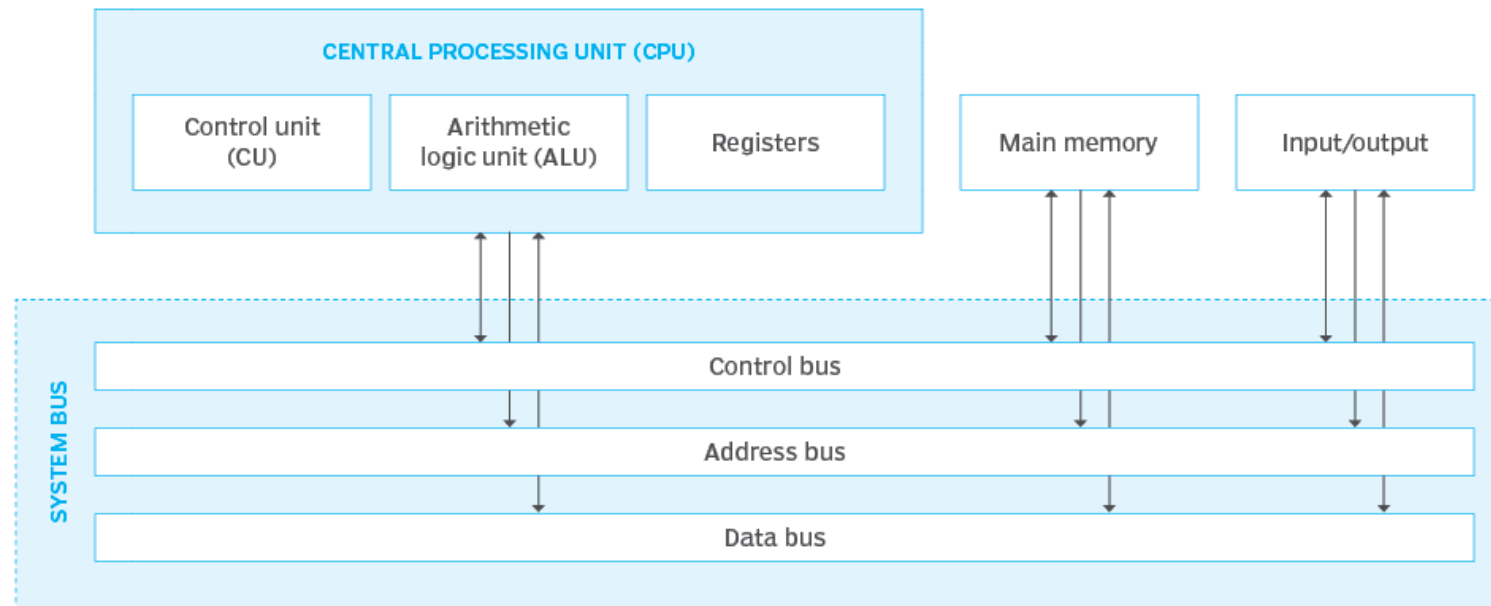





The von Neumann architecture



SOURCE: ROBERT SHELTON

©2022 TECHTARGET, ALL RIGHTS RESERVED 



Von Neumann Architektur: Prozessor und Speicher sind auf gemeinsamen
Datenbus verbunden

Bus transferiert alle Daten zwischen den Neumann-Komponenten

→ Engpass von Daten auf dem Bus

Prozessor schnell → verbingt meißte Zeit durch Warten auf die nötigen Daten aus
dem Speicher

