

3. Speicherarchitekturen

Je näher am Prozessor, desto kleiner aber schneller

Direkt im Prozessor: Register (4 bis 128)

Auf Prozessorschaltkreis 1-Level-Cache

Extern 2-Level-Cache (zwischen Hauptspeicher und 1-Level-Cache)

Hauptspeicher (einige Kilobyte

(Microcontroller) bis mehrere Gbyte im PC-Bereich)

Hintergrundspeicher (Festplatte, Optische Laufwerke)

LAN-Server größere Speicher, über lokales Netz, zeitgeteilte Nutzung durch mehrere Nutzer

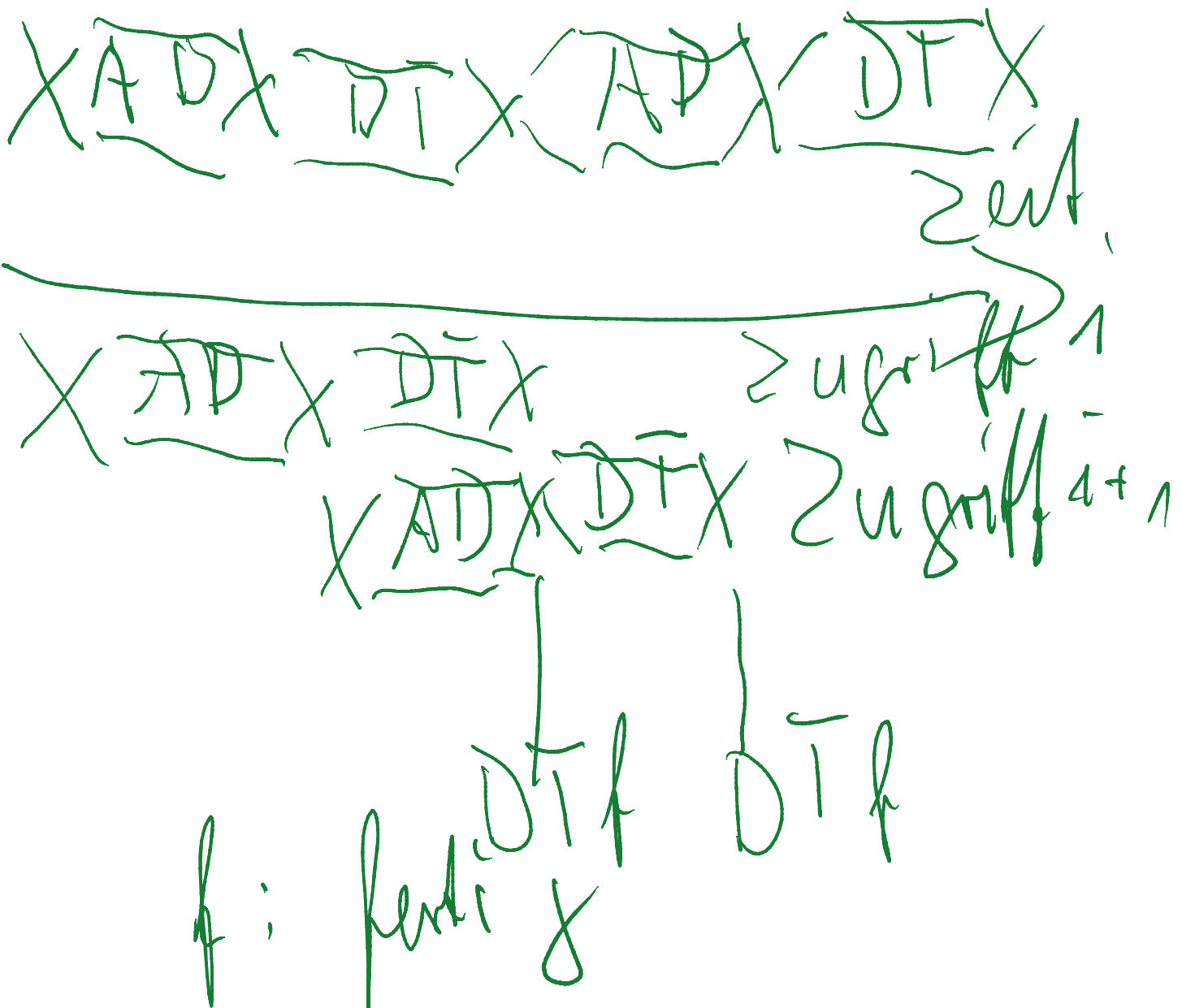
Internet: Zusammenstellung verschiedenster und unterschiedlich zu nutzenden Daten

3.1. Konventionelle Maßnahmen zum schnellen Speicherzugriff

Adresspipelinig:

→ Zerlegen des Speicherzugriffs in Phasen:

