

➤ **Anmeldung zur TI-Klausur bis spätestens 27. August 2003 durch**

- Einwurf der Zulassungsbescheinigung in den Briefkasten „Technische Informatik“ im Untergeschoss des Informatik-gebäudes am Fasanengarten. Es handelt sich um den gleichen Briefkasten, in den die Übungsblätter eingeworfen werden.

UND

- **On-line-Anmeldung auf der TI-Homepage**



Kapitel 5

➤ **Digitale Signalprozessoren**

- Aufbau eines digitalen Signalverarbeitungssystems
- Basisarchitektur digitaler Signalprozessoren

➤ **Mikrocontroller**



Mikrocontroller

Mikrocontroller: Mikrorechner auf einem Chip

- Für spezielle Anwendungsfälle zugeschnitten
- Meist Steuerungs- oder Kommunikationsaufgaben
- Anwendung oft einmal programmiert und für die Lebensdauer des Mikrocontrollers auf diesem ausgeführt
- Anwendungsfelder sind breit gestreut
- Oft unsichtbar in uns umgebenden Geräten verborgen



Mikrocontroller

Anwendungsfelder

➤ im Haushalt

- die Steuerung der Kaffeemaschine,
- der Waschmaschine,
- des Telefons,
- des Staubsaugers,
- des Fernsehers, ...

➤ in der Automatisierung

- das Steuern und Regeln von Prozessen,
- das Überwachen von Prozessen,
- das Regeln von Materialflüssen,
- die Steuerung von Fertigungs- und Produktionsanlagen, ...

➤ in der KFZ Technik

- das Motormanagement,
- das Antiblockiersystem,
- das Stabilitätsprogramm,
- die Traktionskontrolle,
- diverse Assistenten, z. B. beim Bremsen, ...

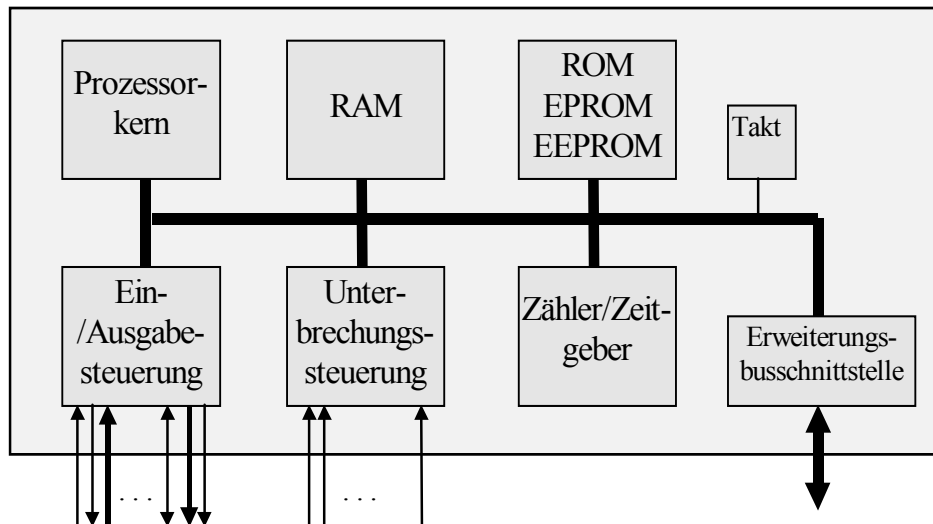


Mikrocontroller

Abgrenzung zu Mikroprozessoren

Ein-Chip Mikrorechner mit aufgabenspezifischer Peripherie

Mikrocontroller

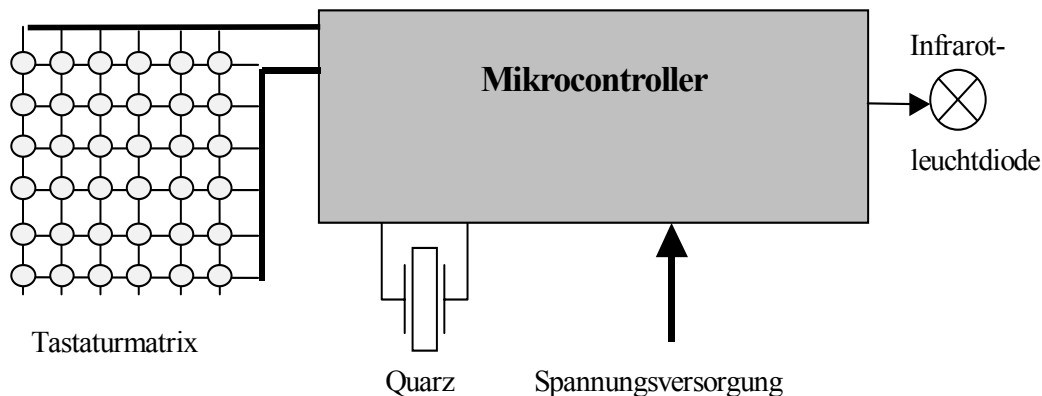


Mikrocontroller

Ziel: Möglichst wenige externe Bausteine für eine Steuerungsaufgabe

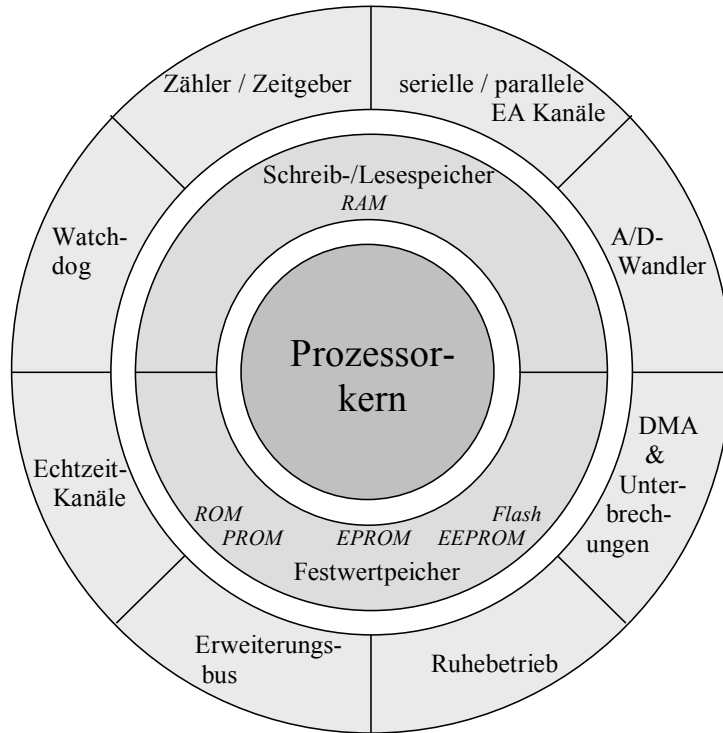
Idealfall: Mikrocontroller, Quarz, Stromversorgung sowie ggf. Treiber und ein Bedienfeld

