



# Rechnerarchitektur, Foliensatz 3 Schnittstellen und Zusatzwerke

G. Kemnitz

Institut für Informatik, Technische Universität Clausthal  
30. Januar 2014



## Inhalt des Foliensatzes

Parallele Schnittstellen

Serielle Schnittstellen

2.1 SPI-Bus

2.2 JTAG (Testbus)

2.3 USART

Zähler und Zeitgeber

3.1 Normal- und CTC-Betrieb

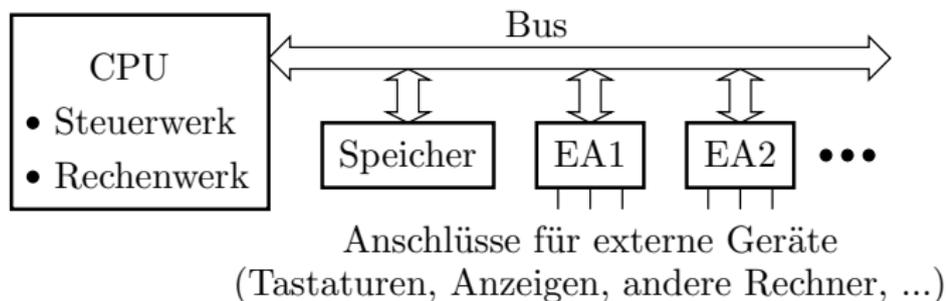
3.2 Zeitmessung

3.3 PWM

Watchdog

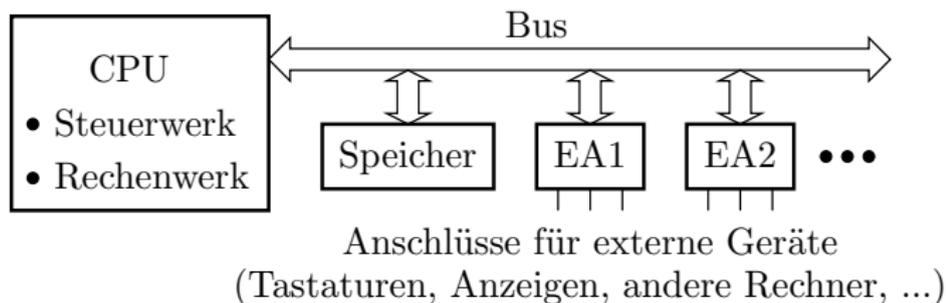
Analog-Digital-Wandler

## Ein- und Ausgabeschnittstellen



Schnittstellen zur Umgebung:

- Digitale Ein- und Ausgänge zur Abfrage von Schaltern, Steuerung von Anzeigen und Aktoren, ...
- Serielle Schnittstellen für den Datenaustausch mit anderen Schaltkreisen und Rechnern.
- Analoge Ein- und Ausgabe z.B. für Sensorwerte.
- Pulsweitenmessung und Pulsweitenmodulation, z.B. für die stufenlose Steuerung von Motoren.



Auf diese Schnittstellen greift der Rechner über EA-Register wie auf den Speicher zu. Über EA-Register hat der Prozessor auch auf weitere Funktionseinheiten Zugriff:

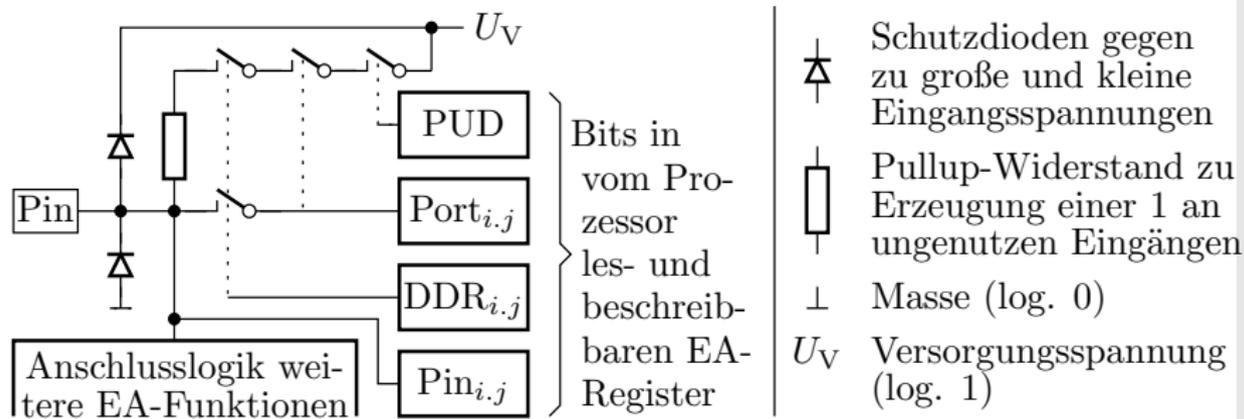
- Timer: Schaltung zur Erzeugung von Zeitereignissen.
- Watchdog: Zeitüberwachungsschaltung zur automatischen Neuinitialisierung bei Programmabsturz.
- Programmierschnittstellen: Schaltungen zum Lesen und Beschreiben des Programmspeicher (Flash), des Daten-EEPROMs und der Fuse-Register.



