

## TG1 A6

CPU Taktfrequenzen, meist in MHz =  $10^6$  Hz, GHz =  $10^9$  Hz

Länge des Taktintervalls

$$= \frac{1}{\text{Taktfrequenz}} = \text{z.B. } \frac{1}{16} \cdot \frac{1}{10^6} = \frac{1}{16 \text{ MHz}} = 62,5 \cdot 10^{-9} = \underline{\underline{62,5 \text{ ns}}}$$

Anzahl der Taktzyklen pro Befehl

$$\frac{\text{Taktfrequenz}}{\text{Befehle pro Sekunde}} \quad \text{im Bsp } \frac{16 \text{ MHz}}{8 \text{ Mio}} = \frac{16}{8} \cdot \frac{10^6}{10^6} = \underline{\underline{2 \text{ TZ/Befehl}}}$$

Anzahl Befehle für ein Programm

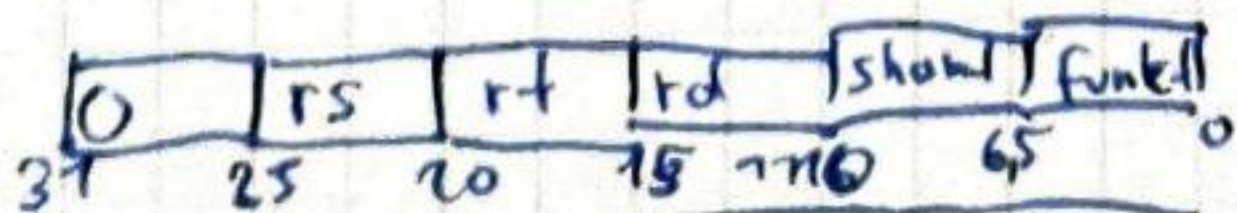
Dauer in Sekunden  $\cdot \frac{\text{Taktfrequenz}}{\text{Taktzyklen pro Befehl}}$

$$\text{im Bsp } 4,5 \text{ s} \cdot \frac{16 \text{ MHz}}{2} = \frac{72}{2} \cdot 10^6 = \underline{\underline{36 \cdot 10^6 \text{ Befehle}}}$$

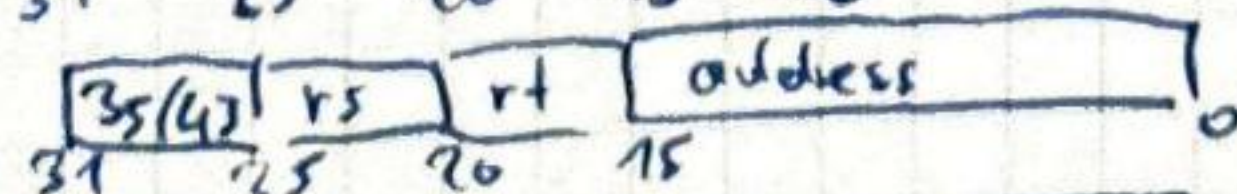
	0	1
RegDst	rt - Zielregister	rd - Register
RegWrite	OFF	Write Register nach write Data
ALUSrc	Read data 2 ON	16 Bit Sign extend
PCSrc		
MemRead	OFF	Adresse an read Data
MemWrite	OFF	Adresse an write Data
MemtoReg	Write Data von ALU	Write Data vom Speicher

Befehlsadresse 32 Bit

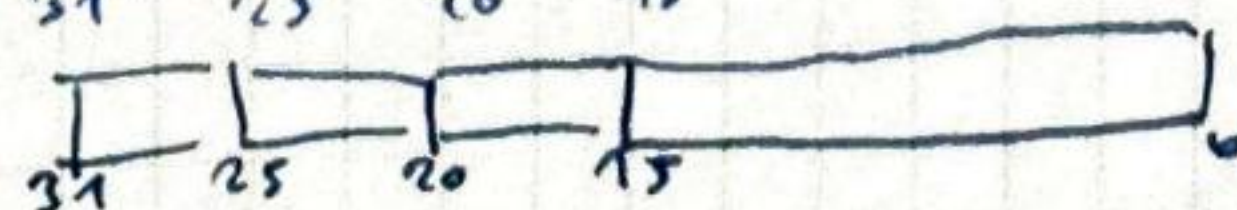
R-Format



I-Format



Verzweigung



Dont Care:  
Design von kompakteren  
Schaltungen,  
größere Blöcke im  
KV-Diagramm

