
MICRO-CONTROLLER



Was ist ein Micro-Controller ?

Controller = Steuerung

Ein Controller wird zur Steuerung eines physischen Prozesses eingesetzt.

Die Realisierung eines Controllers kann auf viele verschiedene Arten erfolgen, z.B. ein Schaltschrank mit Relais, ein analoger Regelkreis oder eine speziell aufgebaute digitale Logikschaltung.

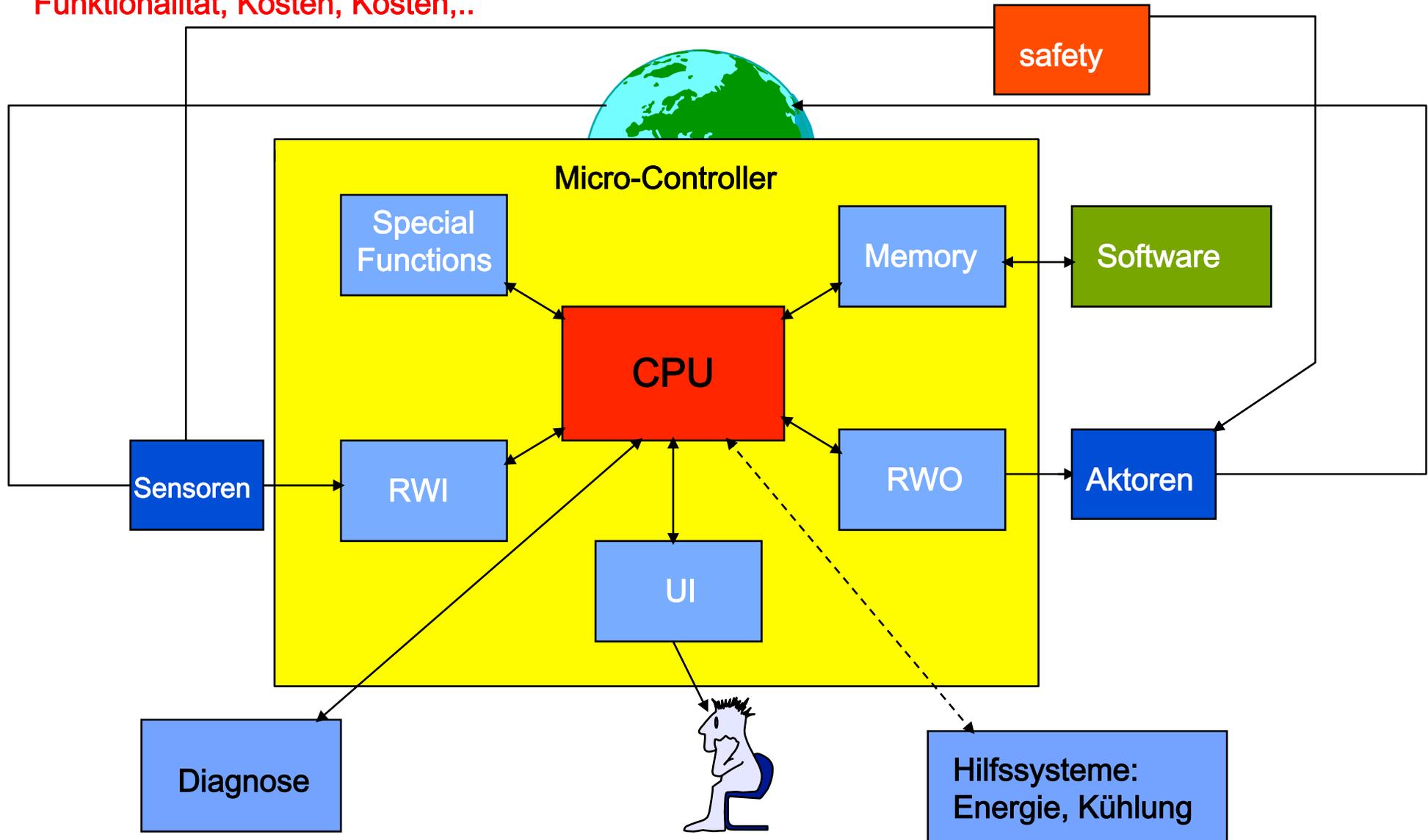
Speicherprogrammierbare Steuerung (SPS)
Programmable Logic Controller (PLC)

Ein Micro-Controller ist eine Steuerungskomponente, deren Funktionen von einem Mikroprozessor kontrolliert werden.



Systemsicht eines Architekten für Kontrollsysteme:

Leistungs-Eigenschaften gemessen in: **Kosten, Time-to-Market, Kosten, Funktionalität, Kosten, Kosten,..**



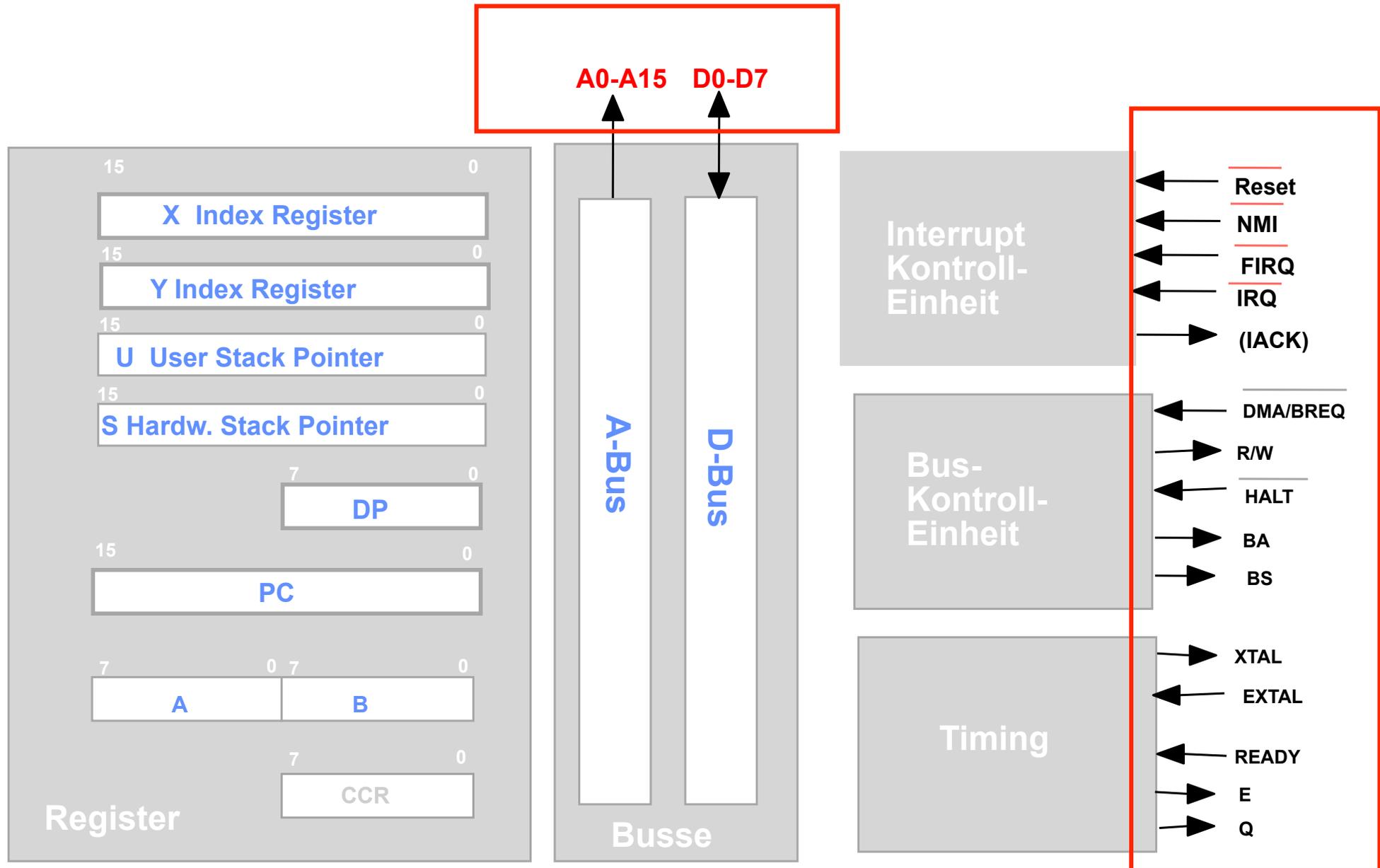
Der CPU-Kern

Architekturen:

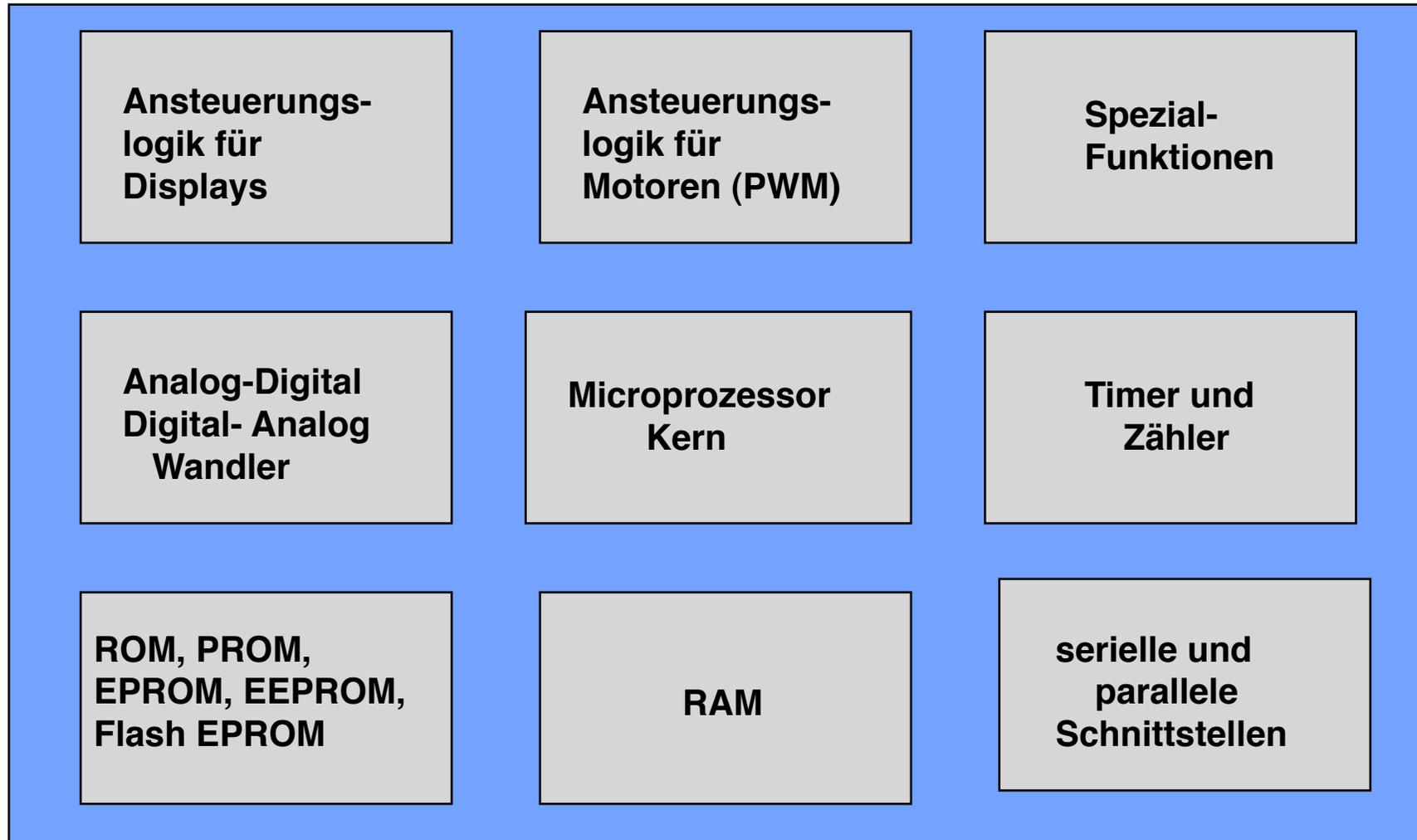
- v. Neumann:** **Sequentielle Abarbeitung, Beispiel 680x, 68kxxx**
- Harvard:** **Parallelität Befehls- und Operandenphase Beispiel: AVR, ARM, StrongARM
besitzen eingeschränkte DSP-Befehle (Multiply/Accumulate)**
- Signalproz.:
(DSP)** **Harvard Architektur, spezielle Multiply/Accumulate Instruktionen, Zirkulare
HW-Puffer, spezielle schnelle ADC-/DAC-Einheiten**
- VLIW:** **MIMD-Architektur, mehrere Verarbeitungseinheiten, gleichzeitige Verarbeitung
mehrerer Instruktionen, Beispiel aus dem DSP Bereich: TMS320C6000, ...**



Schnittstelle eines Microprozessors: Speicherbus



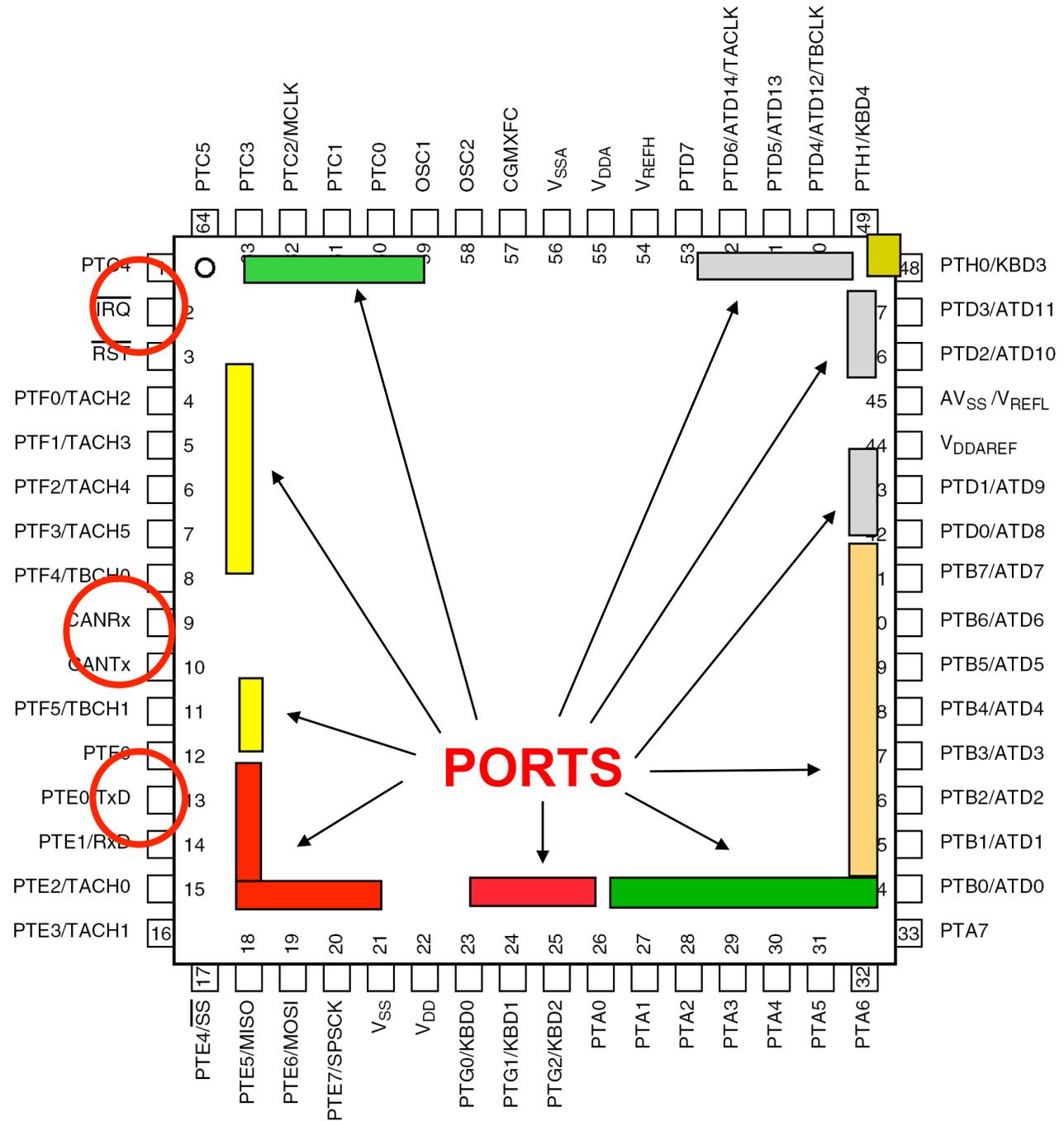
Micro-Controller



Schnittstelle zu
einem
Micro-Controller:

PORTS

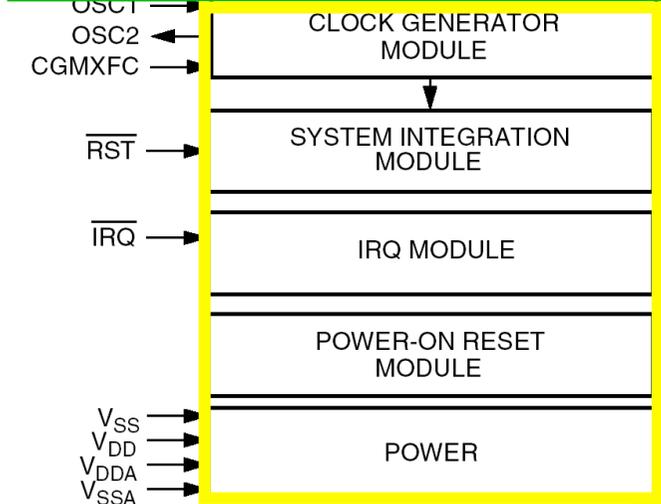
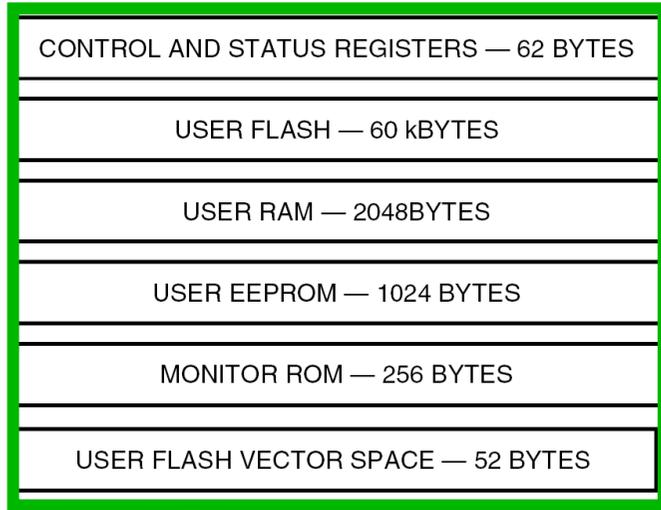
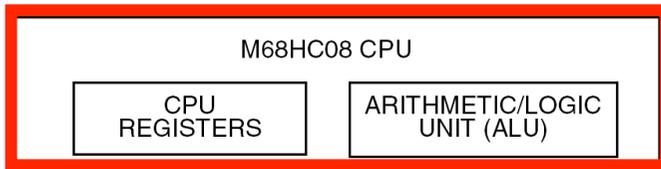
Beispiel:
Motorola
MC 68HC908AZ60A



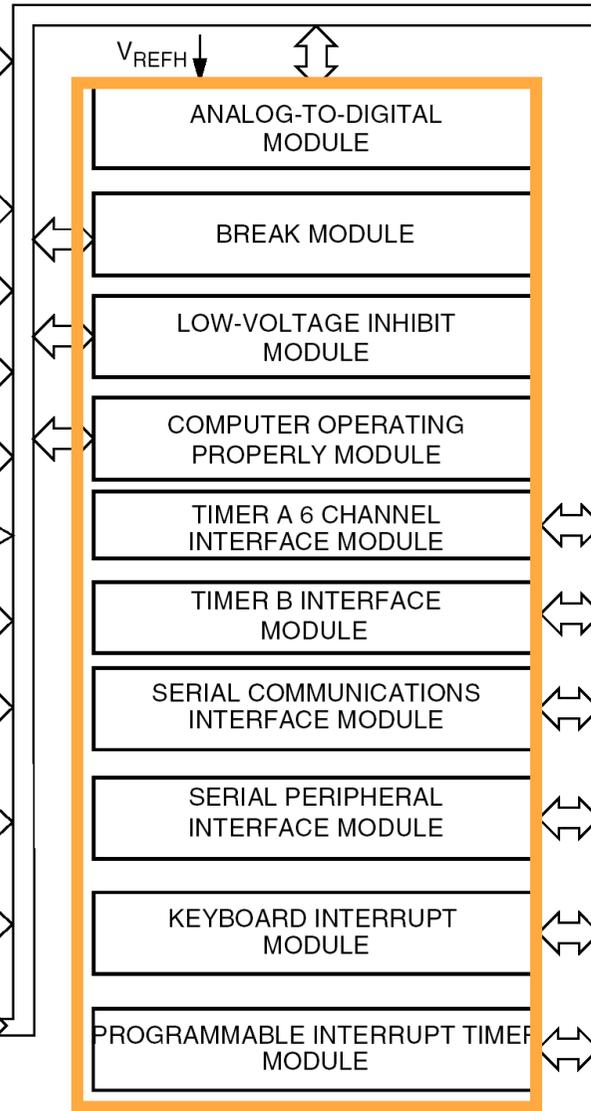
CPU

Memory

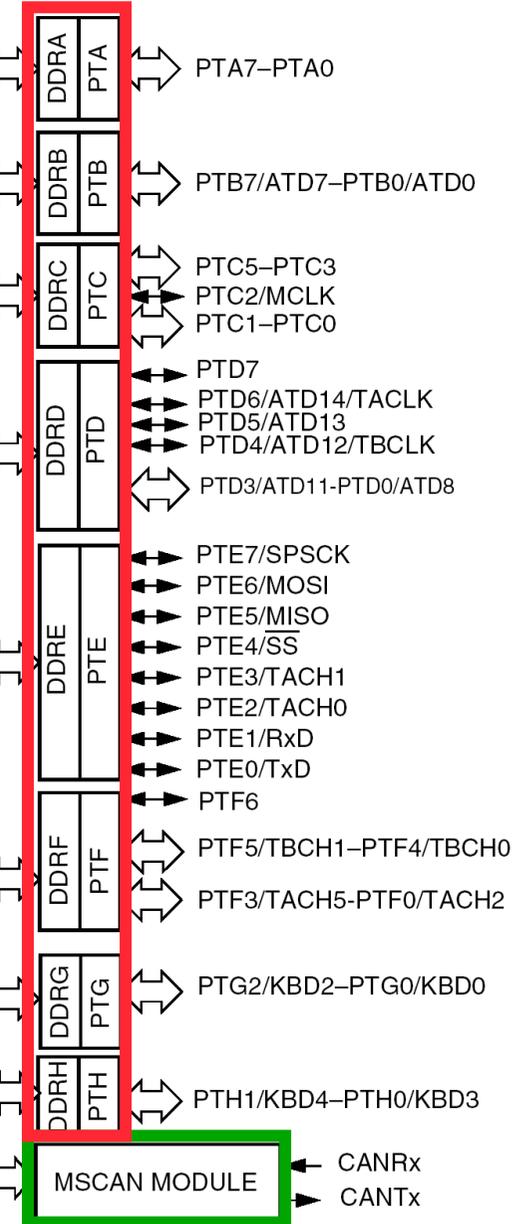
Ports



System-Module



Anwender-Module



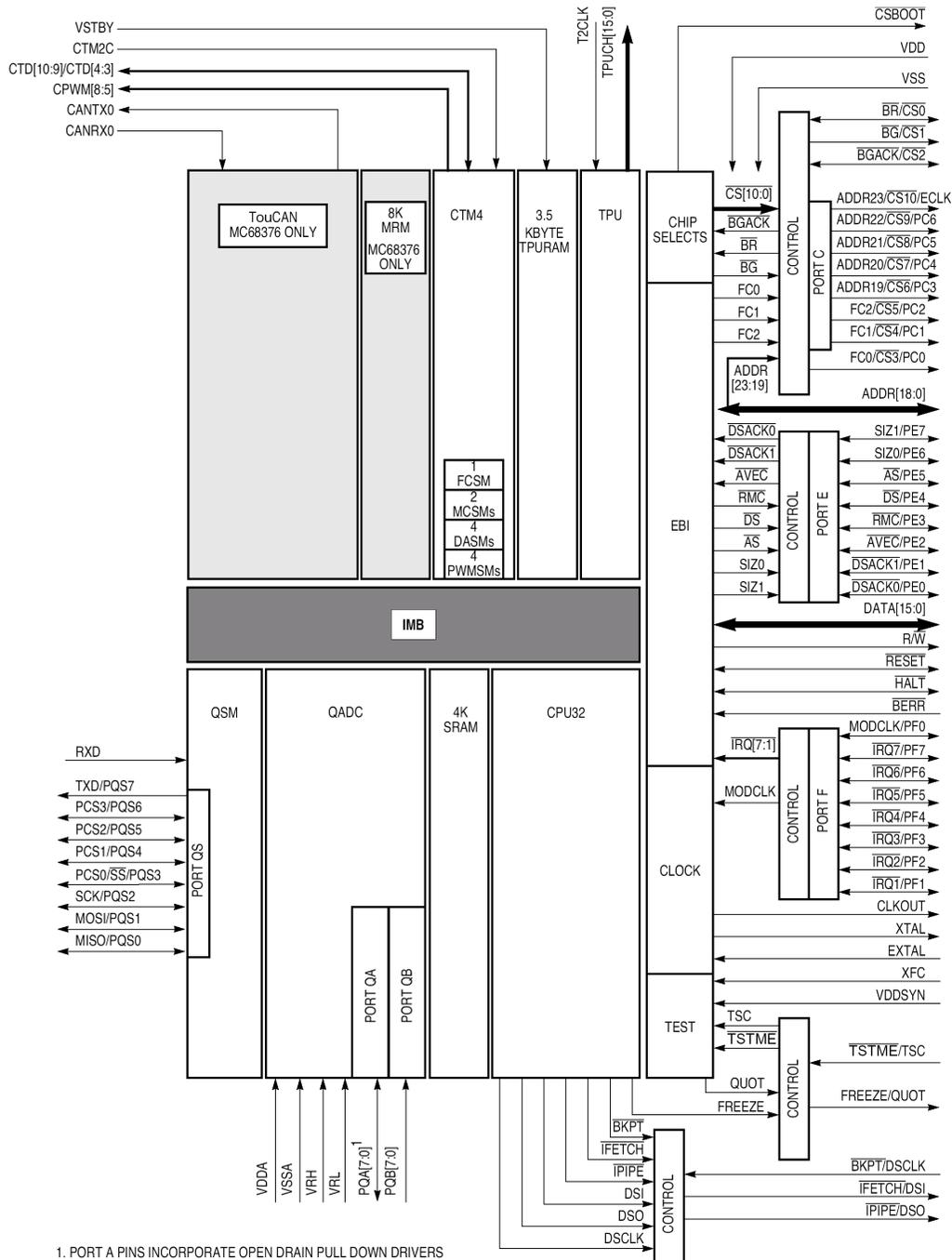
Netzwerk

Micro-Controller-Familien :

