



ESTHI

RECHNER ARCHITEKTUR

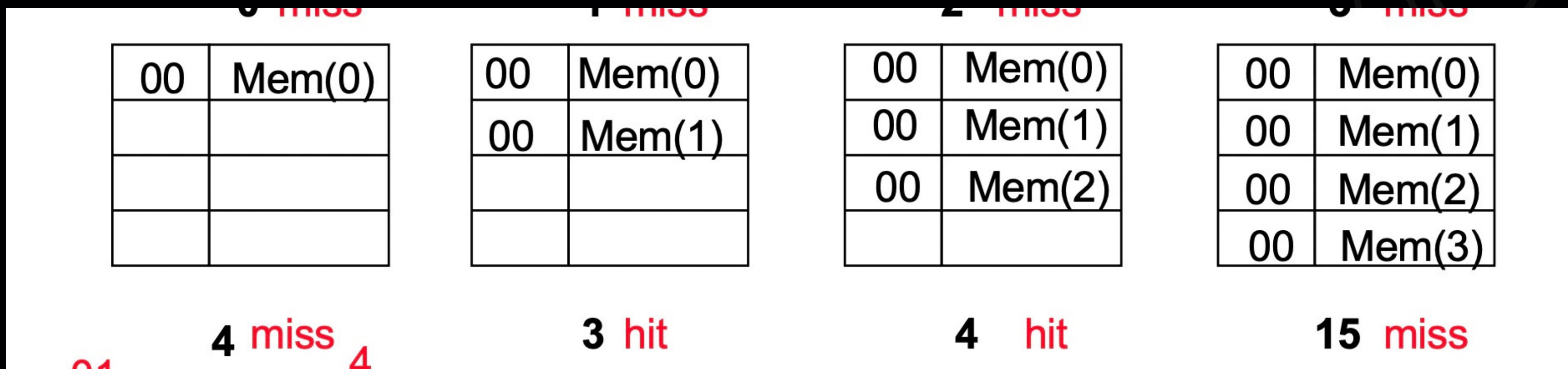
CACHING: EINE EINFACHES
ERSTES BEISPIEL



SALFORD & CO.

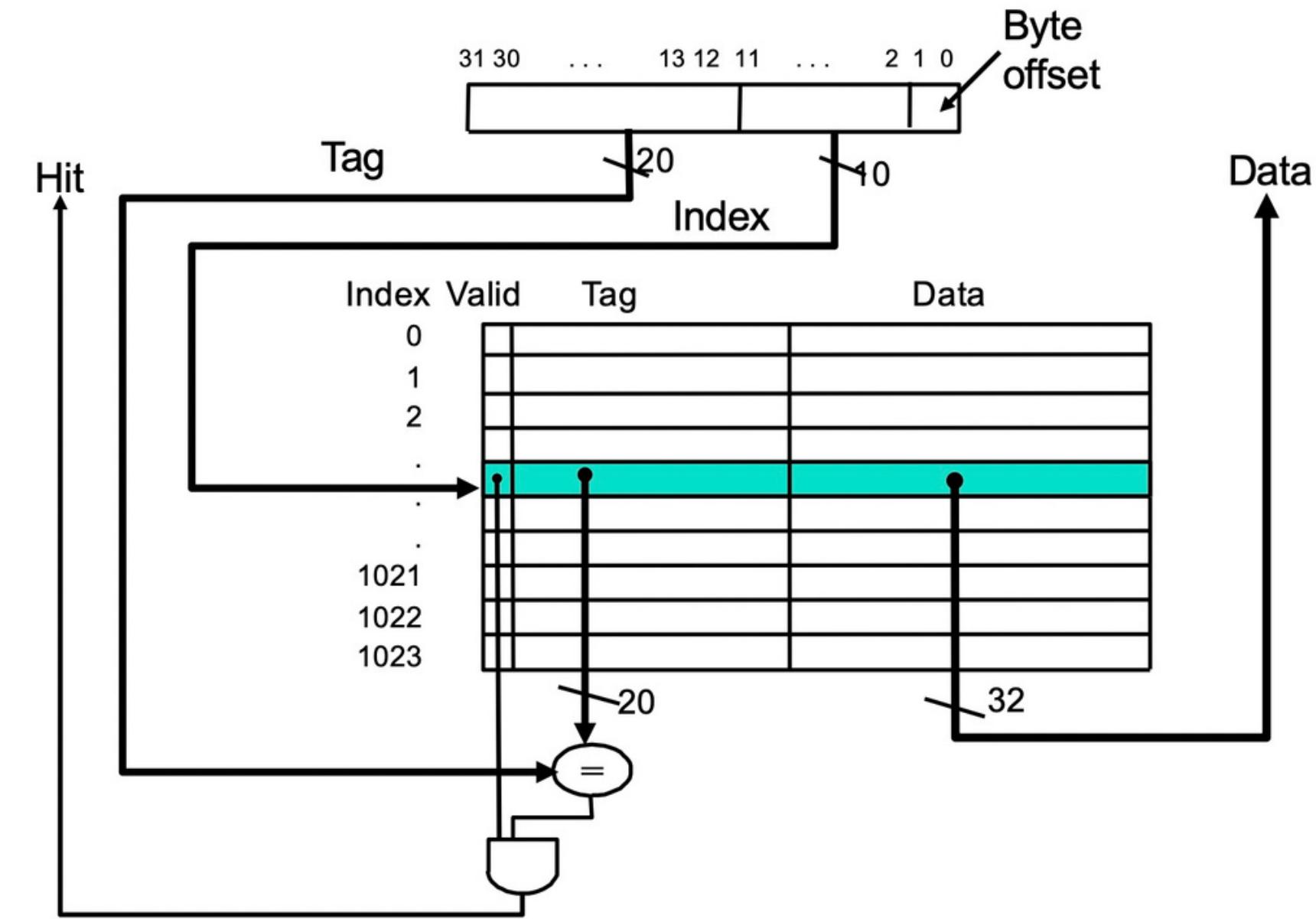
CACHING: EINES EINFACHES ERSTES BEISPIEL





MIPS DIRECT MAPPED CACHE BEISPIEL

ein wort/block, Mapped Cache Beispiel
Ein Wort/Block, Cachegröße = 1K Wörter



MIPS DIRECT MAPPED CACHE BEISPIEL

Ein Wort/Block,
Cachegröße = 1K Wörter



Cache Hits

(I\$ und D\$)

das was wir wollen!

