MICRO-CONTROLLER

Was ist ein Micro-Controller ?

Controller = Steuerung

Ein Controller wird zur Steuerung eines physischen Prozesses eingesetzt.

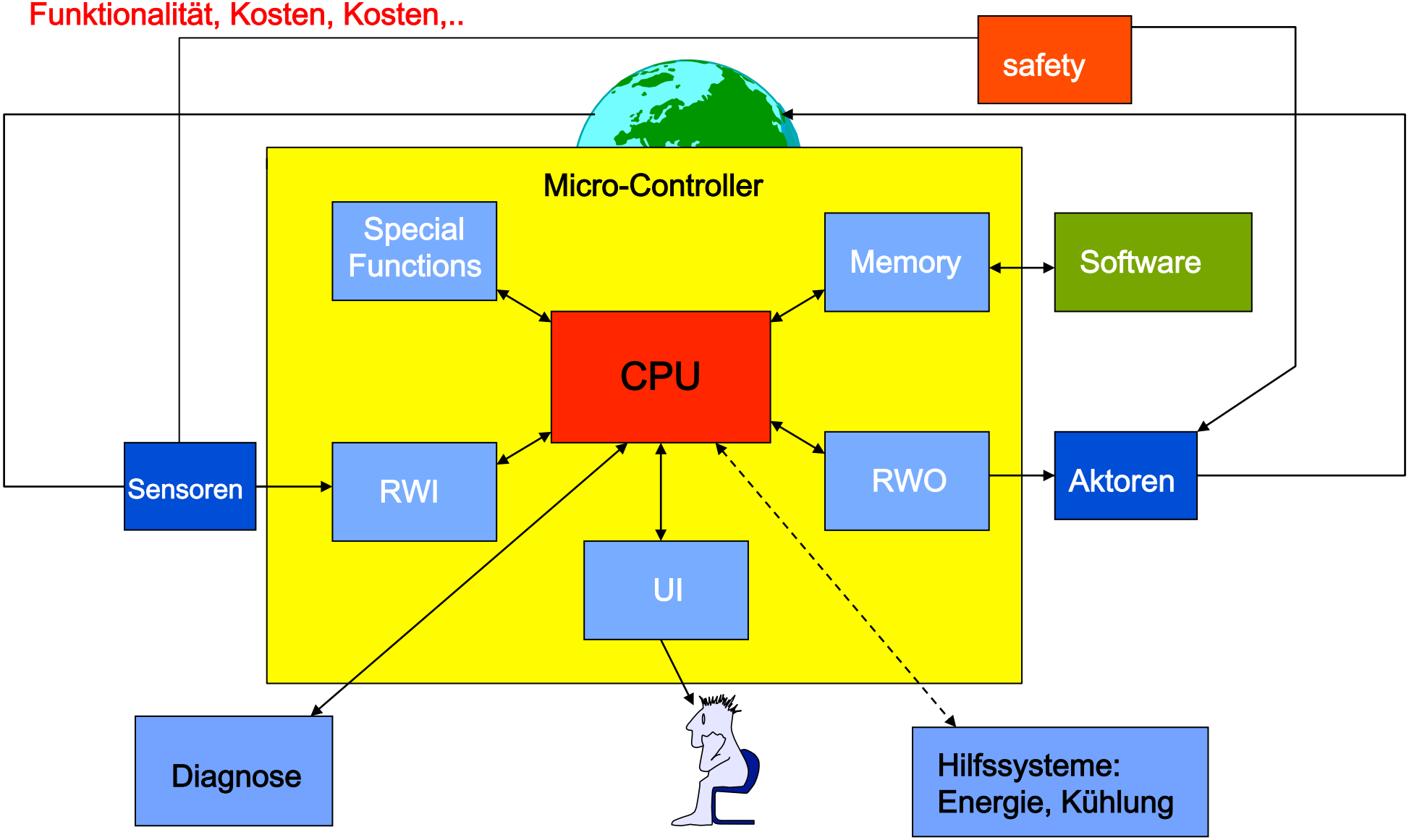
Die Realisierung eines Controllers kann auf viele verschiedene Arten erfolgen, z.B. ein Schaltschrank mit Relais, ein analoger Regelkreis oder eine speziell aufgebaute digitale Logikschaltung.