Zielanwendung	Spez. Komponenten	Nutzung
General Purpose	Timer, A/D, EEPROM, Input Capture/Output Compare serielle und synchr.Schnittstellen	logic replacement,
Automotive	EEPROM, CAN A/D, On-Chip Spannungs- stabilisator	Electric Seat Control Klima, Radio, Alarm, IR-Schlüssel Zündung, Air Bag, etc.
Computer	Monitorsteuerung (hor./vert. Sync), PWM	Tastatur-, Maus-, Monitorkontrolle
Consumer	Multitask Support, LCD-Treiber	Waschmaschinen, CD-Spieler, Handy Fernsteuerung
Industrial	EEPROM, A/D, Timer, PWM, CAN	SPS, Motor-Kontrolle, Thermostat
Telecommunications	EEPROM, DTMF-receiver +generator A/D, D/A, Tongenerator	Digitale Übertragung, Handy-Kontrolle
TV + Video	EEPROM, On-Screen-Display-Supp. LCD- und andere Anzeigetreiber	Videorecorder-Kontrolle, Bildschirm-Menues

PIN-Count, Preis, Störuneempfindlichkeit, Anpaßbarkeit





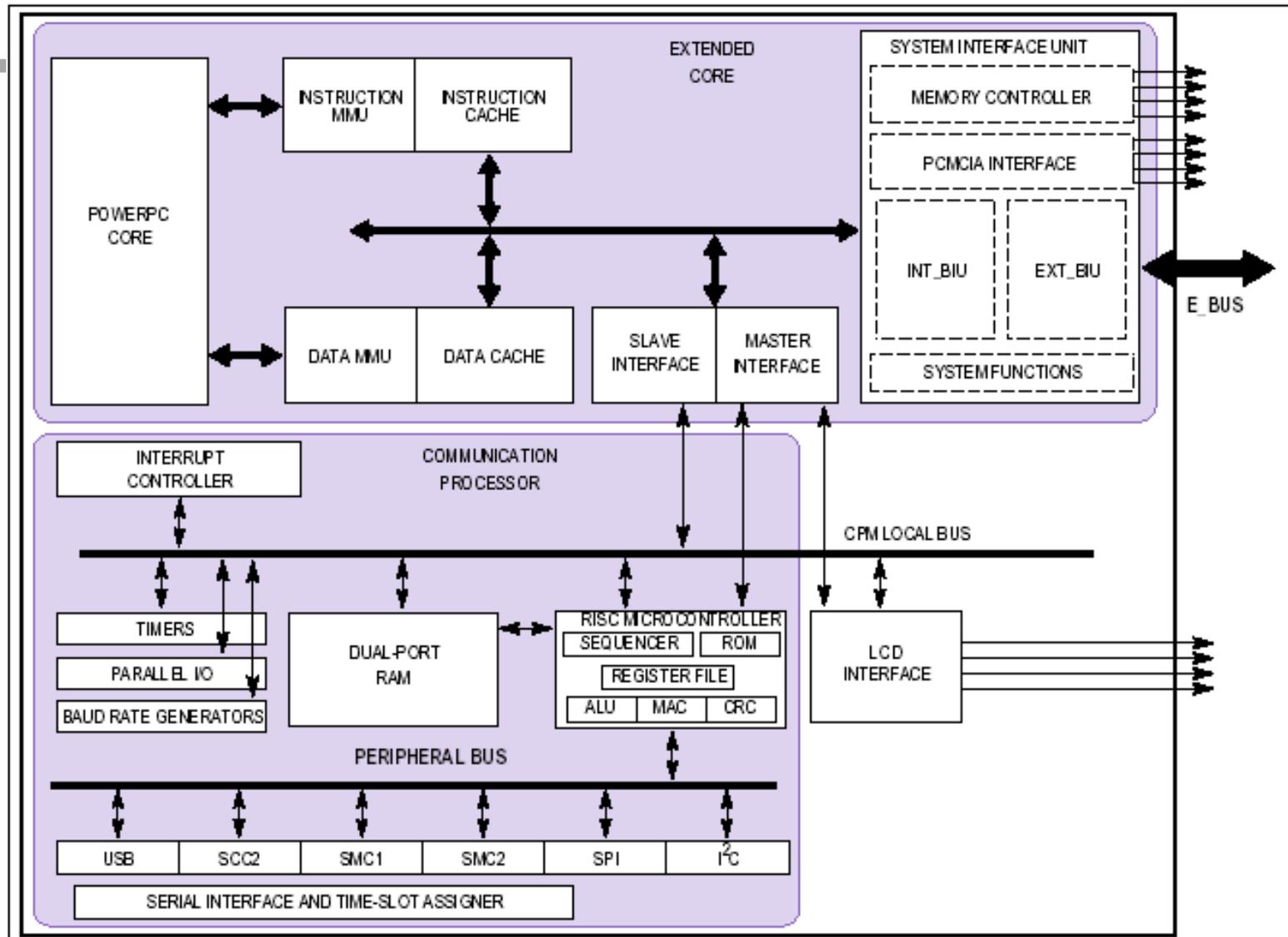
1. PORT A PINS INCORPORATE OPEN DRAIN PULL DOWN DRIVERS

336/376 BLOCK

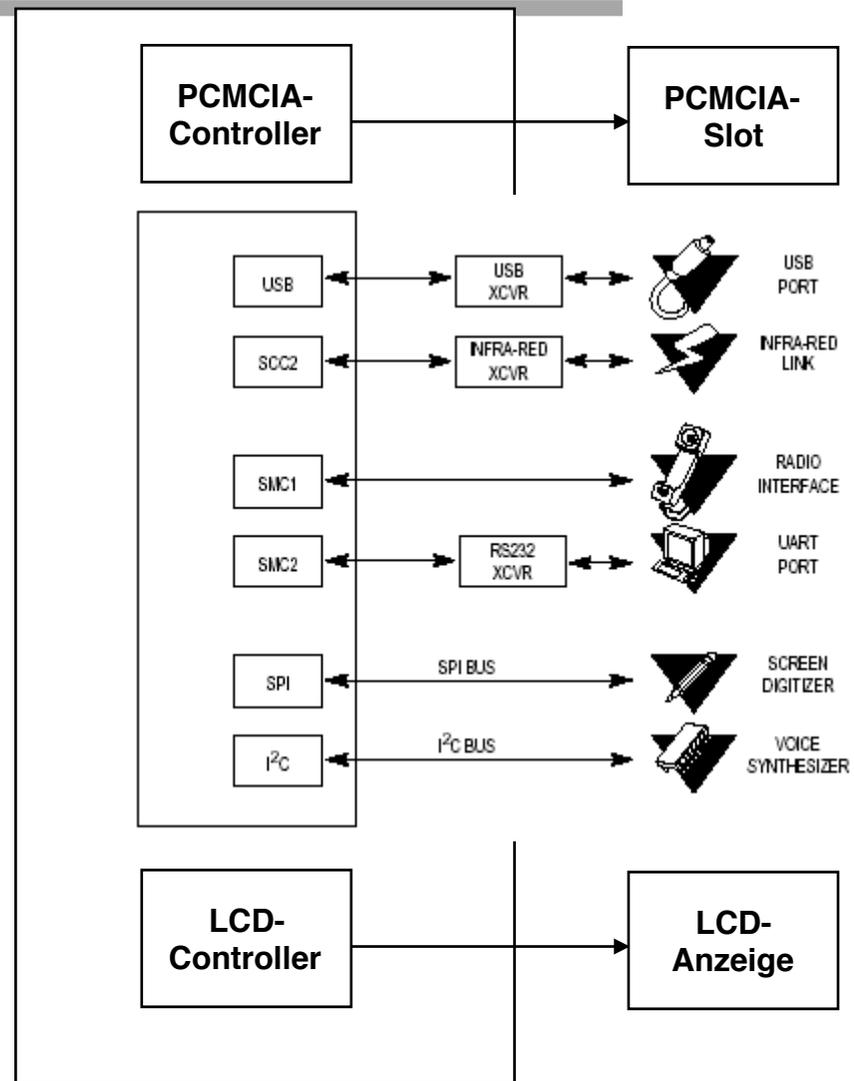
Blockschaltbild: MC 68376

IMB: Inter Module Bus
CTM: Config. Timer Module
QSM: Queued Serial Module
TPU: Time Processing Module
QADC: Queued ADC
EBI: Extended Bus Interface
TouCAN: CAN-Bus 2.0
MRM: Masked ROM Module

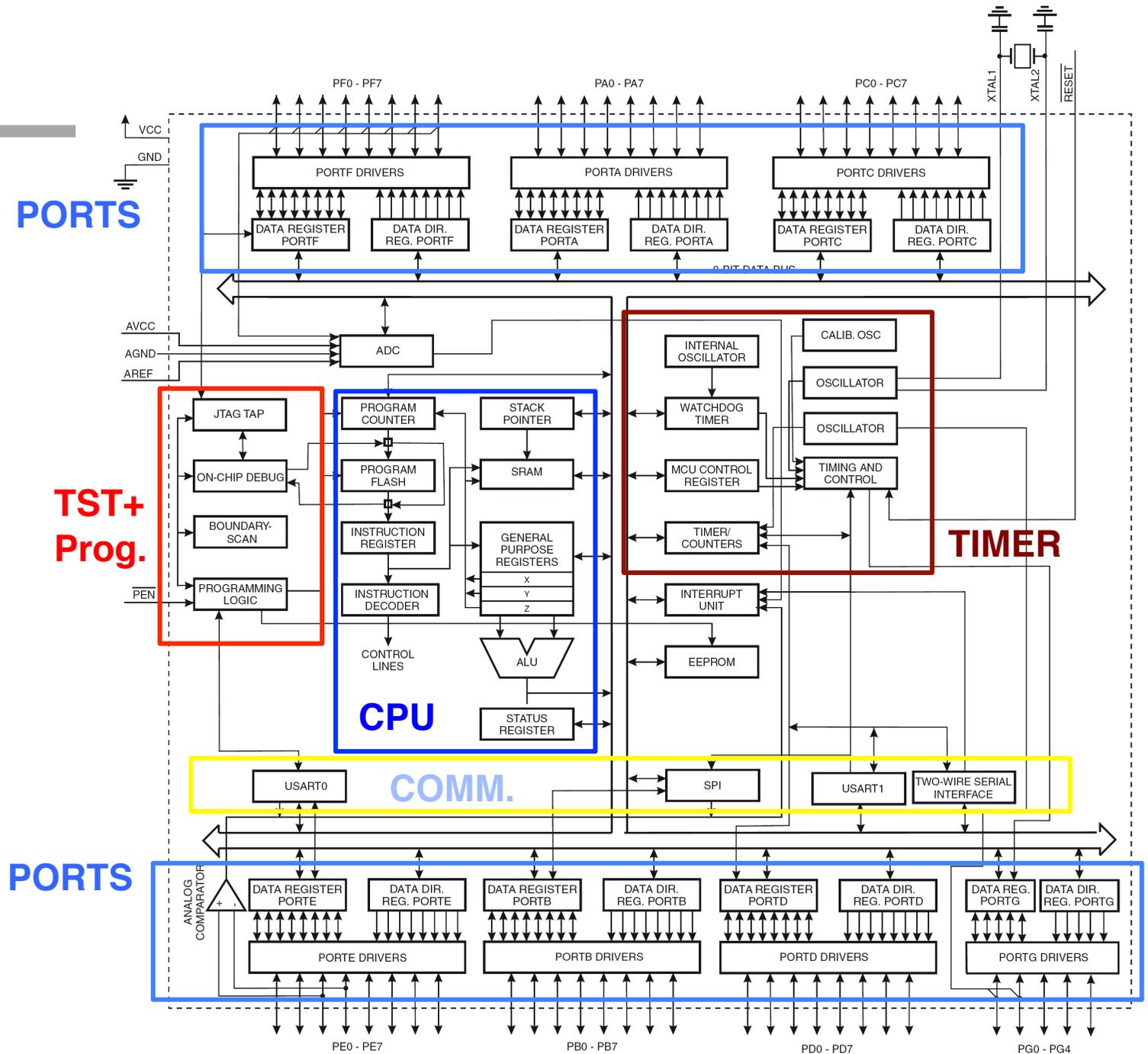
Power PC 823 embedded controller



Beispiel für einen PDA



ATmega 128 Block Schaltbild



Data Sheet:
Rev. 2467M-AVR-11/04

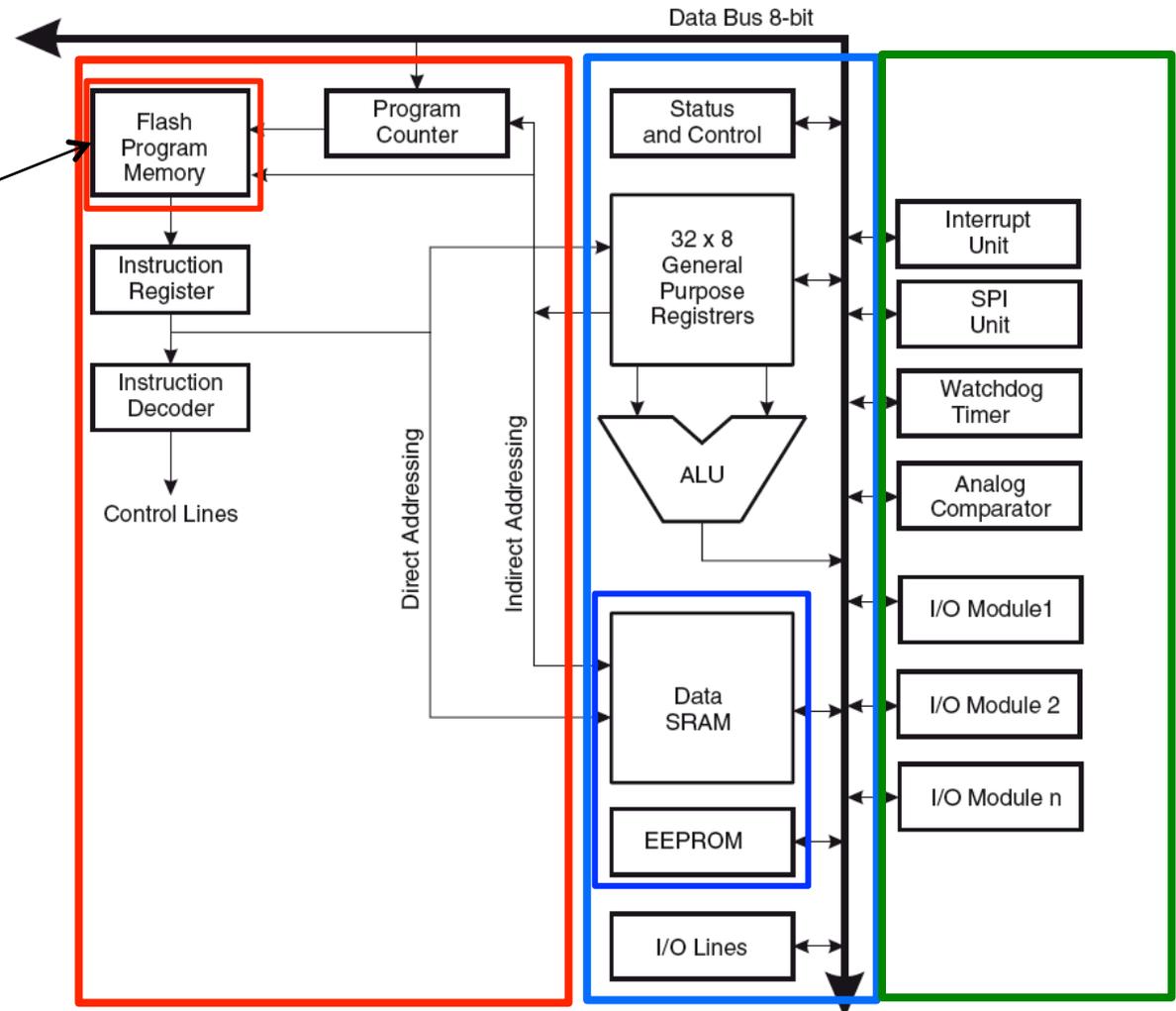


PKES
Wintersemester 10/11

Blockschaltbild der AVR CPU

Programm-Speicher

**Harvard
Architektur**



Kontrolle

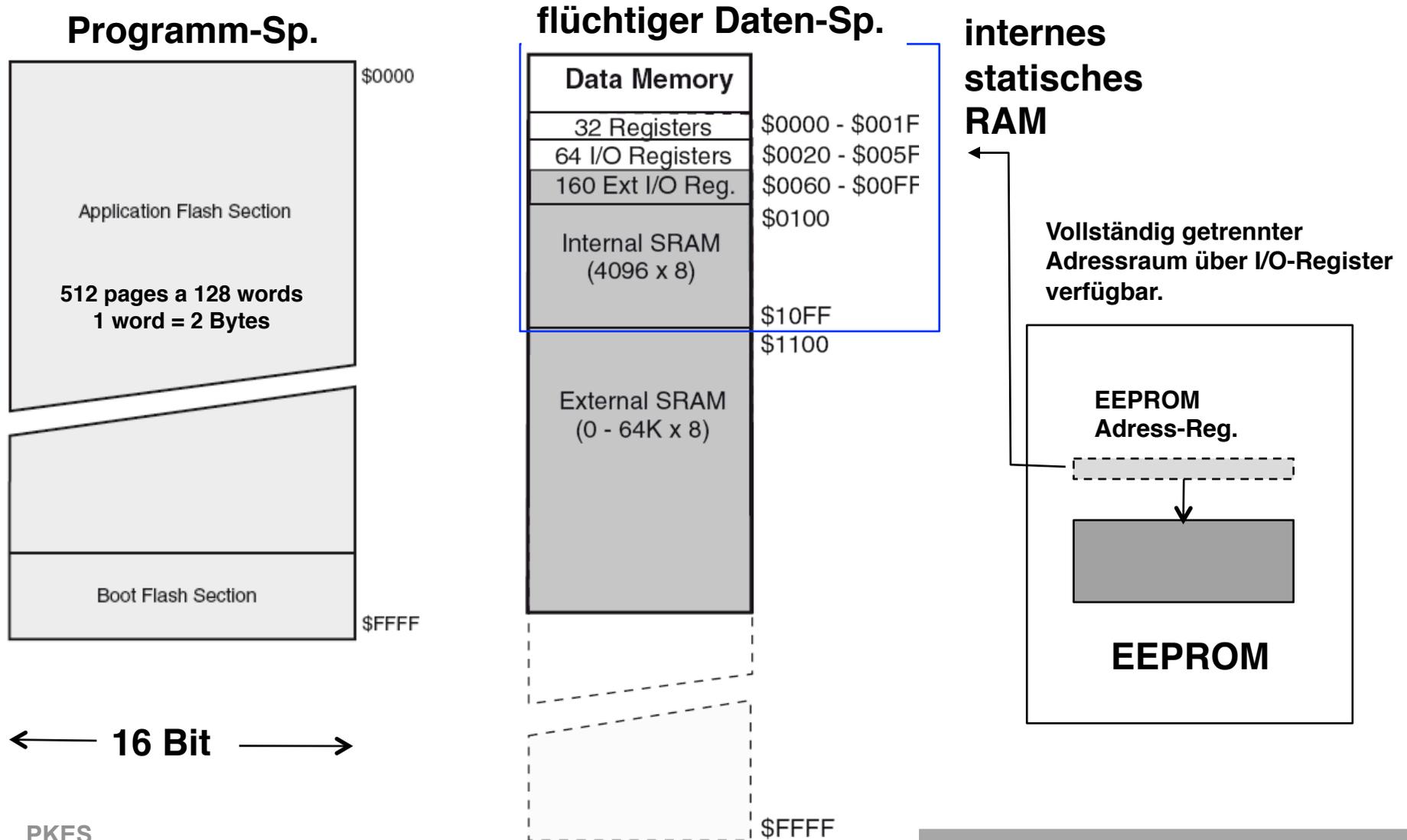
Daten

on-chip "Geräte"

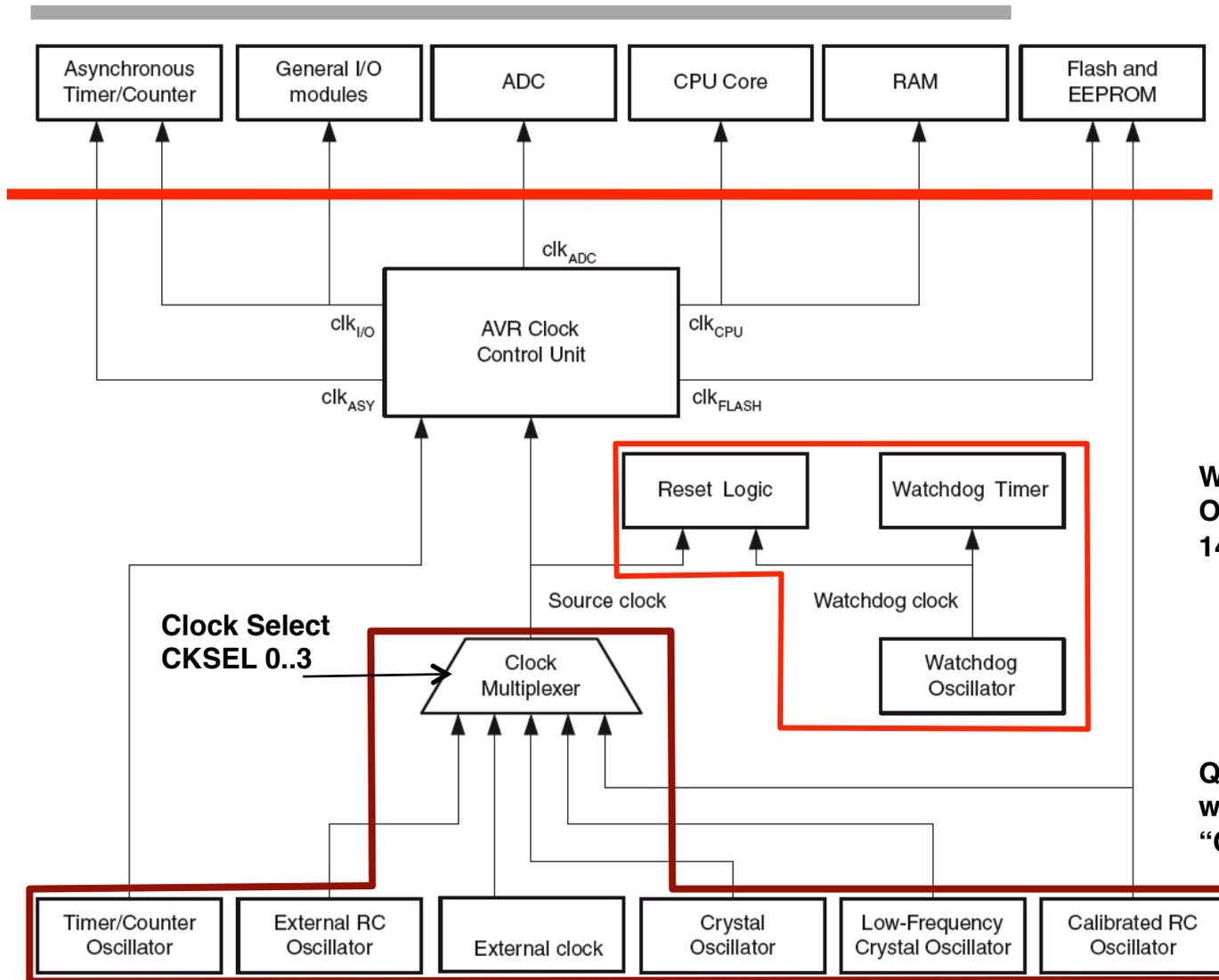
Data Sheet:
Rev. 2467M-AVR-11/04



Speicherorganisation des AVR (ATmega128)



AVR Clock-System



System-Module

Watchdog hat separaten on-Chip Oszillator. Periode kann zwischen 14ms und 1,9 Sek. eingestellt werden.