(Nur D\$)

Cache und Speicher **inkonsistent** zu sein

reib die Daten nur in den Cache-Block (**write-back**) des Cache
ts zum nächst tieferen Level in der Cache Hierarchie **nur wenn**
öhere Cache-Block rausfliegt (Cache Eviction))

erfordert ein **dirty** Bit für jeden Cache-Block welches sagt ob die
n zurück in den Speicher geschrieben werden müssen sobald
er dieser rausfliegt

nen, dass Cache und Speicher **konsistent** sind

reib die Daten in den Cache-Block und den nächsten Level in der
cherhierarchie (**write-through**) → benötigt kein dirty Bit

schreiben läuft mit der Geschwindigkeit des nächsten Level in der
cherhierarchie – sehr langsam! – oder es kann ein **Schreibpuffer**
tzt, dann wird ein stall erst benötigt wenn der Schreibpuffer voll

HANDLING CACHE HITS

I\$
read hits

D\$
Cache-Block

READ HITS

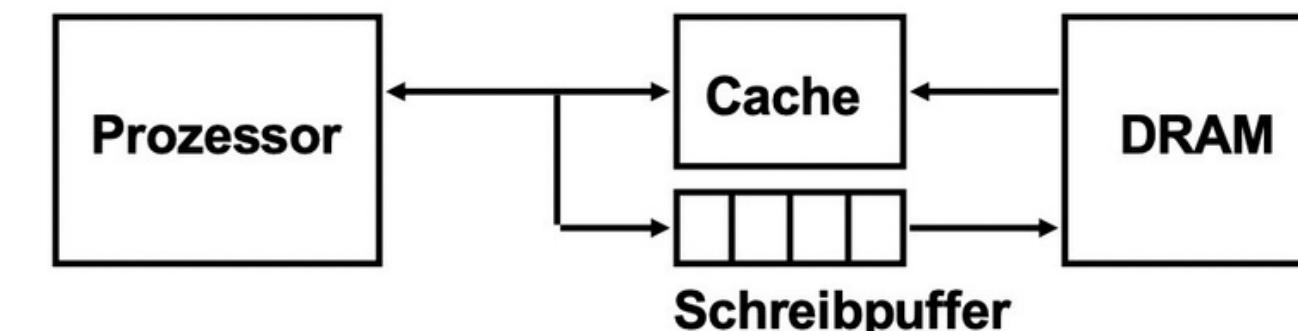


SCHREIBPUFFER FÜR WRITE-THROUGH CACHING

Schreibpuffer zwischen Cache und Hauptspeicher

- Prozessor: schreibt Daten in den Cache und Schreibpuffer
- Speichersteuerung: Schreibt Inhalt des Schreibpuffer in den Hauptspeicher

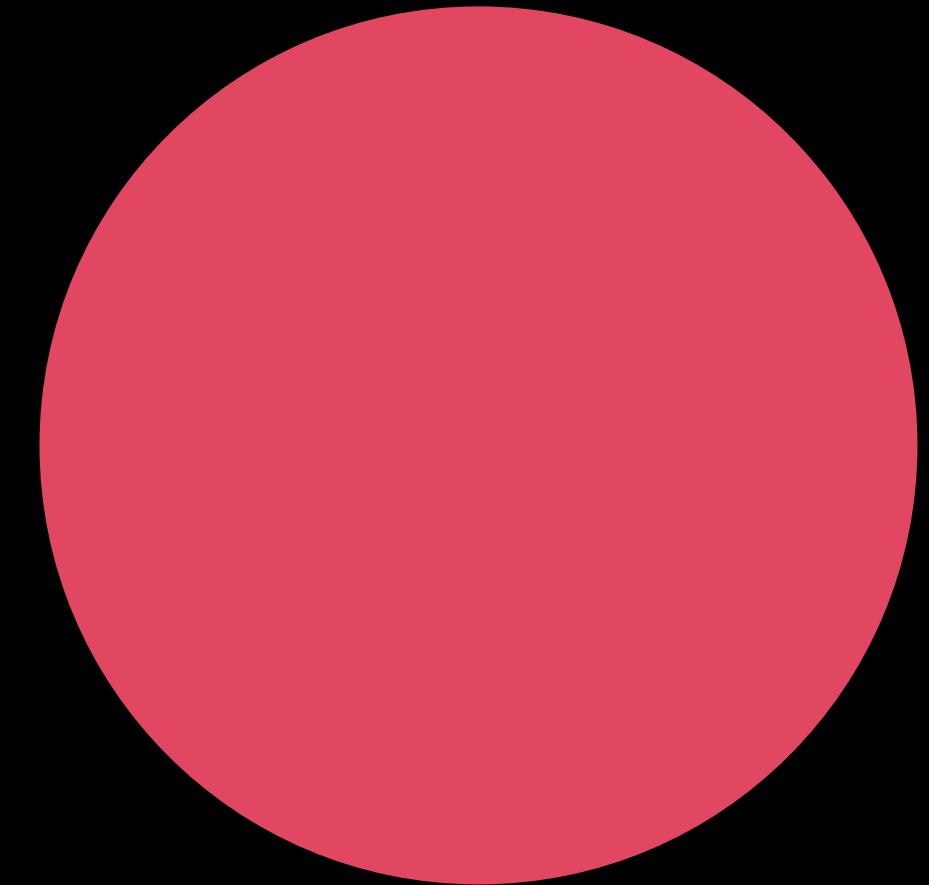
LEARN MORE



“
*The
Schreibpuffer ist
FIFO*
”