	RegDst	ALUSrc	MemtoReg	RegW	MemR	MemW	Branch	ALUOP
R-Format	1	0	0	1	0	0	0	10
Lw	0	1	1	1	1	0	0	00
sw	x	1	x	0	0	1	0	00
beq	x	0	x	0	0	0	1	01

Instruction Memory wird immer gebraucht

Data Memory ist an wenn Werte ~~aus~~ ~~R~~ ausgelesen werden (lw)

Register werden gebraucht bei read & write Prozessen

add: r&w

beq: r

lw: r&w

Sign extend macht aus 16 Bit 32 Bit, wird für Sprung oder

Speicheradressen gebraucht. Bei lw für die Speicheradresse, beq: Sprung adresse

ALU Control wird immer gebraucht wenn die Haupt ALU gebraucht wurde

# A7 TGI

1 ns = 1000 ps

Länge eines Taktzyklus = Summe aller Stufen (ohne Pipeline)  
 = Dauer der längsten Stufe (mit Pipeline)

Benötigt ein Befehl alle Stufen braucht er  
 (Anzahl der Stufen) \* (längste Stufe) Länge (mit Pipeline).

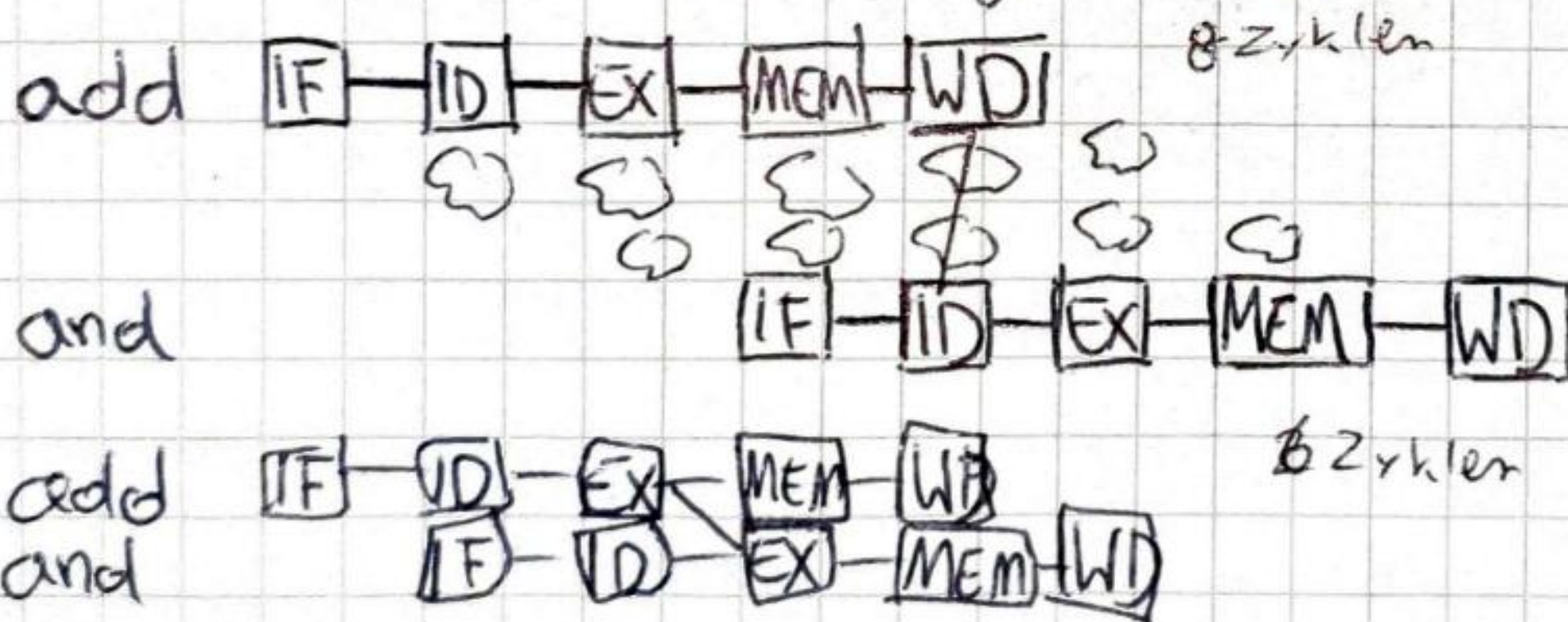