Quelle für Clock kann programmiert werden. Änderungen erfordern ein "Chip Erase" Befehl.

Data Sheet:
Rev. 2467M-AVR-11/04

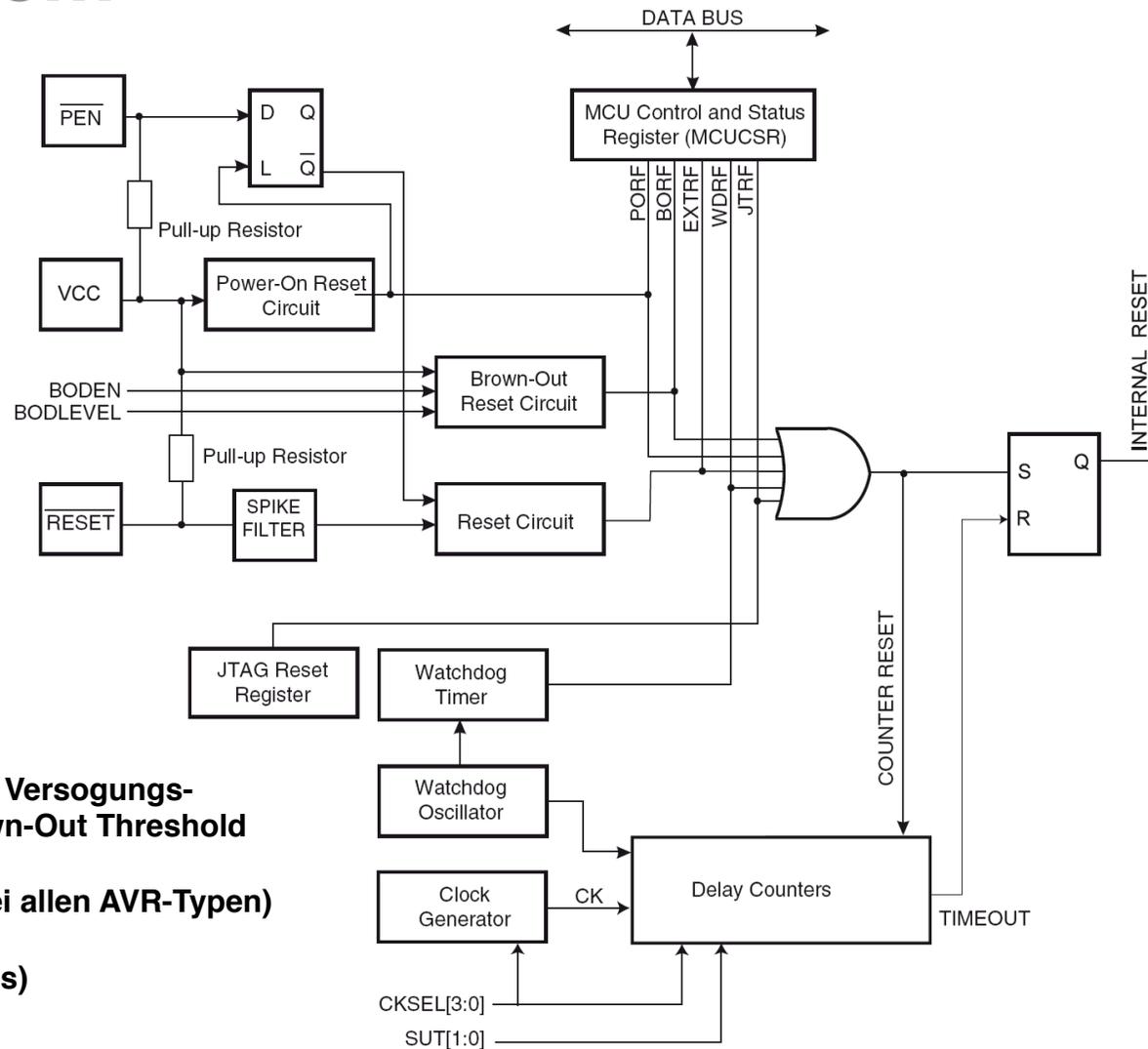


AVR Reset-System

Quellen für Reset:

1. Power-on Reset
2. External Reset
3. Watchdog Reset
4. Brown-out Reset
5. JTAG AVR Reset

- PEN:** Programming Enable
BODEN: Brown-Out Enable
 Brown-Out detektiert, dass die Versorgungsspannung unter die BOT (Brown-Out Threshold Voltage) gesunken ist.
BODLEVEL: Einstellbare Schwelle (nicht bei allen AVR-Typen)
CKSEL 3..0: Clock Select
SUT 1..0: Select Start-Up Time (4,1 -65 ms)



Reset Characteristics

Symbol	Parameter	Condition	Min	Typ	Max	Units
V_{POT}	Power-on Reset Threshold Voltage (rising)			1.4	2.3	V
	Power-on Reset Threshold Voltage (falling) ⁽¹⁾			1.3	2.3	V
V_{RST}	\overline{RESET} Pin Threshold Voltage		$0.2 V_{CC}$		$0.85 V_{CC}$	V
t_{RST}	Pulse width on \overline{RESET} Pin		1.5			μs
V_{BOT}	Brown-out Reset Threshold Voltage ⁽²⁾	BODLEVEL = 1	2.4	2.6	2.9	V
		BODLEVEL = 0	3.7	4.0	4.5	
t_{BOD}	Minimum low voltage period for Brown-out Detection	BODLEVEL = 1		2		μs
		BODLEVEL = 0		2		μs
V_{HYST}	Brown-out Detector hysteresis			100		mV

Notes: 1. The Power-on Reset will not work unless the supply voltage has been below V_{POT} (falling)

Start Up Times for external clock selection

SUT1..0	Start-up Time from Power-down and Power-save	Additional Delay from Reset ($V_{CC} = 5.0V$)	Recommended Usage
00	6 CK	–	BOD enabled
01	6 CK	4.1 ms	Fast rising power
10	6 CK	65 ms	Slowly rising power
11	Reserved		



Reduktion des Stromverbrauch durch “Schlaf-Modi”

Idle: CPU wird angehalten (clk_{cpu} und $\text{clk}_{\text{Flash}}$). Andere Komponenten, wie ADC, SPI, USART, Watchdog (WTD), Timer/Counter laufen weiter.

ADC Rauschunterdrückung: Hält CPU und I/O an (clk_{cpu} , $\text{clk}_{\text{Flash}}$ und $\text{clk}_{\text{I/O}}$). ADC, WTD, Interrupt System, Timer/Counter 0 laufen weiter,

Power Down: Hält alle generierten Clocks an. Operation nur noch für asynchrone Komponenten (externe Interrupts, Zweidraht serielles Interface, WTD, Reset und Brown-Out Reset (Spannungsüberwachung)). Dadurch kann der Prozessor aus dem Power-Down geweckt werden.

Power Save: wie Power Down mit dem Unterschied, dass Timer 0 durchläuft. Bei Überlauf wird der Prozessor geweckt.

Stand-By: wie Power-Down aber der Oszillator läuft durch. Dadurch “Wecken” in weniger Zeit (6 Clock Zyklen). Extended Stand-By identisch nur andere Quelle für die Clock.



Reduktion des Stromverbrauch durch “Schlaf-Modi”

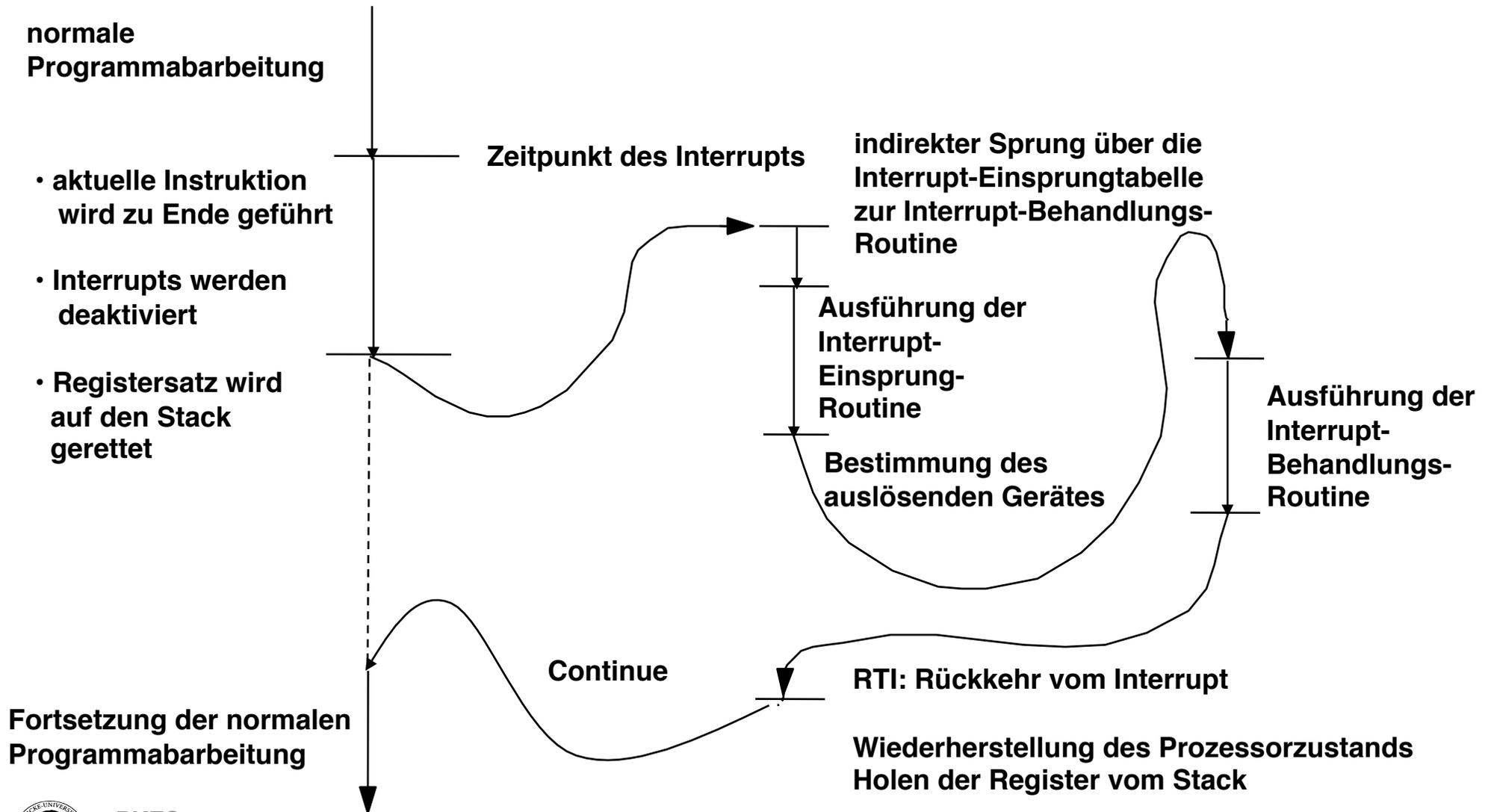
Symbol	Parameter	Condition	Min	Typ	Max	Units	
I_{CC}	Power Supply Current	Active 4 MHz, $V_{CC} = 3V$ (ATmega128L)			5.5	mA	
		Active 8 MHz, $V_{CC} = 5V$ (ATmega128)			19	mA	
		Idle 4 MHz, $V_{CC} = 3V$ (ATmega128L)			2.5	mA	
		Idle 8 MHz, $V_{CC} = 5V$ (ATmega128)			11	mA	
	Power-down mode	WDT enabled, $V_{CC} = 3V$			< 15	25	μA
		WDT disabled, $V_{CC} = 3V$			< 5	10	μA

Einsparungen bis ca. 1:1000 möglich

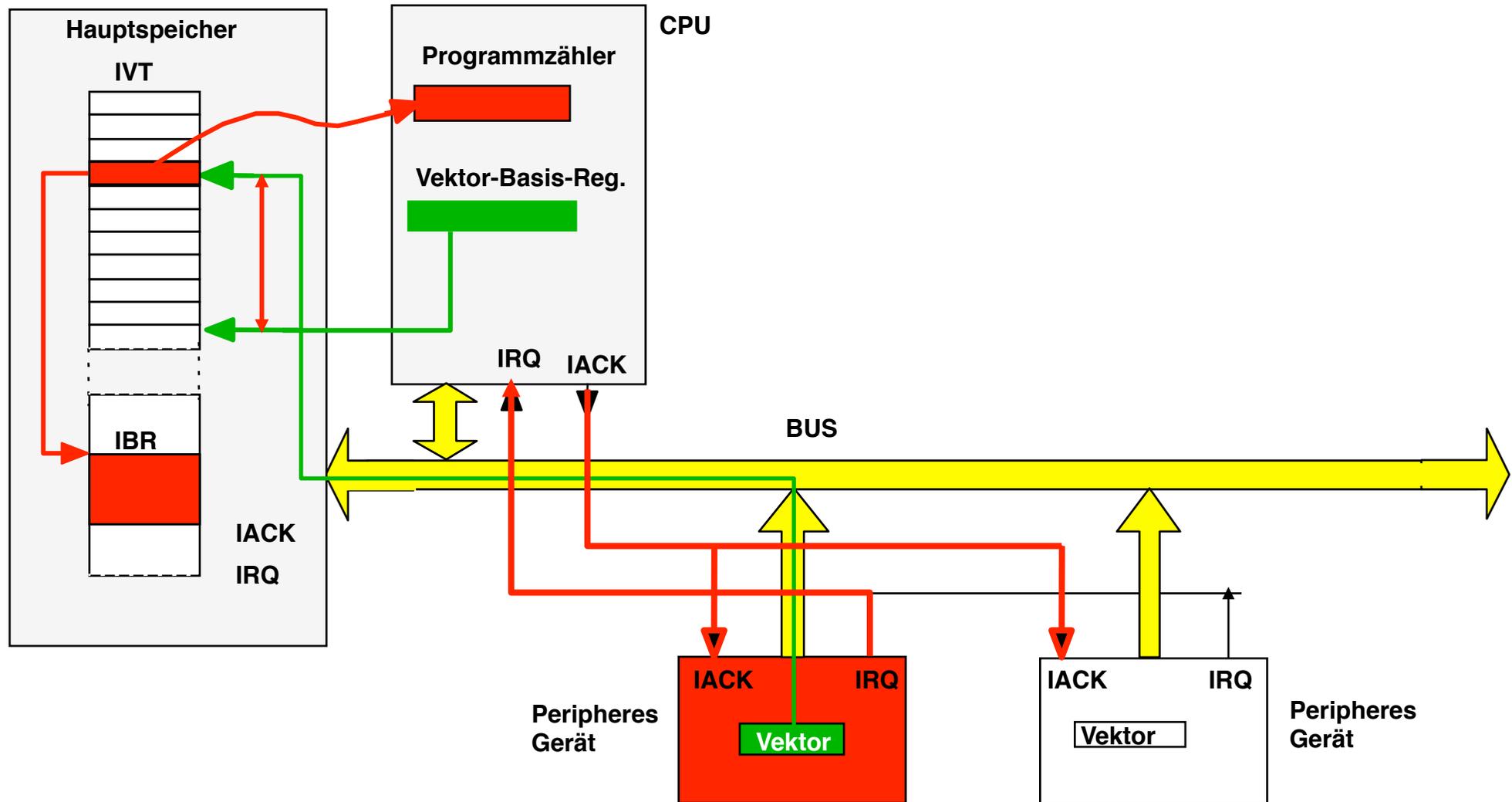
**Nur solche Komponenten aktiviert lassen, die tatsächlich gebraucht werden.
Selektives Abschalten von ADC, Analogkomparator, WTD, Brown-Out, interner
Spannungsreferenz, Port Pins.**



Abarbeitung eines Interrupts



Vektorisierte Unterbrechungsbearbeitung



AVR - Unterbrechungsbearbeitung

8 Interruptvektoren für externe Interrupts an Port INT 0...7

25 Interruptvektoren für interne Ereignisse (Timer, ADC, Comm, Memories)

1 Reset Interruptvektor

Die Priorität der Interrupts ist durch die Position in der Interrupt-Vektor-Map ($0000_{16} - 0044_{16}$) festgelegt. Diese wird im Programmspeicher (Flash) angelegt. Es kann gewählt werden, ob sie am Anfang (0000_{16}) stehen soll oder vor der “Boot-Loader” Sektion.

Prioritäten:

- Reset**
- ext.Interrupts 0-7**
- Timer (hohe Prio)**
- Kommunikation**
- ADC**
- Analog-Komparator**
- Timer (niedrigere Prio)**
- Kommunikation (niedrigere Prio)**



Most typical and general program setup for the Reset and Interrupt Vector Addresses in ATmega128:

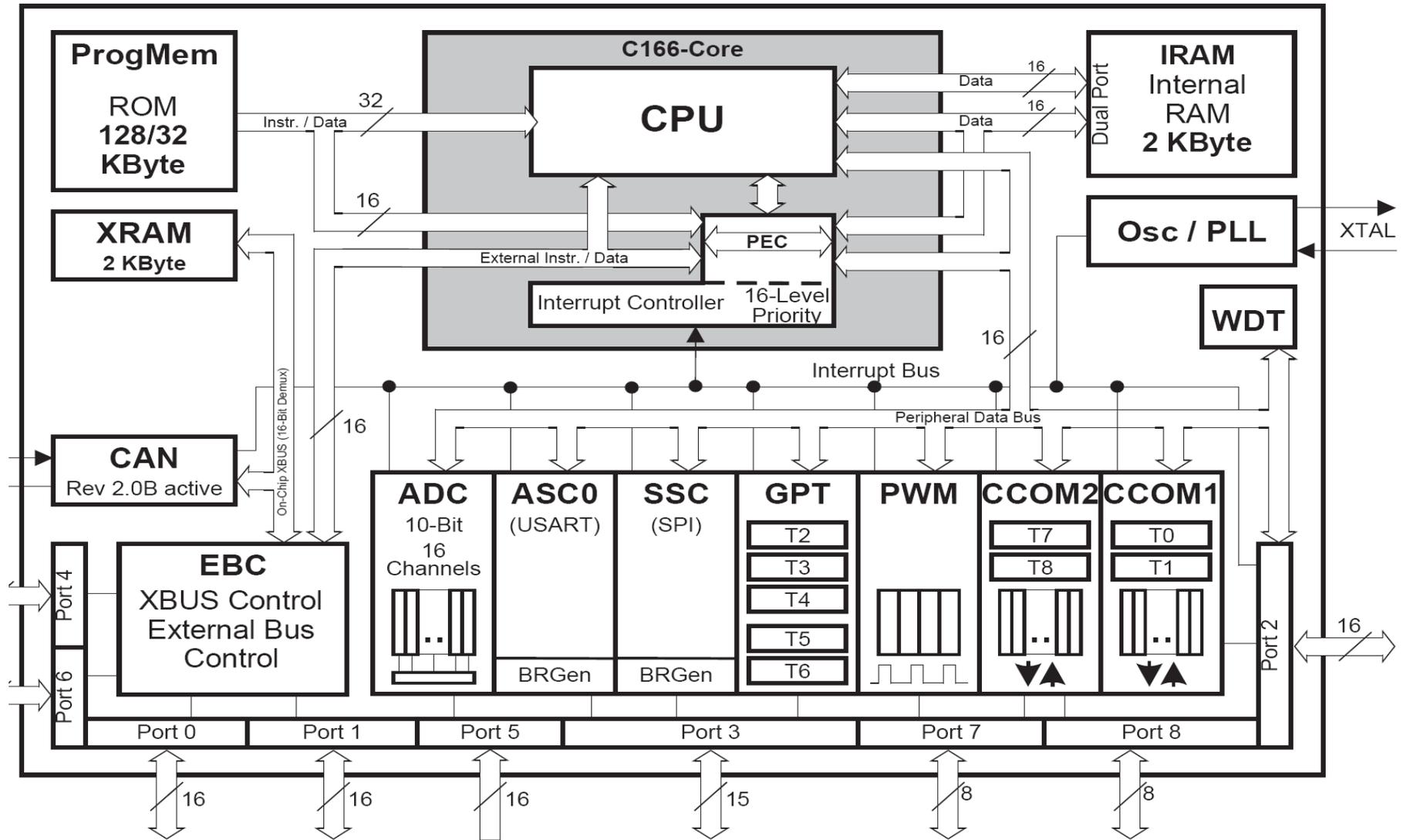
```

Addr.  LabelsCode  Comments
$0000 jmp RESET ; Reset Handler
$0002 jmp EXT_INT0 ; IRQ0 Handler
$0004 jmp EXT_INT1 ; IRQ1 Handler
$0006 jmp EXT_INT2 ; IRQ2 Handler
$0008 jmp EXT_INT3 ; IRQ3 Handler
$000A jmp EXT_INT4 ; IRQ4 Handler
$000C jmp EXT_INT5 ; IRQ5 Handler
$000E jmp EXT_INT6 ; IRQ6 Handler
$0010 jmp EXT_INT7 ; IRQ7 Handler
$0012 jmp TIM2_COMP; Timer2 Compare Handler
$0014 jmp TIM2_OVF ; Timer2 Overflow Handler
$0016 jmp TIM1_CAPT ; Timer1 Capture Handler
$0018 jmp TIM1_COMPA ;Timer1 CompareA Handler
$001A jmp TIM1_COMPB; Timer1 CompareB Handler
$001C jmp TIM1_OVF ; Timer1 Overflow Handler
$001E jmp TIM0_COMP ; Timer0 Compare Handler
$0020 jmp TIM0_OVF ; Timer0 Overflow Handler
$0022 jmp SPI_STC ; SPI Transfer Complete Handler
$0024 jmp USART0_RXC; USART0 RX Complete Handler
$0026 jmp USART0_DRE; USART0,UDR Empty Handler
$0028 jmp USART0_TXC; USART0 TX Complete Handler
$002A jmp ADC ; ADC Conversion Complete Handler
$002C jmp EE_RDY ; EEPROM Ready Handler
$002E jmp ANA_COMP ; Analog Comparator Handler
$0030 jmp TIM1_COMP; Timer1 CompareC Handler
$0032 jmp TIM3_CAPT ; Timer3 Capture Handler
$0034 jmp TIM3_COMPA; Timer3 CompareA Handler
$0036 jmp TIM3_COMPB; Timer3 CompareB Handler
$0038 jmp TIM3_COMP; Timer3 CompareC Handler
$003A jmp TIM3_OVF ; Timer3 Overflow Handler
$003C jmp USART1_RXC; USART1 RX Complete Handler
$003E jmp USART1_DRE; USART1,UDR Empty Handler
$0040 jmp USART1_TXC; USART1 TX Complete Handler
$0042 jmp TWI ; Two-wire Serial Interface Interrupt
Handler
$0044 jmp SPM_RDY ; SPM Ready Handler
;
$0046 RESET:ldir16, high(RAMEND); Main program start
$0047 out SPH,r16 ; Set stack pointer to top of RAM
$0048 ldi r16, low(RAMEND)
$0049 out SPL,r16
$004A sei ; Enable interrupts
$004B <instr> xxx

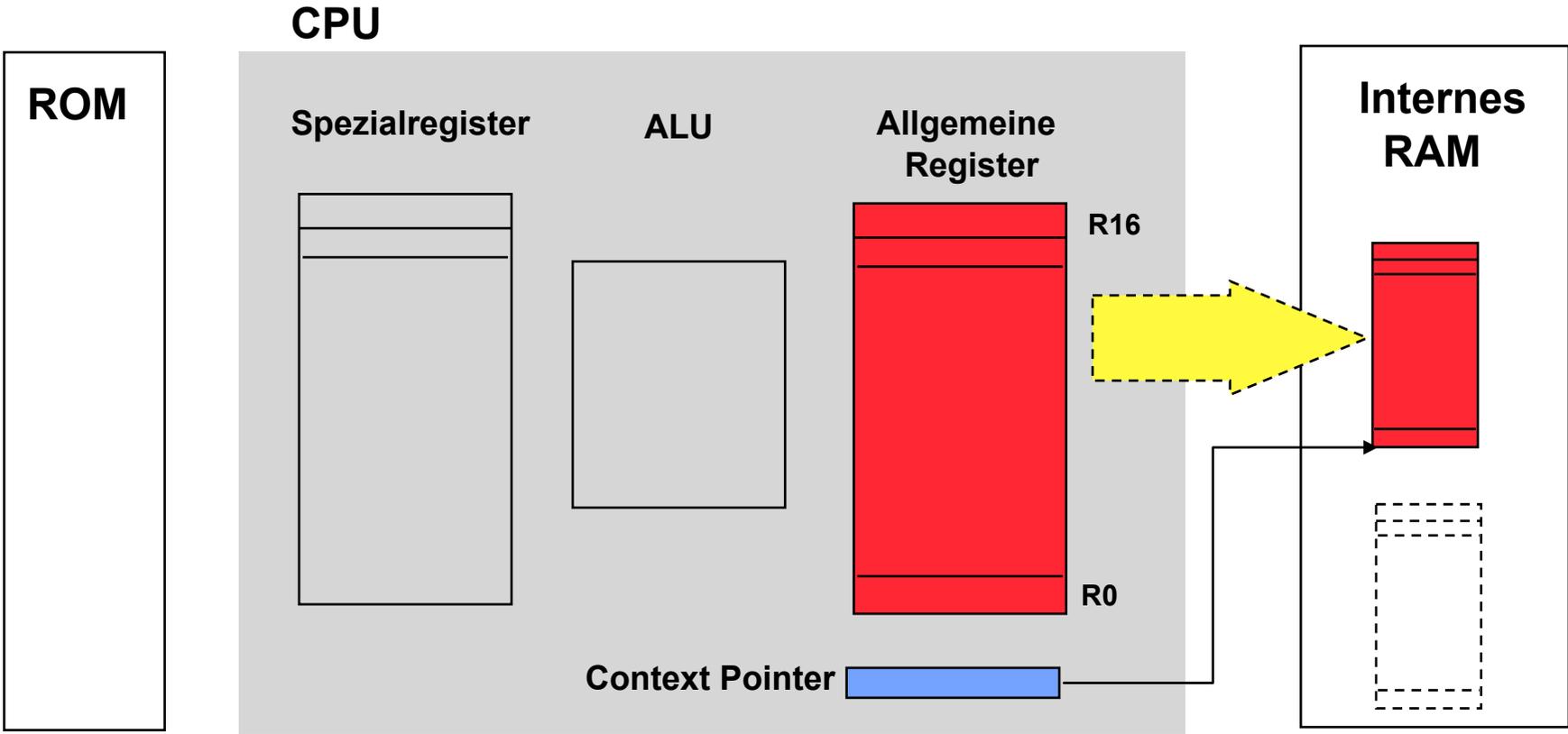
```



Infinion CR167



Blockdiagramm der CPU C166/167



(Context Switch 100ns)



Interner RAM Bereich und Bereich für die Special Functional Registers (SFRs)

Internes RAM wird genutzt:

- für System Stack
- für allgemeine Register (mehrere Bänke)
- für den PEC (Peripheral Event Controller)
- als RAM

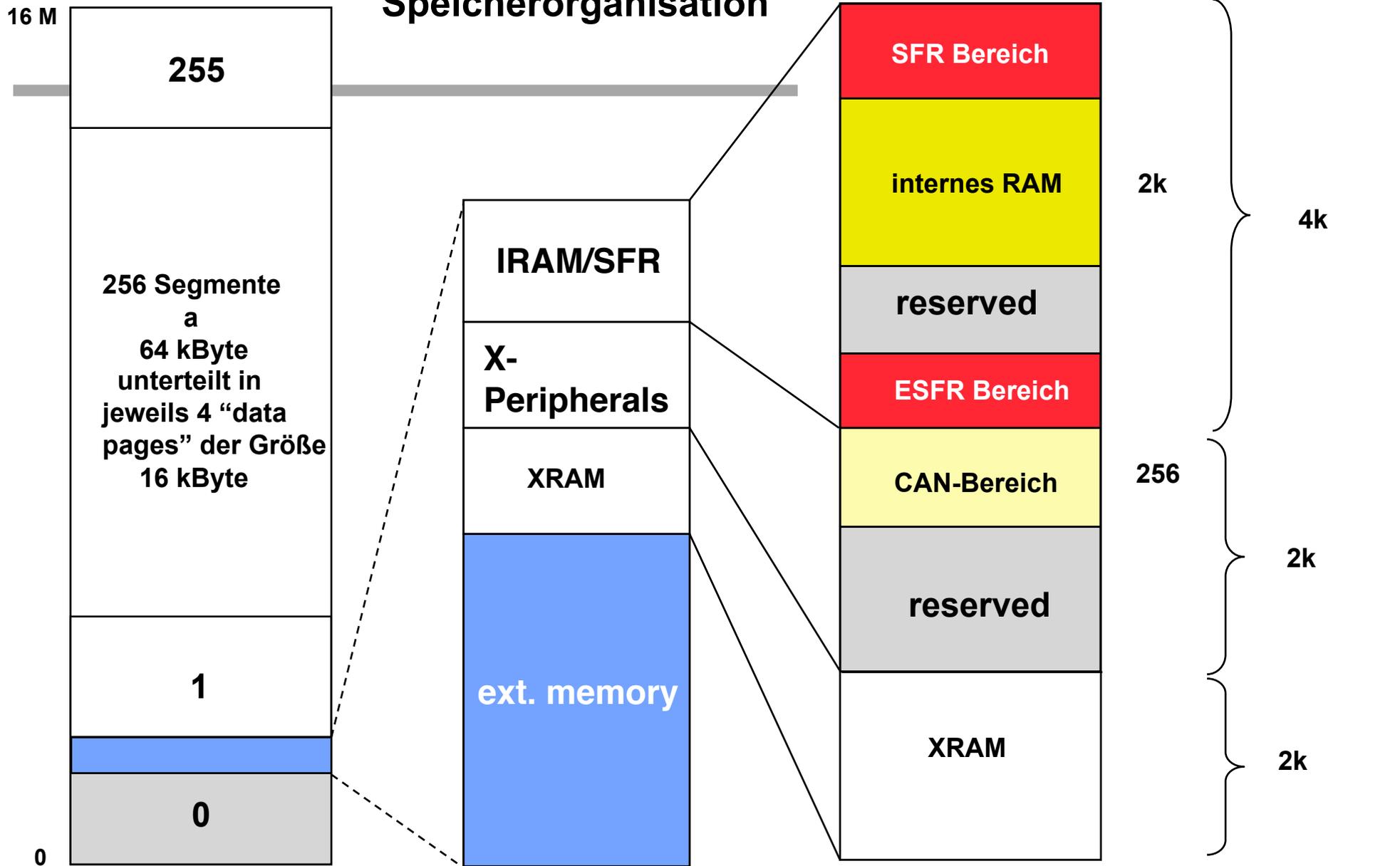
SFRs:

kontrollieren die gesamten on-chip Spezialfunktionen wie:

- Ein-Ausgabe,
- A/D-Wandler,
- Kommunikation (CAN)
- Timer



Speicherorganisation



Interruptverarbeitung

- **Reaktion auf externe Ereignisse**
- **Reaktion auf interne Funktionseinheiten, z.B.**
 - **ADC**
 - **Timer**
 - **Kommunikationseinheiten**

**Ziele: Nebenläufige Bearbeitung verschiedener Vorgänge
Minimale Beanspruchung der CPU**



Interrupt und Trap Funktionen im C167

Normale Interrupt-Verarbeitung

- Interrupts von On-Chip Komponenten

Interrupt Verarbeitung über den PEC (Peripheral Event Controller)

- Minimale Interferenz mit der CPU (nur 1 Befehlszyklus)
- Kein Abspeichern des CPU-Status notwendig

Trap - Funktionen

- Software Traps
- Hardware Traps ausgelöst durch Befehlesausführung (z.B. illegal Opcode, Overflow, etc.)

Externe Interrupts

- Capture Input / Compare Output Leitungen
- Timer Input
- Fast External Interrupts (werden alle 50 ns gesampled, normal 400 ns bei einer 20 MHz Taktrate)



Source of Interrupt or PEC Service Request	Request Flag	Enable Flag	Interrupt Vector	Vector Location	Trap Number
CAPCOM Register 0	CC0IR	CC0IE	CC0INT	00'0040 _H	10 _H / 16 _D
CAPCOM Register 1	CC1IR	CC1IE	CC1INT	00'0044 _H	11 _H / 17 _D
CAPCOM Register 2	CC2IR	CC2IE	CC2INT	00'0048 _H	12 _H / 18 _D
CAPCOM Register 3	CC3IR	CC3IE	CC3INT	00'004C _H	13 _H / 19 _D
CAPCOM Register 4	CC4IR	CC4IE	CC4INT	00'0050 _H	14 _H / 20 _D
CAPCOM Register 5	CC5IR	CC5IE	CC5INT	00'0054 _H	15 _H / 21 _D
CAPCOM Register 6	CC6IR	CC6IE	CC6INT	00'0058 _H	16 _H / 22 _D
CAPCOM Register 7	CC7IR	CC7IE	CC7INT	00'005C _H	17 _H / 23 _D
CAPCOM Register 8	CC8IR	CC8IE	CC8INT	00'0060 _H	18 _H / 24 _D
CAPCOM Register 9	CC9IR	CC9IE	CC9INT	00'0064 _H	19 _H / 25 _D
CAPCOM Register 10	CC10IR	CC10IE	CC10INT	00'0068 _H	1A _H / 26 _D
CAPCOM Register 11	CC11IR	CC11IE	CC11INT	00'006C _H	1B _H / 27 _D
CAPCOM Register 12	CC12IR	CC12IE	CC12INT	00'0070 _H	1C _H / 28 _D
CAPCOM Register 13	CC13IR	CC13IE	CC13INT	00'0074 _H	1D _H / 29 _D
CAPCOM Register 14	CC14IR	CC14IE	CC14INT	00'0078 _H	1E _H / 30 _D
CAPCOM Register 15	CC15IR	CC15IE	CC15INT	00'007C _H	1F _H / 31 _D
CAPCOM Register 16	CC16IR	CC16IE	CC16INT	00'00C0 _H	30 _H / 48 _D
CAPCOM Register 17	CC17IR	CC17IE	CC17INT	00'00C4 _H	31 _H / 49 _D
CAPCOM Register 18	CC18IR	CC18IE	CC18INT	00'00C8 _H	32 _H / 50 _D
CAPCOM Register 19	CC19IR	CC19IE	CC19INT	00'00CC _H	33 _H / 51 _D
CAPCOM Register 20	CC20IR	CC20IE	CC20INT	00'00D0 _H	34 _H / 52 _D
CAPCOM Register 21	CC21IR	CC21IE	CC21INT	00'00D4 _H	35 _H / 53 _D
CAPCOM Register 22	CC22IR	CC22IE	CC22INT	00'00D8 _H	36 _H / 54 _D
CAPCOM Register 23	CC23IR	CC23IE	CC23INT	00'00DC _H	37 _H / 55 _D
CAPCOM Register 24	CC24IR	CC24IE	CC24INT	00'00E0 _H	38 _H / 56 _D
CAPCOM Register 25	CC25IR	CC25IE	CC25INT	00'00E4 _H	39 _H / 57 _D
CAPCOM Register 26	CC26IR	CC26IE	CC26INT	00'00E8 _H	3A _H / 58 _D
CAPCOM Register 27	CC27IR	CC27IE	CC27INT	00'00EC _H	3B _H / 59 _D
CAPCOM Register 28	CC28IR	CC28IE	CC28INT	00'00F0 _H	3C _H / 60 _D
CAPCOM Register 29	CC29IR	CC29IE	CC29INT	00'0110 _H	44 _H / 68 _D
CAPCOM Register 30	CC30IR	CC30IE	CC30INT	00'0114 _H	45 _H / 69 _D
CAPCOM Register 31	CC31IR	CC31IE	CC31INT	00'0118 _H	46 _H / 70 _D

Source of Interrupt or PEC Service Request	Request Flag	Enable Flag	Interrupt Vector	Vector Location	Trap Number
CAPCOM Timer 0	T0IR	T0IE	T0INT	00'0080 _H	20 _H / 32 _D
CAPCOM Timer 1	T1IR	T1IE	T1INT	00'0084 _H	21 _H / 33 _D
CAPCOM Timer 7	T7IR	T7IE	T7INT	00'00F4 _H	3D _H / 61 _D
CAPCOM Timer 8	T8IR	T8IE	T8INT	00'00F8 _H	3E _H / 62 _D
GPT1 Timer 2	T2IR	T2IE	T2INT	00'0088 _H	22 _H / 34 _D
GPT1 Timer 3	T3IR	T3IE	T3INT	00'008C _H	23 _H / 35 _D
GPT1 Timer 4	T4IR	T4IE	T4INT	00'0090 _H	24 _H / 36 _D
GPT2 Timer 5	T5IR	T5IE	T5INT	00'0094 _H	25 _H / 37 _D
GPT2 Timer 6	T6IR	T6IE	T6INT	00'0098 _H	26 _H / 38 _D
GPT2 CAPREL Register	CRIR	CRIE	CRINT	00'009C _H	27 _H / 39 _D
A/D Conversion Complete	ADCIR	ADCIE	ADCINT	00'00A0 _H	28 _H / 40 _D
A/D Overrun Error	ADEIR	ADEIE	ADEINT	00'00A4 _H	29 _H / 41 _D
ASC0 Transmit	S0TIR	S0TIE	S0TINT	00'00A8 _H	2A _H / 42 _D
ASC0 Transmit Buffer	S0TBIR	S0TBIE	S0TBINT	00'011C _H	47 _H / 71 _D
ASC0 Receive	S0RIR	S0RIE	S0RINT	00'00AC _H	2B _H / 43 _D
ASC0 Error	S0EIR	S0EIE	S0EINT	00'00B0 _H	2C _H / 44 _D
SSC Transmit	SSCTIR	SSCTIE	SSCTINT	00'00B4 _H	2D _H / 45 _D
SSC Receive	SSCRIR	SSCRIE	SSCRINT	00'00B8 _H	2E _H / 46 _D
SSC Error	SSCEIR	SSCEIE	SSCEINT	00'00BC _H	2F _H / 47 _D
PWM Channel 0...3	PWMIR	PWMIE	PWMINT	00'00FC _H	3F _H / 63 _D
CAN Interface	XP0IR	XP0IE	XP0INT	00'0100 _H	40 _H / 64 _D
X-Peripheral Node 1	XP1IR	XP1IE	XP1INT	00'0104 _H	41 _H / 65 _D
X-Peripheral Node 2	XP2IR	XP2IE	XP2INT	00'0108 _H	42 _H / 66 _D
PLL Unlock	XP3IR	XP3IE	XP3INT	00'010C _H	43 _H / 67 _D

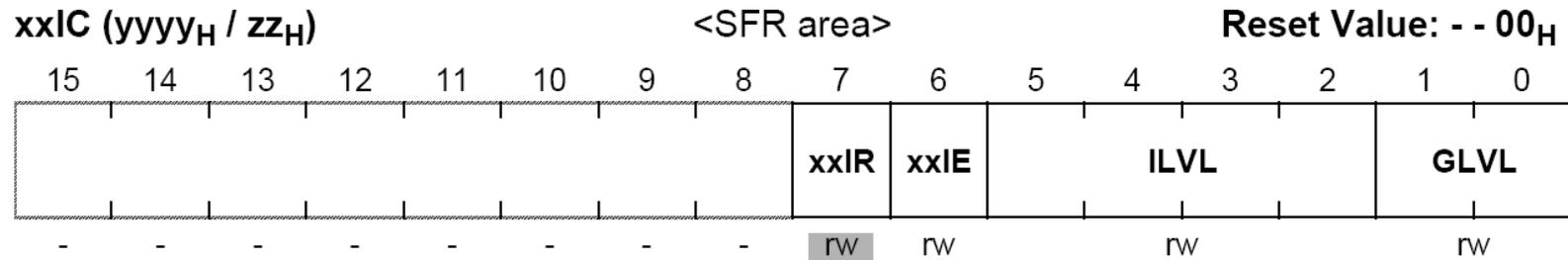
Note: Each entry of the interrupt vector table provides room for two word instructions or one doubleword instruction. The respective vector location results from multiplying the trap number by 4 (4 bytes per entry).

For devices which do not incorporate a CAN Module or a PLL the respective interrupt nodes may be used for software triggered interrupts (see X-Peripheral node n).

Interrupts und Traps des C 167



Interrupt Kontrolle: Einteilung in Gruppen und individuelle Prioritäten



Bit	Function
GLVL	Group Level Defines the internal order for simultaneous requests of the same priority. 3: Highest group priority 0: Lowest group priority
ILVL	Interrupt Priority Level Defines the priority level for the arbitration of requests. F _H : Highest priority level 0 _H : Lowest priority level
xxIE	Interrupt Enable Control Bit (individually enables/disables a specific source) '0': Interrupt request is disabled '1': Interrupt Request is enabled
xxIR	Interrupt Request Flag '0': No request pending '1': This source has raised an interrupt request



Beispiel für Gruppen- und individuelle Interruptprioritäten

Priority Level		Type of Service	
ILVL	GLVL	COUNT = 00H	COUNT ≠ 00 _H
1 1 1 1	1 1	CPU interrupt, level 15, group priority 3	PEC service, channel 7
1 1 1 1	1 0	CPU interrupt, level 15, group priority 2	PEC service, channel 6
1 1 1 0	1 0	CPU interrupt, level 14, group priority 2	PEC service, channel 2
1 1 0 1	1 0	CPU interrupt, level 13, group priority 2	CPU interrupt, level 13, group priority 2
0 0 0 1	1 1	CPU interrupt, level 1, group priority 3	CPU interrupt, level 1, group priority 3
0 0 0 1	0 0	CPU interrupt, level 1, group priority 0	CPU interrupt, level 1, group priority 0
0 0 0 0	X X	No service!	No service!



Interrupt Classes

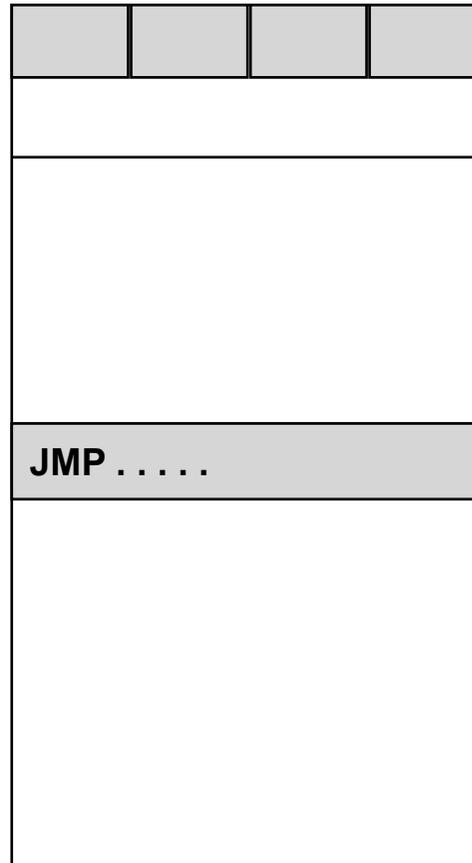
* ILVL: Interrupt Level Vector List
 ** GLVL: Group Level Vector List

ILVL* (priority)	GLVL**				Interpretation
	11	10	01	00	
15					PEC Service on up to 8 channels
14					
13					
12	x	x	x	x	Interrupt Class 1
11	x				5 sources on 2 levels
10					
9					
8	x	x	x	x	Interrupt Class 2
7	x	x	x	x	9 sources on 3 levels
6	x				
5	x	x	x	x	Interrupt Class 3
4	x				5 sources on 2 levels
3					
2					
1					
0					no service



Fast Interrupt Handling

IVT
Interrupt Vector Table



512 Byte
128 x 4-Byte Einträge

Bei Interrupt werden ein Befehl (4 Byte) oder 2 Befehle (2 Byte) ausgeführt. Normalerweise wird ein JMP zur Behandlungsroutine ausgeführt.