## Parallele Schnittstellen



## Schaltung eines einzelnen Anschlusses

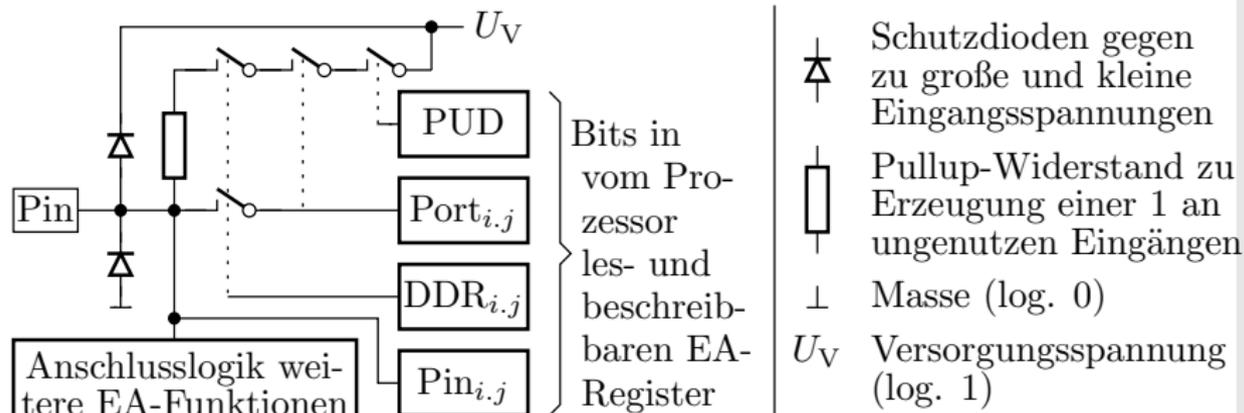


Nutzung als Ausgang:

- Richtungsbit eins setzen:  $DDR_{i,j} \leftarrow 1$ :
- Ausgabe eines Wertes:  $PORT_{i,j} \leftarrow Rd$
- Rücklesen des Ausgabewertes:  $Rd \leftarrow PORT_{i,j}$
- Lesen des Werts am Pin:  $Rd \leftarrow PIN_{i,j}$ .  $PIN_{i,j} \neq PORT_{i,j}$  ist möglich und deutet auf Programmier- oder Schaltungsfehler.



# 1. Parallele Schnittstellen



Nutzung als Eingang:

- Richtungsbit null setzen:  $DDR_{i,j} \leftarrow 0$ :
- Werte zwischen 0 und 1, z.B. bei ungenutzten Anschlüssen verursachen erhöhte Stromaufnahme.
- Ausgabewert eins ( $PORT_{i,j} \leftarrow 1$ ) und SFR-Bit »PUD« nicht gesetzt, zieht ungenutzte Eingänge über einen Widerstand auf eins. Zu empfehlen für alle ungenutzte Eingänge.
- Bei externer Signalquelle und vor allem für analoge Eingänge Pullup-Widerstand mit ( $PORT_{i,j} \leftarrow 0$ ) deaktivieren.



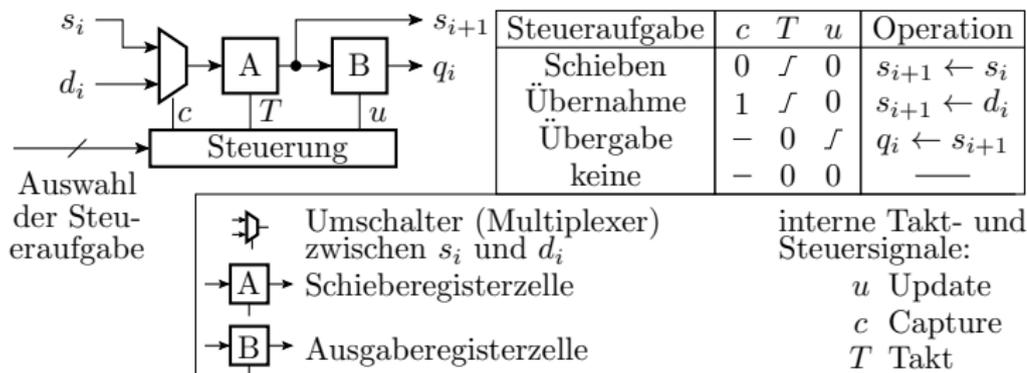
# Serielle Schnittstellen

### Serieller Datenaustausch

Rechner und Rechnerbausteine tauschen ihre Daten oft seriell<sup>1</sup> aus. Grundbaustein Schieberegister mit den Funktionen

- parallele Übernahme der zu übertragenden Daten
- serielle Übertragung und
- paralleler Übergabe.

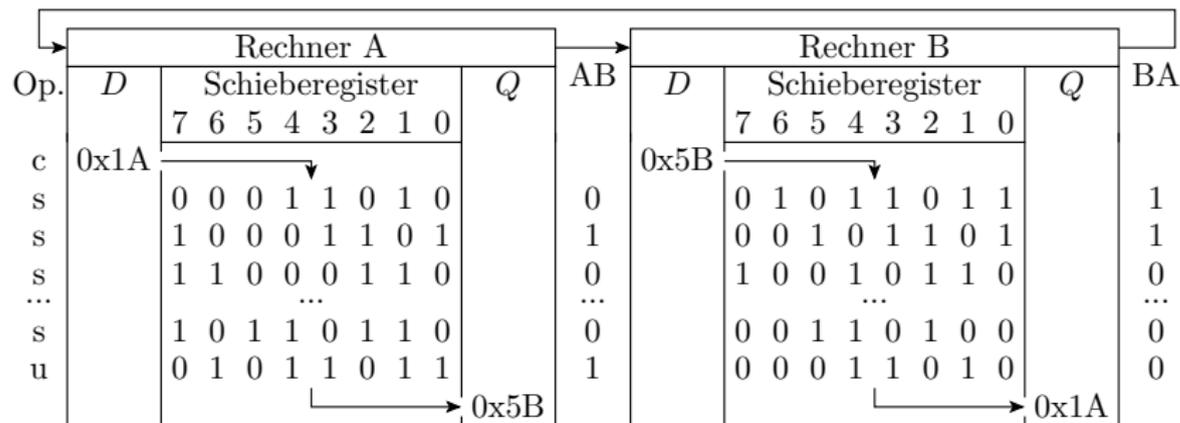
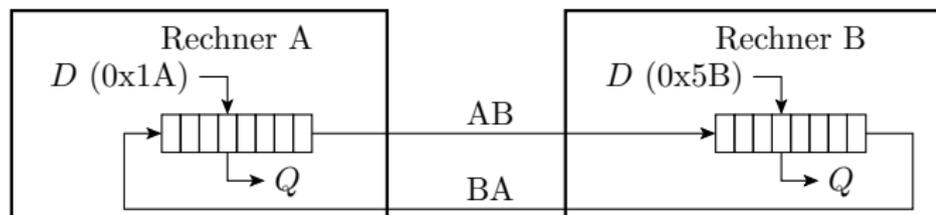
Schaltung einer solchen Schieberegisterzelle:



<sup>1</sup>Seriell, d.h. hintereinander über eine, statt parallel über viele Leitungen.