## MIPS PIPELINE

add  $\boxed{\$1}$ ,  $\$3$ ,  $\$4$   
 and  $\$2$ ,  $\boxed{\$1}$ ,  $\$4$

## MIPS STUFEN

- IF : Befehl aus dem Speicher holen
- ID : Lesen der Register
- EX : Operation ausführen
- MEM : Zugriff auf den Speicher
- WB : Ergebnis ins Register schreiben

Treten zwei von einander abhängige Befehle auf, so braucht es  
 zwischen diesen Befehlen ~~für~~ 2 andere Befehle oder 2 Bubbles,  
 wenn es kein Forwarding

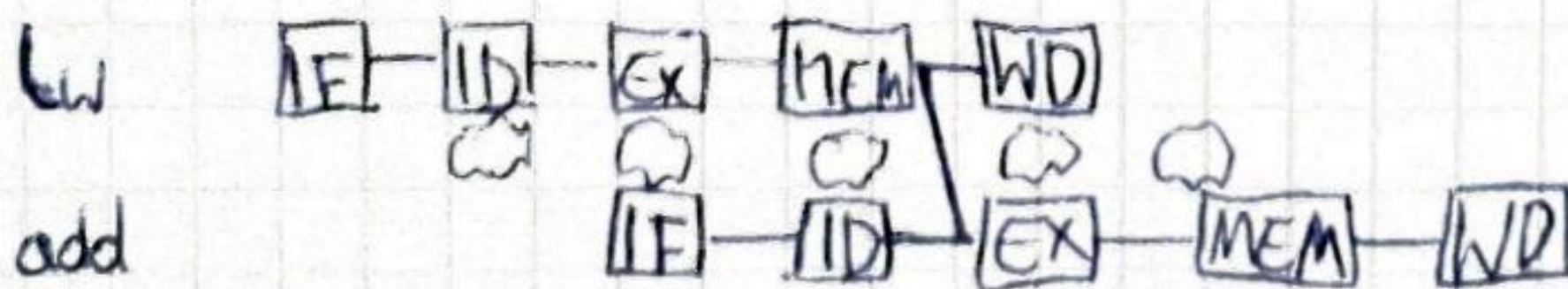


Mit Forwarding kann  
 das Ergebnis nach  
 EX zwischen gespeichert  
 und weiter geleitet werden  
 = keine Bubbles

## LOAD TO USE

Lw  $\boxed{\$3}$ , 8( $\$2$ )  
 add  $\$1$ ,  $\$2$ ,  $\boxed{\$3}$

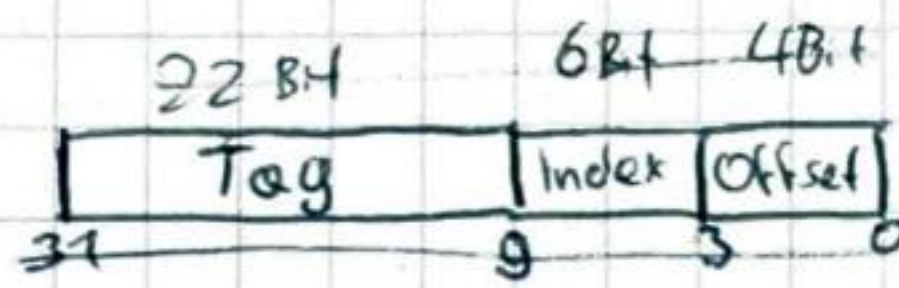
Hier kommt das Ergebnis aus MEM, daher auch  
 einen Leertakt (trotz Forwarding)