PEC: Peripheral Event Controller

- **Schnelle Alternative zur normalen Interruptverarbeitung**
- **Erlaubt den Transfer eines einzelnen Datums mit minimaler CPU-Belastung**
 - **CPU-Aktivität wird nur für einen einzigen Zyklus unterbrochen**
 - **kein interner Zustand muß gerettet werden**
 - **Prioritätsebenen 14 oder 15.**



Datentransfer mit PEC

PEC-Cntrl.-Reg



Feld:

INC kontrolliert die SRC und DST PEC-Pointer

00: keine Modifikation

01: Incrementiere DSTx um 1 oder 2 (Byte oder Wort)

10: Incrementiere SRCx um 1 oder 2 (Byte oder Wort)

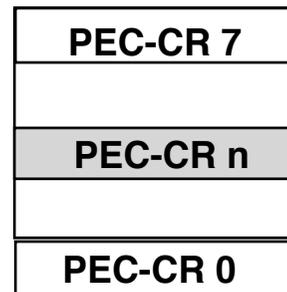
11: Reserviert

BWT: 0: Transferiere Wort

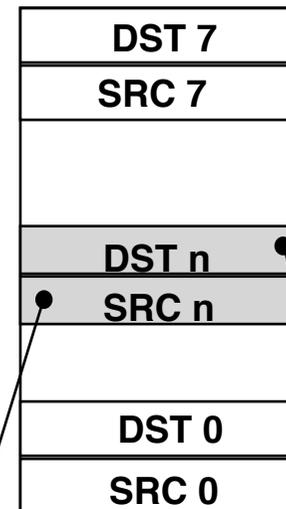
1: Transferiere Byte

Count: Anzahl der zu transferierenden Bytes bzw. Worte

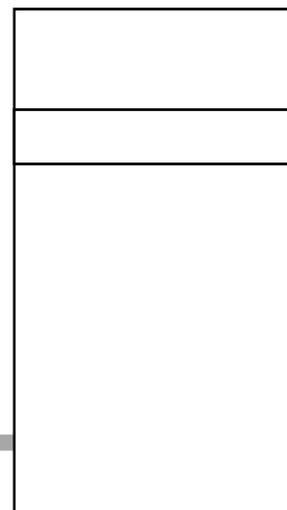
PEC-Cntrl.-Reg



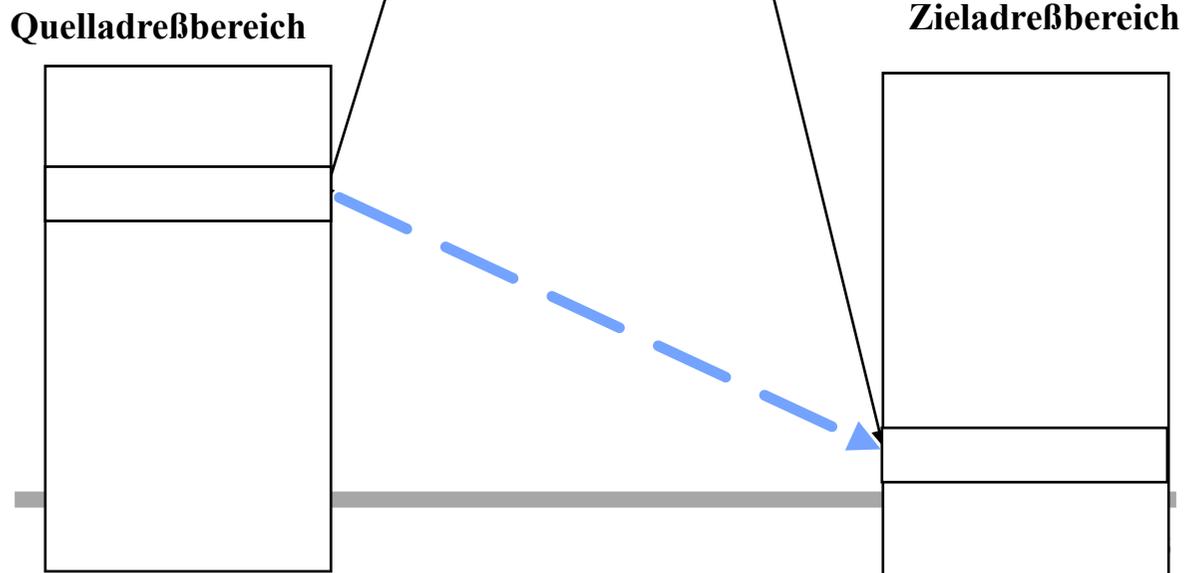
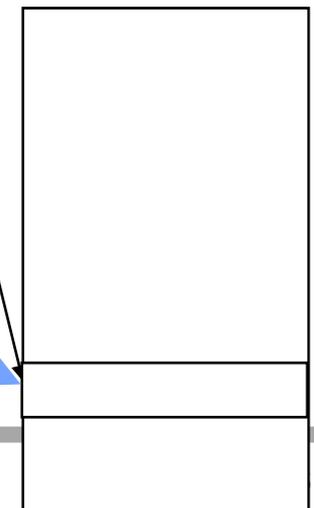
PEC-Pointer



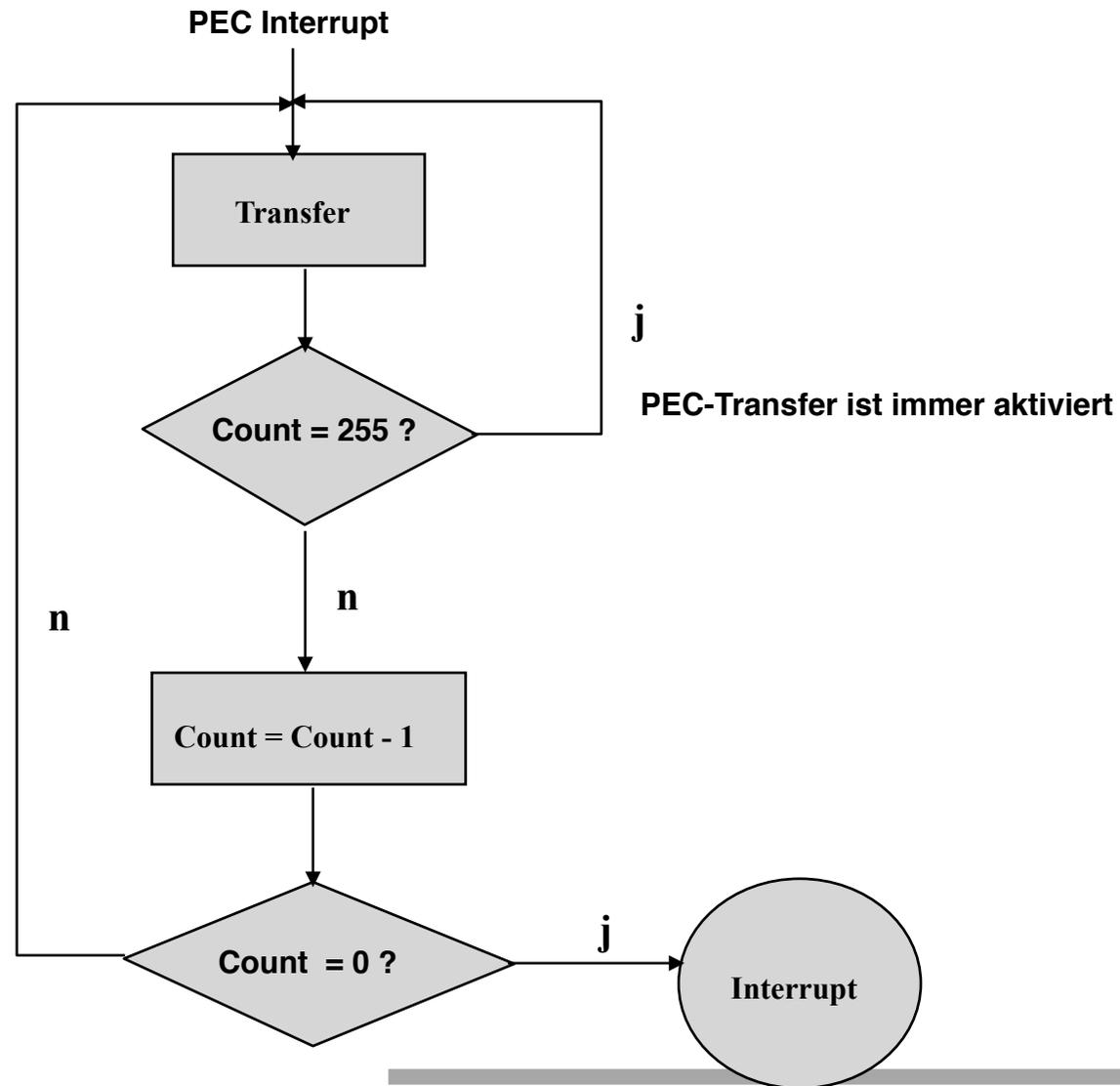
Quelladreibereich



Zieladreibereich

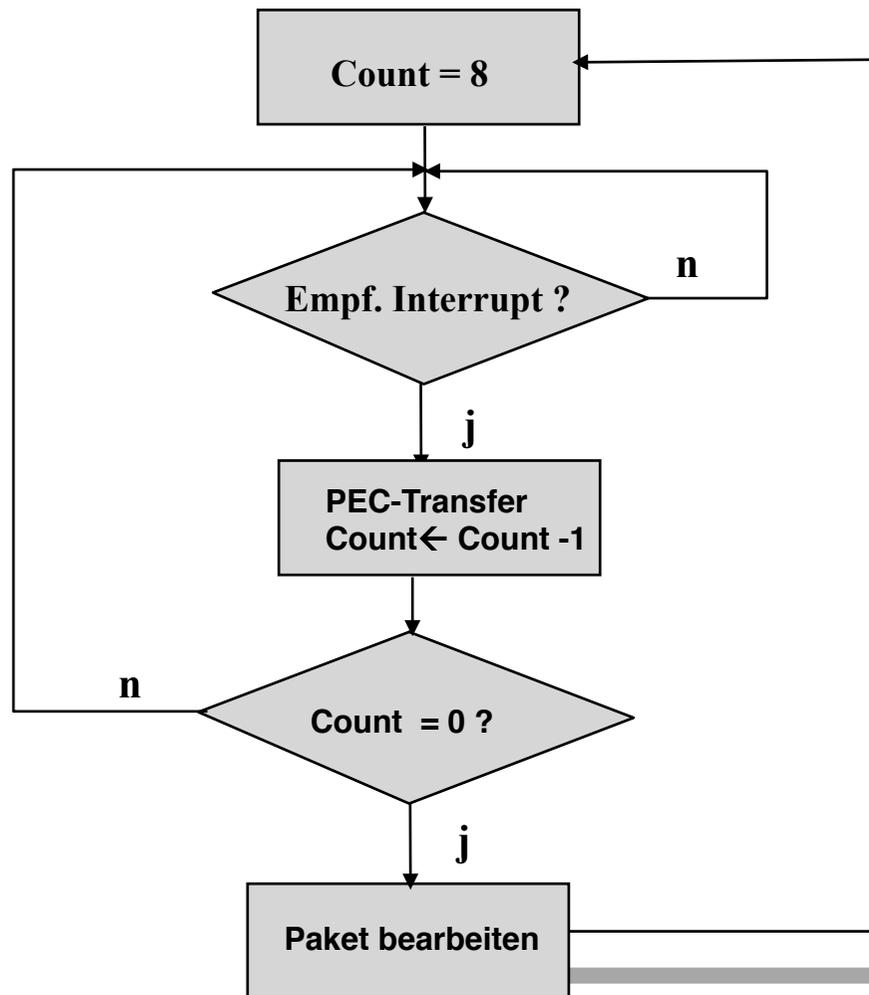
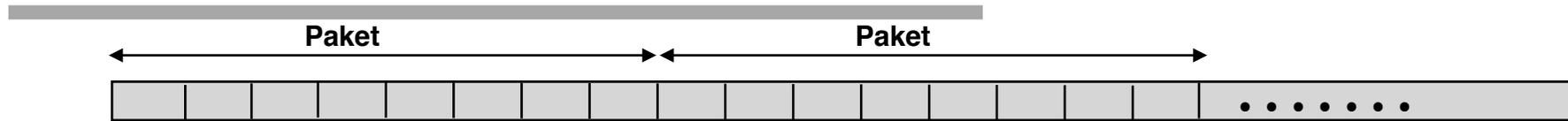


Steuerung des PEC- Datentransfers



PEC-Anwendungsbeispiel: asynchrone Kommunikation

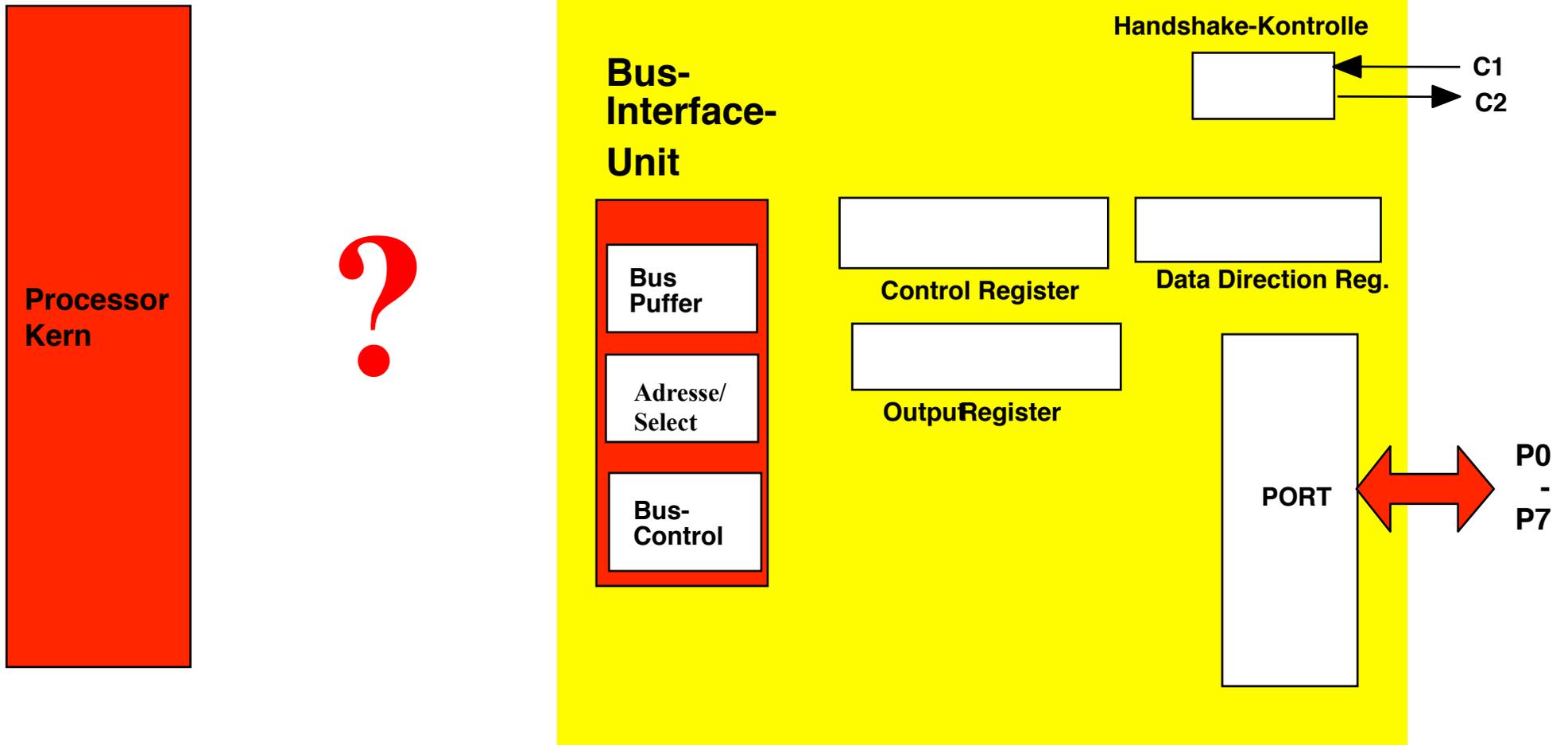
- jeweils 8 Byte-Pakete sollen empfangen werden.