### Bidirektionale Kopplung zweier Rechner



c Übernahme (Capture)

s Schieben (Shift)

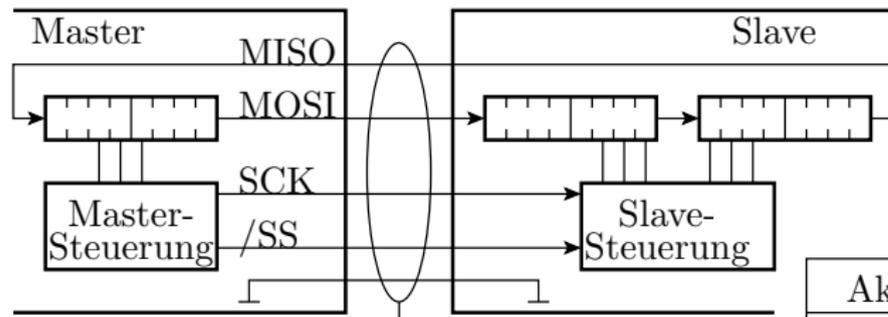
u Übergabe (Update)



# SPI-Bus

### SPI-Bus

Serieller Bus zur Vernetzung von Schaltkreisen.



Leitungen zwischen den Schaltkreisen

MOSI Master Out Slave In

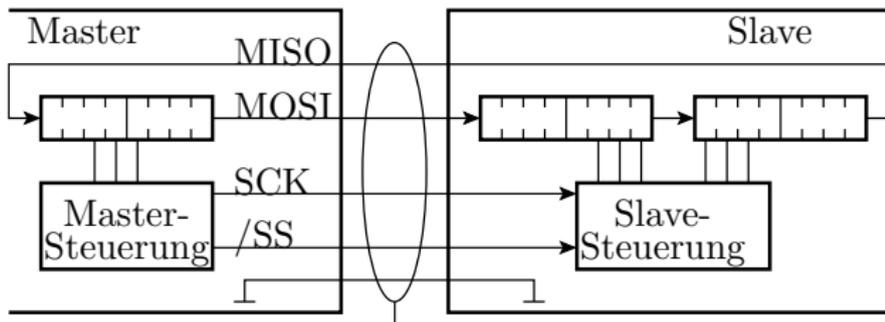
MISO Master In Slave Out

SCK Schiebetakt

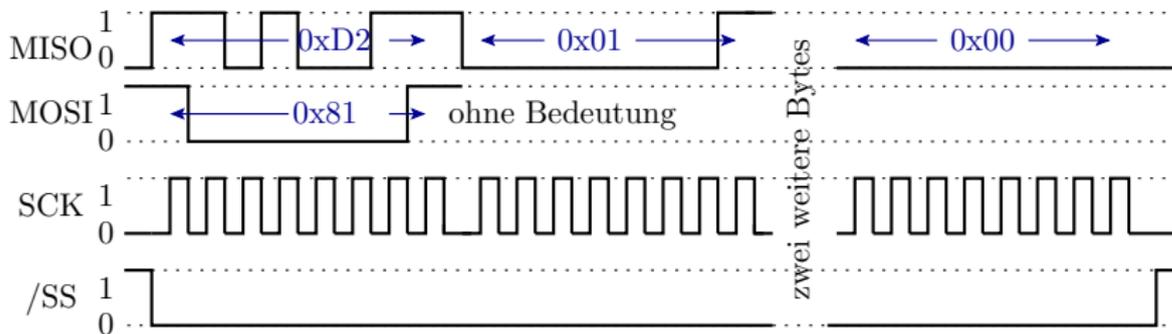
/SS Slave-Auswahl

Aktion	/SS	SCK
Übernahme		
Schieben	0	
Ausgabe		
keine	sonst	

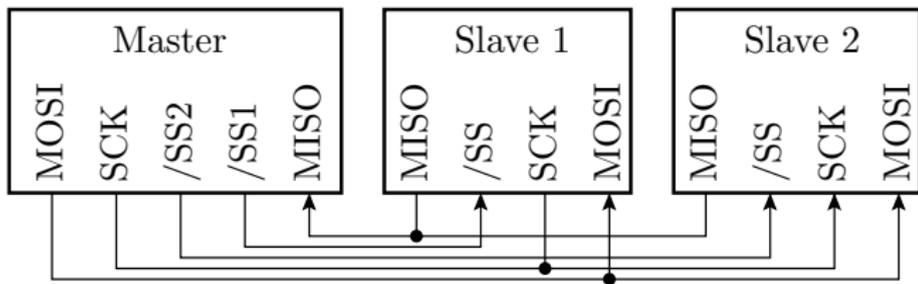
- Ein Schaltkreis ist der Master, der den Takt SCK und die Slave-Auswahlsignale erzeugt, die anderen sind Slaves, die diese Signale vom Master erhalten.



Beispiel für die Übertragung 0x81 vom Master zum Slave und von 0xD201...00 vom Slave zum Master:



- Eine Übertragung beginnt mit Aktivierung von  $/SS=0$  (Übernahme), gefolgt von  $n$  Schiebetakten und endet mit Deaktivierung  $/SS=1$ .
- Das  $/SS$ -Signal des Masters wird über einen Ausgang einer parallelen Schnittstelle gesteuert.
- An einen Master können parallel mehrere Slaves mit unterschiedlichen  $/SS$ -Signalen angesteuert werden.