SPEICHERFREQUENZ
(BZGL. ZEIT) \ll 1 /
DRAM SCHREIBZYKLUS





SPEICHERFREQUENZ (BZGL. ZEIT) \ll 1 / DRAM SCHREIBZYKLUS

Alptrum eines Speichersystem Designers ● Wenn
Speicherfrequenz (bzgl. Zeit) \rightarrow 1 / DRAM Schreibzyklus führt zu
Schreibpuffer Sättigung - E.g. Nutze ein write-back Cache



0 miss			1 hit			2 miss		
00	Mem(1)	Mem(0)	00	Mem(1)	Mem(0)	00	Mem(1)	Mem(0)
						00	Mem(3)	Mem(2)
3 hit			4 miss			3 hit		
00	Mem(1)	Mem(0)	01	Mem(1)	Mem(0)	01	Mem(5)	Mem(4)
00	Mem(3)	Mem(2)	00	Mem(3)	Mem(2)	00	Mem(3)	Mem(2)
4 hit			15 miss			15 miss		
01	Mem(5)	Mem(4)	11	Mem(5)	Mem(4)	01	Mem(5)	Mem(4)
00	Mem(3)	Mem(2)	00	Mem(3)	Mem(2)	00	Mem(3)	Mem(2)

● 8 requests, 4 misses

ANNAHME: HAUPTSPEICHER WORT ADDRESS-SEQUENZ



URSACHEN FÜR CACHE MISSES

compulsory miss
conflict miss
q Capacity miss:

