

1 Grundlagen

von Neumann: gemeinsamer Speicher

Harvard: getrennter Speicher

RISC: Reduced Instruction Set Computer

CISC: Complex Instruction Set Computer

2 Single Cycle MIPS

RISC, Register Machine, 32 Bit

2.1 Befehlsformate

2.1.1 Register Type

OP	RS	RT	RD	shamt	func
6	5	5	5	5	6

OP-Code: legt die Funktion des Befehls fest

function: legt die Funktion der ALU fest

Register Source: Quellregister 1

Register Target: Quellregister 2

Register Destination: Zielregister

2.1.2 Immediate Type

OP	RS	RT	Imme
6	5	5	16

Immediate: zur Übergabe einer Konstanten

Register Source: Quell Operand

Register Target: Zielregister

2.1.3 Jump Type

OP	Target
6	26

3 Multi Cycle MIPS

4 Ausnahmebehandlung

interne Ausnahmen: - arithmetischer Überlauf
externe - Interrupts

4.1 Hazards

4.1.1 Structure Hazard

Es stehen nicht genug Ressourcen zur Verfügung.
Im Normalfall nicht vorhanden bei verwendeter Architektur

4.1.2 Data Hazards

Datenabhängigkeit

4.1.3 Control Hazard

Die Befehlsreihenfolge ist für die nächsten n Takte bekannt.

5 Cache

Hit Rate: Hits pro Zugriffe

Miss Rate: $1 - HR$

$Zeile = \lfloor WA \div Words \rfloor \bmod Blocks$

$x \bmod y = x - \lfloor \frac{x}{y} \rfloor \cdot y$

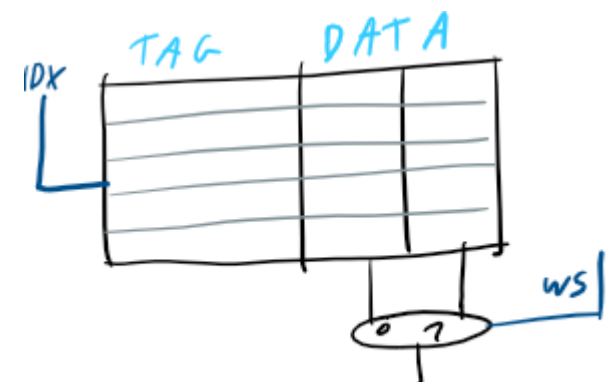
BS = Immer die letzten zwei

HIT	Write Through Wert im HS aktualisieren	Write Back nur im Cache update, Dirty Bit
Miss	Write Allocate Daten aus HS in Cache laden	Write Around nur in HS aktualisieren

M	miss
S	spatial hit
T	temporal hit
W	write to cache & memory
WC	write to cache
WM	write to memory
B	write back to memory before replace
F	fetch on write miss

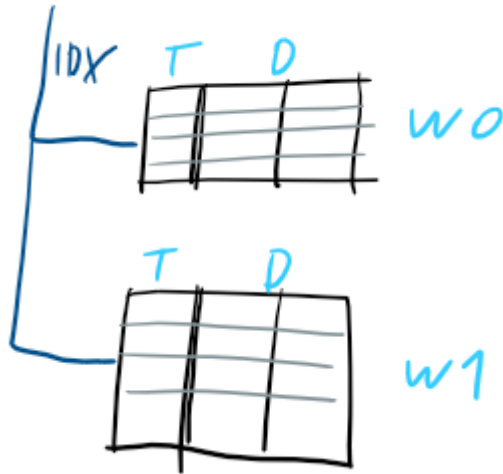
5.1 Direct Mapped Cache

+	-
einfacher schneller	jeder Index einmal abgebildet



5.2 Set-Associative Cache

+	-
höhere Hit Rate schneller	komplexer



Page: Physikalischer Abschnitt

Frame: Virtueller Abschnitt

Frames = $PAR \div PageSize$

Pages = $VAR \div PageSize$

Breite Frame # = $\log_2(\#Frames)$

Breite Page # = $\log_2(\#Pages)$

Offset = $\log_2(PS)$

$L1 \div L2 = BreitePage\# \div 2$ (Der größere Teil geht zu L2)

KiB	2^{10}
MiB	2^{20}
GiB	2^{30}

5.2.1 Least Recently Used

Laden von Zeile i	$C_i = 0$
Hit auf Zeile i	$C_i = 0$
Hit auf Zeile $j \neq i$ wenn $C_i \leq C_j$	$C_i = C_i + 1$
Miss	ersetzen der kleinsten Zeile alle anderen +1

6 Virtuelle Speicherverwaltung

VAR: Virtueller Adressraum

PAR: Physikalischer Adressraum