### Konfiguration des SPI-Controllers

Name	Address	Value	Bits	
SPCR	0x4C	0x53	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	
SPIE	0x00	0x00	<input type="checkbox"/>	
SPE	0x01	0x01	<input checked="" type="checkbox"/> <input type="checkbox"/>	SPI aktivieren
DORD	0x00	0x00	<input type="checkbox"/>	Bit 7 zuerst senden
MSTR	0x01	0x01	<input checked="" type="checkbox"/> <input type="checkbox"/>	als Master
CPOL	0x00	0x00	<input type="checkbox"/>	Steuersignale wie Folie zuvor
CPHA	0x00	0x00	<input type="checkbox"/>	
SPR	0x03	0x03	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	Bittakt = CPU-Takt durch 128
SPSR	0x4D	0x00	<input type="checkbox"/>	
SPIF	0x00	0x00	<input type="checkbox"/>	Ereignisbit
WCOL	0x00	0x00	<input type="checkbox"/>	Fehlerbit für Senedatenüberschreiben
SPI2X	0x00	0x00	<input type="checkbox"/>	Verdopplung der Bitrate
SPDR	0x4E	0x00	<input type="checkbox"/>	SPI-Datenregister

- Einschalten als Master oder Slave.
- Festlegen der Bitrate und Protokollparameter.
- Pins für /SS Signale konfigurieren, beim Master als Ausgänge mit Wert eins, beim Slave als Eingänge.

## Algorithmus für den Datenaustausch

Name	Address	Value	Bits
SPSR	0x4D	0x00	<input type="checkbox"/>
SPIF		0x00	<input type="checkbox"/>
SPDR	0x4E	0x00	<input type="checkbox"/>

Für jede  $n$ -Byte-Übertragung

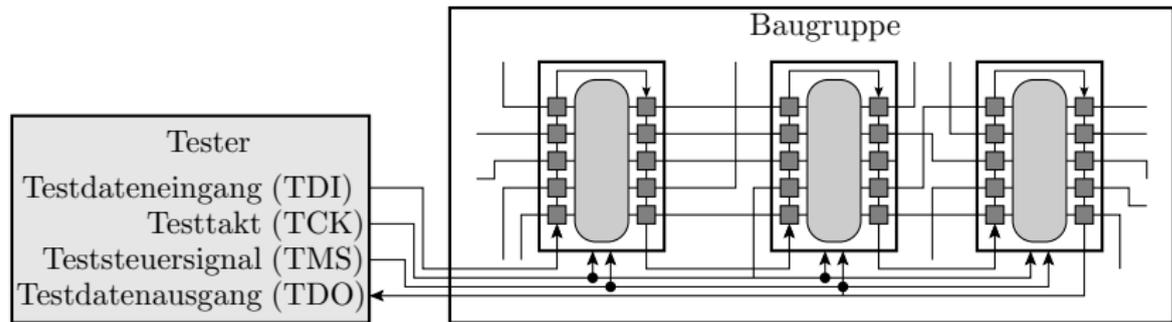
- Aktiviere das Slave-Auswahlsignal /SS (Master) bzw. warte auf /SS=0 (Slave).
- Für jedes Byte
  - Schreibe Sendewert in das Datenregister
  - warte bis Ereignisbit SPIF eins ist
  - Lese empfangenes Byte aus und schreibe nächstes zu sendende Byte in das SPI-Datenregister SPDR.
- Deaktiviere das Slave-Auswahlsignal.



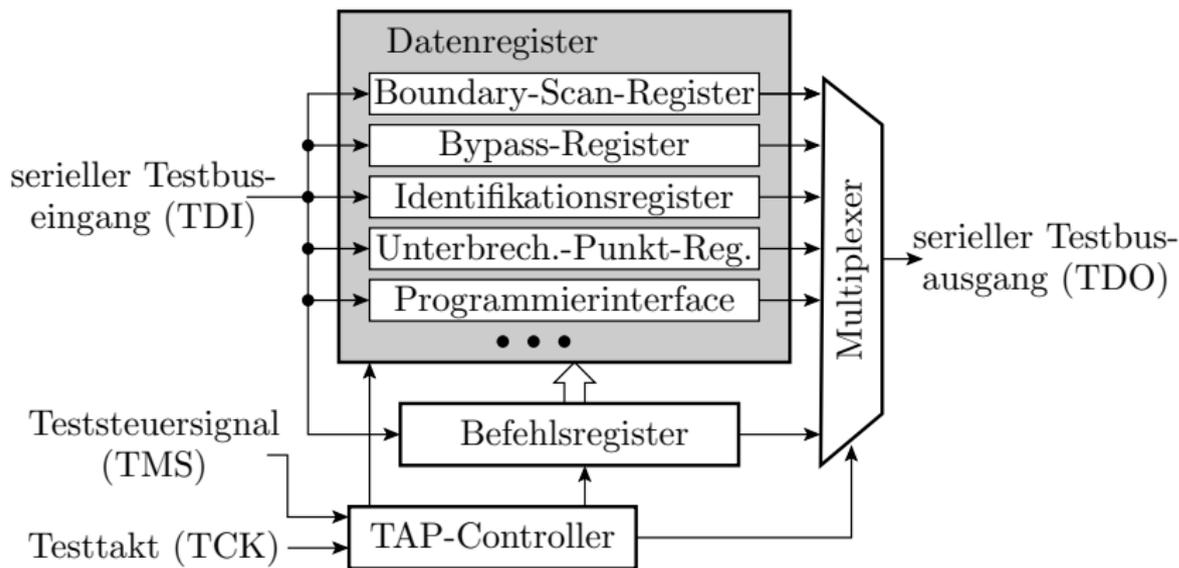
# JTAG (Testbus)

## JTAG

Test-, Diagnose-, Debug- und Programmierbus.