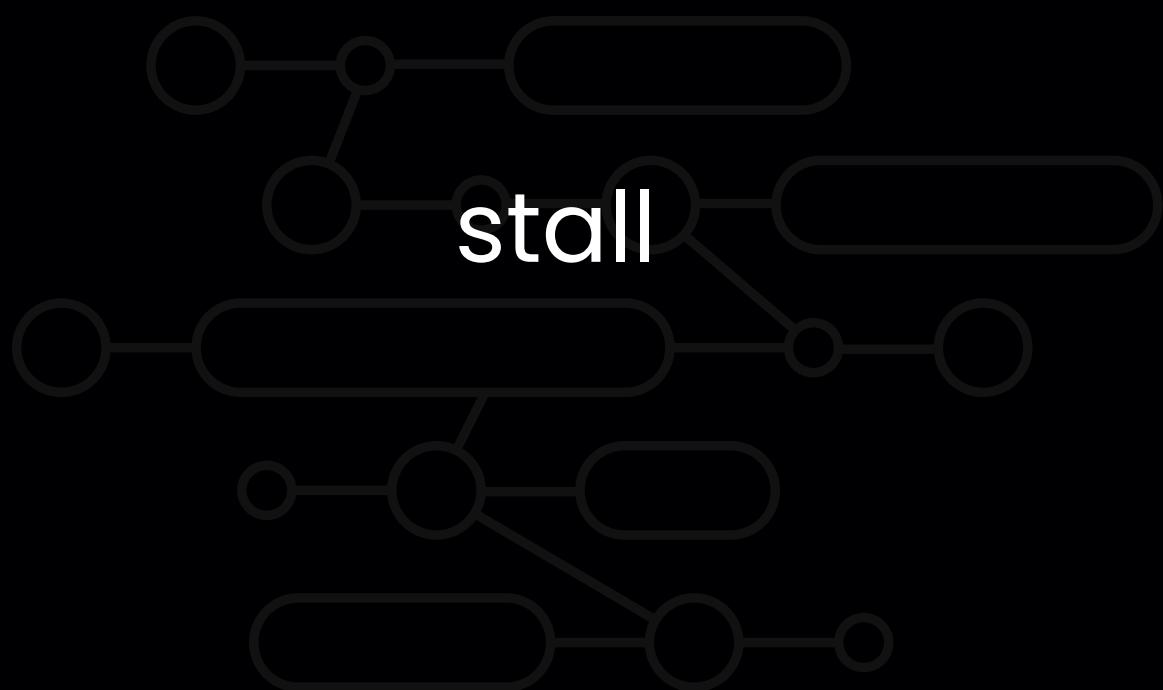


ESTHI



BEHANDELN VON CACHE MISSES

read misses

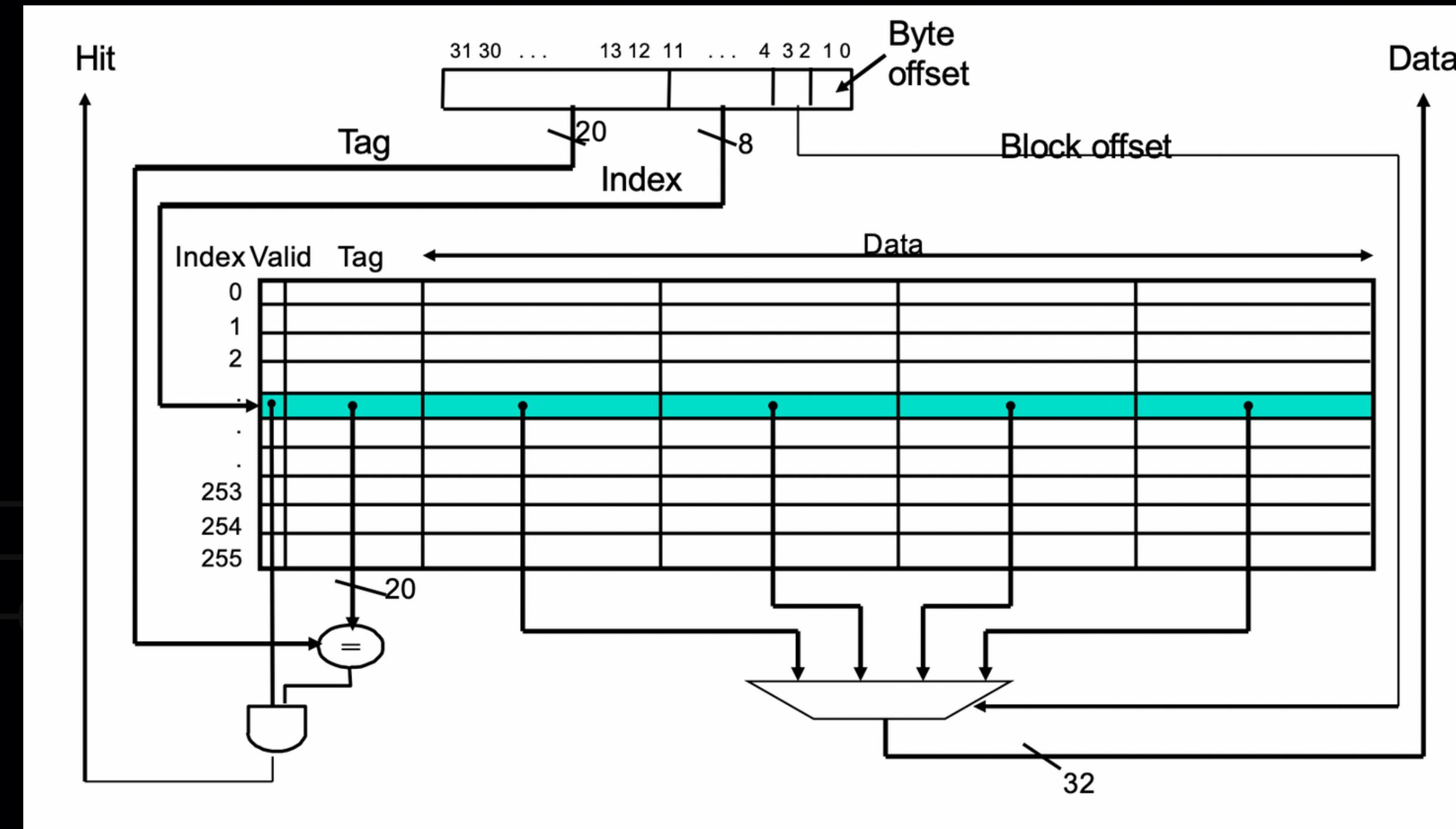




BEHANDELN VON MULTI-WORD BLOCK DIRECT MAPPED CACHE MISSES

read misses

stall





RÄUMLICHE LOKALITÄT NUTZEN

0 miss

00	Mem(1)	Mem(0)

1 hit

00	Mem(1)	Mem(0)

2 miss

00	Mem(1)	Mem(0)