→ Ineffiziente Nutzung der Rechenleistung



Cachespeichersysteme

➔ Zugriff auf nötige Daten brauchen durchschnittlich signifikant weniger Zeit

Grundlagenpraktikum Rechnerarchitektur

Mehrstufige, direkt assoziative Caches

Emirhan Afşin, Nirbheek Singh Setia, Kartik Krishnan

Inhaltsverzeichnis

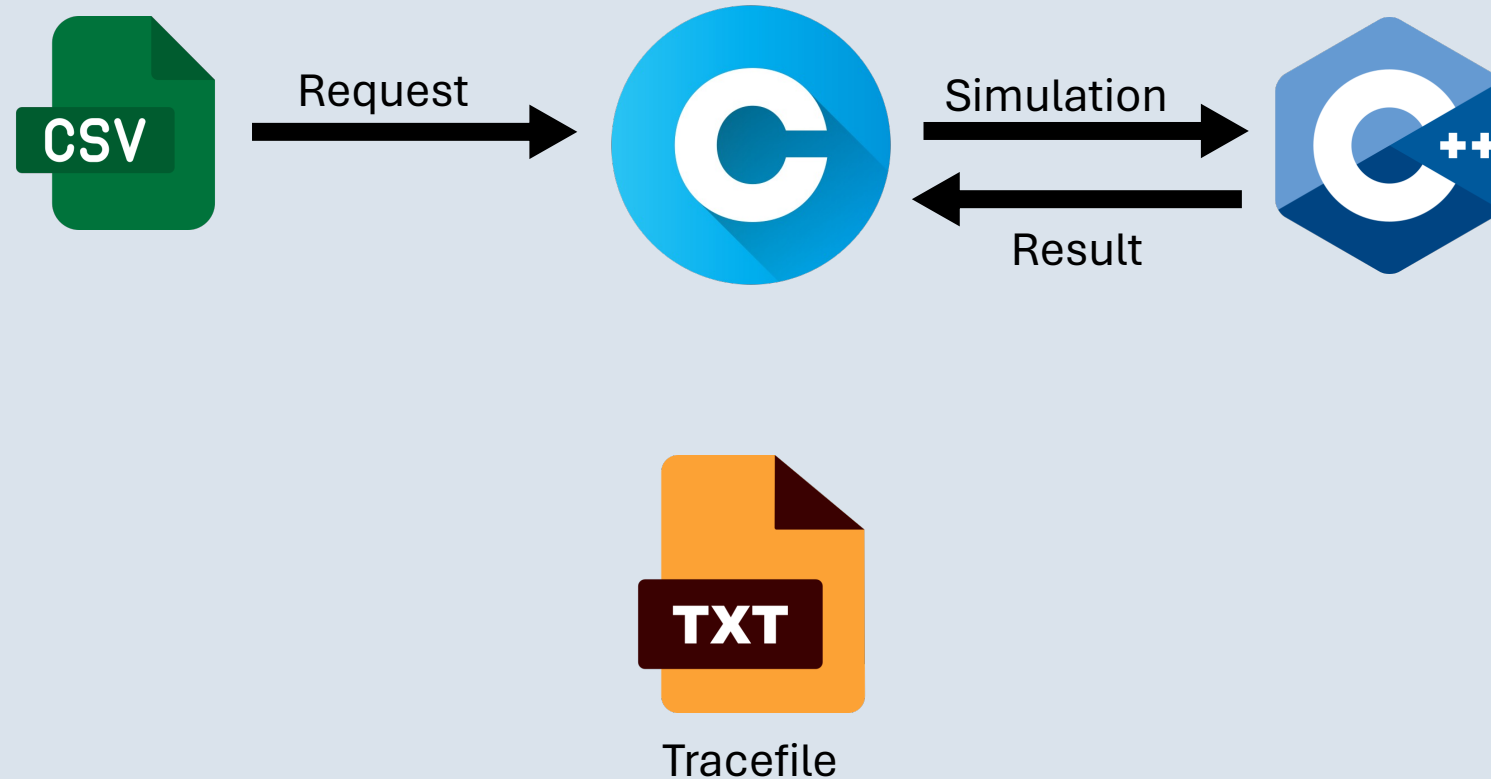
1. Problemstellung
2. Lösungsansätze und Optimierung
3. Korrektheit / Genauigkeit
4. Schaltkreisanalyse
5. Zusammenfassung und Ausblick