Speicherprogrammierbare Steuerung (SPS)
Programmable Logic Controller (PLC)

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

Systemsicht eines Architekten für Kontrollsysteme:

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

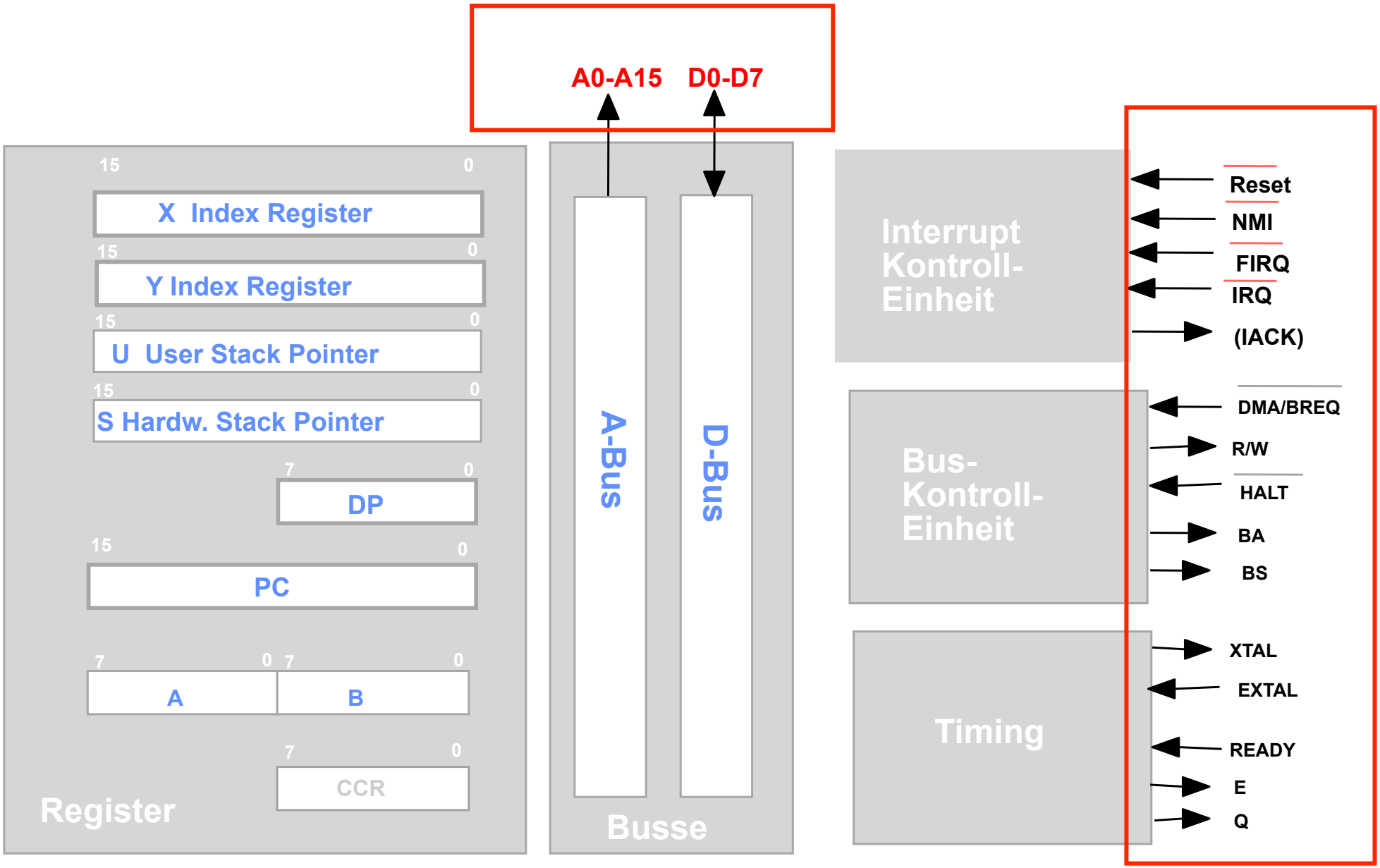


Der CPU-Kern

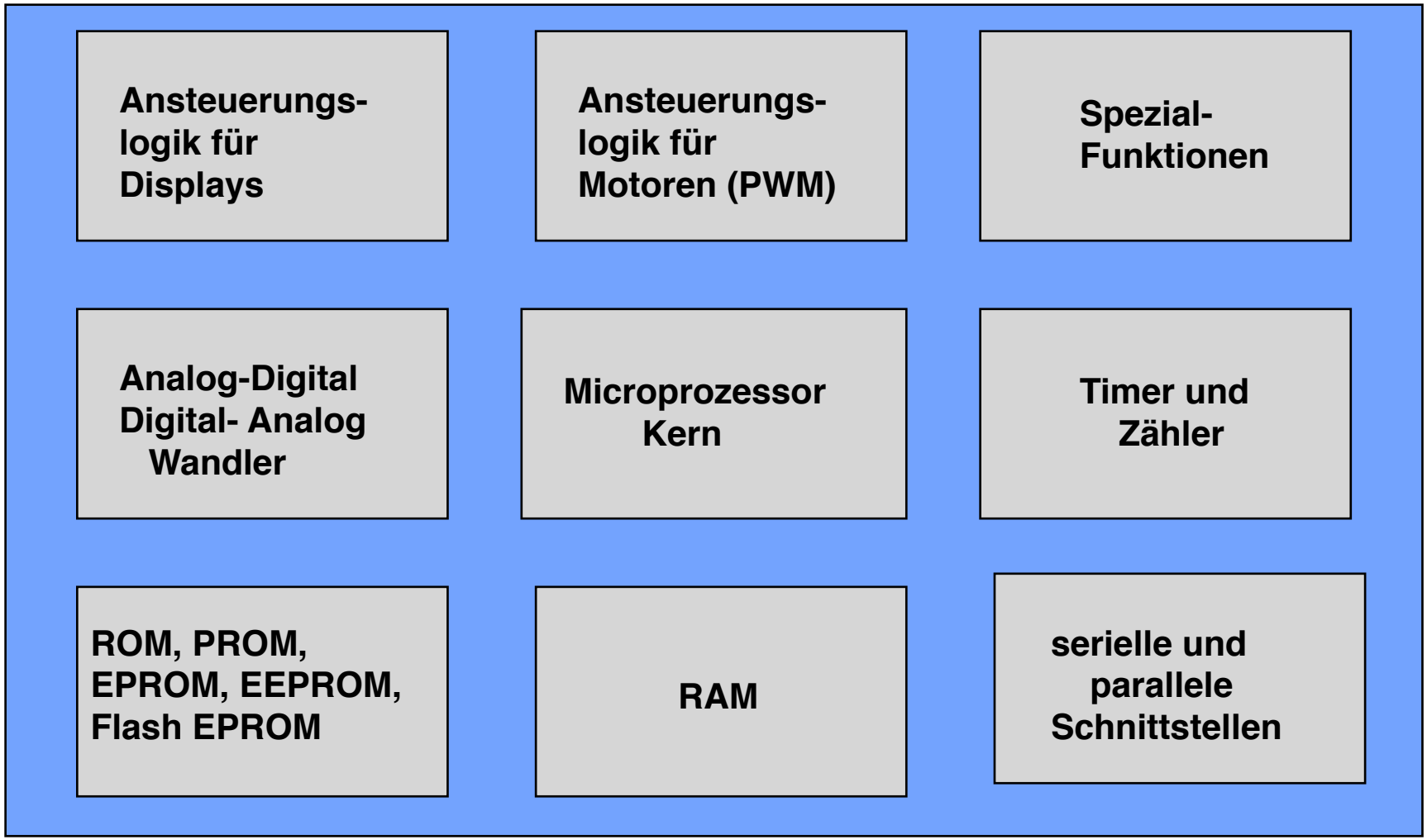
Architekturen:

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



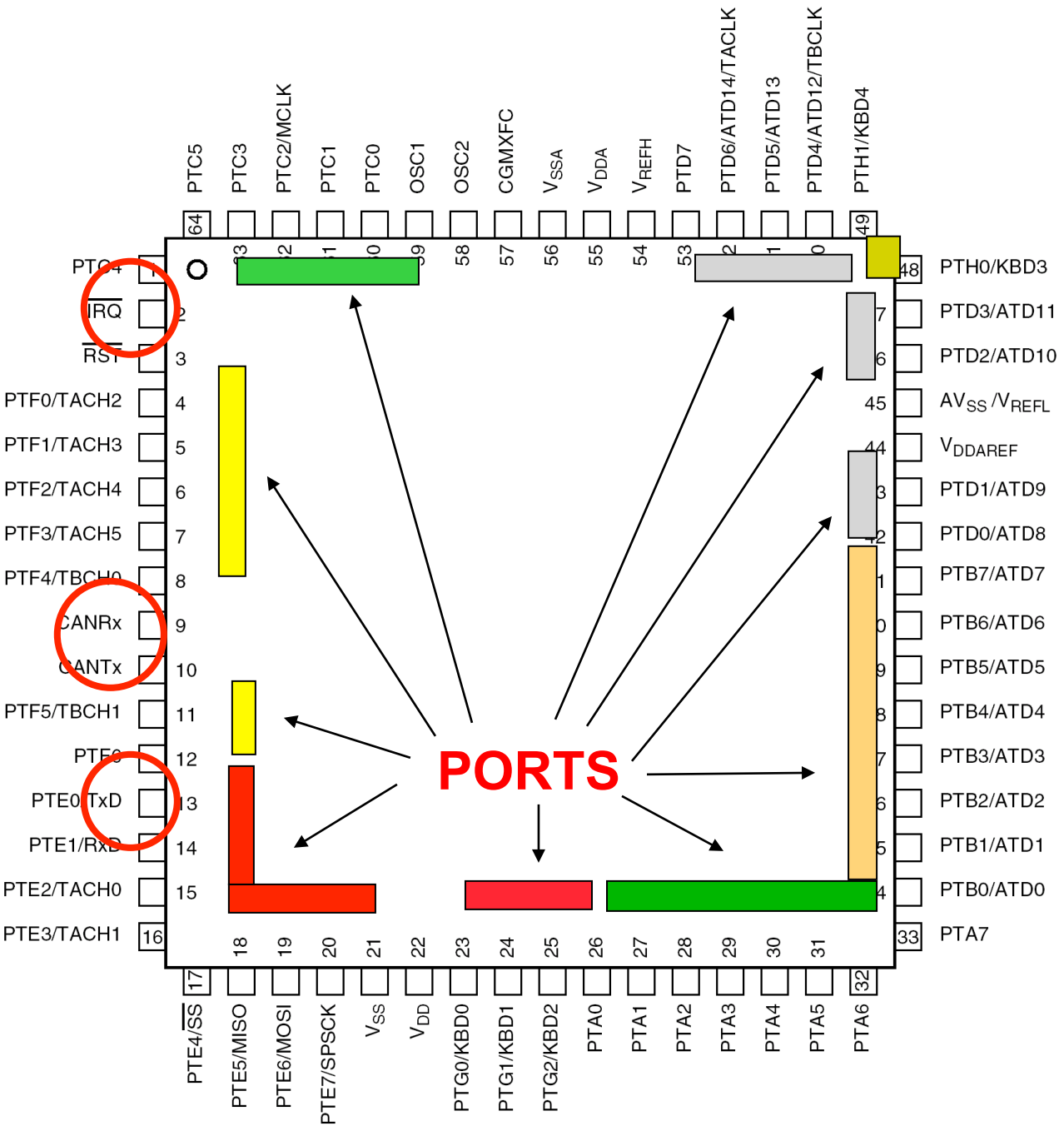