00	Mem(3)	Mem(2)

3 hit

00	Mem(1)	Mem(0)
00	Mem(3)	Mem(2)

4 miss

01	Mem(1)	Mem(0)
00	Mem(3)	Mem(2)

3 hit

01	Mem(5)	Mem(4)
00	Mem(3)	Mem(2)

4 hit

01	Mem(5)	Mem(4)
00	Mem(3)	Mem(2)

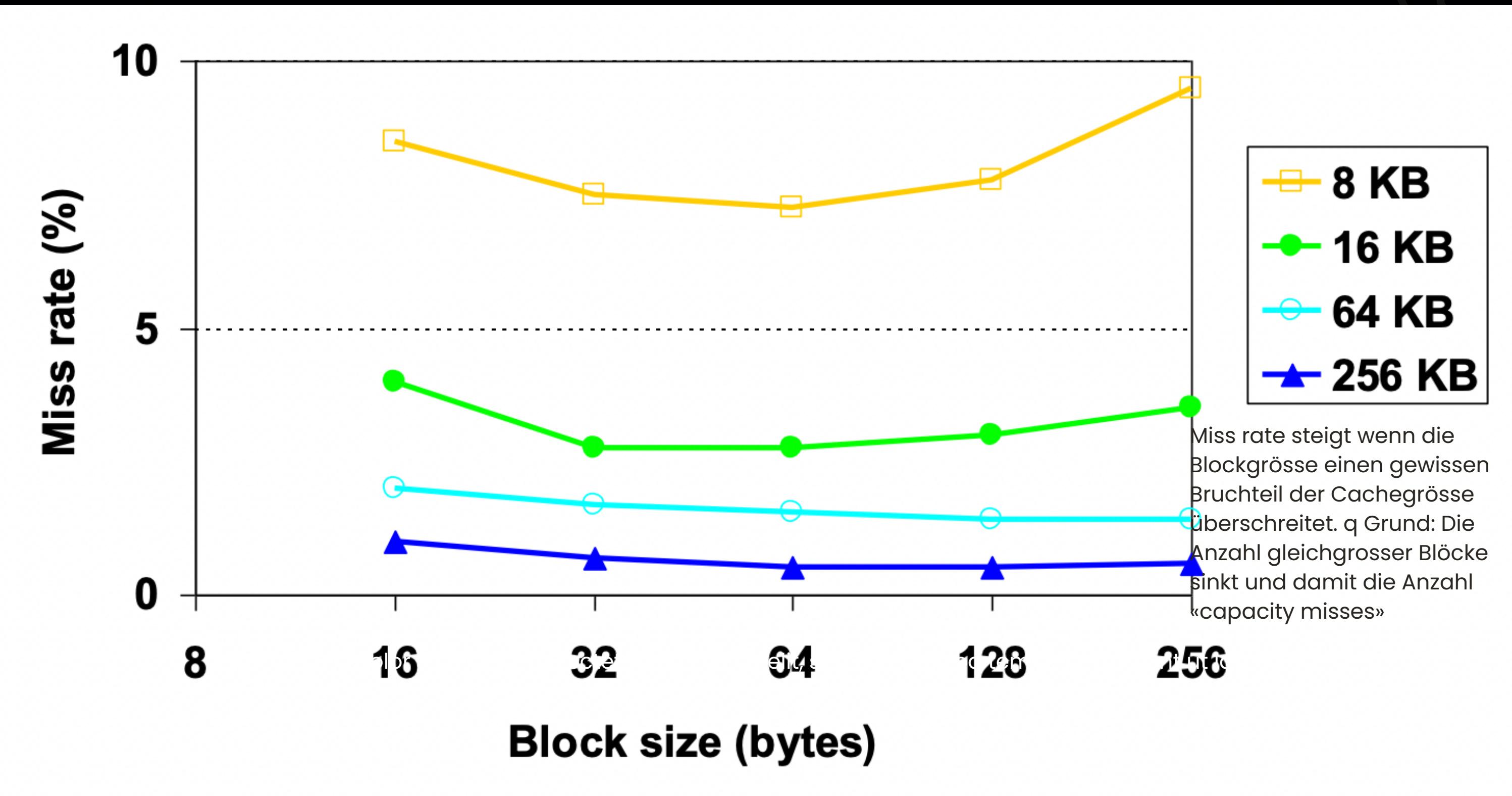
15 miss

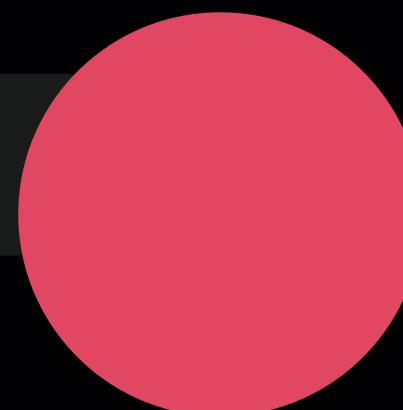
11	Mem(5)	Mem(4)
00	Mem(3)	Mem(2)