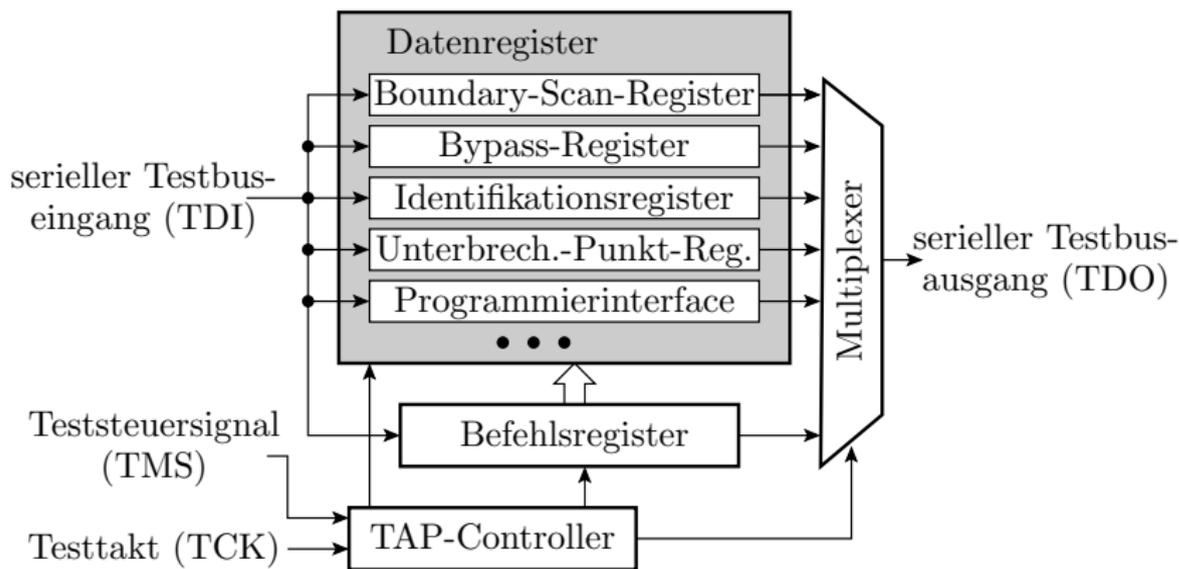


- Verschaltung aller Schaltkreise zu einer Kette.
- In der Ursprungsidee (Boundary-Scan) waren die Schieberegister mit den Funktionen Übernahme, Schieben und Übergabe am Schaltkreisrand angeordnet, um die Verbindungen zwischen den Schaltkreisen ohne mechanische Kontaktierung testen zu können.



Ein Schaltkreis mit JTAG-Bus hat mehr als ein Befehlswort auswählbare Datenregister:

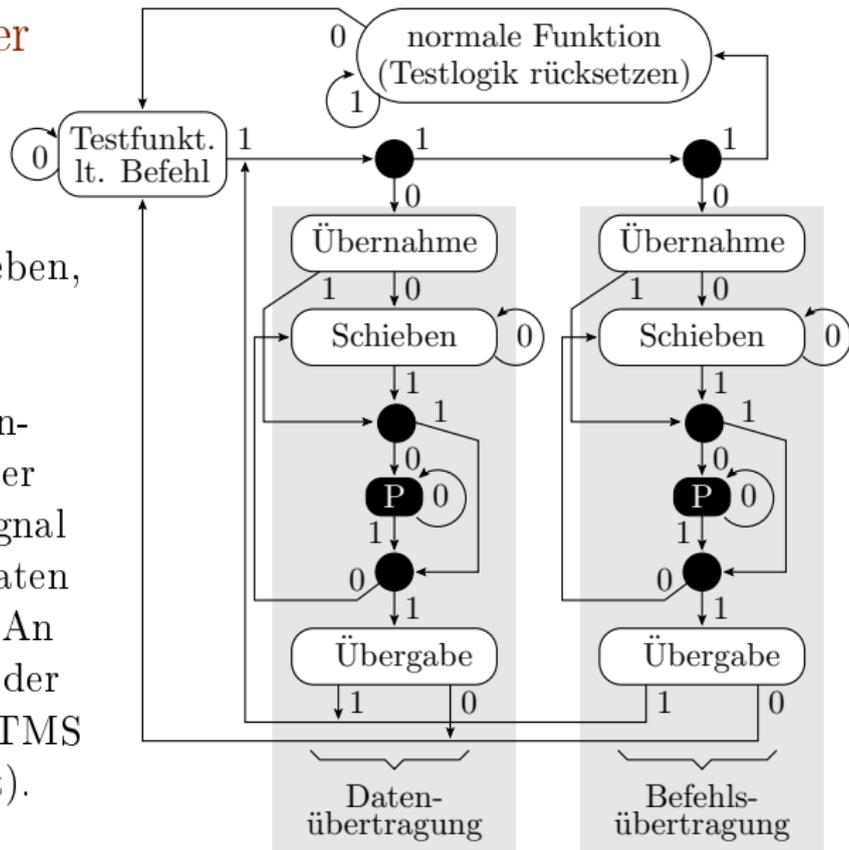
- Bypass-Register zur Verkürzung der Länge des Schieberegisters durch den Schaltkreis auf 1 Bit.
- Identifikationsregister mit Hersteller- und Bauteilnummer.



- **Programmierinterface:** Schnittstelle zum Lesen und schreiben des Befehls-Flashes, des Daten-EEPROMs und der Fuse-Register.
- **Schnittstellenregister** zum OCD (On-Chip Debugger), ...

## TAP-Controller

Die Auswahl der 6 Busaktionen: Übernahme, Schieben, Übergabe für des Befehls- und das ausgewählte Datenregister erfolgt über ein 1-Bit-Steuersignal und einen Automaten mit 8 Zuständen. An den Kanten steht der Wert des Signals TMS (Test Mode Select).





Von der JTAG-Implementierung in unserem Prozessor sind nur die standardisierten Testfunktionen, die für den Bestückungstest von Baugruppen vorgesehen sind, öffentlich zugänglich dokumentiert. Die Befehle für die Programmierung und den OCD (On-Chip Debugger) sind in den Dokumentationen nicht beschrieben.