Micro-Controller

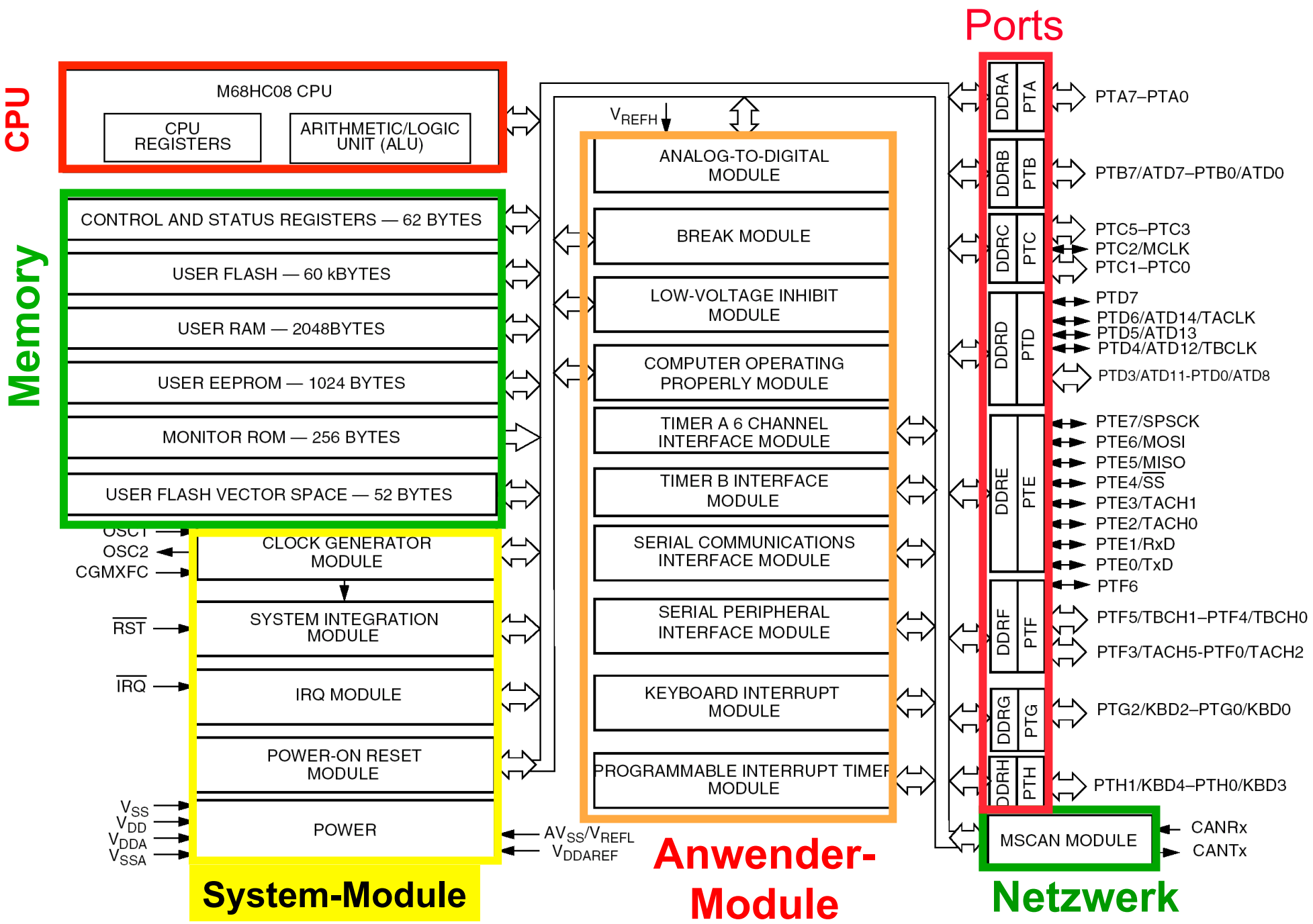


Schnittstelle zu einem Micro-Controller:

PORTS

Beispiel:
Motorola
MC 68HC908AZ60A

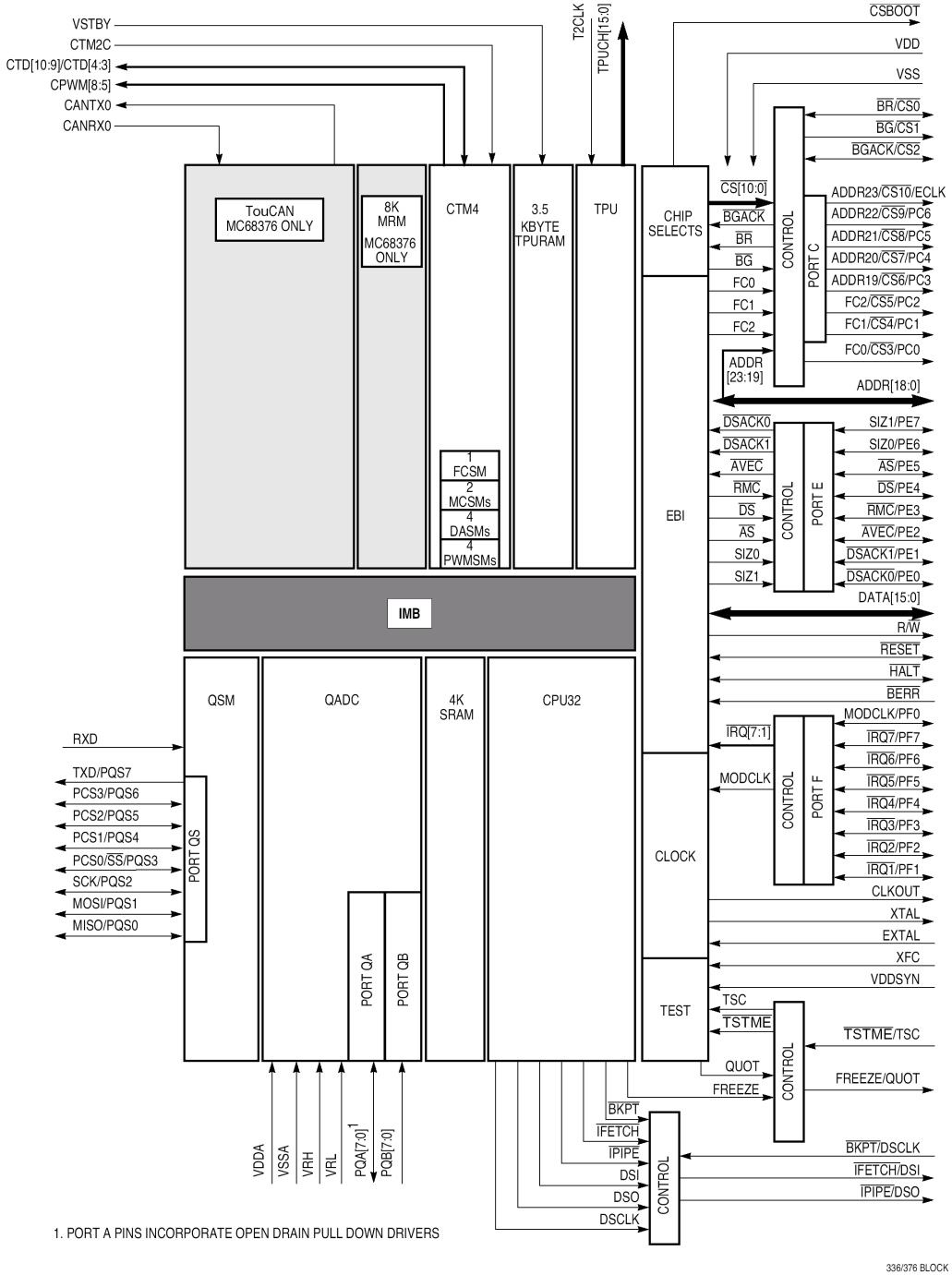



CPU
M68HC08 CPU
CPU REGISTERS
ARITHMETIC/LOGIC UNIT (ALU)
Memory
CONTROL AND STATUS REGISTERS — 62 BYTES
USER FLASH — 60 kBYTES
USER RAM — 2048BYTES
USER EEPROM — 1024 BYTES
MONITOR ROM — 256 BYTES
USER FLASH VECTOR SPACE — 52 BYTES
OSC1
OSC2
CGMXFC
CLOCK GENERATOR MODULE
SYSTEM INTEGRATION MODULE
RST
IRQ
IRQ MODULE
POWER-ON RESET MODULE
VSS
VDD
VDDA
VSSA
POWER
System-Module
VREFH
ANALOG-TO-DIGITAL MODULE
BREAK MODULE
LOW-VOLTAGE INHIBIT MODULE
COMPUTER OPERATING PROPERLY MODULE
TIMER A 6 CHANNEL INTERFACE MODULE
TIMER B INTERFACE MODULE
SERIAL COMMUNICATIONS INTERFACE MODULE
SERIAL PERIPHERAL INTERFACE MODULE
KEYBOARD INTERRUPT MODULE
PROGRAMMABLE INTERRUPT TIMER MODULE
AVSS/VREFL
VDDAREF
Anwender-Module
Ports
DDRA
PTA
PTA7–PTA0
DDRB
PTB
PTB7/ATD7–PTB0/ATD0
DDRC
PTC
PTC5–PTC3
PTC2/MCLK
PTC1–PTC0
DDRD
PTD
PTD7
PTD6/ATD14/TACLK
PTD5/ATD13
PTD4/ATD12/TBCLK
PTD3/ATD11-PTD0/ATD8
DDRE
PTE
PTE7/SPSCK
PTE6/MOSI
PTE5/MISO
PTE4/SS
PTE3/TACH1
PTE2/TACH0
PTE1/RxD
PTE0/TxD
DDRF
PTF
PTF6
PTF5/TBCH1–PTF4/TBCH0
PTF3/TACH5-PTF0/TACH2
DDRG
PTG
PTG2/KBD2–PTG0/KBD0
DDRH
PTH
PTH1/KBD4–PTH0/KBD3
MSCAN MODULE
CANRx
CANTx
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