- 8 requests, 4 misses

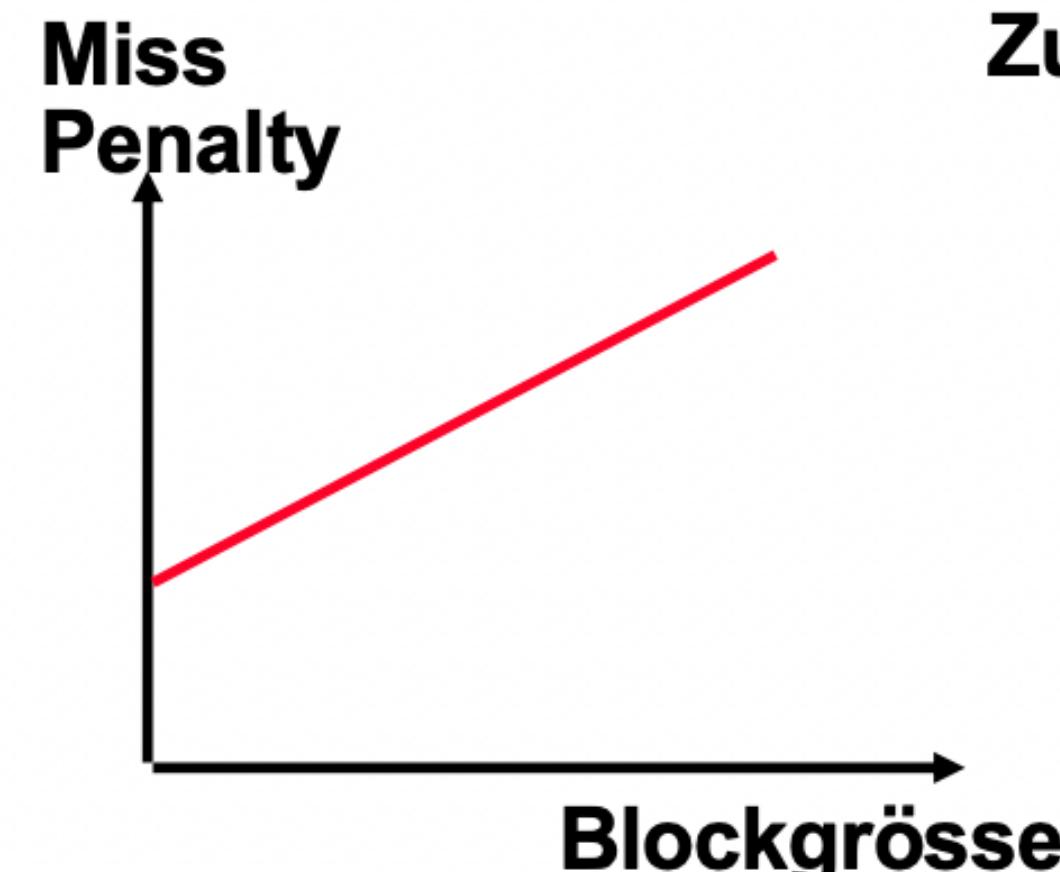
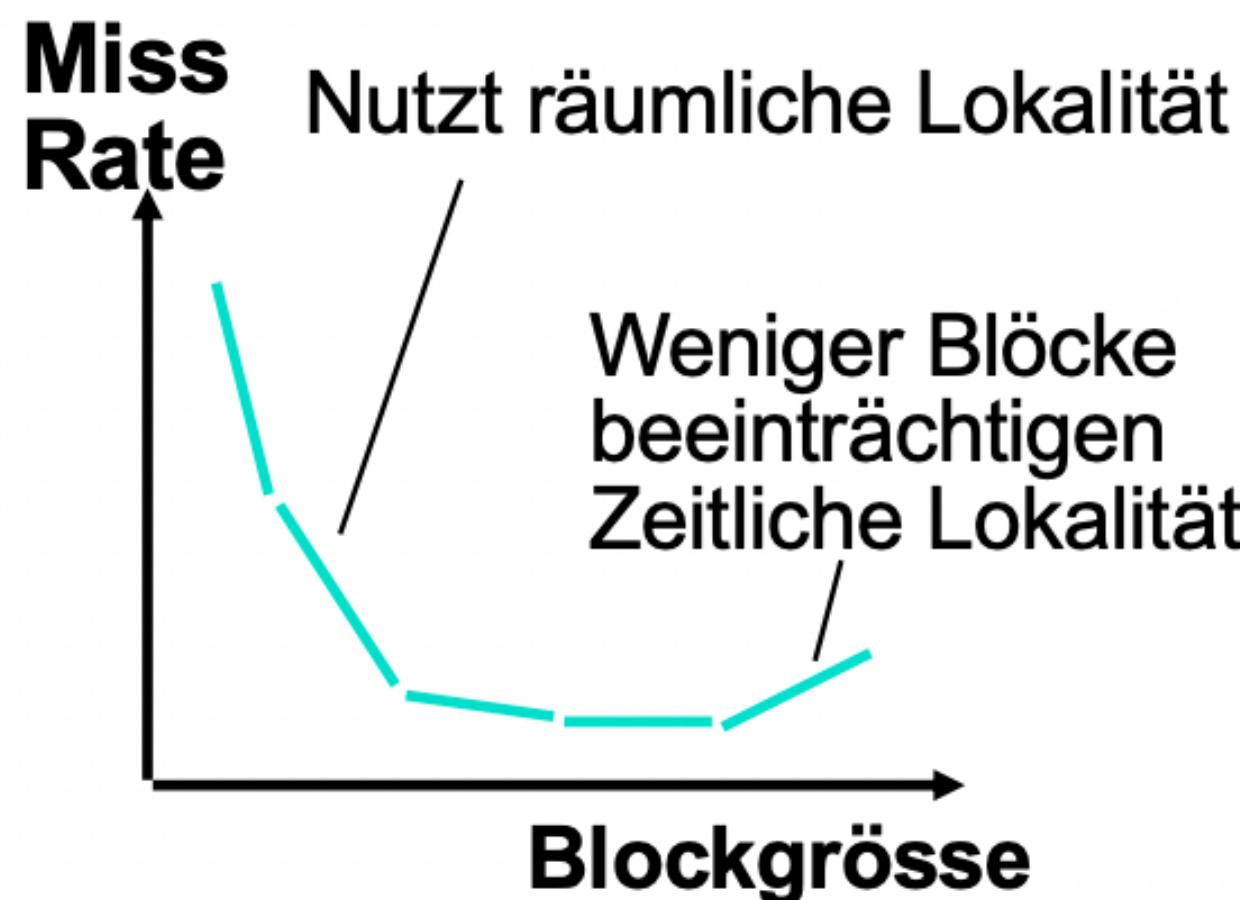
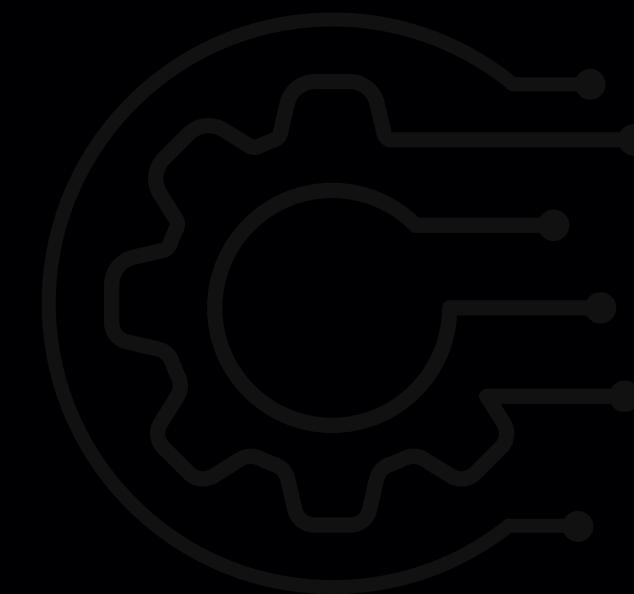