# USART





Baudrate  $b$  als Funktion der Prozessortaktfrequenz  $f_{\text{Proz}}$  und dem programmierbaren Teilerfaktor  $T$ :

$$b = \frac{f_{\text{Proz}}}{16 \cdot (T + 1)}$$

Teilerfaktor für  $f_{\text{Proz}} = 8 \text{ MHz}$  und  $b = 9600 \text{ Baud}$ :

$$T = \frac{8 \text{ MHz}}{16 \cdot 9600 \text{ Hz}} - 1 = 51 = 0x33$$

Initialisierung: 9600 Baud, 8 Daten-, 1 Stoppbit, keine Parität:

	UCSR0C	0xC2	0x06	<input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input type="checkbox"/>	
	UMSEL0		0x00	<input type="checkbox"/> <input type="checkbox"/>	asynchron
	UPM0		0x00	<input type="checkbox"/> <input type="checkbox"/>	kein Paritätsbit
	USBS0		0x00	<input type="checkbox"/>	1 Stoppbit
	UCSZ0		0x03	<input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	8 Datenbit
	UCPOL0		0x00	<input type="checkbox"/>	
	UBRR0	0xC4	0x0033	<input type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/> <input checked="" type="checkbox"/>	Baudrate 9600

**UBRR0** = 0x33;

Teilerwert für 9600 baud

**UCSR0B** = 0b00011000;

Sender und Empfänger einschalten

**UCSR0C** = 0b00000110;

Datenformat 8n1

## Sende- und Empfang



Funktionen für den Empfang und das Versenden eines Bytes

```

uint8_t getChar(){
    while (!(UCSR1A & (1<<RXC0))){}; // Warte auf Empfang
    return UDR0;                      // Rückgabe Empfangsbyte
}

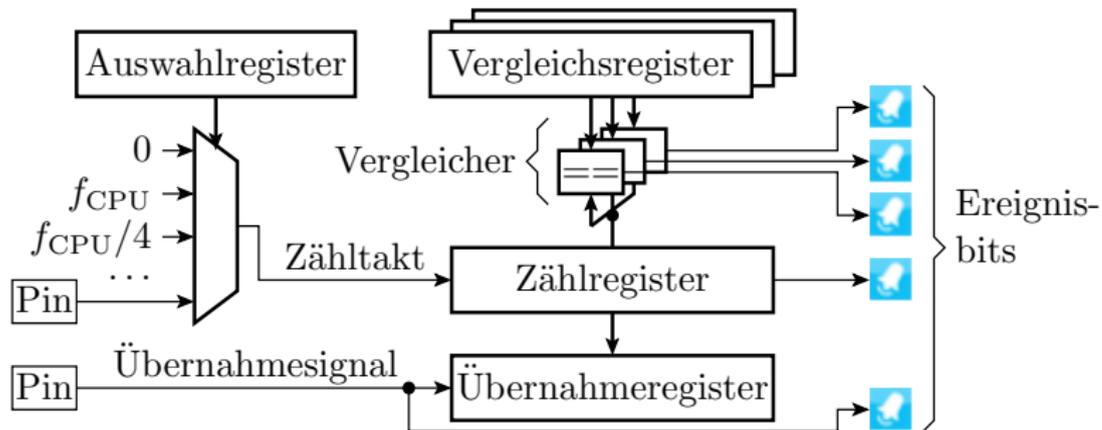
void putChar(uint8_t c){
    while (!(UCSR1A & (1<<UDRE0))){}; // Warte bis Sendepuffer frei
    UDR0 = c;                          // frei. Byte versenden
}
    
```



# Zähler und Zeitgeber

## Zähler-/Zeitgeber

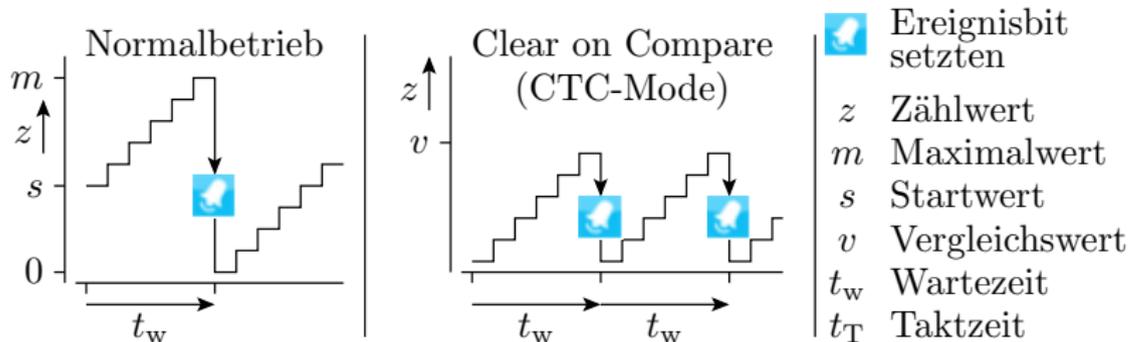
Multifunktionsbaustein mit einem Zähler, auswählbare Quelle der zu zählenden Pulse, Vergleichsregistern, Registern für die Übernahme von Zählwerten ...