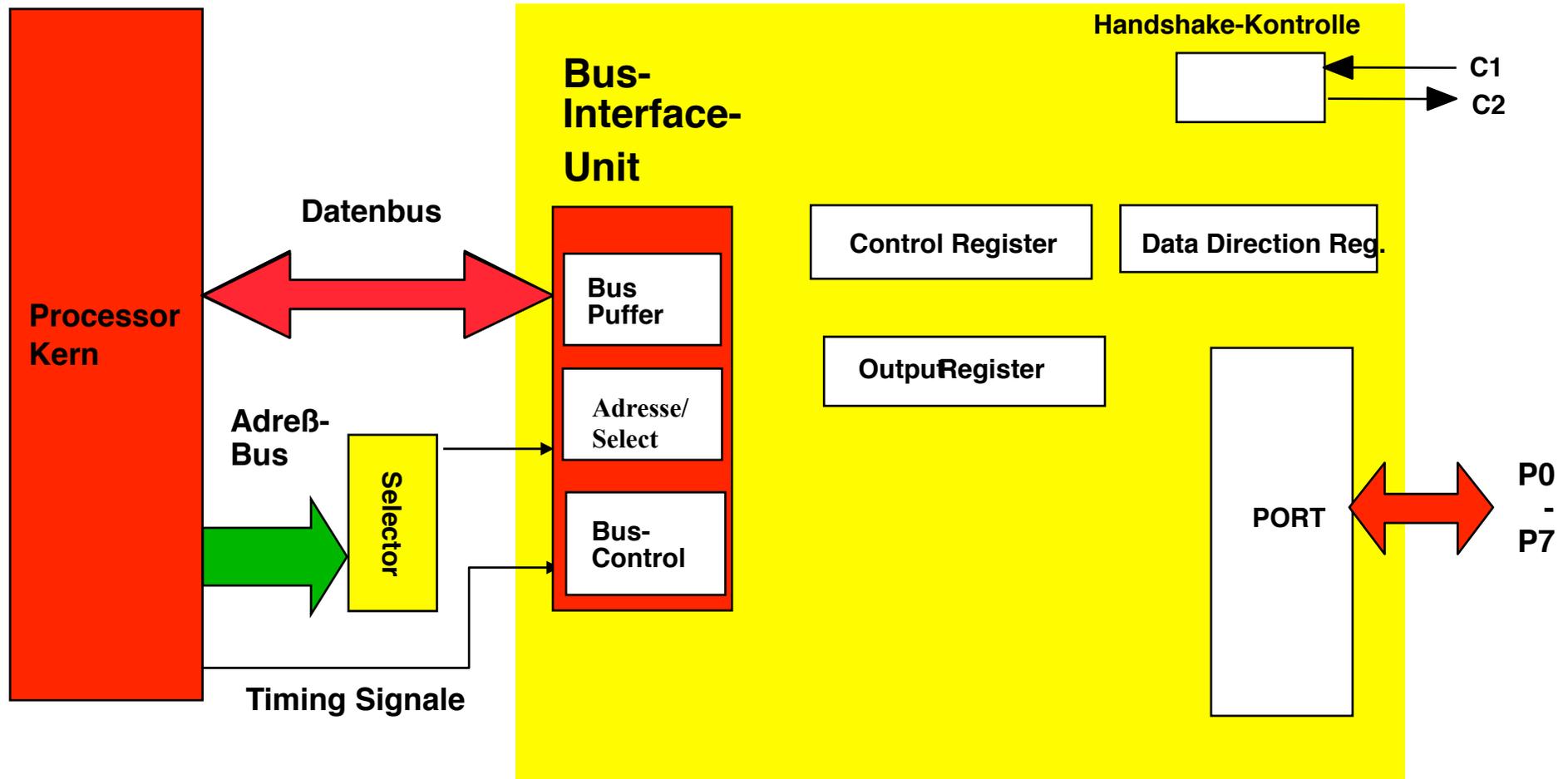
Ports, die Schnittstelle zur Peripherie

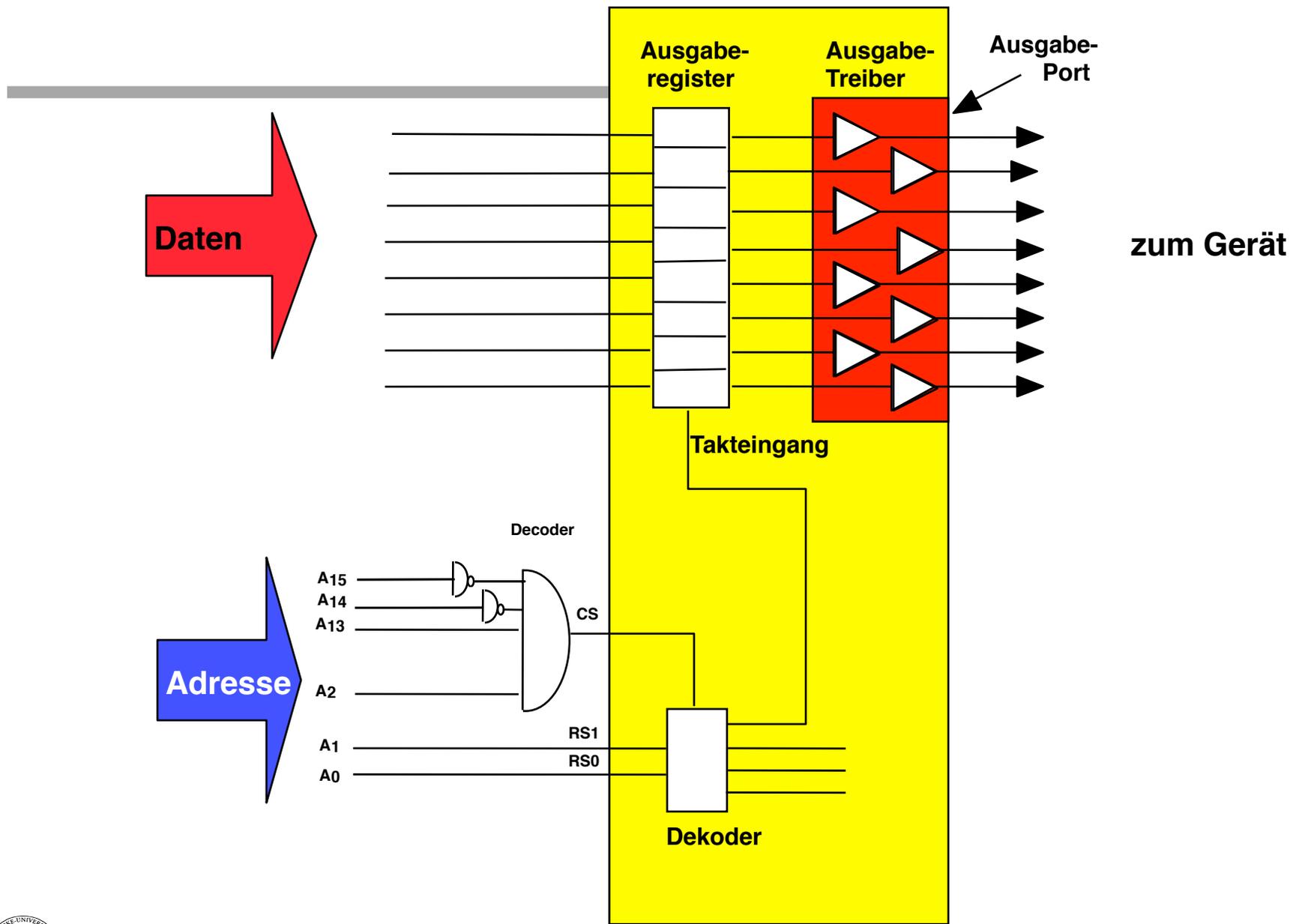


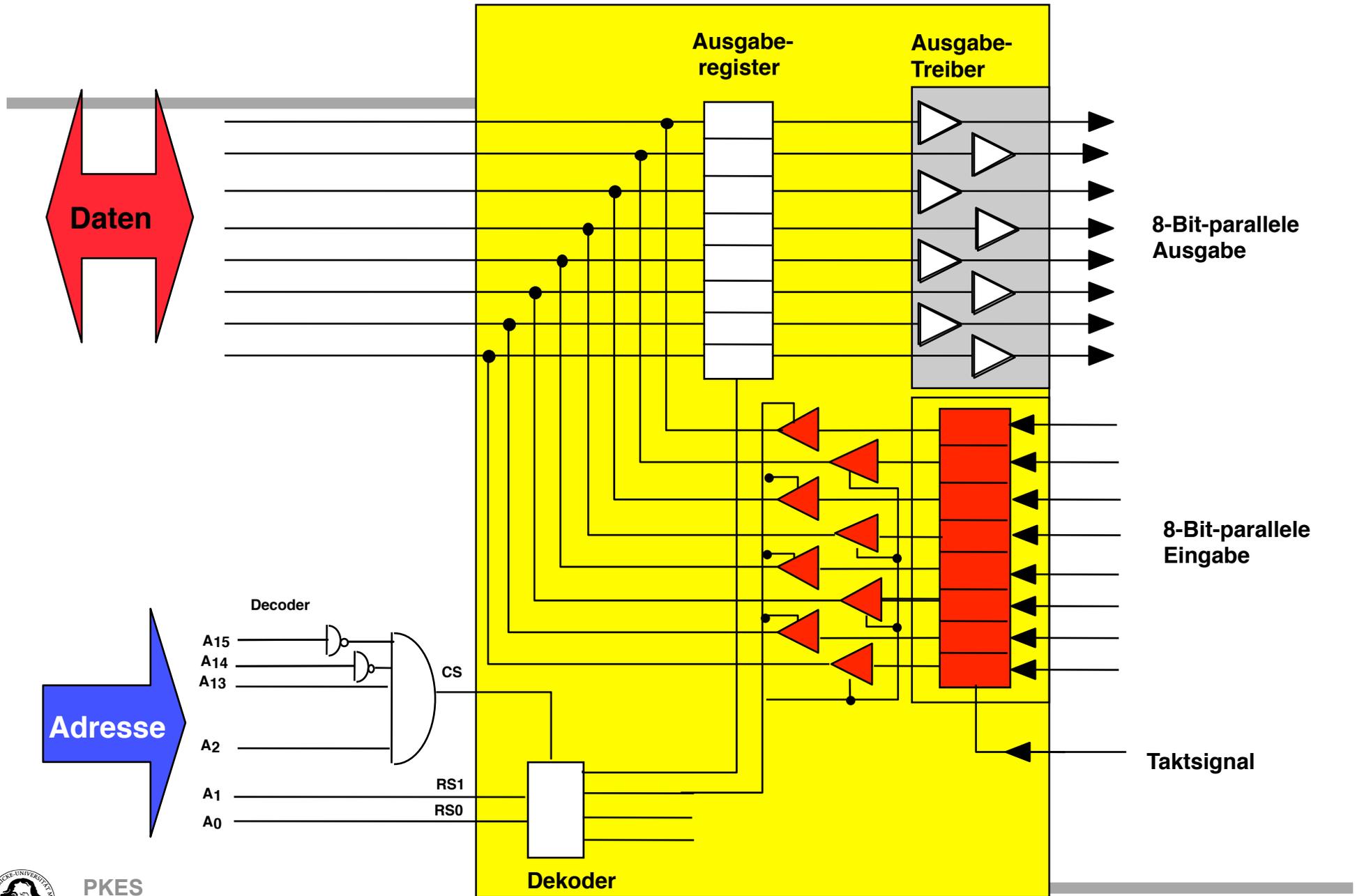
Eine Byte-parallele Geräteschnittstelle



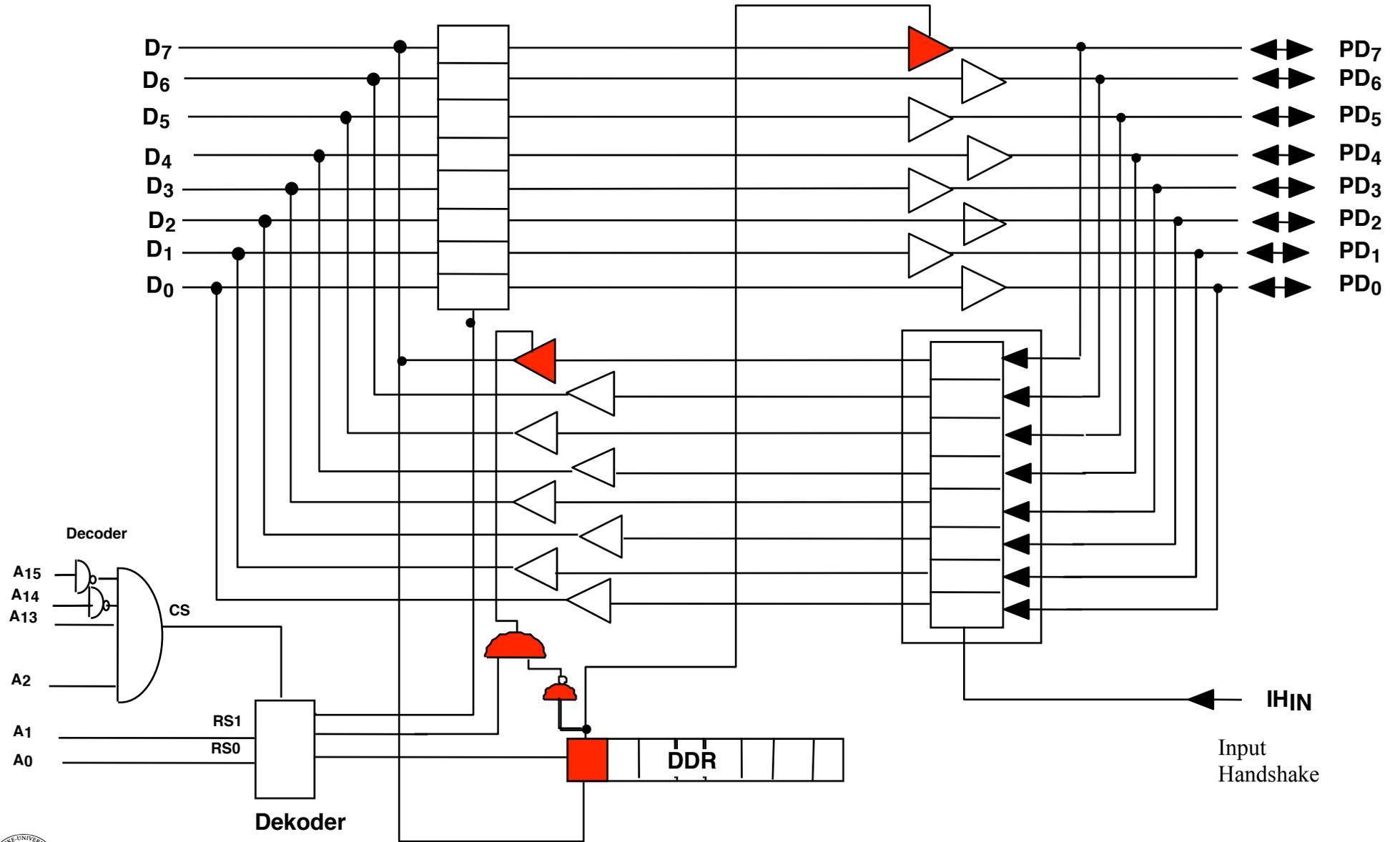
Eine Byte-parallele Geräteschnittstelle

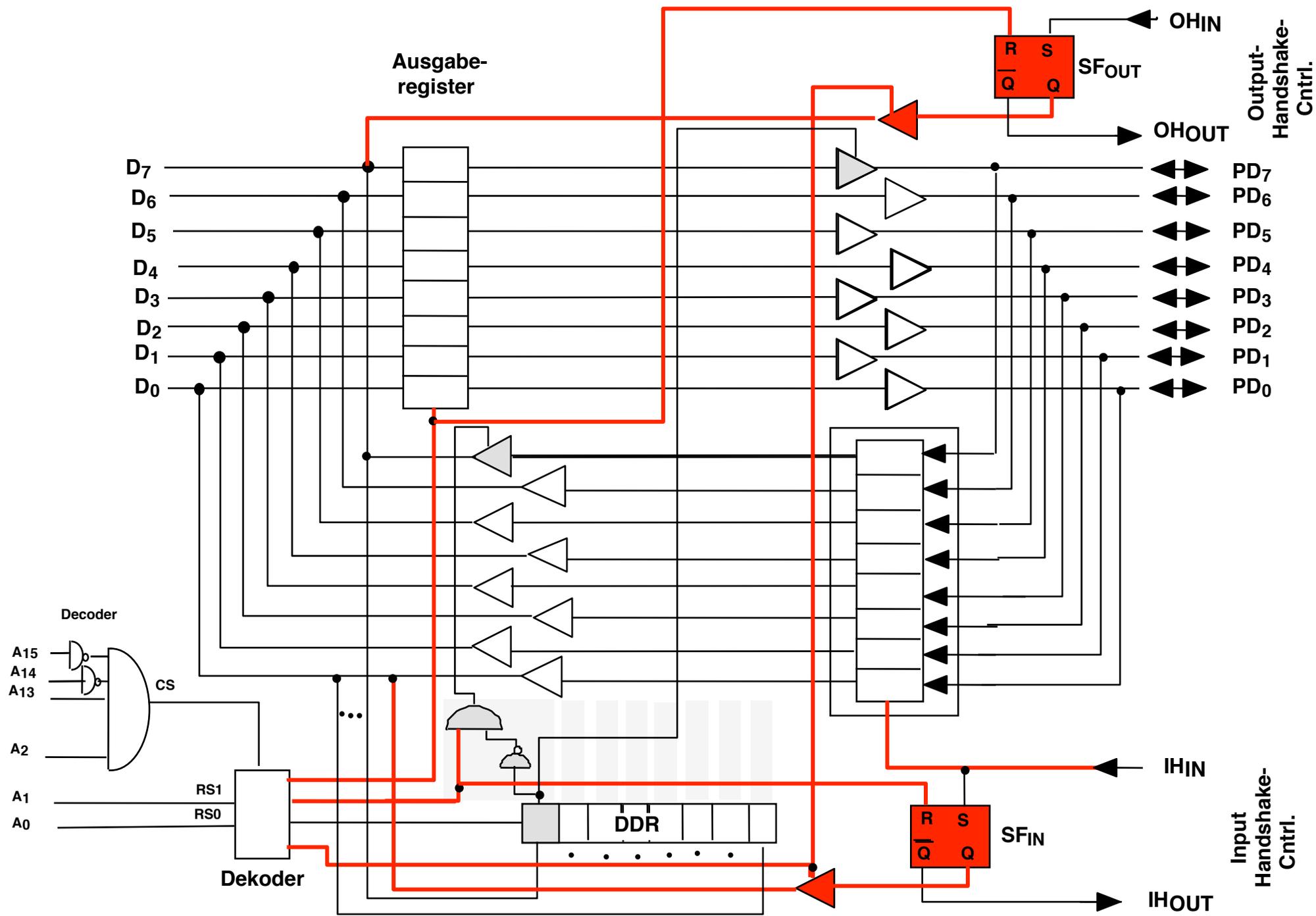






Ausgabe- register





Infineon C 167 I/O-Ports

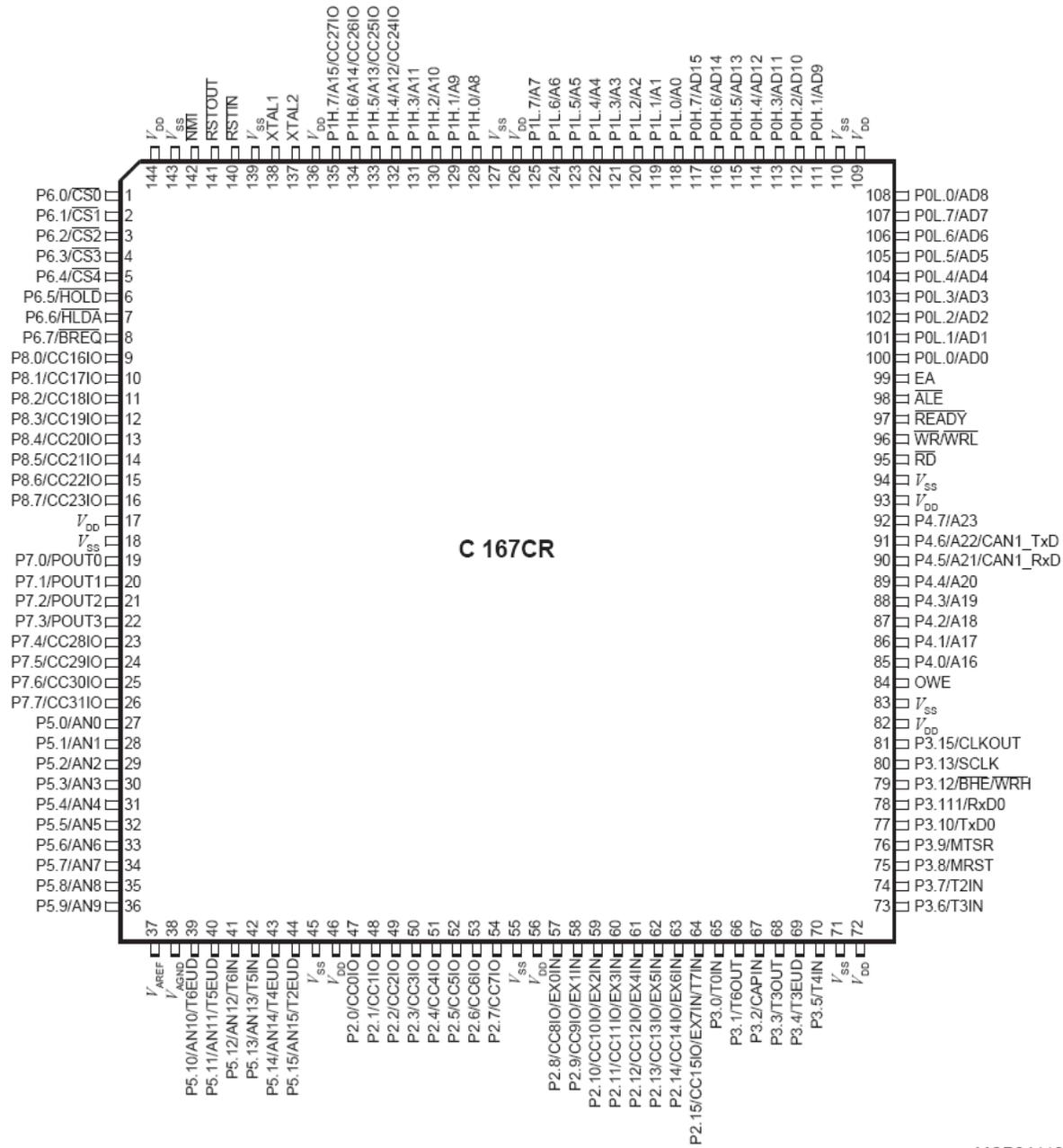
111 I/O-Leitungen organisiert in:

- 1 16-Bit I/O-Port (Port 2)**
- 2 2x8-Bit I/O-Ports (Port 0 (P0H, P0L), Port 1 (P1H, P1L))**
- 4 8-Bit I/O-Ports (Port 4, 6, 7, 8)**
- 1 15-Bit I/O-Port (Port 3)**
- 1 16-Bit Input Port (Port 5, Analoge Eingänge)**

Ports können meist in einer Vielzahl von Modi sehr flexibel den Anforderungen angepaßt werden.



C 167 Pins



Bedeutung für Port-Pins für Port 0:

- **General Purpose I/O-Pins**
- **8-Bit Datenbus**
- **16-Bit Adreßbus**
- **8-Bit Daten-/16-Bit Adreßbus (multiplexed mode)**
- **16-Bit Daten-/16-Bit Adreßbus (multiplexed mode)**

Spezialfunktionen der übrigen Ports (Alternate Functions) :

Port 1: Capture & Compare

Port 2: Fast External Interrupt Inputs

Port 3: Timer Input & Output, Serial synch. & async. Communication Channels

Port 4: CAN TxD & RxD

Port 5: Analog In

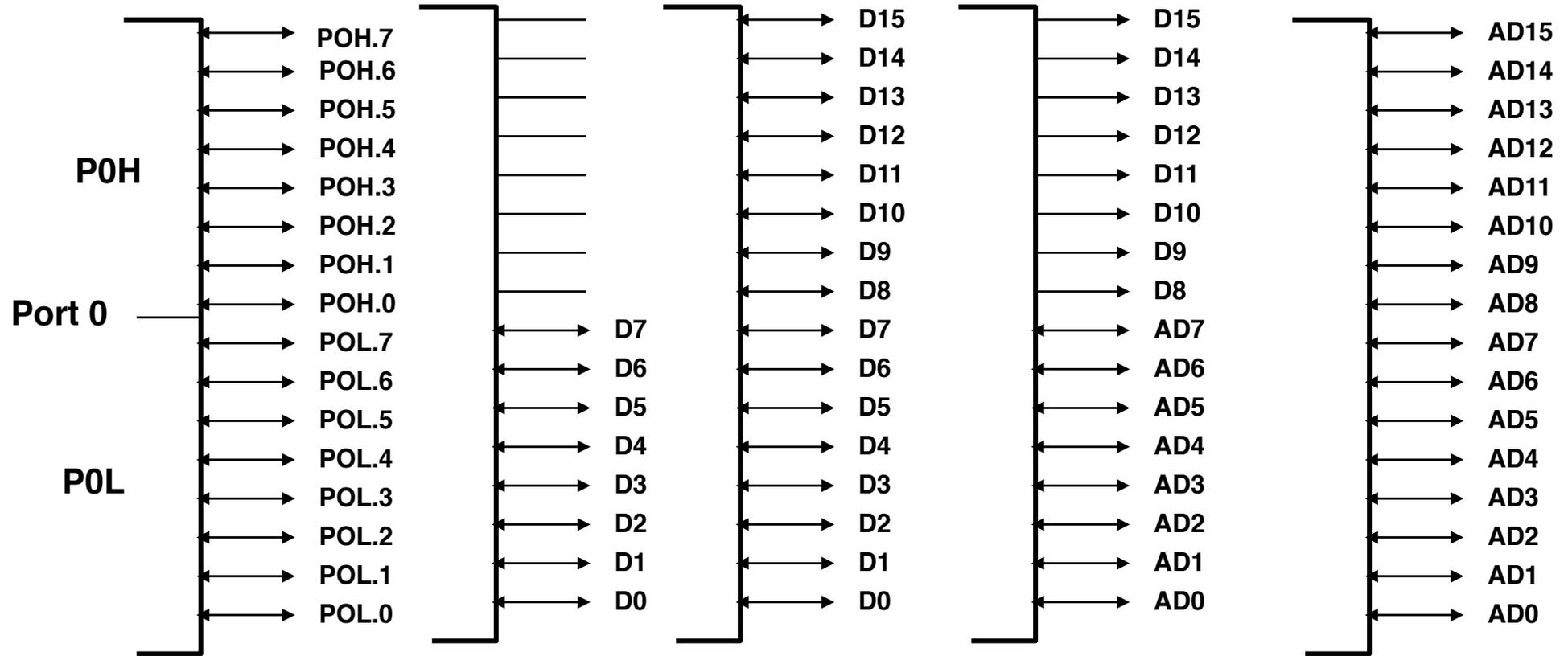
Port 6: Chip Select Lines for Memory Extensions