MISS RATE VS BLOCKGRÖSSE VS CACHEGRÖSSE

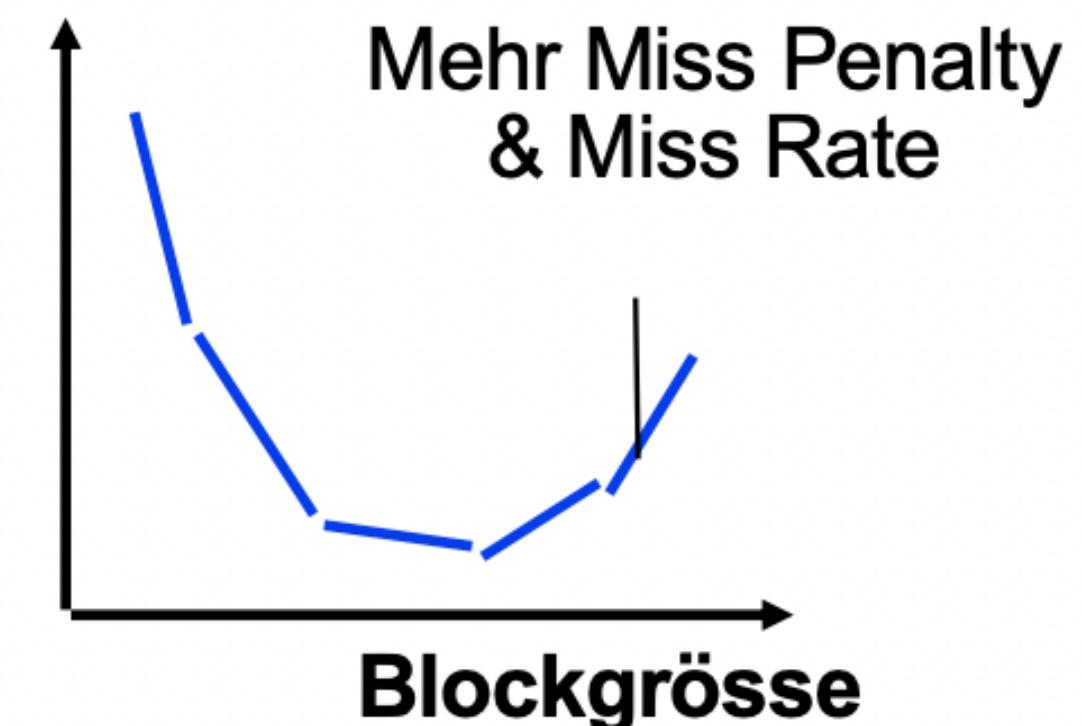




BLOCKGRÖSSE TRADEOFF



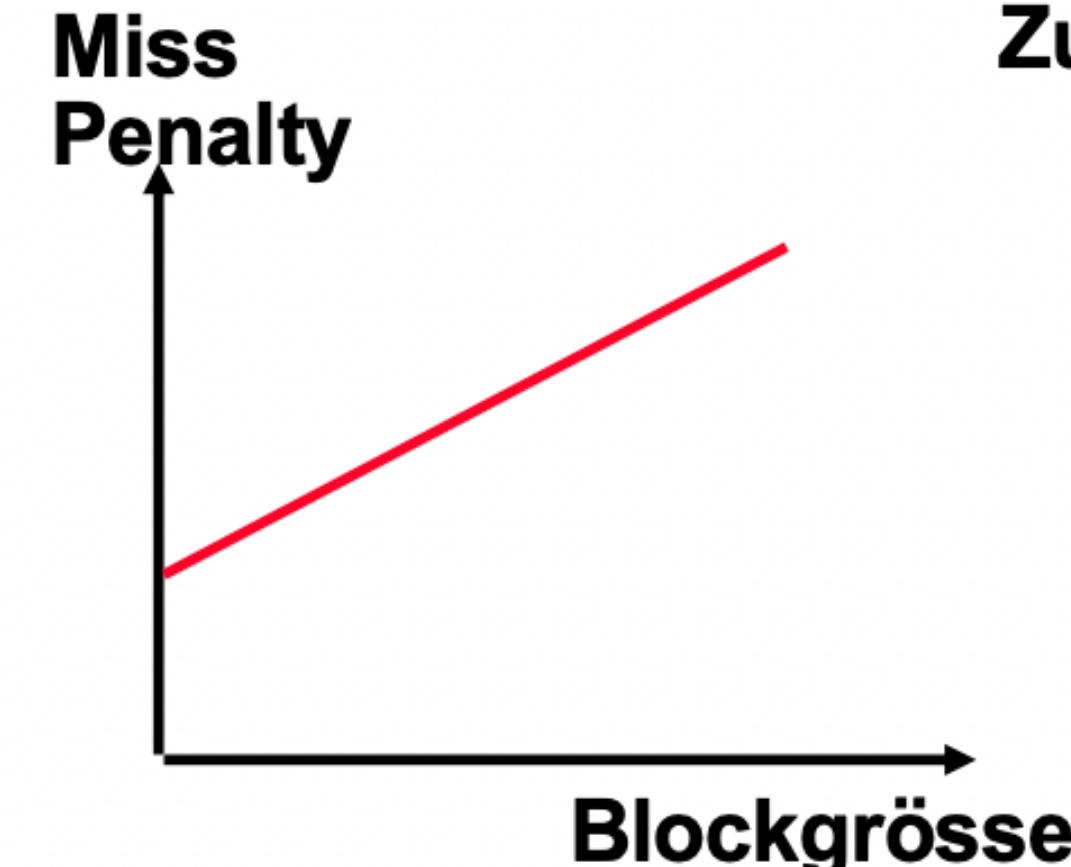
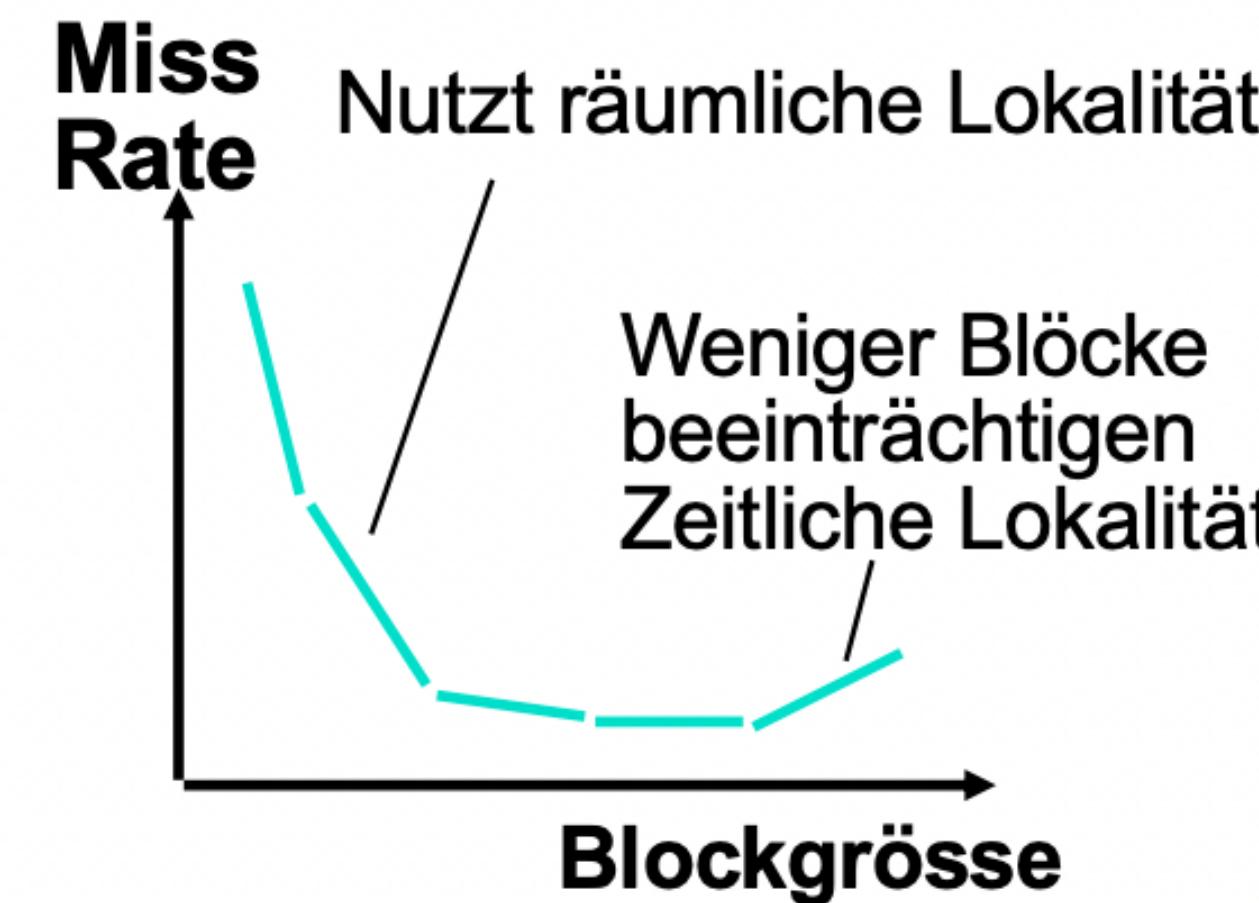
Durchschnittliche Zugriffszeit