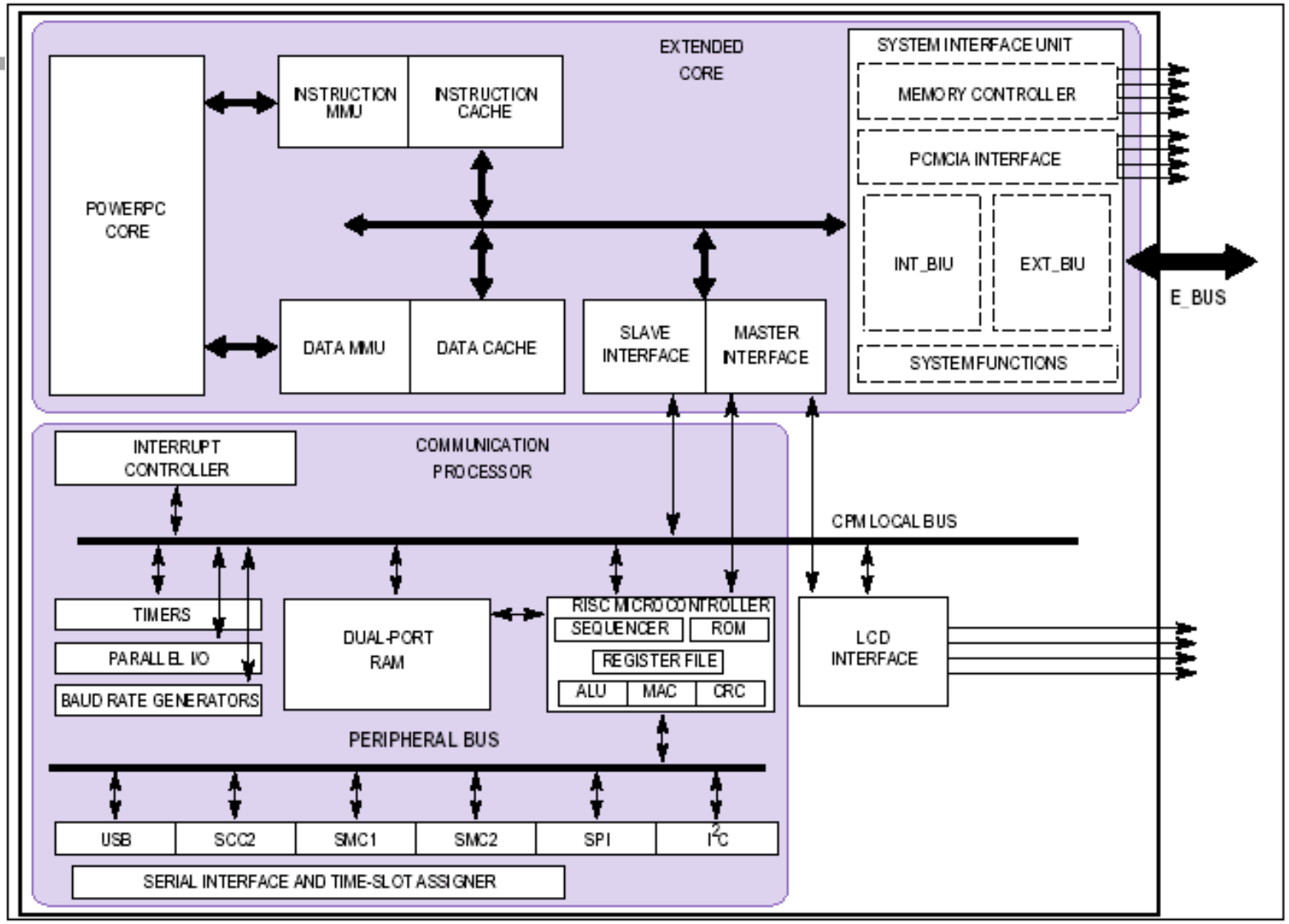
PIN-Count, Preis, Störunempfindlichkeit, Anpaßbarkeit



