacher werden, weniger Energie und Fläche verbrauchen, wenn die Anwendung, die sie nutzt, Lokalität enthält)
 - Prozessorvereinfachung (komplexer Prozessorkern durch einfacheren Kern ersetzt, der weniger Transistoren benötigt)
 - Spezialisierung der Hardware auf einen bestimmten Anwendungsbereich