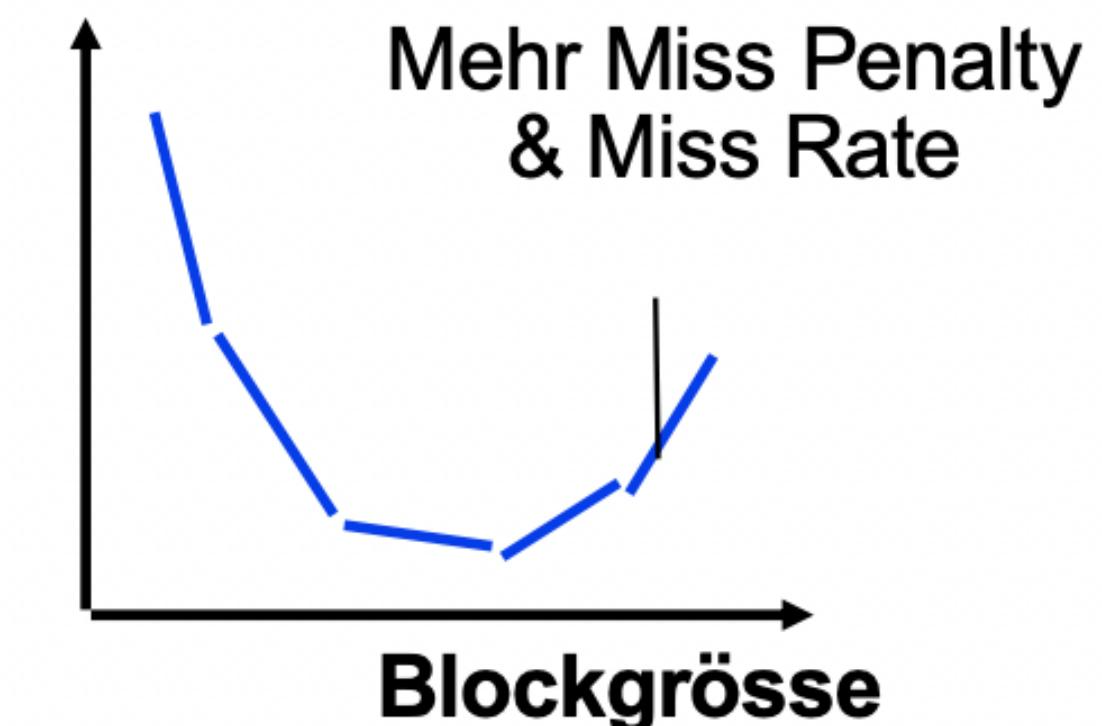
Allgemein, Durchschnittliche Speicherzugriffszeit = Hit Time + Miss Penalty x Miss Rate



MULTIWORT BLOCK ÜBERLEGUNGEN



Durchschnittliche Zugriffszeit



Allgemein, Durchschnittliche Speicherzugriffszeit = Hit Time + Miss Penalty x Miss Rate



ESTHI

CACHE ZUSAMMENFASSUNG

DAS LOKALITÄTSPRINZIP