Problemstellung

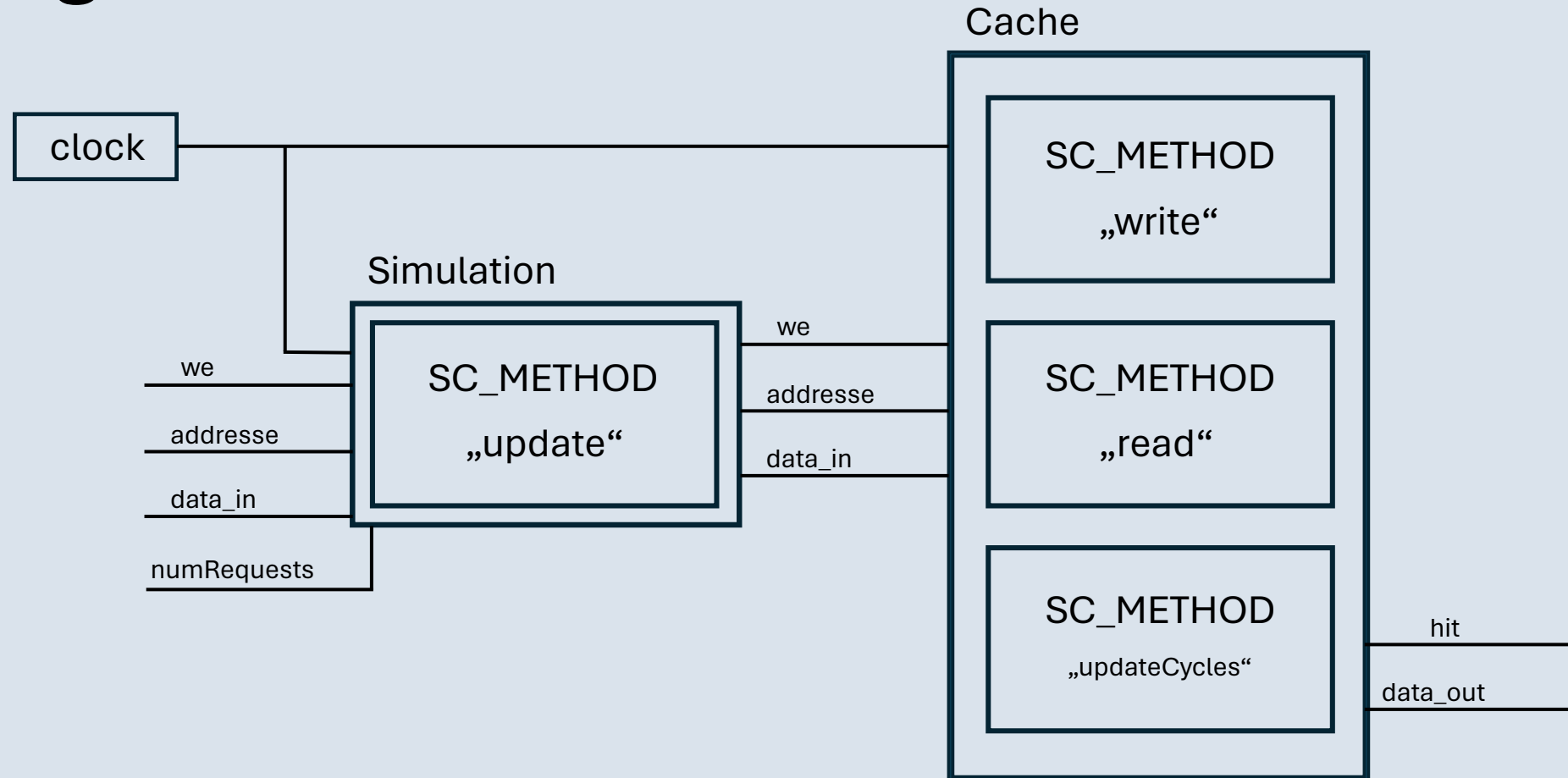
- Untersuchung des Einflusses von mehrstufigen Caches auf Laufzeit und Latenz
- Unterteilung des Projektes:
 - Recherche (Theoretisch)
 - Implementierung (praktisch)
- Relevanz des Themas hervorbringen und unter Beweis stellen

