



## Projekt: Cache Controller

Moritz Nöltner-Augustin,  
Tim Schneider,  
Dennis Sebastian Rieber

# Gliederung



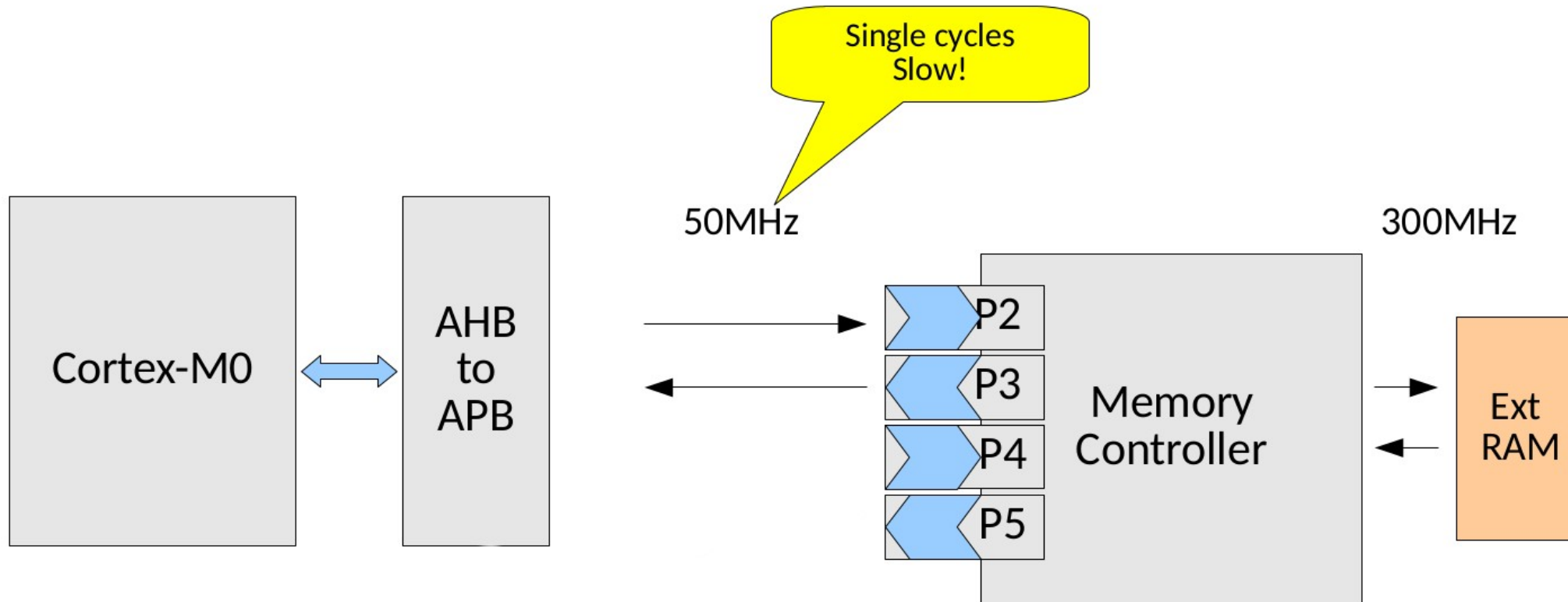
UNIVERSITÄT  
HEIDELBERG  
ZUKUNFT  
SEIT 1386