1. Adressieren und Dekodieren (AD)
2. Datentransfer (DT)

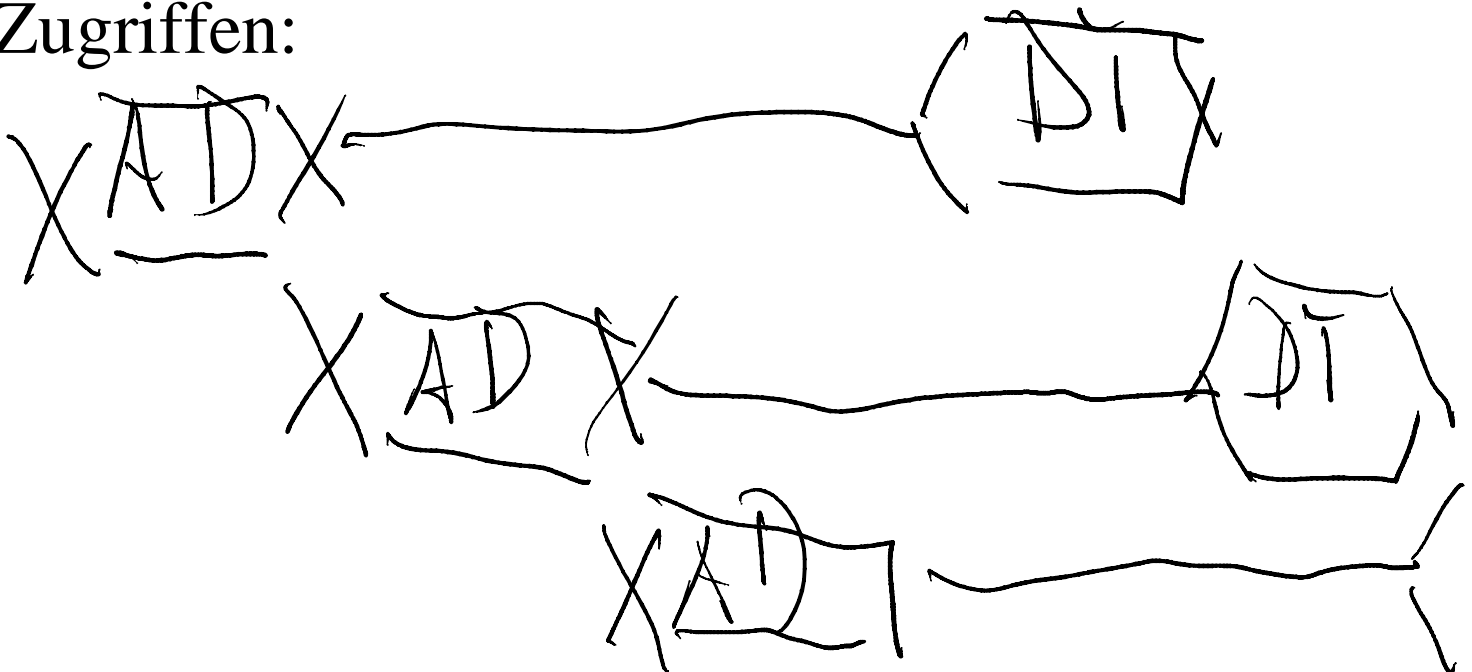


- Einzelner Speicherzugriff wird nicht schneller
- Aber im Mittel werden doppelt so viele Zugriffe möglich

Interleaving:

Es wird benutzt, dass oft aufeinander folgende Speicherzellen benötigt werden.

Zeitverlauf bei aufeinander folgenden Zugriffen:



$$B0 = /A2 \ \& \ /A3$$

$$B1 = A2 \ \& \ /A3$$

$B2 = \neg A2 \ \& \ A3$

$B3 = A2 \ \& \ A3$

A1 und A0 werden für die Byteauswahl im Speicherwort genutzt (hier 32-bit-Speicherwort)

Burst-Zyklusverfahren:

Zugriff auf aufeinander folgende Speicherzellen.

Zeitgewinn durch Reduzierung der Dekodierung, da der schaltkreis nach einer Dekodierung die nächsten Speicherinhalte auch zur Verfügung stellt.

Interleavig und Burstzyklus: F27 unten: Relativ schnelle aufeinander folgende Ausgabe der Adressen, verzögerte Rückgabe der Daten im Block (Burst) durch mehrere Bänke können die Folgeadressen eher ausgegeben werden.

3.2. Cachespeicher

Ziel ist, in einem kleinen und damit schnellen Speicher die Befehle und Daten zu speichern, die mit großer Wahrscheinlichkeit wieder gebraucht werden.

Cachezugriff F28:

1. Prozessor gib Adresse $A(j)$ an Speicher lesen bzw Schreiben aus
2. Übergabe der Adresse $A(j)$ an den Cache
3. Vergleich der Adresse $A(j)$ mit allen im Cache vorhandenen Adresse
4. Datentransfer mit dem Cache
5. Bei Lesen: Daten der Hauptspeicherkopie aus dem Cache übernehmen. Mit dem Hauptspeicherwert muss nichts erfolgen.
6. Beim Schreiben wird die Hauptspeicherkopie im Cache und der Hauptspeicherwert überschrieben.

Problem:

Schneller Vergleich der aktuell vom Prozessor ausgegebenen Adresse mit allen im Cache vorhandenen Adressen parallel ist mit normalen Aufwand nicht zu realisieren.

Dieses Prinzip ist in F29 dargestellt. (jede Hauptspeicheradresse im Cache möglich.)

Vergleicherprinzip: F30/31

Für jeden Speichereintrag ein Vergleicher.

Reduktion des Vergleicheraufwandes:

Direct Mapped Cache:

- Aufteilen der Prozessoradresse in zwei Teile: Tag wird zum Vergleich benutzt
- Index adressiert direkt (F33)

Zwei-Wege-Cache: Funktion von zwei Einwegecaches (F34) auch n-Wege möglich, und beim größten möglichen n (n ist Zahl der Speicherzellen) -> vollassoziativer Cache

Tracecache: Spezieller Cache, nur für Befehle.

Grundidee: Befehlsfolgen werde oft mehrfach durchlaufen (z.B. in einer Schleife) (F36)