Beispiel: Fernbedienung



Mikrocontroller

Schalenmodell eines Mikrocontrollers:



Mikrocontroller

Prozessorkern:

- prinzipiell kein Unterschied zum Kern eines Mikroprozessors
- Kosten spielen jedoch meist die dominante Rolle
→ **einfacher als der Kern eines Mikroprozessors**

Varianten:

1. Eigens für den Mikrocontroller entwickelter einfacher Kern



2. Verwendung älterer Kerne von Mikroprozessoren

- bewährte Technik, Kompatibilität, reduzierte Kosten
- Leistungsvermögen meist ausreichend
- Modifikationen:
 - Stromsparmodus
 - kein Cache
 - keine virtuelle Speicherverwaltung

➔ Reduktion des Stromverbrauchs, Verbesserung des Echtzeitverhaltens



Speicher

- integrierter Festwert- und Schreiblesespeicher
- Aufnahme von Daten und Programmen
- Vorteil: Einsparung von Anschlüssen und Decodierlogik bei vollständiger interner Speicherung
- Größe und Typ des Speichers unterscheiden oft verschiedene Untertypen desselben Mikrocontrollers
- z. B. je nach Stückzahl der Anwendung unterschiedlicher Typ des Festwertspeichers (ROM, PROM, EPROM, EEPROM, FLASH)



Mikrocontroller

Serielle und parallele Ein-/Ausgabekanäle

- grundlegende digitale Schnittstellen eines Mikrocontrollers
- seriell oder parallel
- synchron oder asynchron

AD/DA-Wandler

- grundlegende analoge Schnittstellen eines Mikrocontrollers
- Anschluss analoger Sensoren und Aktoren
- Auflösung und Wandlungszeit sind die wichtigsten Größen
- AD-Wandler sind häufiger anzutreffen als DA-Wandler



Mikrocontroller

Zähler und Zeitgeber

- im Echtzeitbereich ein wichtiges Hilfsmittel
- für eine Vielzahl unterschiedlich komplexer Anwendungen einsetzbar

Bsp:

- Zählen von Ereignissen, Messen von Zeiten kommen mit einem Zähler bzw. Zeitgeber aus
- Pulsweitenmodulation, Frequenz- oder Drehzahlmessung, Schrittmotorsteuerungen benötigen mehrere Einheiten

➔ Die bei Mikrocontrollern verfügbare Bandbreite reicht von einfachen Up-/Downcountern über Capture-Compare-Einheiten bis zu autonomen Zeitgeber-Coprozessoren



Watchdog

- „Wachhund“ zur Überwachung der Programmaktivitäten eines Mikrocontrollers
- Programm muss in regelmäßigen Abständen Lebenszeichen liefern
- Bleiben diese aus, so nimmt der Wachhund einen Fehler im Programmablauf an → **Reset**

Beispiel: Mars Sojourner Mission



Echtzeitkanäle

- Echtzeiterweiterung der parallelen E/A-Kanäle
 - Kopplung eines parallelen Kanals mit einem Zeitgeber
- der Ein-/Ausgabezeitpunkt wird von der Hardware und nicht der Software bestimmt



Unterbrechungen (*Interrupts*)

- Unterbrechung des Programmablaufs bei Ereignissen
- Schnelle, vorhersagbare Reaktion auf Ereignisse
- Insbesondere wichtig bei Echtzeitanwendungen
- Behandlung eines Ereignisses durch eine *Interrupt-Service-Routine*
- Mikrocontroller kennen meist externe Unterbrechungsquellen (Eingangssignale) und interne Unterbrechungsquellen (Zähler, Zeitgeber, E/A-Kanäle, ...)



DMA (*Direct Memory Access*)

- Direkter Datentransfer zwischen Peripherie und Speicher ohne Beteiligung des Prozessorkerns
- Höhere Datenraten durch spezielle Transferhardware
- Entlastung des Prozessorkerns
- Prozessorkern muss lediglich die Randbedingungen des Transfers festlegen
- Meist in Mikrocontrollern gehobener Leistungsklasse zu finden



Ruhebetrieb (*Standby Mode*)

- Oft begrenzter Energievorrat oder Wärmeemission bei Mikrocontroller-Anwendungen
- Ruhebetrieb zur Reduktion des Energieverbrauchs
- Abschaltung nicht benötigter Peripheriekomponenten und Festwertspeicher
- Erhaltungsspannung am Schreib-/Lesespeicher
- Statische Prozessorkerne erlauben einen Takt von 0 Hz



Erweiterungsbus

- Reichen die internen Komponenten eines Mikrocontrollers für eine Anwendung nicht aus → Erweiterungsbus zum Anschluss externer Komponenten
- Ein Bus benötigt viele Anschlüsse (z. B. 16-Bit-Adressen, 8-Bit-Daten, 4 Steuersignale = 28 Anschlüsse) →
 - Multiplexing (z. B. Daten-/Adressmultiplexing: Daten und Adressen teilen sich die gleichen Leitungen)
 - Bus muss sich Anschlüsse mit internen Peripheriekomponenten teilen (Ressourcenkonflikt)
 - Stufenweise Reduktion des Adressraums



Prozessorkerne

- Einfache RISC- oder CISC Prozessorkerne
- Benötigen wenig Fläche
- Verhalten und Eigenschaften sind wohl bekannt
- Im Low-Cost-Bereich oft einfache 8-Bit-Kerne ohne Pipeline
 - ➔ einfacher Aufbau
 - ➔ sehr einfache zeitliche Vorhersagbarkeit



Beispiel:

<i>Befehl</i>		<i>Taktzyklen</i>
LOOP: IN	A,(10)	2
LD	(IX),A	5
INC	IX	2
DEC	B	2
JNZ	LOOP	5
Gesamt:		16

⇒ bei einer Taktfrequenz von 10 MHz
Ausführungszeit pro Schleifendurchlauf: $16 / 10 \text{ MHz} = 1,6 \mu\text{sec}$



Mikrocontroller Komponenten

Je komplexer ein Prozessorkern, desto schwerer wird die Vorhersage des Zeitverhaltens

- Pipeline → Pipeline-Konflikte
- Cache → Cache Misses
- Spekulation → Fehlspekulation

In Echtzeitsystemen interessiert die *Worst Case Execution Time* (WCET)

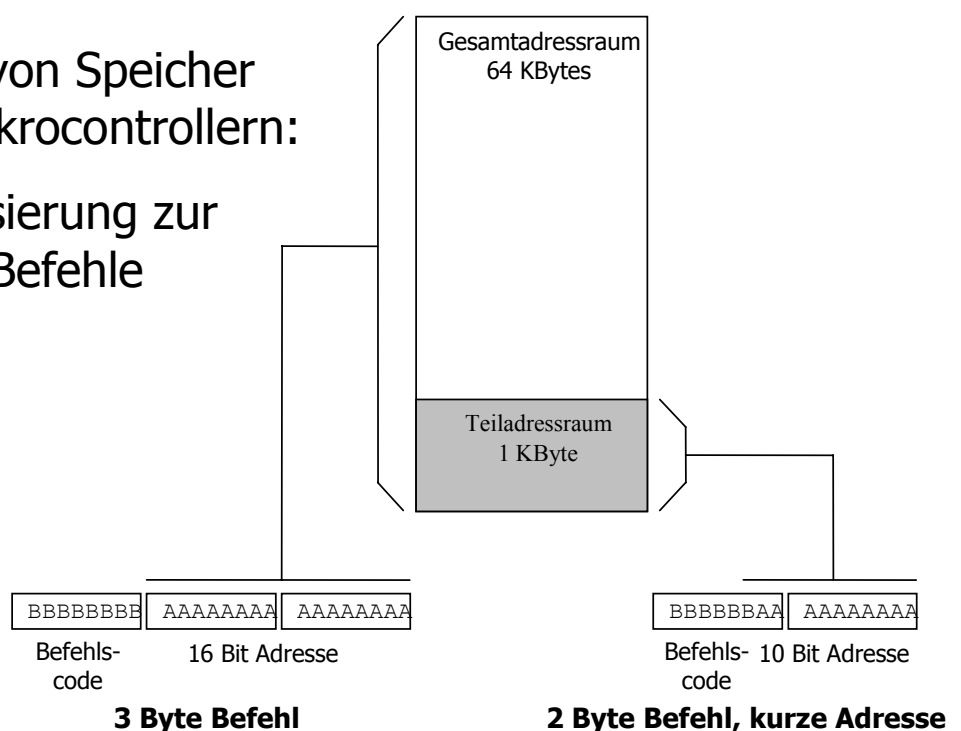
Aufwändige Analysen sind erforderlich (und nicht immer möglich)



Mikrocontroller Komponenten

Zur Einsparung von Speicher bei einfachen Mikrocontrollern:

Verkürzte Adressierung zur Verkürzung der Befehle



Mikrocontroller Komponenten

Zusätzlich:

- kurze Befehlscodes für die am häufigsten benutzten Befehle
- längere Befehlscodes für seltene Befehle

Besonders bei einfachen Mikrocontrollern ist Speicher meist eine knappe Ressource und muss optimal genutzt werden

Der Prozessorkern kann dies durch die genannten Maßnahmen unterstützen



Mikrocontroller Komponenten

1. Ein-/Ausgabeeinheiten

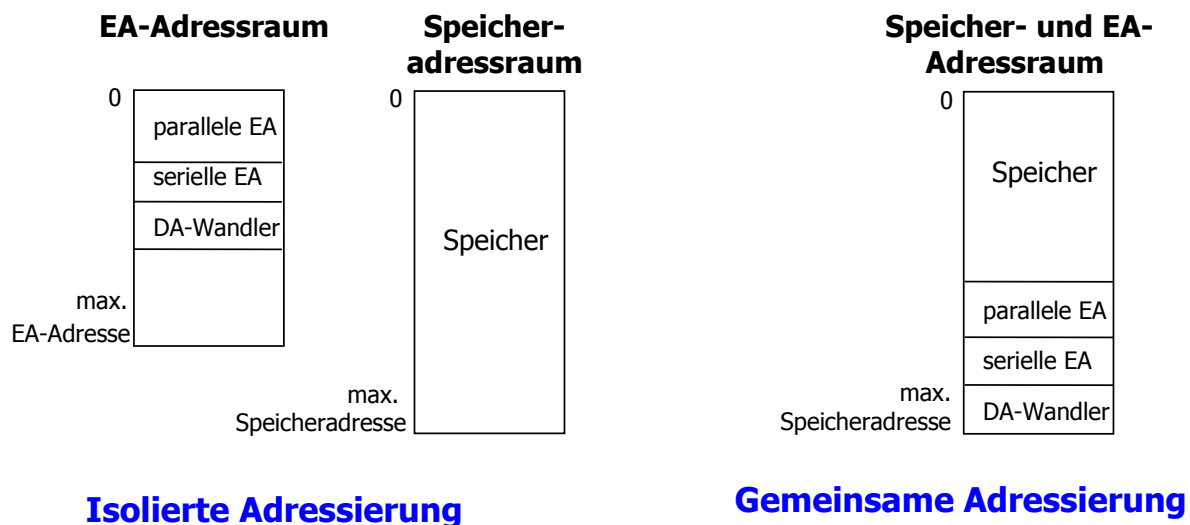
Bindeglied des Mikrocontrollers zur Umwelt

- analog/digital
- seriell/parallel
- Übertragungsraten
- Übertragungsformate
- Übertragungsaufwand

...