## Normal- und CTC-Betrieb

## Betriebsarten als Zeitgeber



- Normalbetrieb: Initialisierung mit einem Startwert. Zähltakt einschalten. Warten, bis Ereignisbit gesetzt ist. Wartezeit:

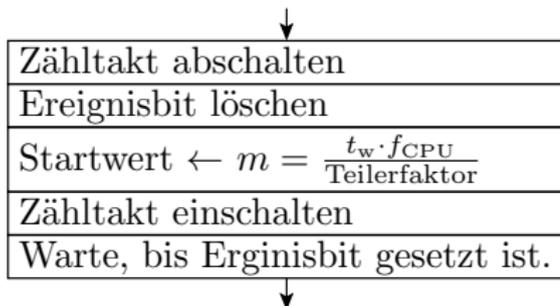
$$t_w = (m - s) \cdot t_T$$

- Erzeugung periodischer Zeitsignale: Vergleichswert einstellen und Zähler rücksetzen. Zähltakt einschalten. Wiederhole, wenn Ereignisbit gesetzt wird. Ereignisperiode:

$$t_w = v \cdot t_T$$



## Wartefunktion mit einem Timer

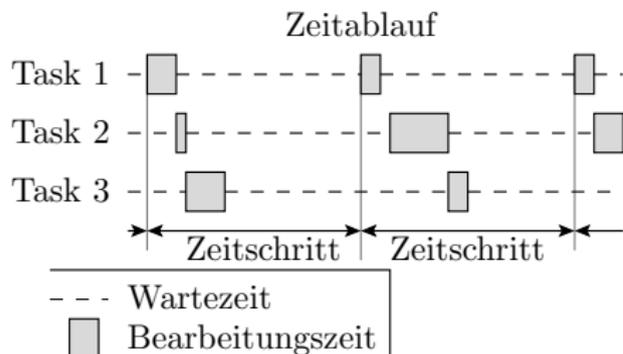


Vorteile gegenüber einer einfachen Programmwarteschleife:

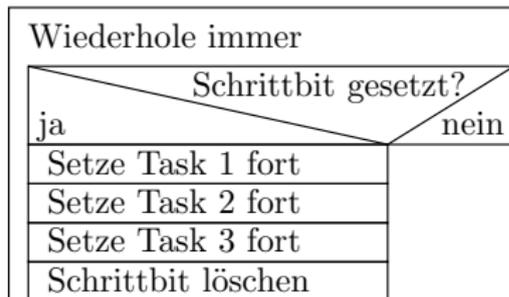
- Insbesondere in C einfacher zu programmieren.
- Wird vom Compiler nicht weg optimiert.
- Wartezeit ändert sich nicht bei Übersetzung mit anderem Compiler oder anderer Compileroptimierungen, ...



## Periodische Dienste



## Programmablauf

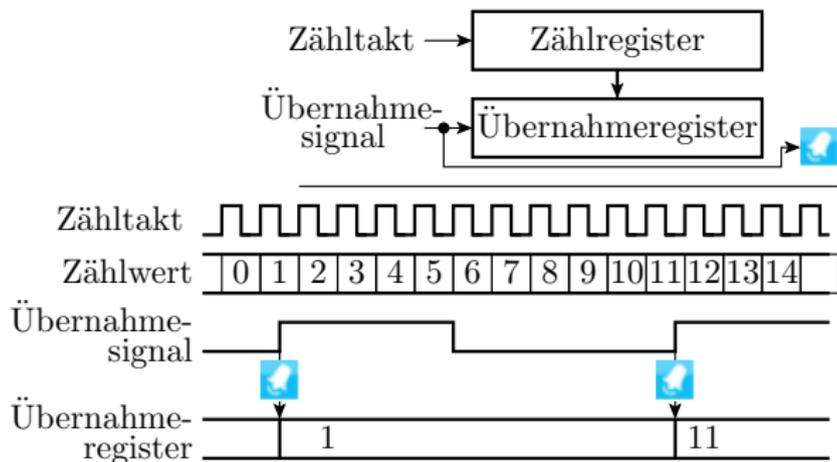


Ein Mikrorechnerprogramm besteht in der Regel aus mehreren Tasks, wie Sensorwerte und andere Eingaben lesen, Berechnungen ausführen und Aktoren ansteuern. Programmierung mit einem Timer im CTC-Modus, der zyklisch mit einer Periodendauer  $t_w$  ein Ereignisbit setzt. Das Programm wartet auf dieses Ereignisbit, führt die Schrittfunktionen aus, z.B. das Weiterschalten einer Uhr, und setzt das Ereignisbit zurück.



# Zeitmessung

### Zeitmessung



#### Initialisierung

- Auswahl des Zähltakts
- Auswahl der Übernahmebedingung (steigende, fallende, beide Flanken des Übernahmesignals)

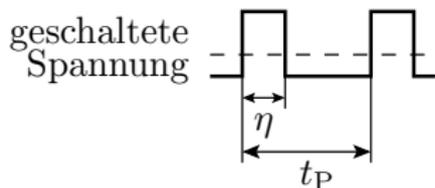
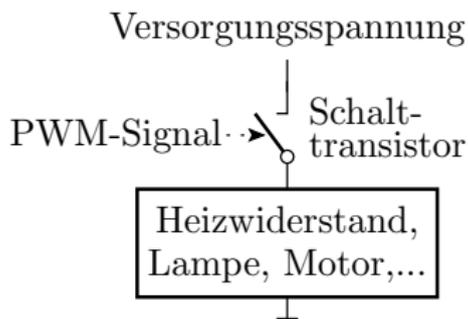
#### Messung von Zeitdifferenzen:

- Warte auf Ereignisbit »Übernahme«
- Lese Übernahmeregister und Subtrahiere den Übernahmewert vom Ereignis zuvor.