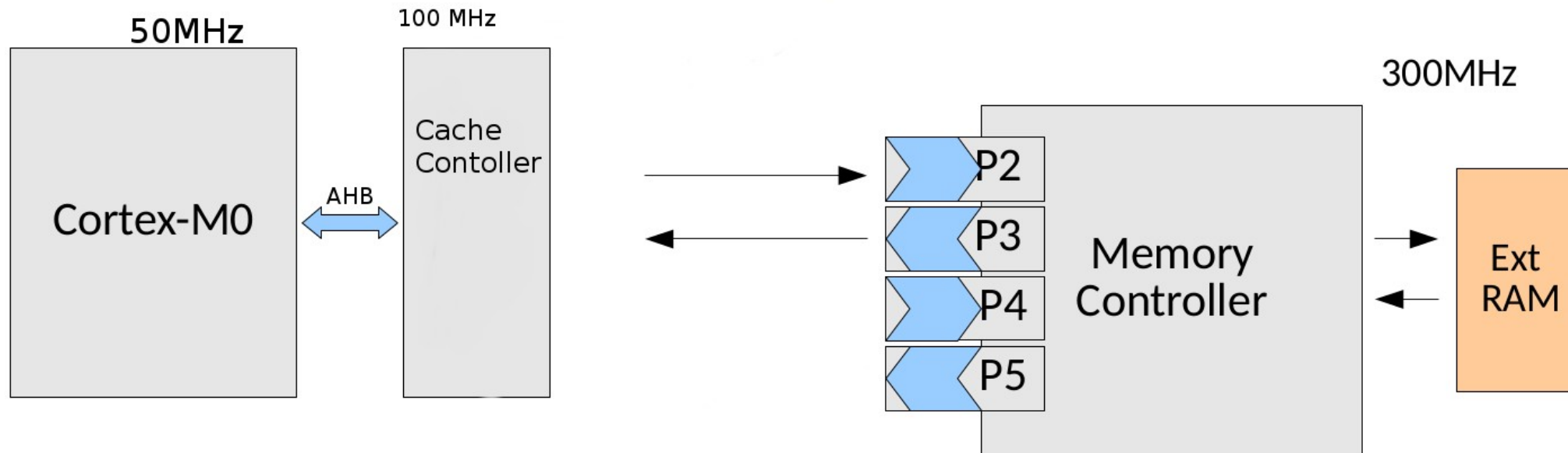
- Einleitung
  - Ausgangssituation und Ziel
- Grundlagen
  - Direct Mapped Cache
  - AHB
  - Memory Controller
- Realisierung
  - Allgemeines Design
  - Read FSM
  - Write FSM
  - Register
  - Integration
- Fazit