„Logische“ Anbindung an den Prozessorkern



Vorteile „isolierte Adressierung“:

- klare Trennung von Speicher- und Ein-/Ausgabezugriffen
- Speicheradressraum wird nicht durch EA-Einheiten reduziert
- schmalere Ein-/Ausgabeadressen → einfachere Hardware

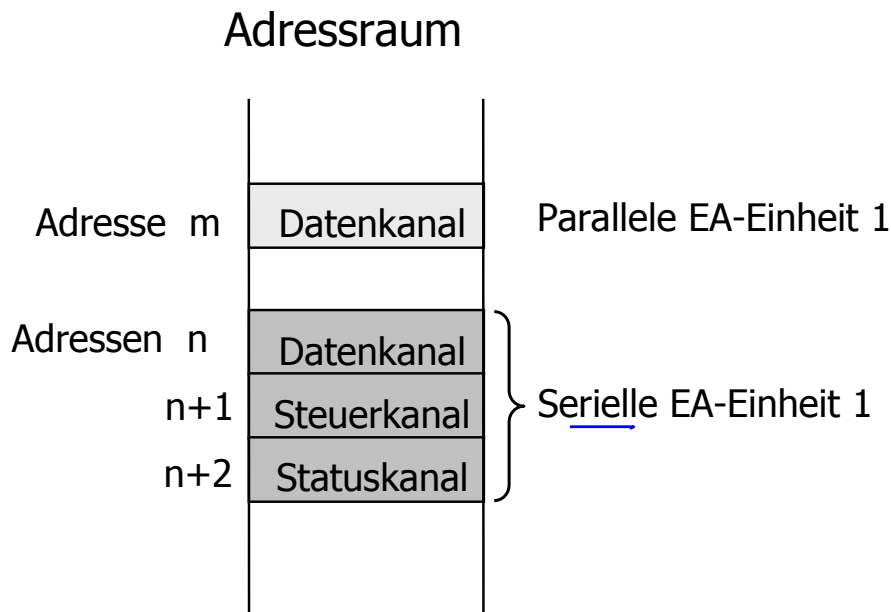
Vorteile „gemeinsame Adressierung“:

- Homogenität
- keine speziellen Befehle zur Ein-/Ausgabe erforderlich
- Alle Speicher-Adressierungsarten können auch zur Ein-/Ausgabe benutzt werden



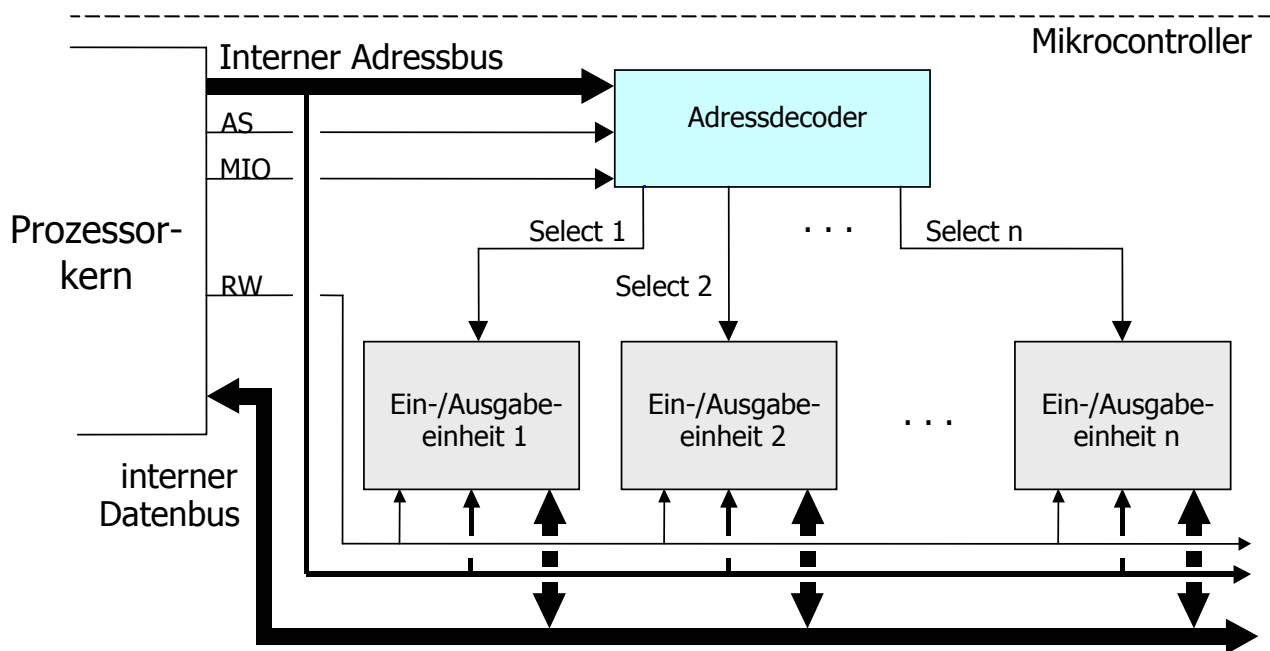
Mikrocontroller Komponenten

Adressraumbedarf unterschiedlich komplexer EA-Einheiten



Mikrocontroller Komponenten

Physikalische Anbindung durch **Adressdecoder**



Zur Anpassung unterschiedlicher Datenübertragungsraten von Prozessorkern und EA-Einheiten : **Synchronisation**

Von Seiten der EA-Einheit:

- Software-Synchronisation (z. B. XON/XOFF-Protokoll)
- Hardware-Synchronisation (z. B. RTS/CTS-Handshake)



2. Digitale parallele Ein-/Ausgabeeinheiten (parallele IO-Ports)

Charakteristika:

- Anzahl parallel übertragener Bits (meist Zweierpotenz)
- Ein-/Ausgaberichtung
- Übertragungsgeschwindigkeit