Port 7: PWM Channels, Capture & Compare

Port 8: Capture & Compare



Alternate Function →

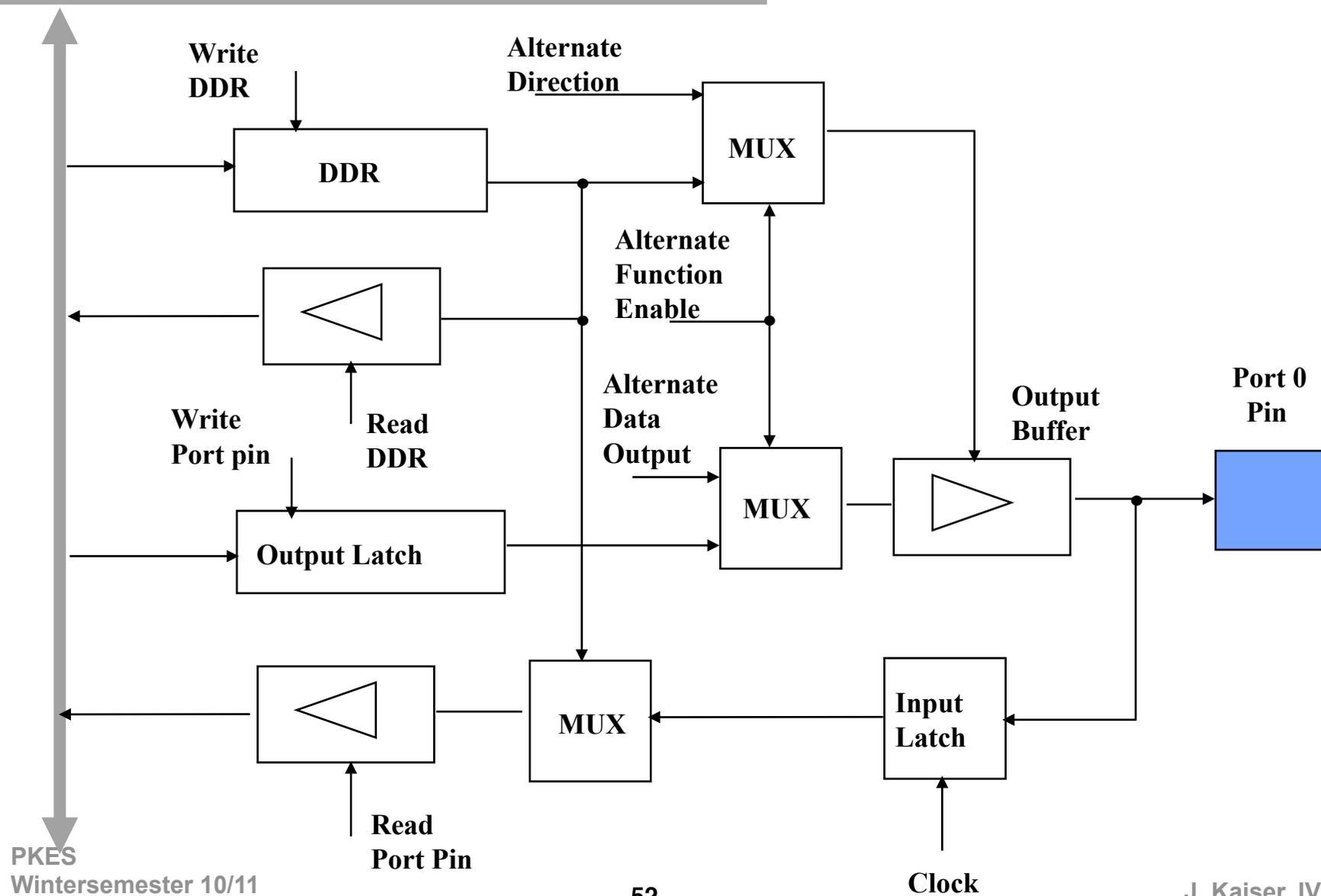


general purpose I/O

PKES Wintersemester 10/11



Beispiel für die Schaltung eines Port Pins

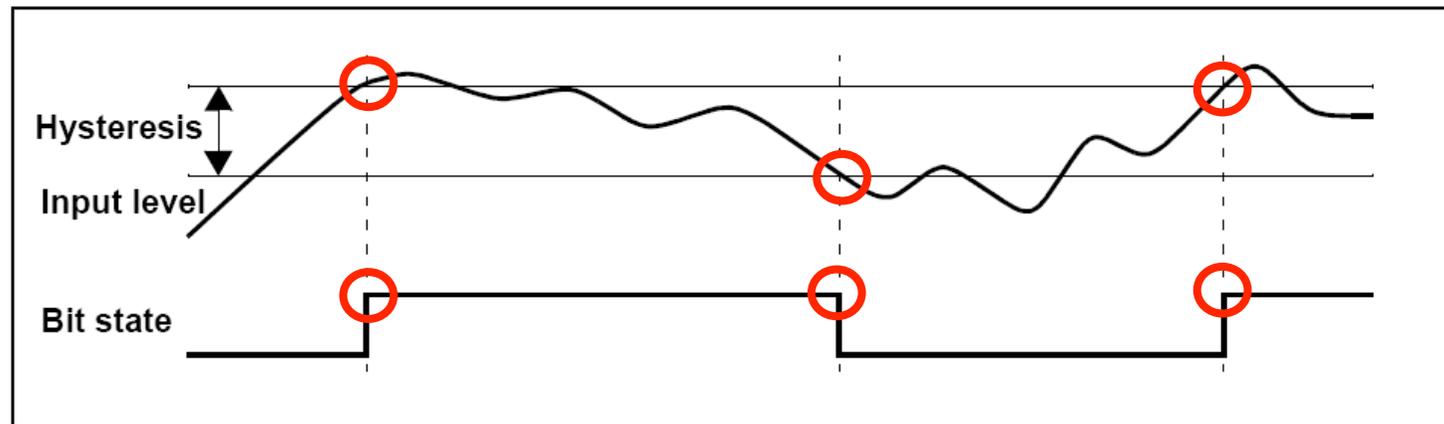


Festlegung der Eingangshysterese

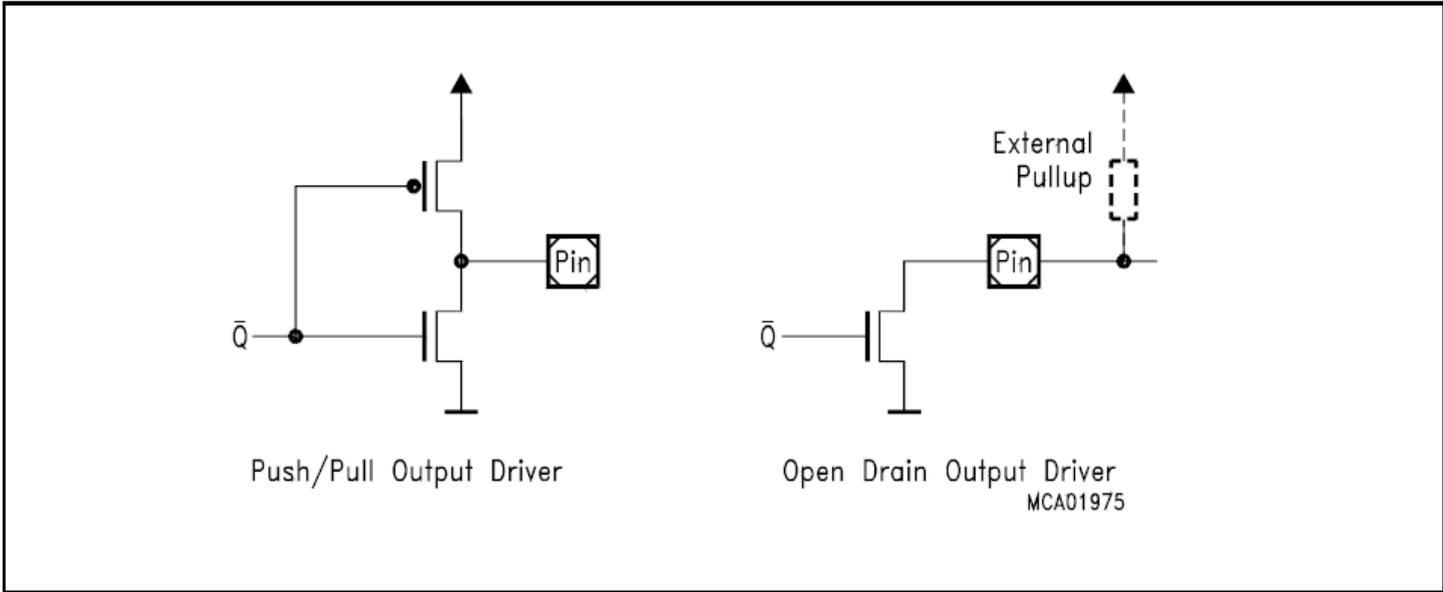
Bit	Function
PxLIN	Port x Low Byte Input Level Selection 0 : Pins Px.7...Px.0 switch on standard TTL input levels 1 : Pins Px.7...Px.0 switch on special threshold input levels
PxHIN	Port x High Byte Input Level Selection 0 : Pins Px.15...Px.8 switch on standard TTL input levels 1 : Pins Px.15...Px.8 switch on special threshold input levels

All options for individual direction and output mode control are available for each pin independent from the selected input threshold.

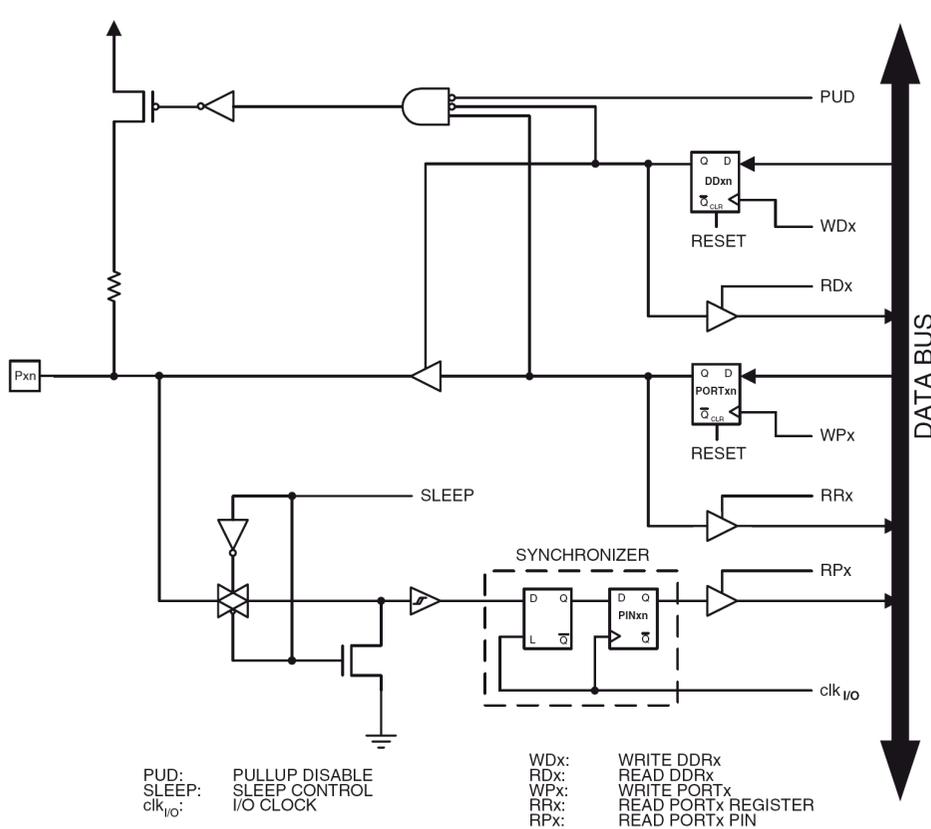
The input hysteresis provides stable inputs from noisy or slowly changing external signals.



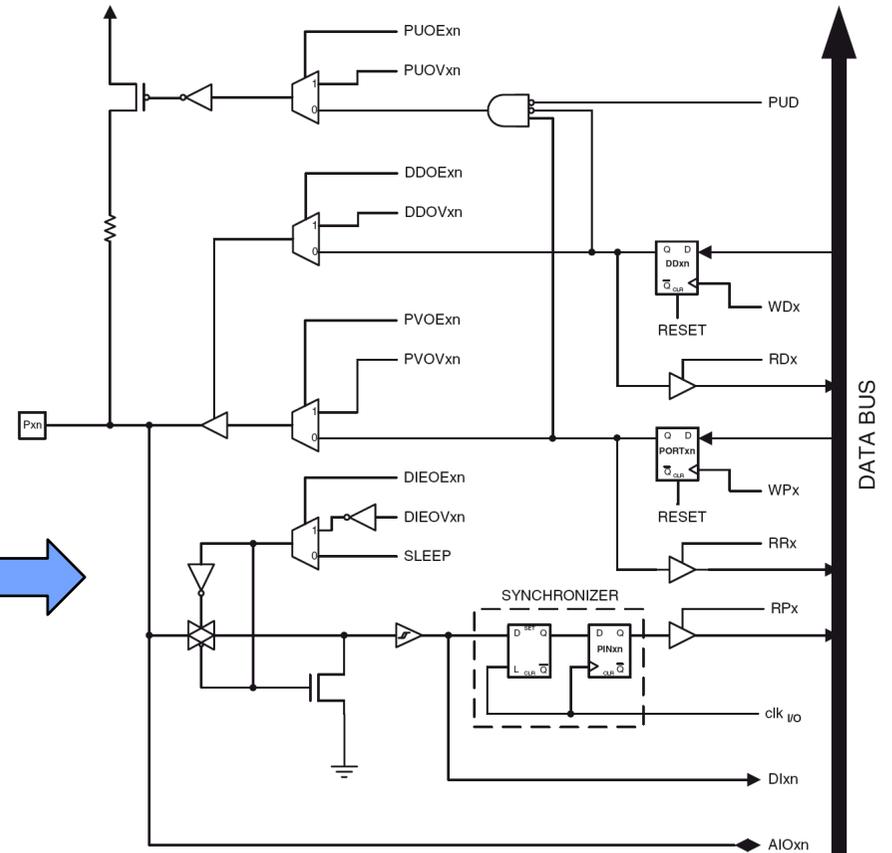
Bestimmung der Ausgangstreibercharakteristik



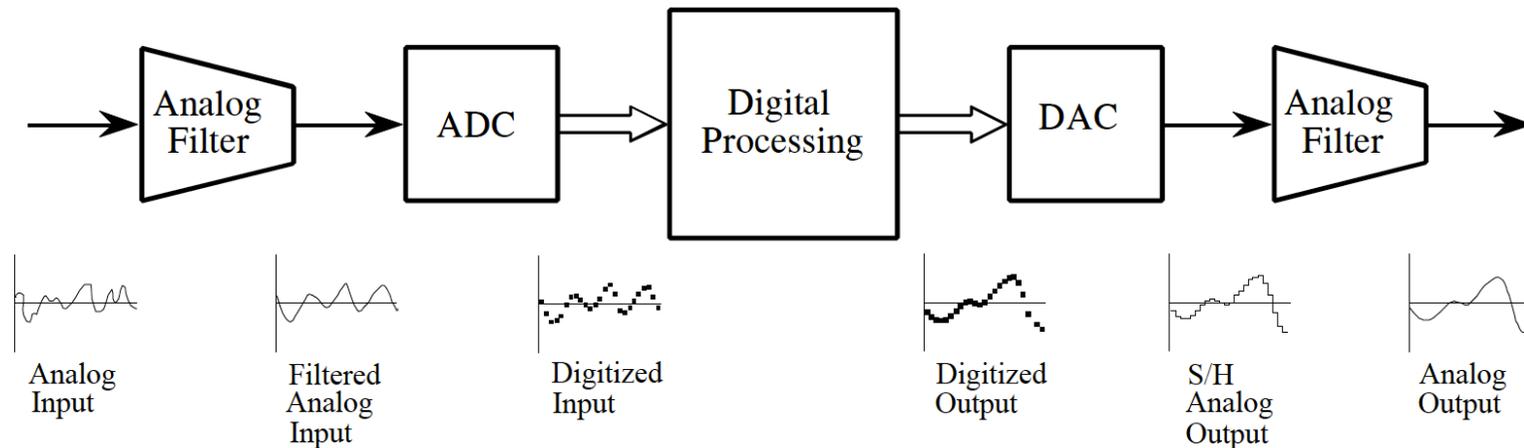
....Port Pins des AVR



digital Standard



Digitale Signalprozessoren



Steven W. Smith, Ph.D. "The Scientist and Engineer's Guide to Digital Signal Processing"

Web-Resource: www.dspguide.com/ (10.2010)



Digitale Signalprozessoren

Typische Funktionen:

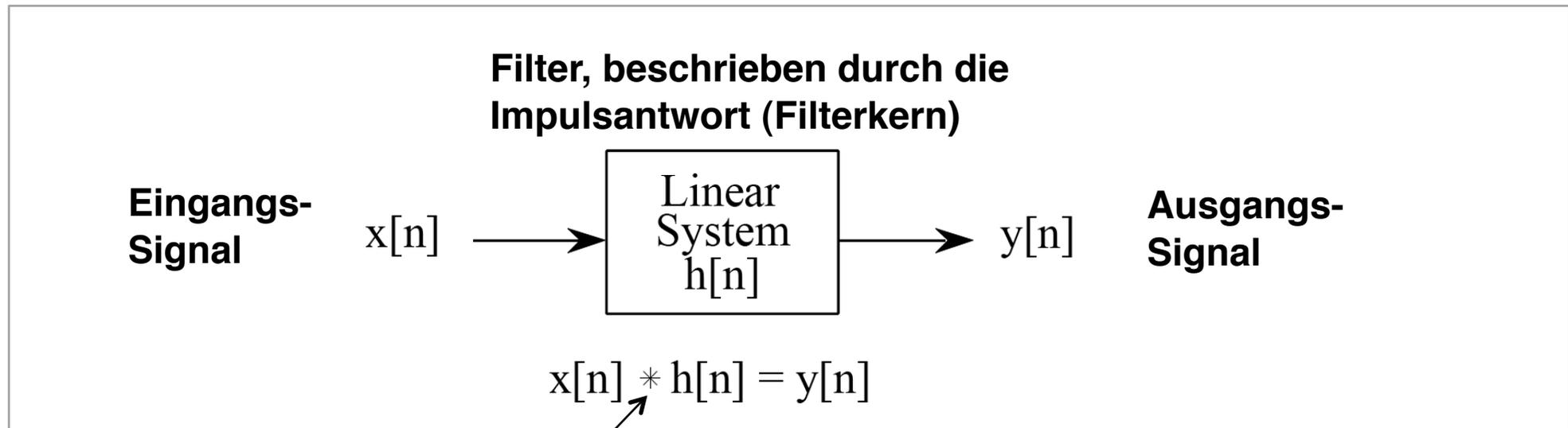
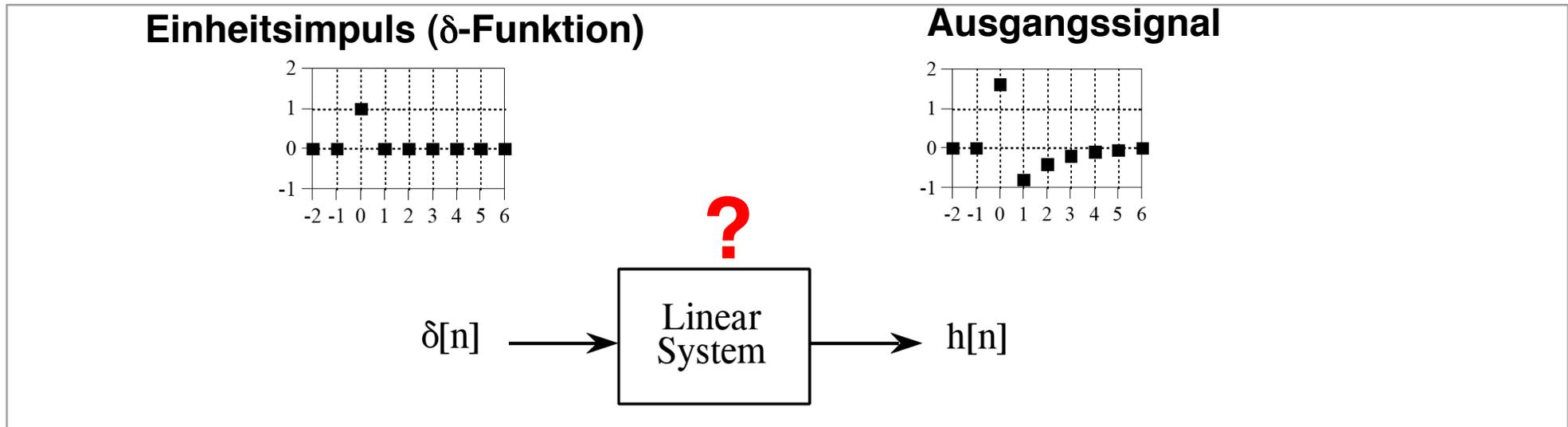
Digitale Filter
Spektralanalyse (Fourieranalyse)
Datenkompression
Signalmultiplexing

In den Anwendungsbereichen:

Radar
Sonar
Medizin: MRT, CT, Ultraschall,
EKG, EEG Analyse
Bildverarbeitung
Audio Verarbeitung



Digitale Filter



Eigenschaften eines linearen Systems

1. Homogenität

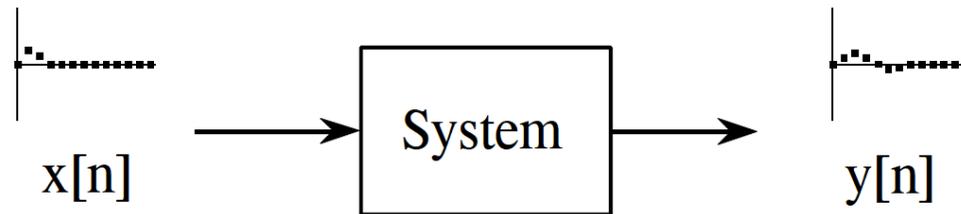
2. Additivität

3. Verschiebungsinvarianz



Homogeneity

IF



THEN

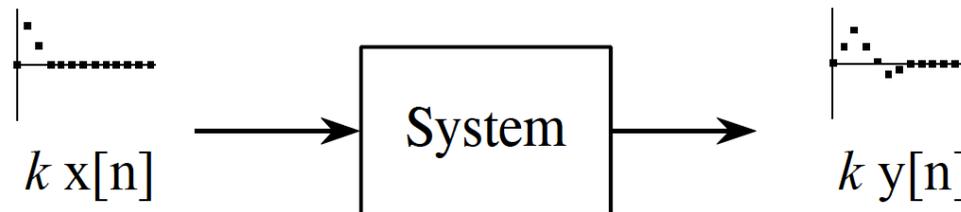


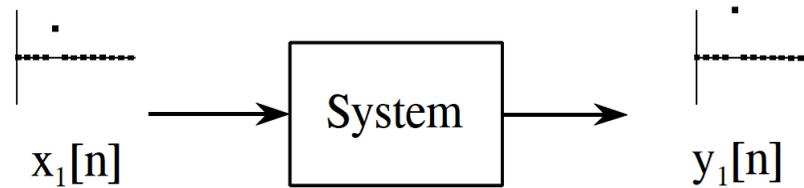
FIGURE 5-2

Definition of homogeneity. A system is said to be *homogeneous* if an amplitude change in the input results in an identical amplitude change in the output. That is, if $x[n]$ results in $y[n]$, then $kx[n]$ results in $ky[n]$, for any signal, $x[n]$, and any constant, k .

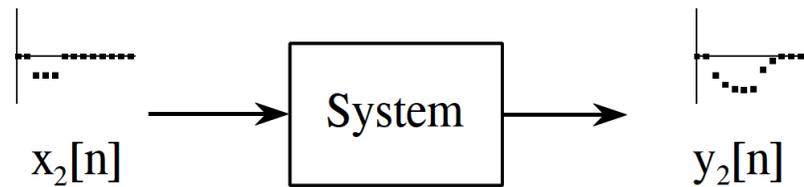


Additivity

IF



AND IF



THEN

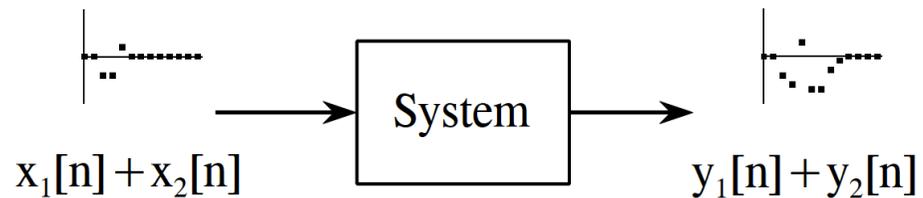


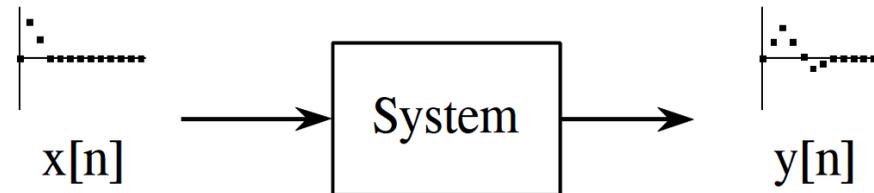
FIGURE 5-3

Definition of additivity. A system is said to be *additive* if added signals pass through it without interacting. Formally, if $x_1[n]$ results in $y_1[n]$, and if $x_2[n]$ results in $y_2[n]$, then $x_1[n] + x_2[n]$ results in $y_1[n] + y_2[n]$.



Shift Invariance

IF



THEN

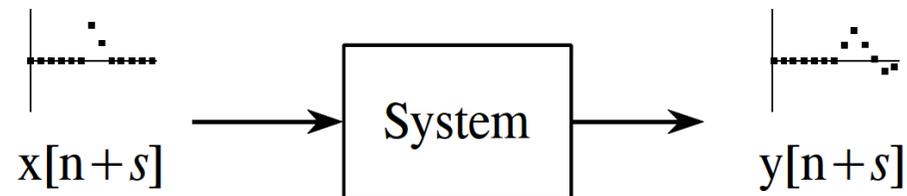


FIGURE 5-4

Definition of shift invariance. A system is said to be *shift invariant* if a shift in the input signal causes an identical shift in the output signal. In mathematical terms, if $x[n]$ produces $y[n]$, then $x[n+s]$ produces $y[n+s]$, for any signal, $x[n]$, and any constant, s .