# Ausgangssituation



UNIVERSITÄT  
HEIDELBERG  
ZUKUNFT  
SEIT 1386





# Gliederung



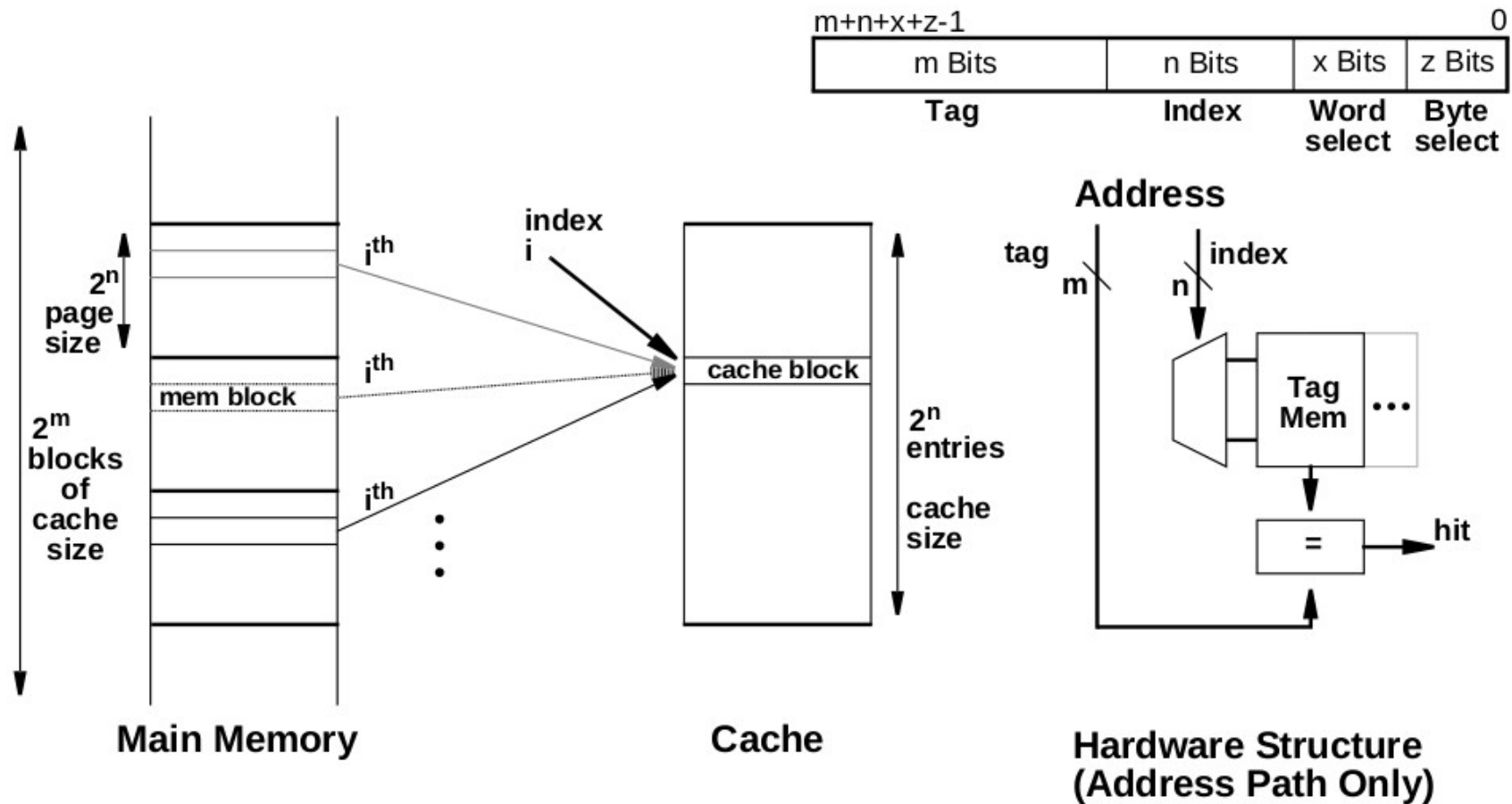
UNIVERSITÄT  
HEIDELBERG  
ZUKUNFT  
SEIT 1386

- Einleitung
  - Ausgangssituation und Problemstellung
- Grundlagen
  - Direct Mapped Cache
  - AHB
  - Memory Controller
- Realisierung
  - Allgemeines Design
  - Read FSM
  - Write FSM
  - Register
  - Integration
- Fazit