Bedingungsavwertung, ob gesprungen wird oder nicht stet erst ab  
MEM zur Verfügung, daher braucht es nach diesem  
Befehl 2 Leerfakte, damit diese Bedingung auch richtig ausgeführt  
wird

A8 TGI

CACHE



$KiB = 2^{10} B$   
 $1 Byte = 8 Bit$

$\frac{\text{Nutzdaten}}{\text{Blockgröße}} = 2^{\text{Index}}$

Bsp  $\frac{256 KiB}{16 B} = 2^{14}$  /  $2^{14}$  Blöcke / 14 Bit Index

$2^{\text{Index}} = \text{Blockanzahl}$   
 $2^{\text{offset}} = \text{Blockgröße (Byte)}$

4 Bit Offset :  $32 - 14 - 4 = 14 \text{ Bit TAG}$

$16 = 2^4$

Speicherbedarf (write through)

Speicher pro <sup>Eintrag</sup> Block : 1 Valid-Bit + Daten + Tag

Im Bsp:  $1 + 14 + 128 = 143$ , ~~143~~  $\neq 143 \text{ Bits} \cdot 2^{14} \text{ Blöcke}$   
 $143 \cdot 2^{-3} \cdot 2^{14} \text{ Byte}$   
 $= 286 \text{ KiB}$

Speicherbedarf (write-back)

Speicher pro Eintrag: 1 Valid-Bit + 1 Dirty-Bit + Daten + Tag

Vorteil: keine permanente Kommunikation mit dem Hauptspeicher  
Nachteil: Mehr Verwaltungsinformationen

Vierfach Satzassoziativ

$\frac{\text{Anzahl Blöcke}}{4} = 2^{\text{neuer Index}}$

Neue Blockanzahl, durch die neue Blockanzahl wird der Tag jetzt um 2 Bit größer.

Ein Cache Set =  $4 \cdot (1 \text{ Valid Bit} + \text{Tag} + \text{Daten})$

Nutzdaten = Cache Set  $\cdot$  Neue Blockanzahl.

Im Bsp.

$\frac{2^{14}}{4} = 2^{12}$

$32 - 12 - 4 = 16 \text{ Tag}$

$4 \cdot (1 + 16 + 128) = 580 \text{ Bits}$

$580 \cdot 2^{12} = 2375680 \text{ Bits} = 290 \text{ KiB}$

Blocknummer = Blockadresse modulo Blockanzahl

Im Bsp

Block	Index		
1	1	Miss	wird geladen
1g	3	Miss	wird geladen
1	1	Hit	vorhanden
3	3	Miss	...

Trefferrate =  $\frac{\text{Hits}}{\text{Zugriffe}} \Rightarrow \frac{1}{4}$

### Vollassoziativer Cache

keine fixe Zuordnung von Speicheradressen und Position

Block kann an jeder Position platziert werden

Tag		Block 1	2	3	4	
0	Miss	0				
1	Miss	1	0			
2	Miss	2	1	0		} Bei Hit Eintrag weiter nach vorne
0	Hit	0	2	1		
4	Miss	4	0	2	1	} <del>1er</del> 1er fällt raus, da nur 4 Blöcke
3	Miss	3	4	0	2	

→ Blockanzahl 2<sup>index</sup>

A9

TGI

## VIRTUELLER SPEICHER

Anzahl Bits für eine physische Adresse

physischer Hauptspeicher = 2<sup>32</sup> <sup>BitAdresse</sup> → Antwort!