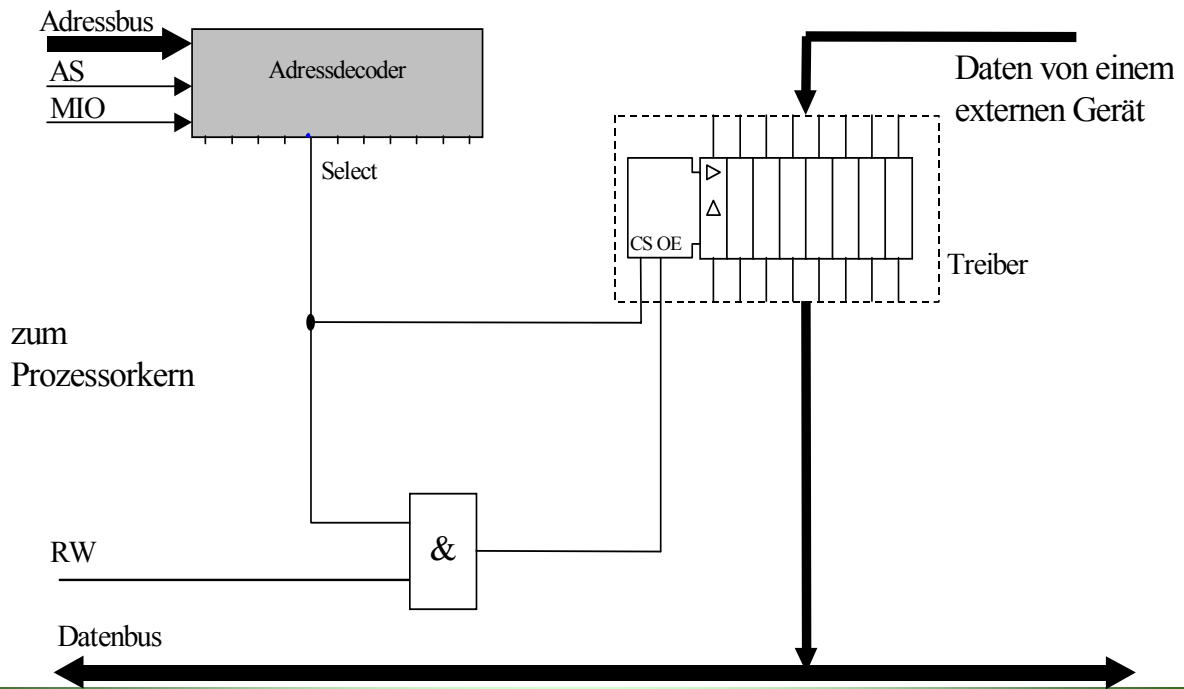
Der meist hohen Übertragungsgeschwindigkeit steht ein großer Bedarf an Anschlüssen gegenüber

→ meist teilen sich die parallelen EA-Einheiten eines Mikrocontrollers die Anschlüsse mit anderen Komponenten



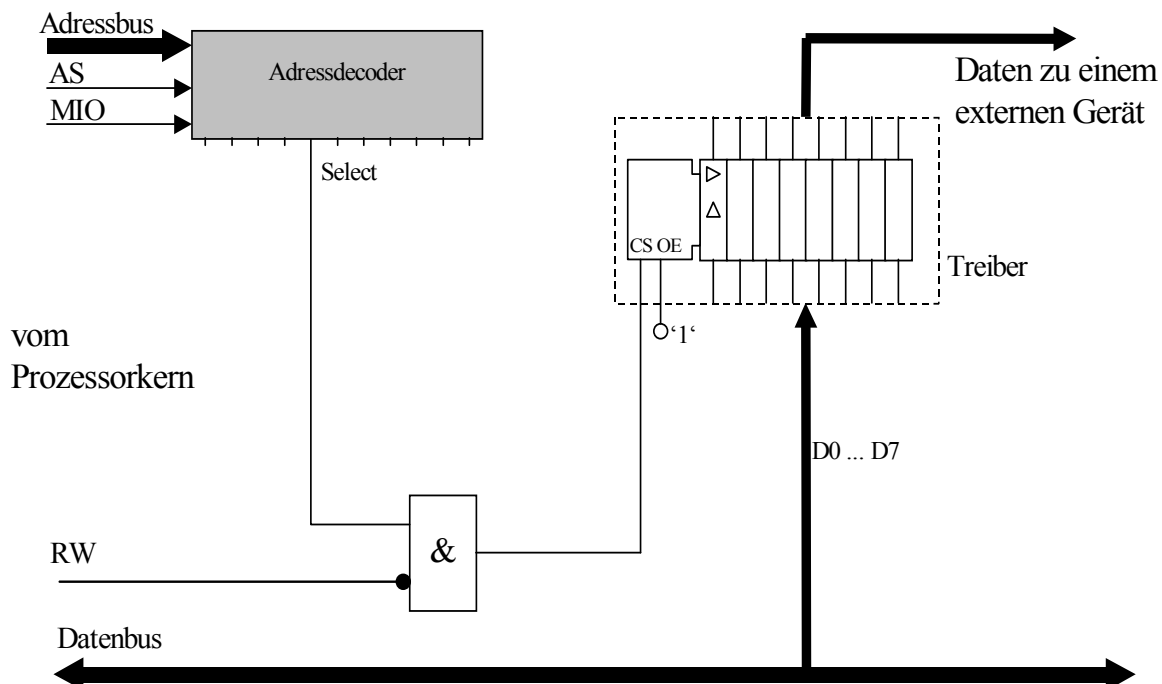
Mikrocontroller Komponenten

Einfache parallele Eingabeeinheit:



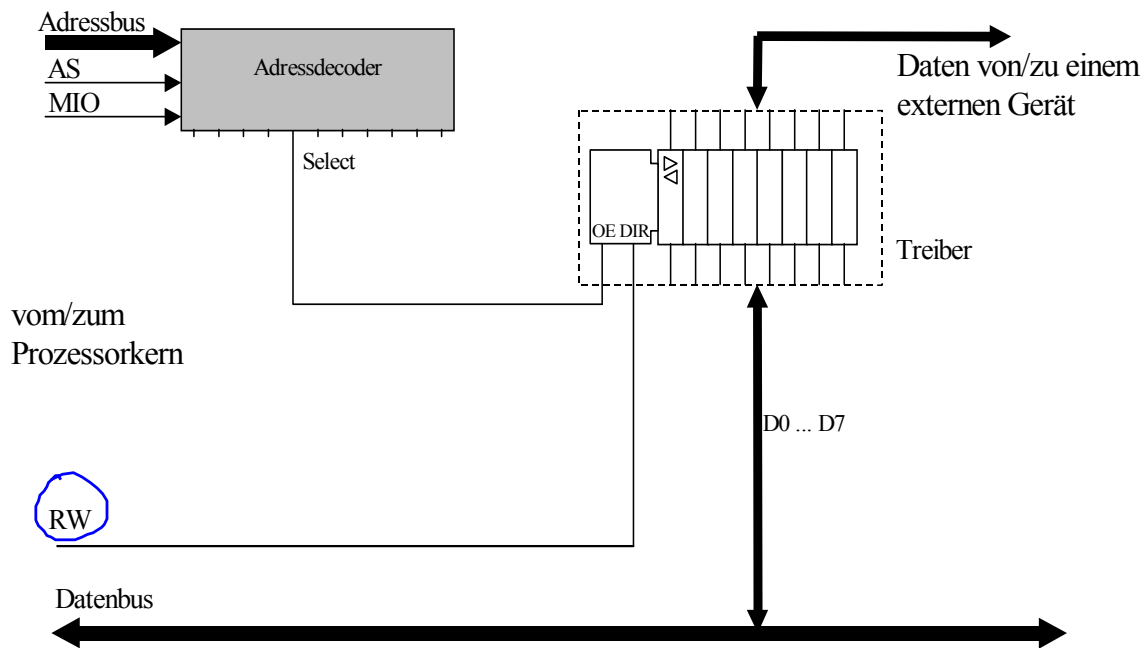
Mikrocontroller Komponenten

Einfache parallele Ausgabeeinheit:



Mikrocontroller Komponenten

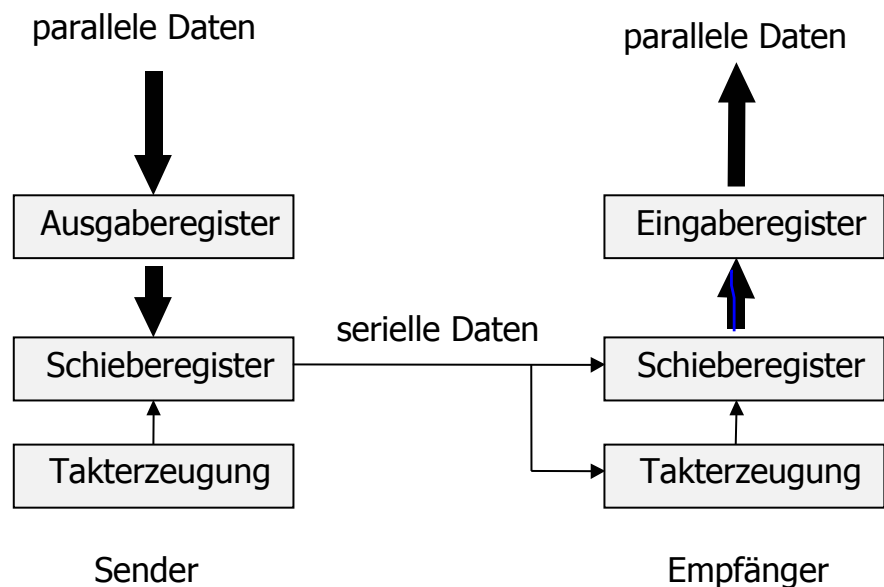
Einfache bidirektionale Einheit:



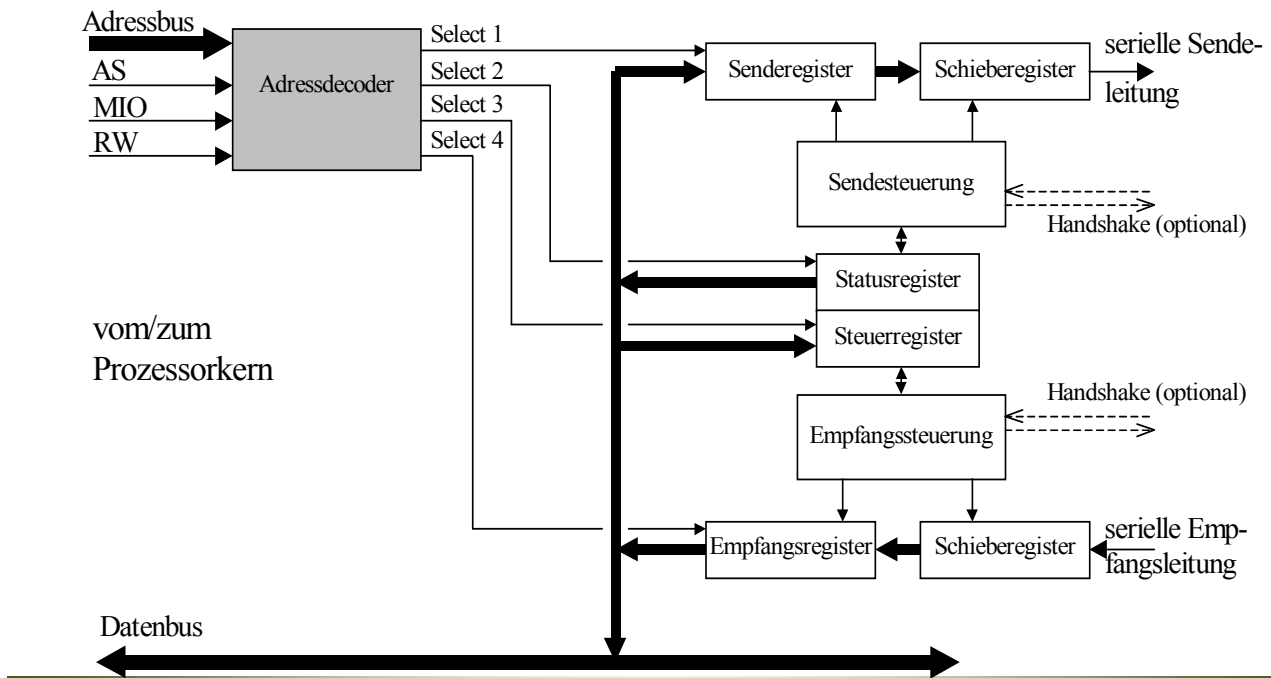
Mikrocontroller Komponenten

3. Digitale serielle Ein-/Ausgabeeinheiten (serielle IO-Ports)

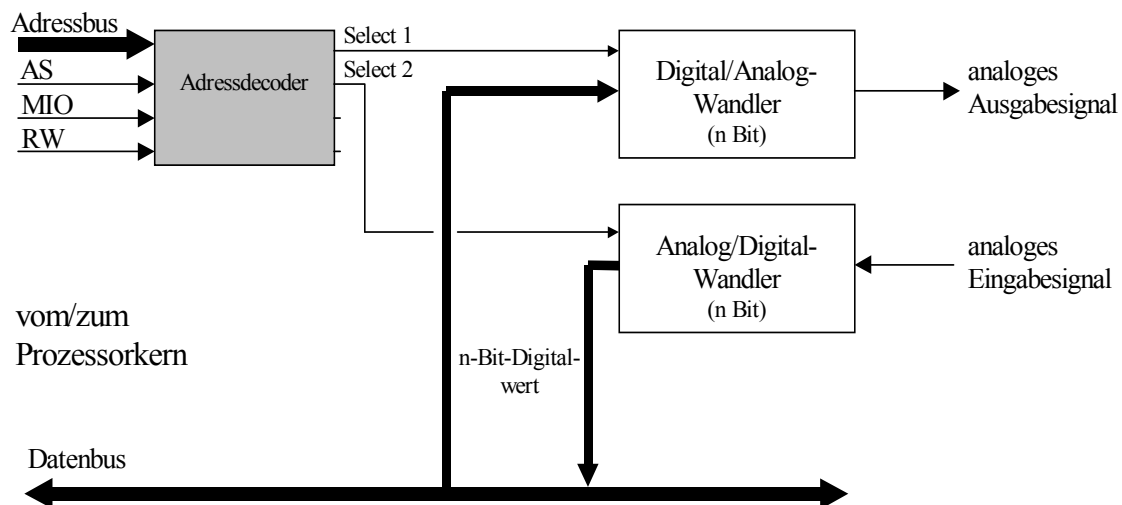
Grundprinzip:



Digitale serielle Ein-/Ausgabeeinheit



4. Analoge Ein-/Ausgabeeinheiten



zur Verarbeitung durch den Prozessorkern müssen analoge in digitale Signale gewandelt werden



Wandlungsfunktion:

bei n Bit Auflösung

→ Aufteilung in 2^n Schritte

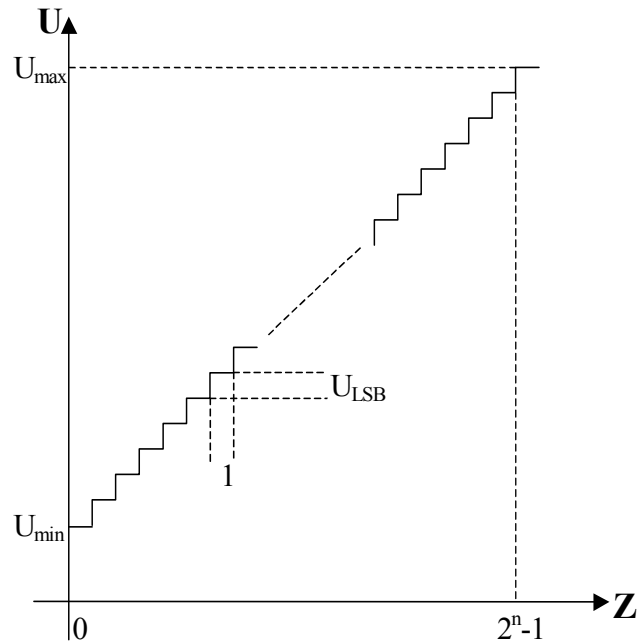
$$U_{LSB} = (U_{max} - U_{min}) / 2^n$$

Digital/Analog-Wandlung

$$U = (Z \cdot U_{LSB}) + U_{min}$$

Analog/Digital-Wandlung

$$Z = (U - U_{min}) / U_{LSB}$$



Beispiel: $U_{max} = 5 \text{ V}$, $U_{min} = 0 \text{ V}$, 12-Bit-Wandlung

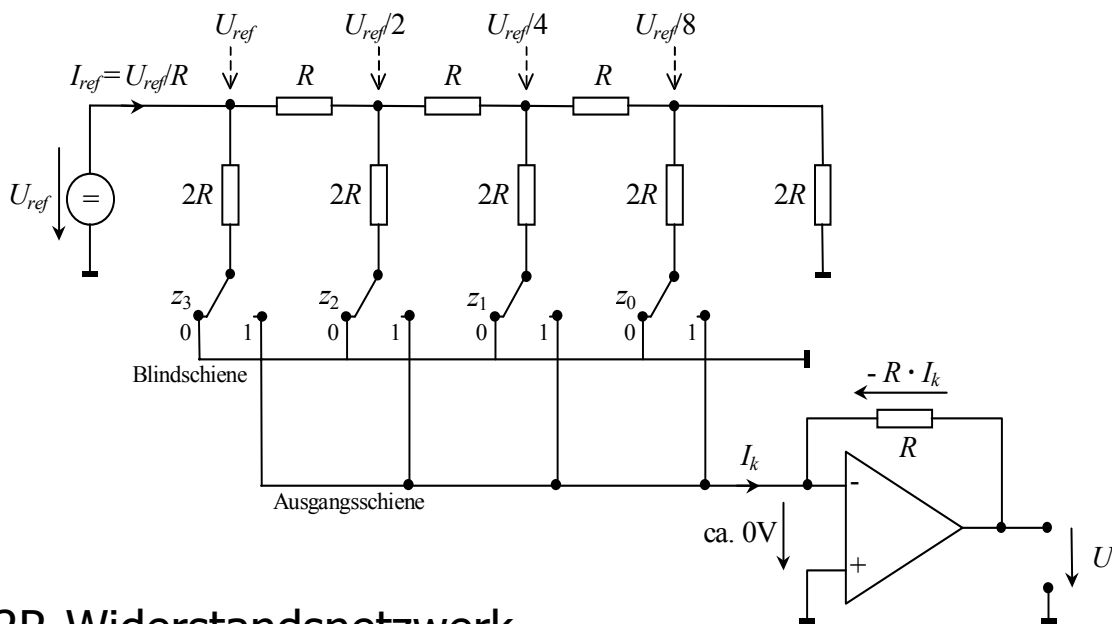
$$\rightarrow U_{LSB} = 1,221 \text{ mV}$$

Wichtigste Kriterien für die Auswahl eines Wandlers

- Auflösung (n Bit)
- Spannungsbereich ($U_{min} - U_{max}$)
- Wandlungszeit
- Wandlungsfehler