# Direct Mapped Cache



UNIVERSITÄT  
HEIDELBERG  
ZUKUNFT  
SEIT 1386



# AHB - Lite Architecture



UNIVERSITÄT  
HEIDELBERG  
ZUKUNFT  
SEIT 1386

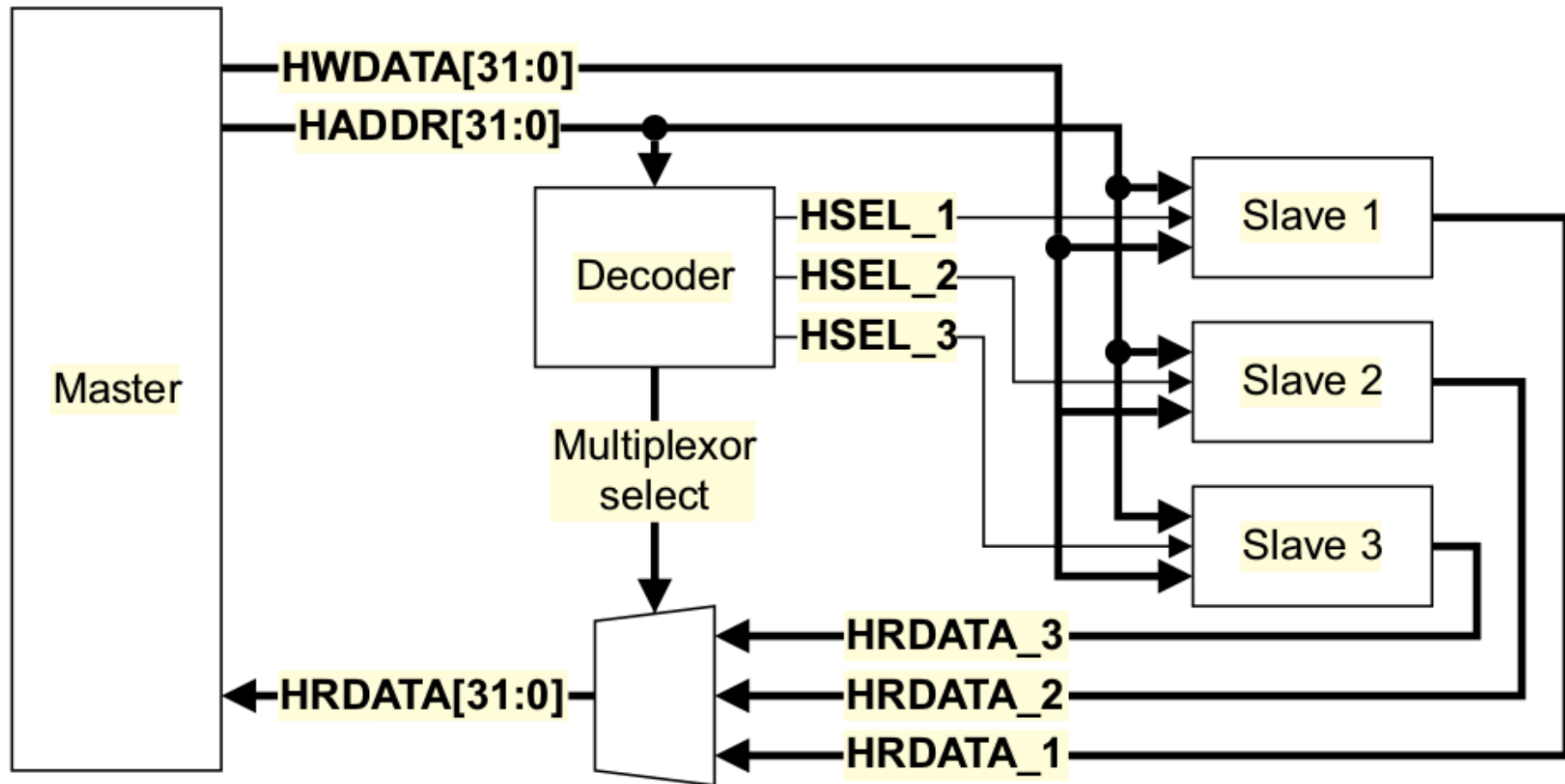
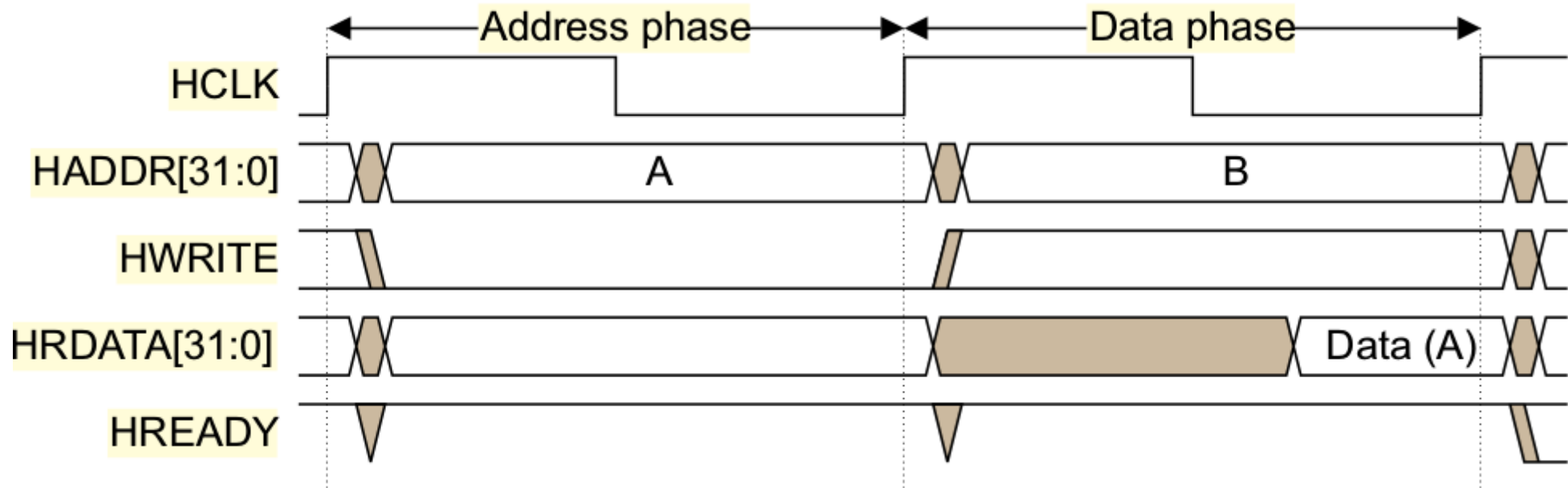