# PWM

## PWM-Signale

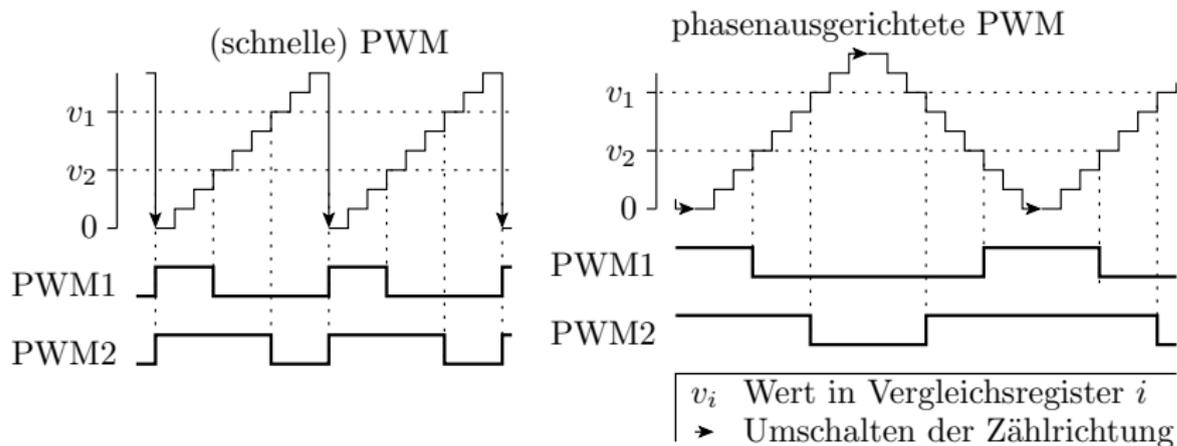


$\eta$	relative Pulsbreite
$t_P$	Periodendauer
- - -	mittlere Spannung, Strom oder Leistung

PWM- (PulsWeitenModulierte) Signale dienen zur

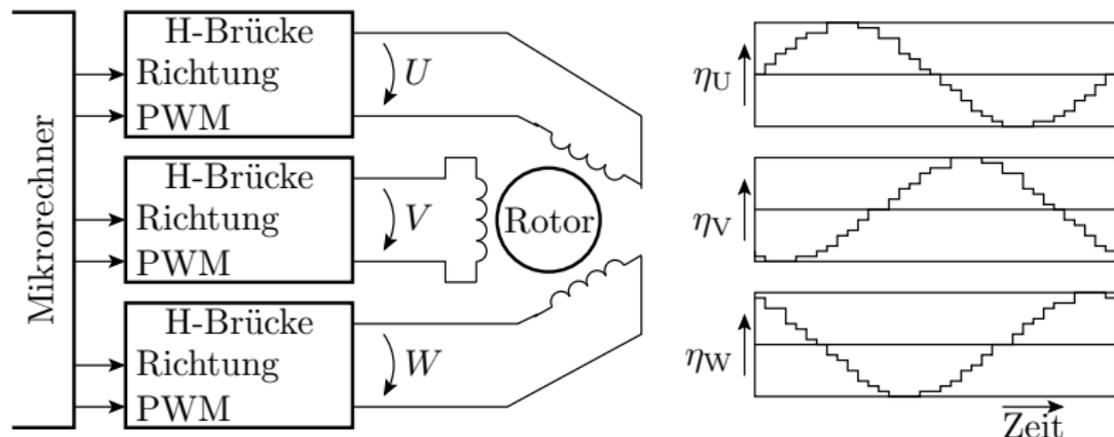
- feingestufteten Steuerung von Aktoren (Heizleistung, Helligkeit, Motordrehzahl) und
- zur Informationsübertragung (Sender sendet PWM-Signale und der Empfänger misst die Pulsbreite).

### Erzeugen von PWM-Signalen



- Ein- und Ausschalten der PWM-Ausgänge ohne Programmunterstützung.
- Ausgabe der PWM-Signale auch negiert möglich.
- Die Periode wird entweder durch den Zählerüberlauf oder einen weiteren Vergleichswert in einem Register bestimmt.

## Typische Motoransteuerung



- Die Erzeugung von 3 sinusförmigen Mittelwertverläufen erfordert eine PWM-Einheit mit drei Vergleichsregistern.
- Ansteuerung über H-Brücken.
- Erlaubt eine stufenlose Positions-, Geschwindigkeits- und Drehmomentsteuerung sogar für Asynchronmaschinen.



# Watchdog



### Watchdog

Jedes größere Programm enthält statisch gesehen Fehler, die unter anderem auch dazu führen können, dass das Programm abstürzt. Bei Absturz erreicht das Programm einen Zustand, den es nicht wider von selbst verlassen kann, z.B. eine endlose Warteschleife.

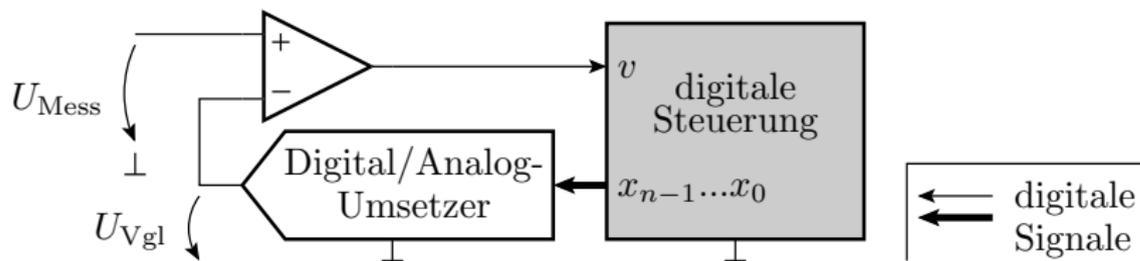
Ein Watchdog ist ein Zeitzähler, der vom Programm in einem bestimmten Zeitabstand rückgesetzt werden muss. Bei Überlauf initialisiert er den Rechner neu. Begrenzt die Dauer der Nichtverfügbarkeit von Rechnern durch Ausfälle auf wenige ms bis s.

Zur Verhinderung, dass Programmierfehler den Watchdog deaktivieren, gibt es nur einen Befehl zum Einschalten des Watchdogs nach Programmstart und zum Rücksetzen des Zählers, aber keinen zum Ausschalten.



# Analog-Digital-Wandler

## Mehrschrittwandler



$$v = \begin{cases} 0 & \text{wenn } U_{\text{Mess}} < U_{\text{Vgl}} \\ 1 & \text{sonst} \end{cases}$$



## Sukzessive Approximation

- schnellster serieller Wandleralgorithmus
- ein Vergleich je Bit