=> Analyse und Implementierung von mehrstufigen Caches

Allgemeiner Lösungsansatz

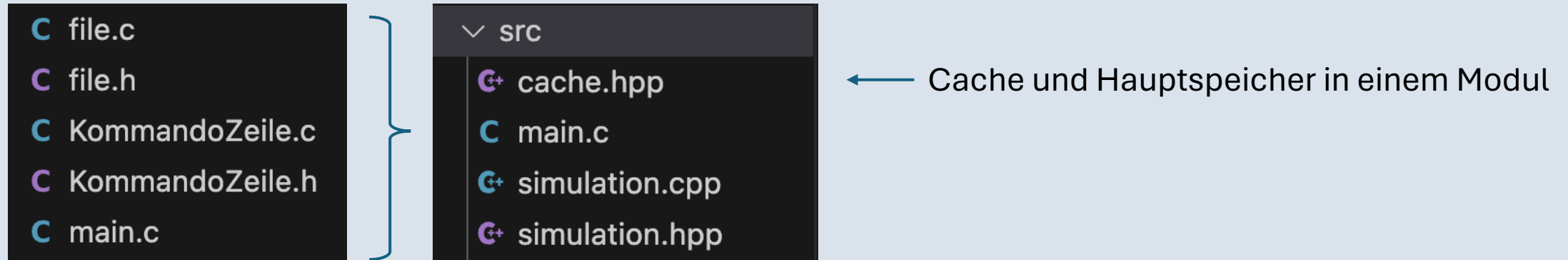


Lösungsansatz



Optimierungen

Strukturelle Optimierung:



Inhaltliche Optimierungen:

SC_THREAD → SC_METHOD

Reduktion der unnötigen Tag Berechnungen

Änderung vom Cache in Byteadressierbarkeit

Korrektheit / Genauigkeit

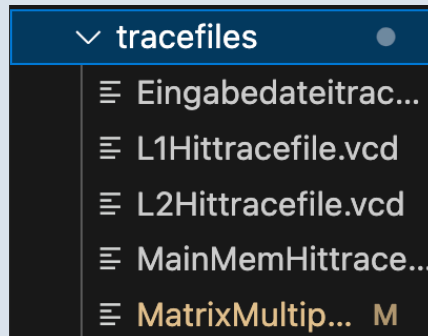
Printstatements wie beispielsweise:

1.

```
// Ausgabe um die Eingabe zu überprüfen
for (int i = 0; i < anfragen->count; i++) {
    if (anfragen->requests[i].we == 0) {
        printf("Operation: %d, Address: 0x%X\n", anfragen->requests[i].we, anfragen->requests[i].addr);
    } else if (anfragen->requests[i].we == 1) {
        printf("Operation: %d, Address: 0x%X, Value: 0x%X\n", anfragen->requests[i].we, anfragen->requests[i].addr, anfragen->requests[i].data);
    }
}
```

Überprüfung der Tracefiles:

2.

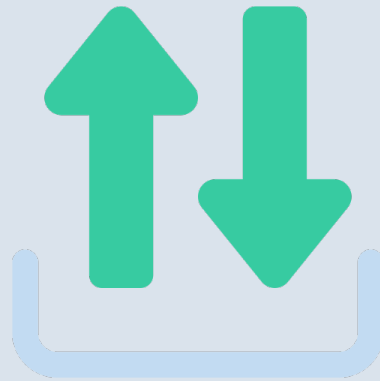


```
$scope module SystemC $end
$var wire 1 aaaaa clk $end
$var wire 32 aaaab we [31:0] $end
$var wire 32 aaaac addr [31:0] $end
$var wire 32 aaaad data_in [31:0] $end
$var wire 32 aaaae data_out [31:0] $end
$var wire 1 aaaf hit $end
$upscope $end
$enddefinitions $end
```

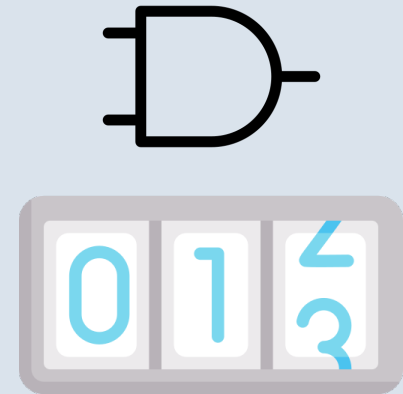
Schaltkreisanalyse



Tracking mit
Tracefile

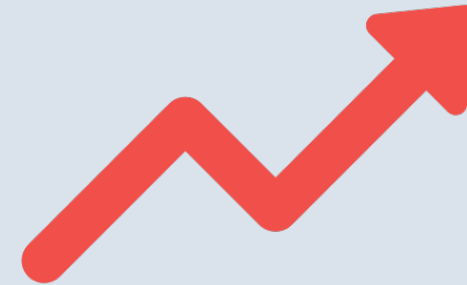


Input and Output



PrimitiveGateCount

Fazit





Haben Sie Fragen?
Vielen Dank für eure Aufmerksamkeit

Emirhan, Nirbheek, Kartik

Quellen

<https://www.techtarget.com/whatis/definition/von-Neumann-bottleneck>

<https://www.flaticon.com>