Figure 1-1 AHB-Lite block diagram

# AHB - Lite



UNIVERSITÄT  
HEIDELBERG  
ZUKUNFT  
SEIT 1386

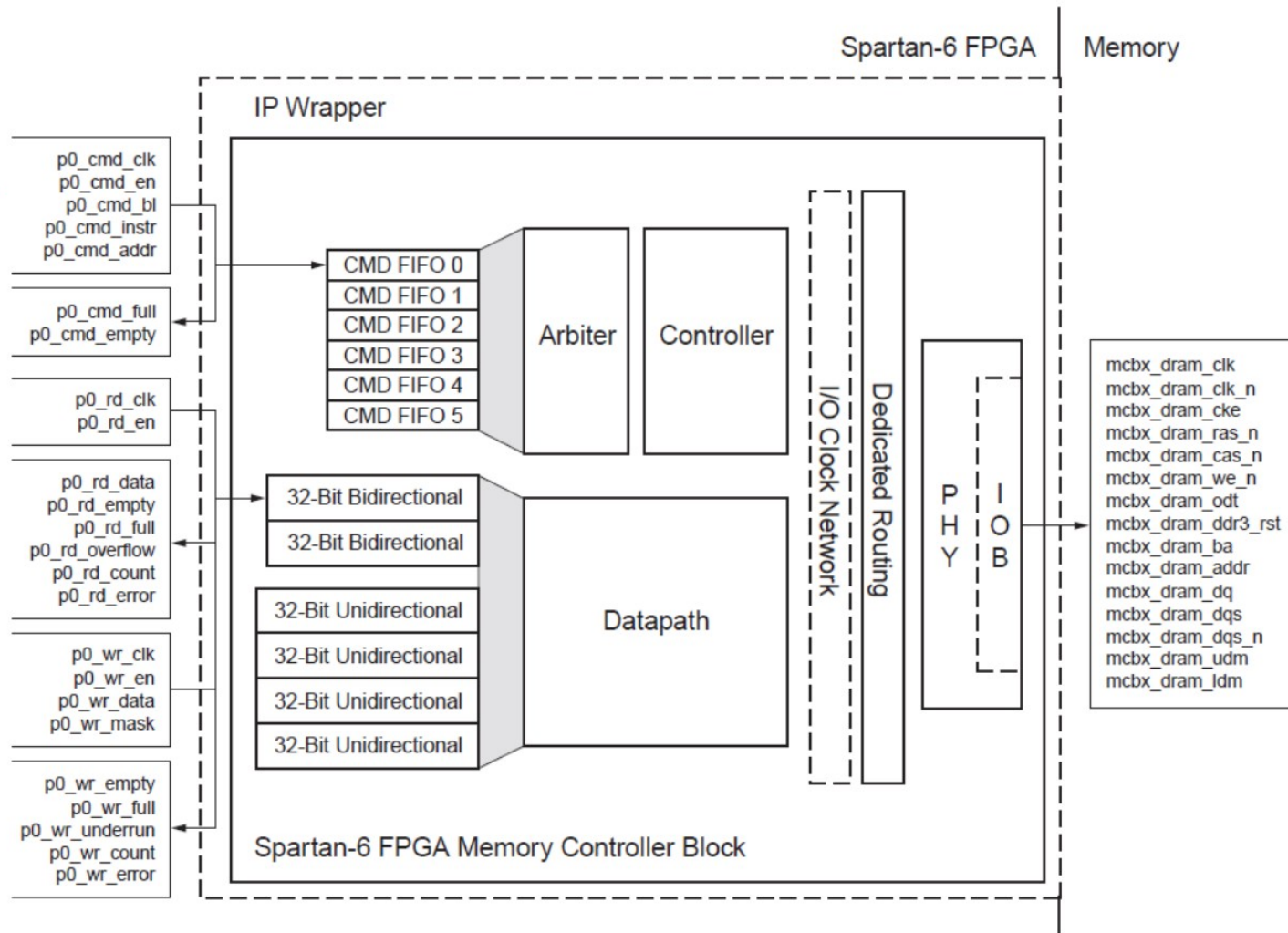




# Memory Controller



UNIVERSITÄT  
HEIDELBERG  
ZUKUNFT  
SEIT 1386



UG388\_c3\_01\_050400

# Gliederung



UNIVERSITÄT  
HEIDELBERG  
ZUKUNFT  
SEIT 1386

- Einleitung
  - Ausgangssituation und Problemstellung
- Grundlagen
  - Direct Mapped Cache
  - AHB
  - Memory Controller
- Realisierung
  - Allgemeines Design
  - Read FSM
  - Write FSM
  - Register
  - Integration
- Fazit