Im Bsp: 4 GiB = 2<sup>32</sup> B ⇒ 32-Bit Adresse

Anzahl virtueller Seiten im System =  $\frac{\text{insgesamte Bytes}}{\text{Byte pro Seite}}$

Anz. virtuelle Seiten = 2<sup>(benötigte Bit zum nummerieren)</sup> / Einträge

Größe Tabelleneintrag mit Dirty und Valid Bit

Tabelleneintrag: physische Seiten Bit + 1 Valid + 1 Dirty

↳ Ergebnis in Bytes • Anzahl Einträge

Im Bsp

18 B + 2 = 20, runden auf ganze Bytes ≈ 24 Bit  
24 Bit = 3 Bytes  
3 · 2<sup>24</sup> Einträge

## Synchroner Bus

Dauer Taktzyklus =  $\frac{1}{\text{Takt}}$

2 · Taktzyklus - T<sub>m</sub> - T<sub>bs</sub> = Dauer Leseoperation

A10

TGI

NETZWERKE

- R ... Verbindung zwischen zwei Hosts, meist bps
- m ... Entfernung — " — , meist Meter
- s ... Ausbreitungsgeschwindigkeit, meist Meter/Sekunde
- L ... Größe eines Datenpakets, meist Bit

Ausbreitungsverzögerung

$$d_{\text{Ausbreitung}} = \frac{m}{s}$$

Übertragungsverzögerung

$$d_{\text{Übertragung}} = \frac{L}{R}$$

$$d_{\text{gesamt}} = \frac{m}{s} + \frac{L}{R}$$

Host A sendet um  $t=0$  Position letztes Bit bei  $t=d_{\text{Übertragung}}$ ?

Verlässt gerade Host A

Bei  $d_{\text{Ausbreitung}} > d_{\text{Übertragung}}$  (für das erste Bit)

hat Host B noch nicht erreicht

Maximale Anzahl an Bits in einer Leitung?

Dauer eines Bits:  $T_{\text{bit}} = \frac{1}{R}$  Sekunden

Abstand zwischen 2 Bits:  $L_{\text{bit}} = s \cdot T_{\text{bit}}$

d... Länge der Leitung (m)

Max. Bits in der Leitung d:  $\frac{d}{L_{\text{bit}}}$

Kurze Formel  
 $BDP = R \cdot \frac{d}{s}$

Im Bsp.

$$R \cdot d_{\text{Ausbreitung}} = R \cdot \frac{d}{s} = 2 \cdot 10^6 \cdot \frac{1000 \cdot 10^3}{2 \cdot 10^8} = \underline{\underline{10.000}}$$



## Dauer bis zum Erhalt einer Nachricht (mit Bestätigung)

Hinweg:  $\frac{\text{Paketgröße}}{R} + d_{\text{Ausbreitung}}$

Rückweg:  $\frac{\text{Bestätigungsgröße}}{R} + d_{\text{Ausbreitung}}$

Dauer für ein Paket + Bestätigung:  $\frac{\text{Paket} + \text{Bestätigung}}{R} + 2 \cdot d_{\text{Ausbreitung}}$

Im Bsp.  $\frac{4000 + 1000}{R} + 2 \cdot d_{\text{Ausbreitung}} = 12,5 \text{ ms pro Paket}$

20 Pakete = 250 ms

## IP Adressen

Netzwerkmaske: Darstellung der Submaske in ~~binär~~ Dezimal

Im Bsp. 175.149.56.0 27 Submaske 27 1er von links weg

1111 1111 1111 1111 1111 1000 0000 0000 = 255.255.248.0

(32-Submaske-2)

Anzahl IP Adressen in einem Netzwerk = 2

32 - Submaske = Bit für Teilnehmer Net-10

Die erste und die letzte Adresse wird nicht vergeben

daher  $2^{\text{Teilnehmer} - 2}$  Adressen möglich

Netzwerkadresse: Host-Bit auf Null

Broadcastadresse: Host-Bit auf 1

10101111 11111111	10101100 11111111 Net-10	00010100 11111000	10000100 00000000 Net-10
----------------------	--------------------------------	----------------------	--------------------------------

## Subnetze einrichten

bei 4 weiteren Subnetzen muss die maske um 2 Bit verlängert werden, da  $2^2 = 4$

Netz 1 00

Netz 2 01

Netz 3 10

Netz 4 11

Bsp 2. Netz Maske

175.149.0011 1100 . 0000 0000

175.149.0010 1000 . 0000 0000

→ 2. Netz