Digital/Analog-Wandlung



R/2R-Widerstandsnetzwerk



Funktion des Wandlers:

$$I_k = z_3 (U_{ref} / 2R) + z_2 (U_{ref} / 4R) + z_1 (U_{ref} / 8R) + z_0 (U_{ref} / 16R)$$

Für den Operationsverstärker gilt näherungsweise:

$$U = -R \cdot I_k$$

Daraus folgt:

$$\begin{aligned} U &= - (z_3 (U_{ref} / 2) + z_2 (U_{ref} / 4) + z_1 (U_{ref} / 8) + z_0 (U_{ref} / 16)) \\ &= - (z_3 2^3 + z_2 2^2 + z_1 2^1 + z_0 2^0) U_{ref} / 2^4 \\ &= - Z \cdot U_{ref} / 2^4 \end{aligned}$$

oder für n Bit: $U = - Z \cdot U_{ref} / 2^n$



Analog/Digital-Wandlung

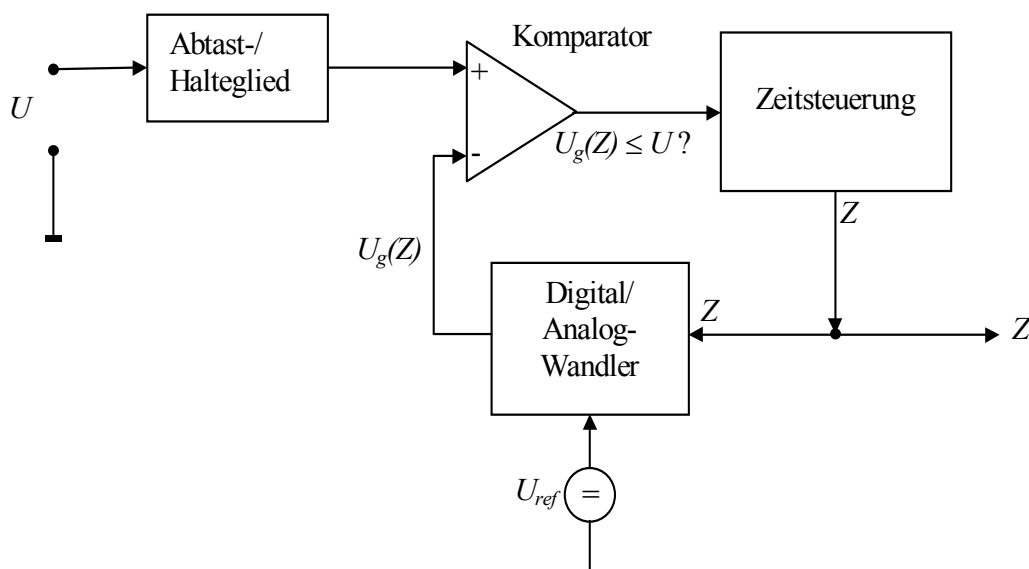
Prinzipielle Wandlungsverfahren:

- Parallelverfahren
sehr schnell, hoher HW-Aufwand
- Wägeverfahren
mittel schnell, mittlerer HW-Aufwand
- Zählverfahren (*Dual Slope*)
langsam, geringer HW-Aufwand, störunempfindlich



Mikrocontroller Komponenten

Beispiel: das Wägeverfahren



wandelt n Bits in n Schritten

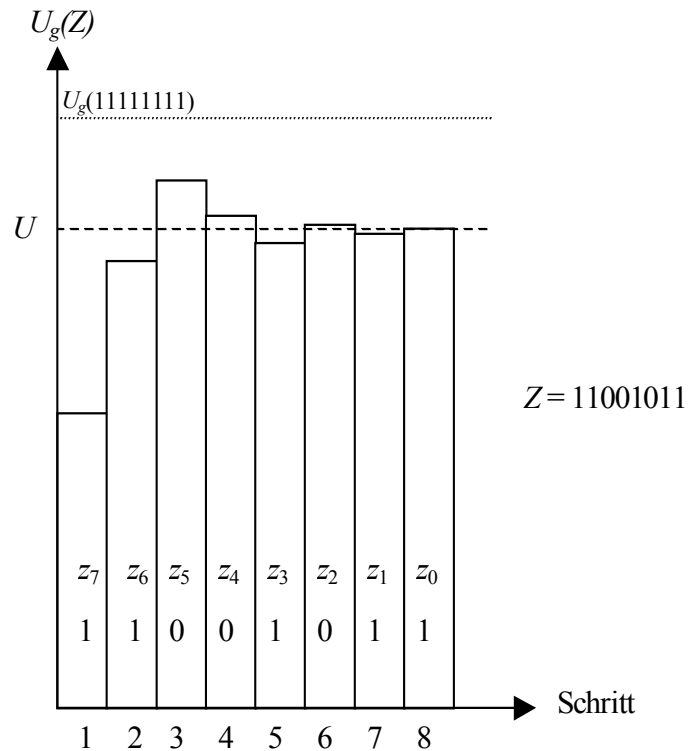


Mikrocontroller Komponenten

Ablauf:
sukzessive Approximation

Wandlerfunktion:

$$\begin{aligned} Z &= \lfloor 2^n \cdot U / U_{ref} \rfloor \\ &= \lfloor U / U_{LSB} \rfloor \end{aligned}$$



Vorlesungen im Hauptdiplom

- Rechnerstrukturen (Karl, SS)
- Systemarchitektur (Liefländer, WS)
- Mikrorechnertechnologie I+II (Brinkschulte, WS+SS)
- Robotik I+II+III (Dillmann, WS, SS, SS)



Klausur am 02. September 2003, 9:00 Uhr

- 8 - 10 Aufgaben
- Zum Bestehen der Klausur sind mindestens 40 aus 90 Punkten erforderlich
- Themen (TI-1):
 - Schaltalgebra (Minimierung, Realisierung, ...)
 - Laufzeiteffekte
 - Entwurf und Analyse von Schaltwerken (Flipflops, ...)
 - Rechnerarithmetik



Klausur am 02. September 2003, 9:00 Uhr

- Themen (TI-2)
 - Mikroprozessoren
 - Funktion, Aufbau
 - Pipelining
 - Assembler (MIPS)
 - Speicher (Organisation, Bausteine)
 - DMA & Interrupts
 - Cache-Speicher, Speicherverwaltung
 - Digitale Signalprozessoren & Mikrocontrollern