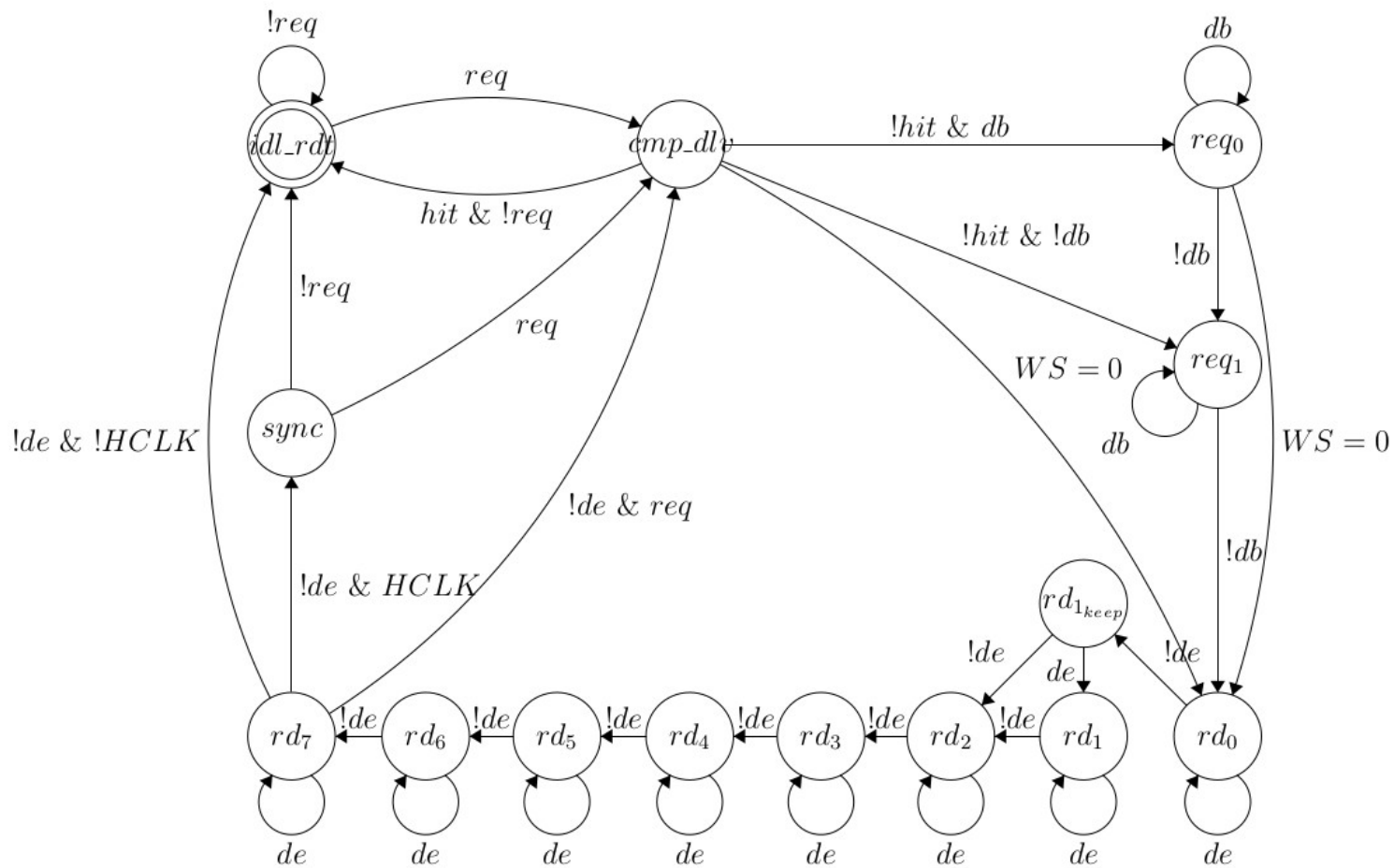
# Realisierung – Allgemeines Design