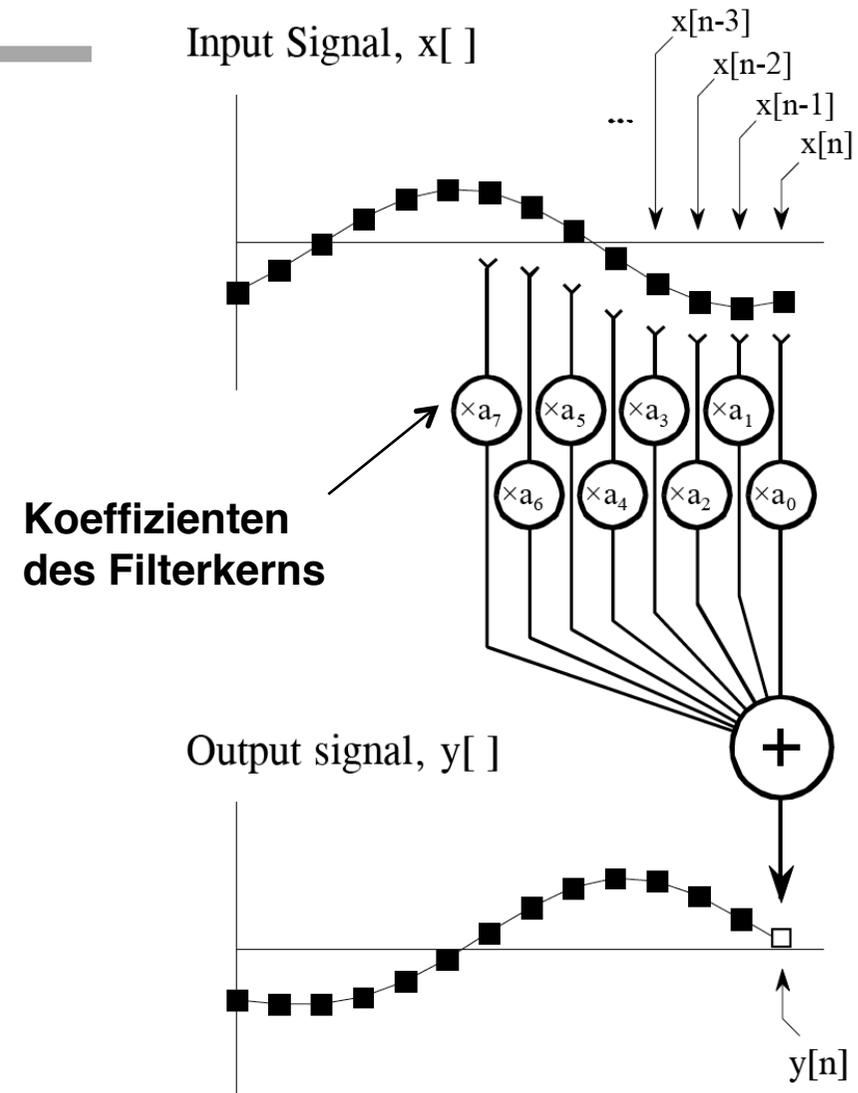
Digitale Filter

Die Faltung des Signals mit dem Filterkern wird beschrieben durch:

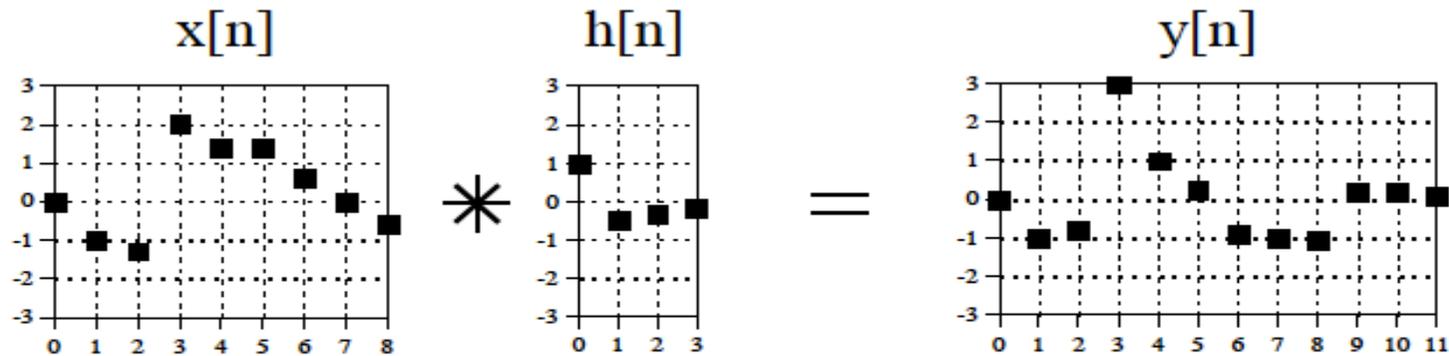
$$y[i] = \sum_{j=0}^{M-1} h[j] x[i-j]$$

Das Ausgangssignal kann als gewichtete Summe über einer Folge von Eingangssignalen betrachtet werden.

Aus: Steven W. Smith, Ph.D.: The Scientist and Engineer's Guide to Digital Signal Processing, <http://www.dspguide.com>

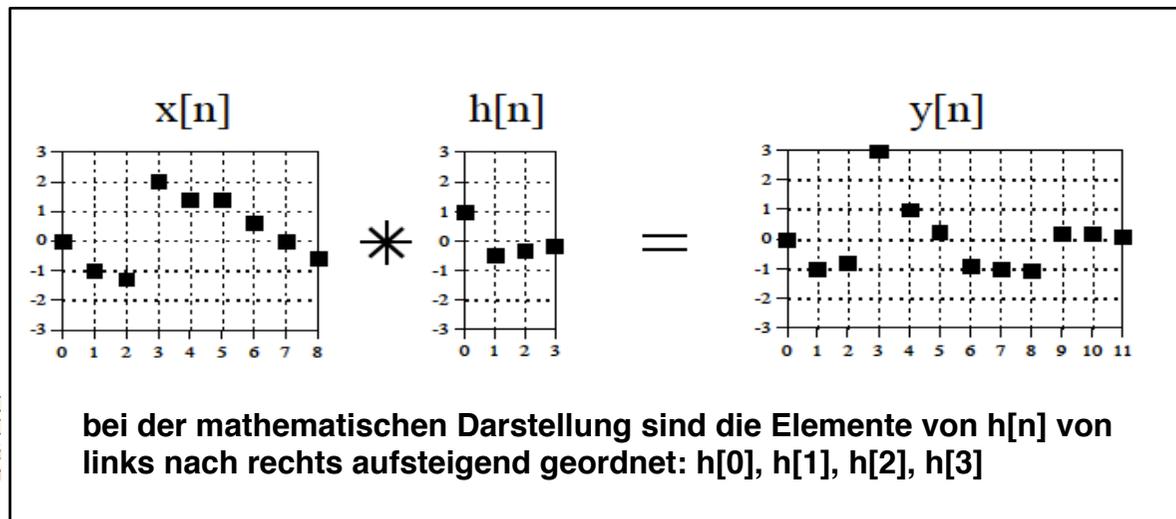
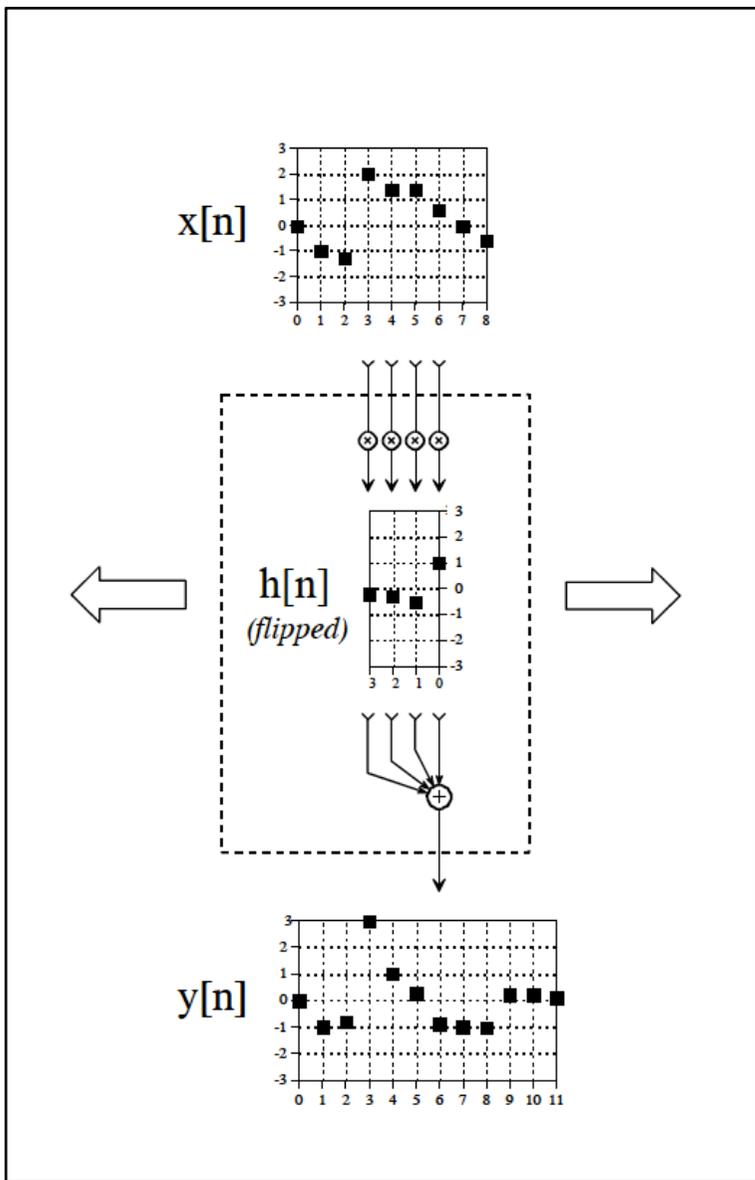


Digitale Filter



Beispiel der Faltung einer Signalfolge mit einem Filterkern

$$y[n] = x[n] * h[n] \quad \rightarrow \quad y[i] = \sum_{j=0}^{M-1} h[j] x[i-j]$$

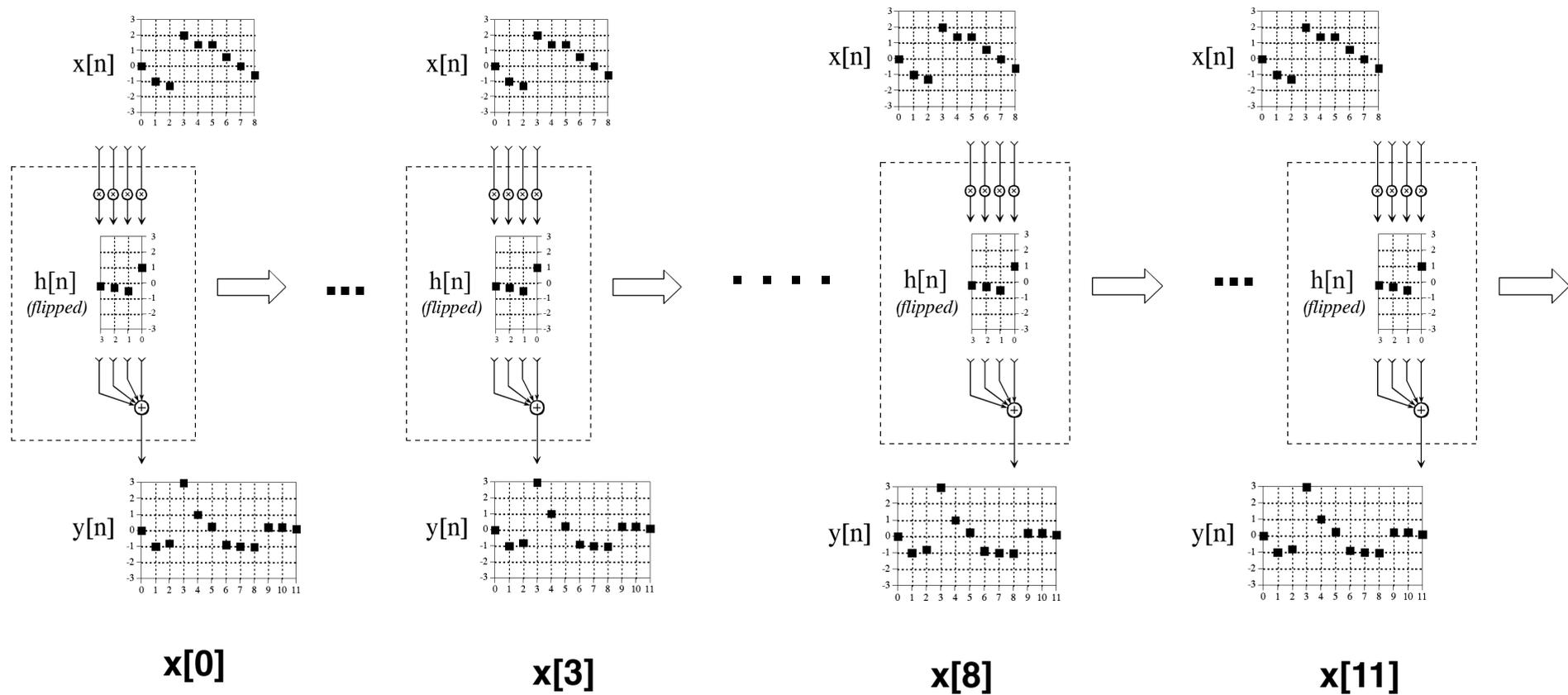


Bei der tatsächlichen Berechnung muss $h[0]$ zuerst angewandt werden. Deshalb wird die Impulse Response "geflippt"

$$y[i] = \sum_{j=0}^{M-1} h[j] x[i-j]$$

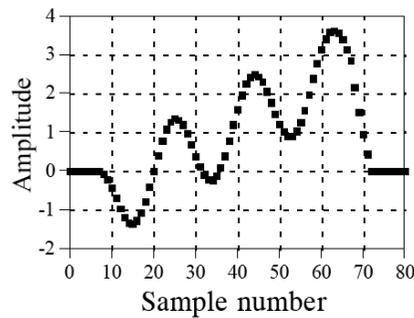


Digitale Filter



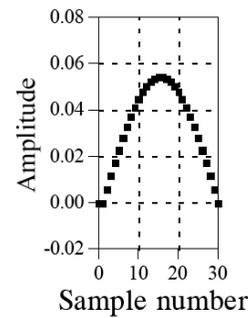
Digitale Filter

Tiefpass-Filter



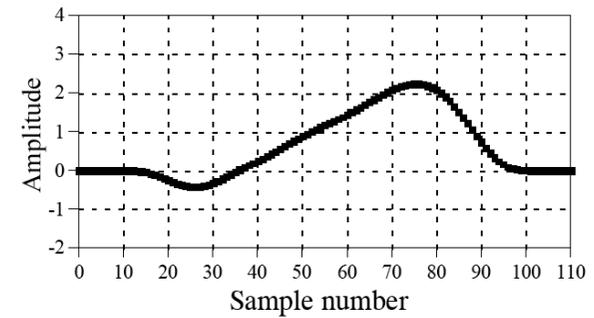
Eingangssignal

*



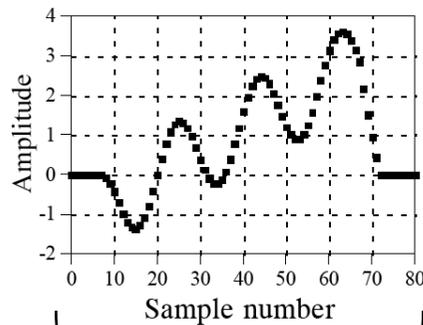
Impulsantwort

=



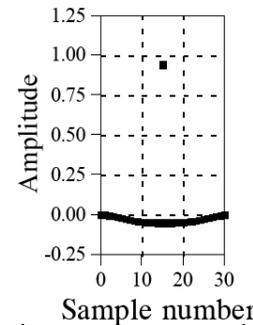
Ausgangssignal

Hochpass-Filter



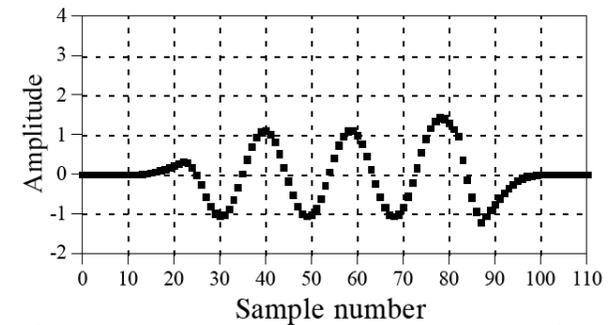
Eingangssignal

*



Impulsantwort

=



Ausgangssignal



Digitale Filter

Befehlsfolge Filter:

Äußere Schleife: Erzeugt Strom von Eingabedaten

1. ADC Interrupt Behandlung (lies ADC-Daten)
2. lies Datum in Eingaberingpuffer
3. aktualisiere Eingabepufferadresse
4. initialisiere Arbeitsregister mit "0"

Innere Schleife: Führt die Faltung mit Filterkern durch

1. lies Koeffizient vom (Ring-)Koeffizienten-Ringpuffer"
 2. aktualisiere Koeffizientenpufferadresse
 3. lies Wert vom Eingaberingpuffer
 4. aktualisiere Eingabepufferadresse
 5. multipliziere Eingabewert mit Koeffizient
 6. addiere Produkt zum Inhalt des Arbeitsregisters
 7. führe Schleife so oft aus bis alle Eingabewerte und Koeffizienten bearbeitet wurden
5. speichere Wert im Arbeitsregister im Speicher oder transferiere es zum DAC
 6. führe äußere Schleife aus bis Abbruchbedingung gegeben wird



Digitale Filter

```
// convolution using output side algorithm

int X[80];      // input signal, 80 points
int H[30];     // impulse response, 30 points
int Y[110];    // output signal, 110 points

load_input_signal_and_impulse_response(X, H);

for (int i=0;i<110;++i) {
    Y[i]=0;
    for (int j=0;j<30;++j) {
        if (((i-j) < 0) || ((i-j)>=80)) continue;
        Y[i]=Y[i] + H[j] * X[i-j];
    }
}

store_output(Y);
```



Digitale Signalprozessoren

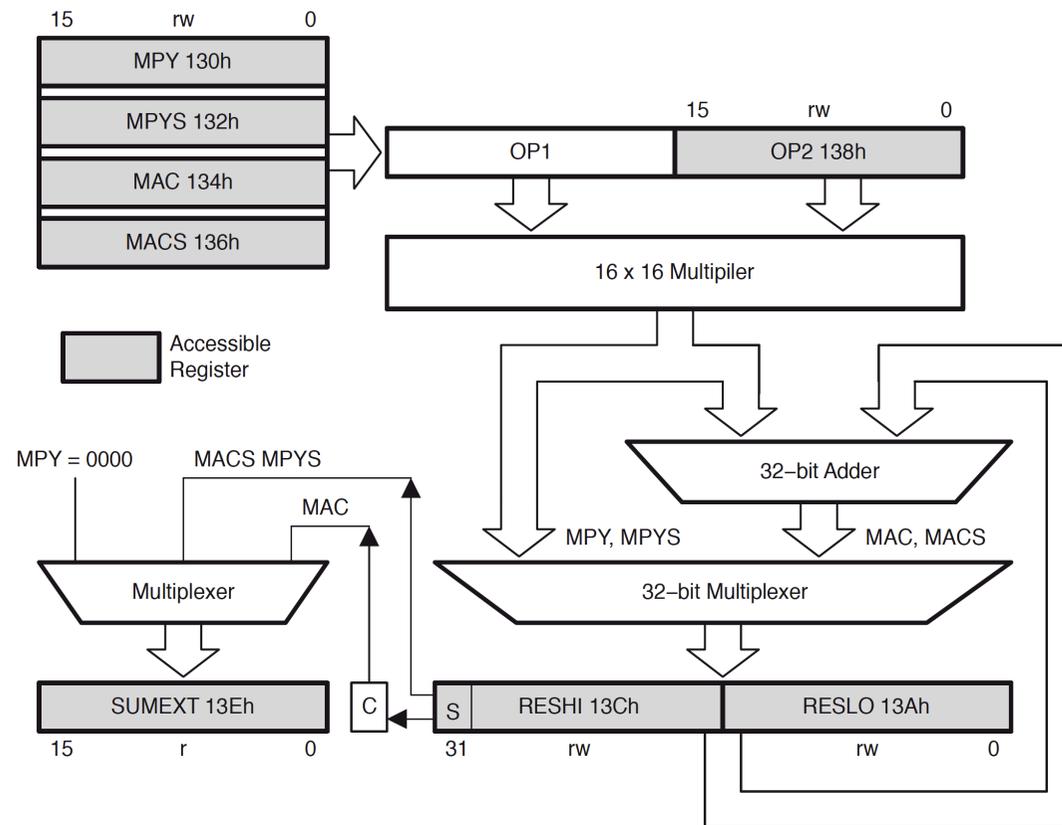
DSPs haben meist Optimierungen in folgenden Bereichen:

- Spezielle synchrone, serielle Schnittstellen für die Ein- und Ausgabe der digitalen Signale
- MAC-Befehle für die gleichzeitige Multiplikation und Addition in einem Maschinenbefehl.
- Adressgeneratoren für die Implementierung von Schleifen und Ringpufferstrukturen ohne softwareseitigen Overhead.
- Implementierung des Prozessors ausschließlich in Harvard-Architektur.
- Existenz eines dedizierten Hardware-Stacks.
- Schnelle Ausführung von Schleifen
- Mehrmaligen Zugriff auf den Speicher in einem Zyklus

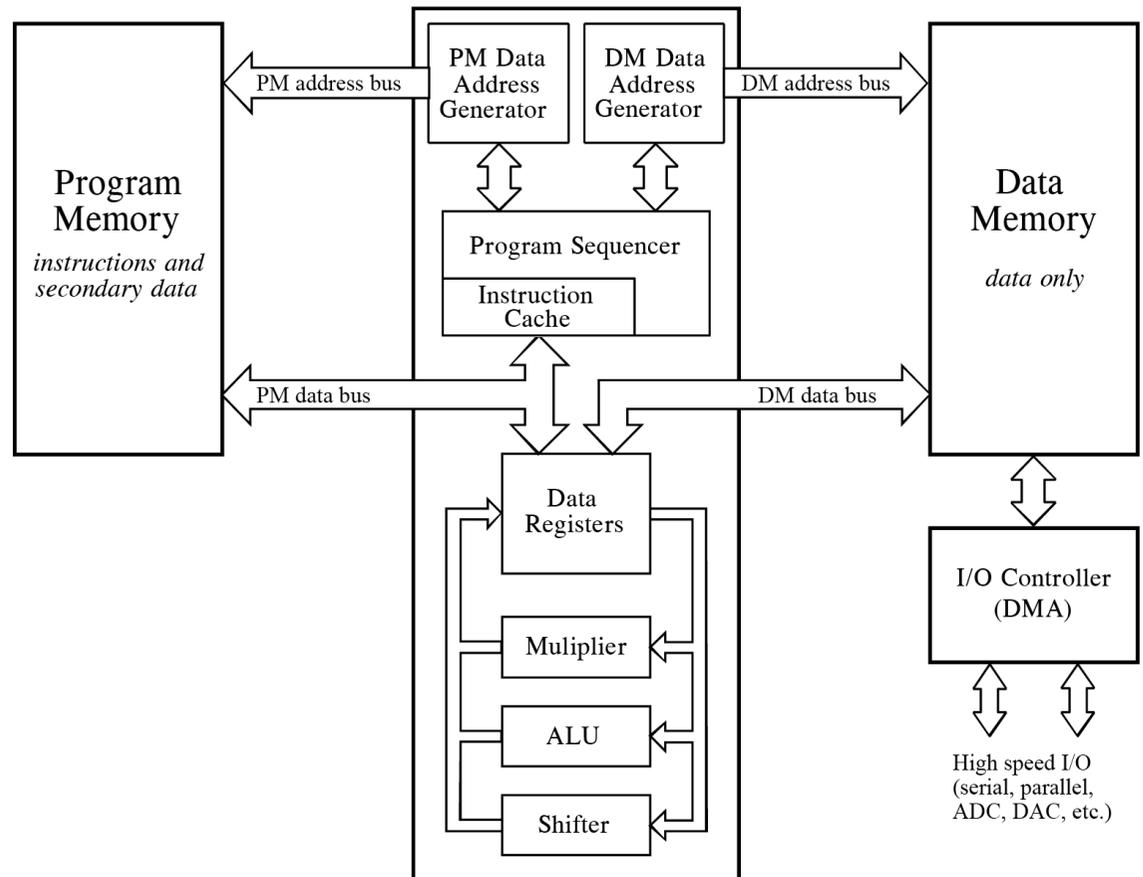


Digitale Signalprozessoren

Multiplikationseinheit des MSP430 (Texas Instruments)



Digitale Signalprozessoren



Vereinfachtes Blockschaltbild des Shark DSPs (Analog Devices)

Aus: Steven W. Smith, Ph.D.:
The Scientist and Engineer's Guide
to Digital Signal Processing,
<http://www.dspguide.com>



Wesentliche Punkte:

Mikro-Controller sind als eingebettete Komponenten konzipiert, die eine CPU mit einer großen Palette interner Funktionsmodule einschließlich Speicherkomponenten aufweisen und deshalb eine minimale Anzahl externer Komponenten benötigen.

Mikro-Controller weisen umfangreiche Möglichkeiten der Konfiguration für Speicher- und Systemkomponenten auf. Sie werden mit einer Standardkonfiguration ausgeliefert.

Im Gegensatz zu Mikro-Prozessoren, die eine Speicherschnittstelle zur Peripherie realisieren, stellen Mikro-Controller konfigurierbare Ports zur Verfügung, die spezielle Ein- und Ausgabe-Funktionen unterstützen.