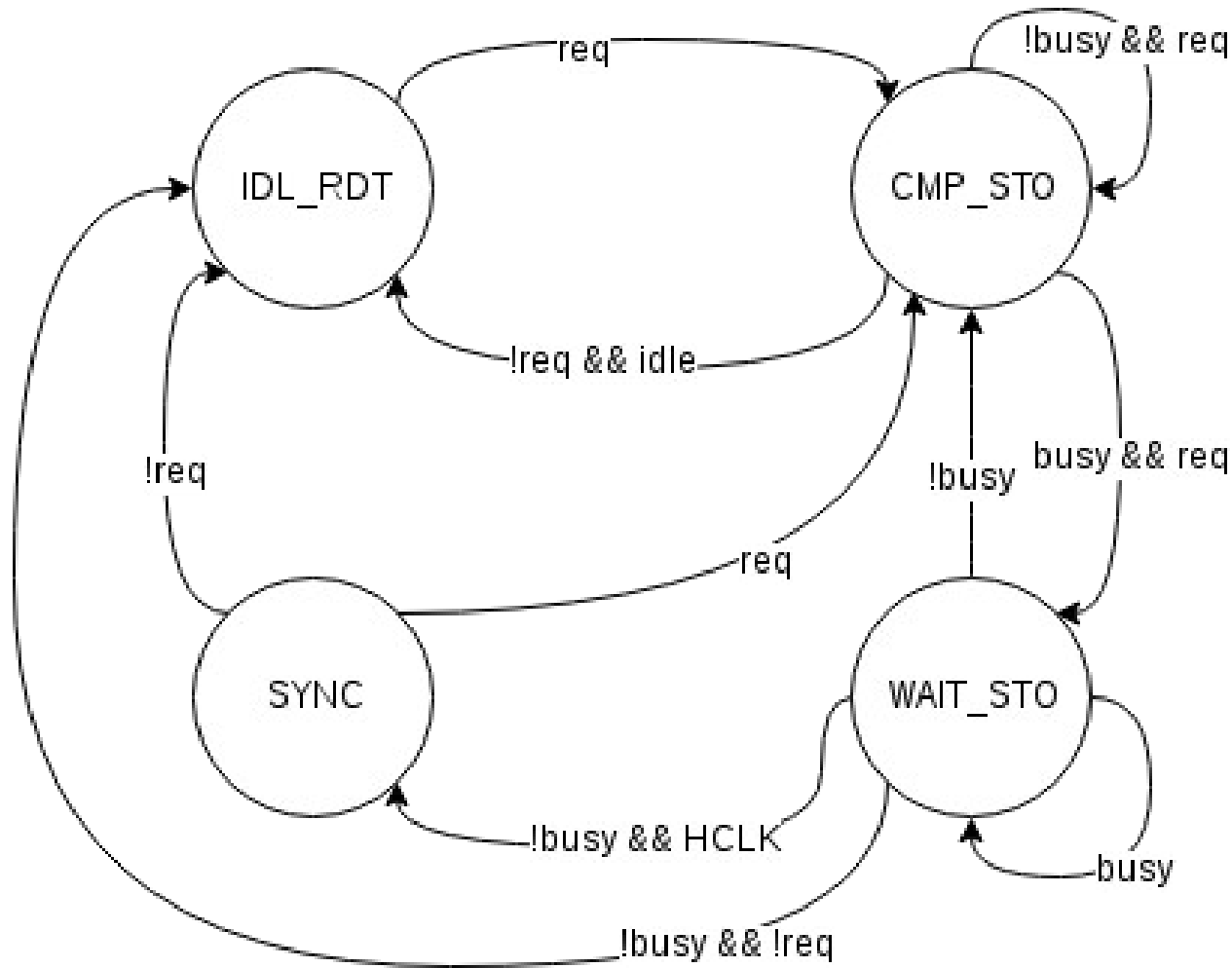
UNIVERSITÄT  
HEIDELBERG  
ZUKUNFT  
SEIT 1386

- Gesamtkapazität: 4kByte
- Cacheline: 8 Wörter á 32 Bit = 32 Bytes
  - 128 Cachelines
  - Tag: 12 Bits
  - Flags: Invalid (Wird am Anfang gesetzt, oder wenn der Prozessor den Cache anweist, [eine] Cachezeile[n] zu invalidieren, oder Reset)
- Cache in Hardware: zwei getrennte Dual Port RAMS für TAG und Daten
- Logik aufgeteilt in zwei FSM
  - Read
  - Write
- Eigene Coherence durch mehrere Devices die den DRAM nutzen können

# Read FSM



# Write FSM





- Registersatz in bestehenden Integriert
- Invalidate Registerset (2 x 32bit)
  - Spezifizieren Addressrange
  - Werden von Cache auf 0xFFFFFFFF gesetzt wenn fertig
- Hit/Miss counter (2 x 32bit)
  - CPU read only
  - Counter-logik im cache-controller



- Cache auf Adressbereich gemappt
- Integration des Controllers in singleARM.vhd als component
- Hxxgen.vhd angepasst

# Gliederung



UNIVERSITÄT  
HEIDELBERG  
ZUKUNFT  
SEIT 1386

- Einleitung
  - Ausgangssituation und Problemstellung
- Grundlagen
  - Direct Mapped Cache
  - AHB
  - Memory Controller
- Realisierung
  - Allgemeines Design
  - Read FSM
  - Write FSM
  - Register
  - Integration
- Fazit



# Fazit – Was tut, was nicht?



UNIVERSITÄT  
HEIDELBERG  
ZUKUNFT  
SEIT 1386

- ✓ Cache Controller
- ✓ Integration in bestehende Architektur
- ✓ Erste Tests mit Software
- ✗ Noch keine Instruktionen über Cache
- ✗ Keine Invalidate Register
- ✗ Keine Hit/Miss Counter

# Fazit – Was war das Problem



UNIVERSITÄT  
HEIDELBERG  
ZUKUNFT  
SEIT 1386

- Performante Realisierung schwerer als angenommen
  - Unterschiedliche Takte (100MHz vs. 50 MHz)
  - Zwei Interfaces mit gänzlich unterschiedlicher Funktionsweise
  - Nebenläufiges Lesen und Schreiben