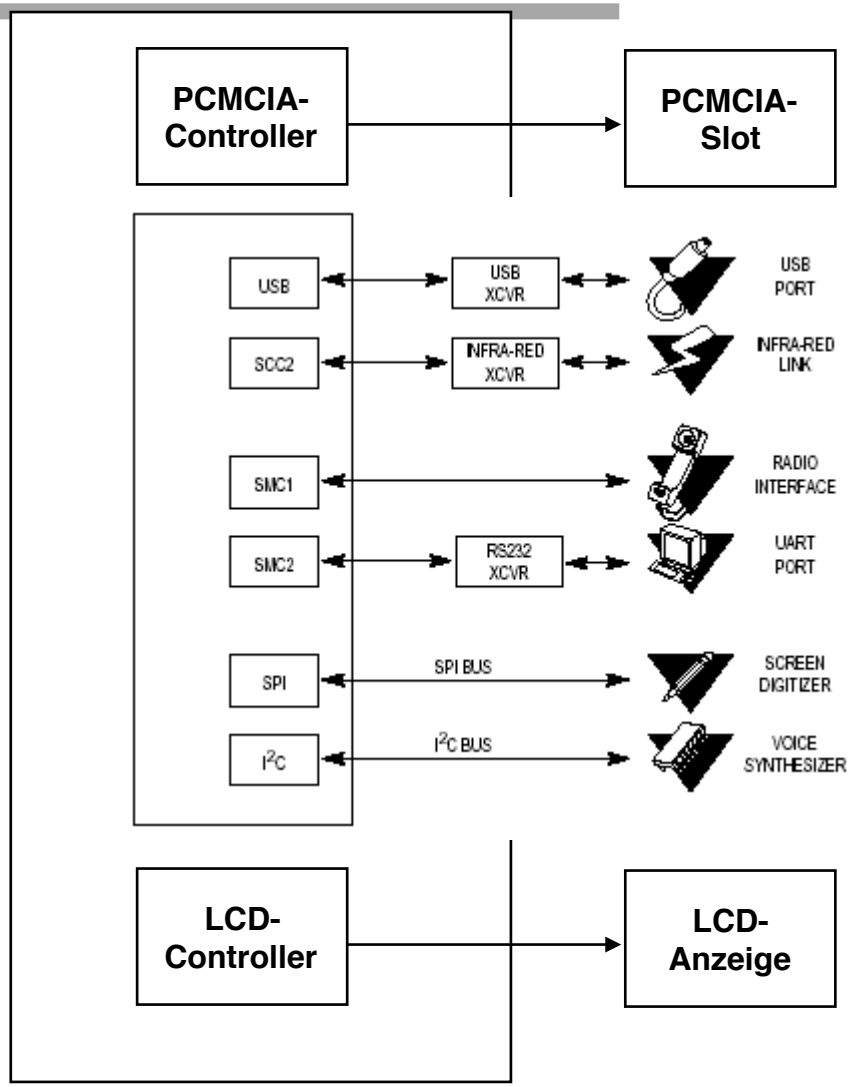
Blockschaltbild: MC 68376

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

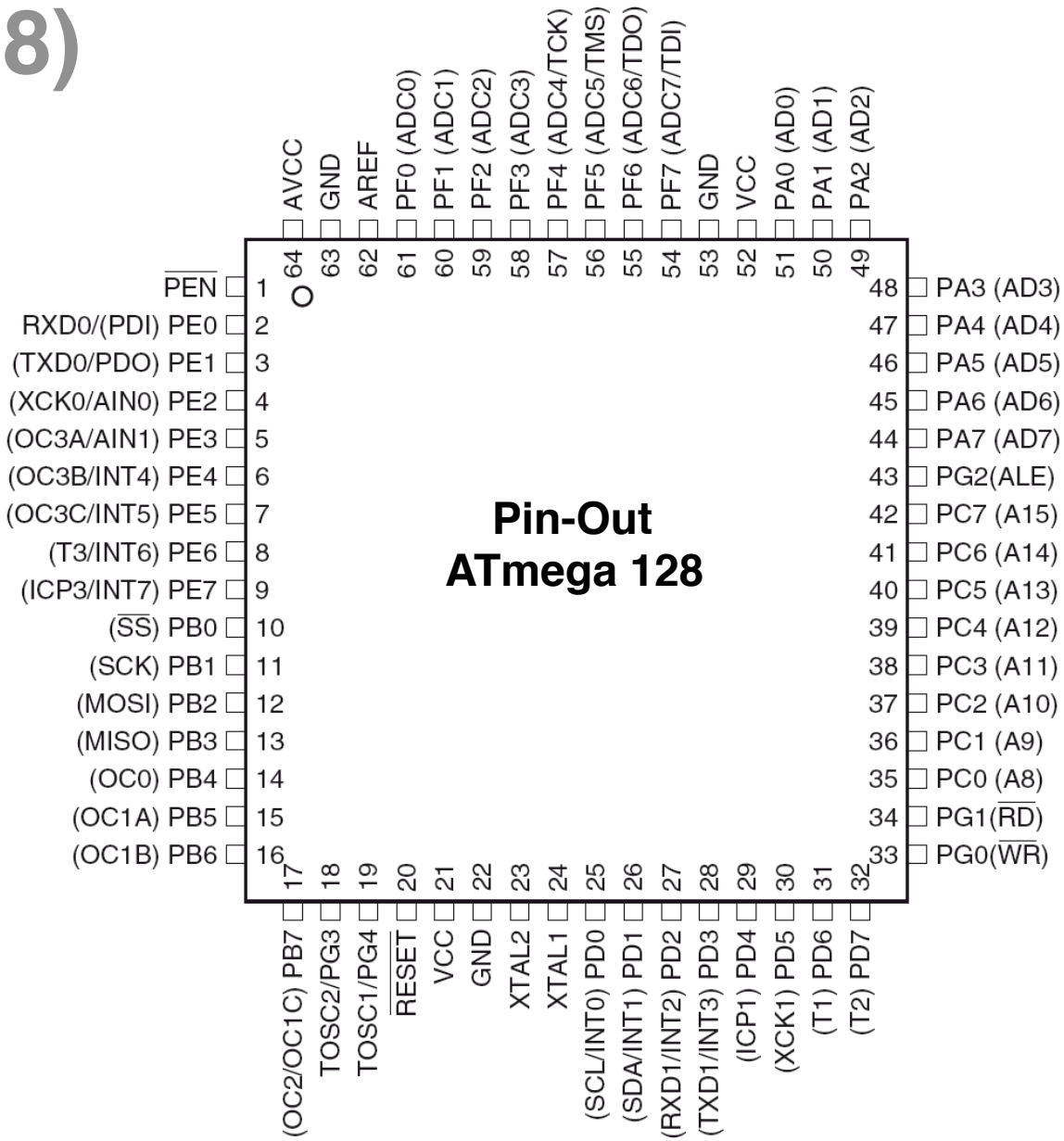


AVR (ATmega128)

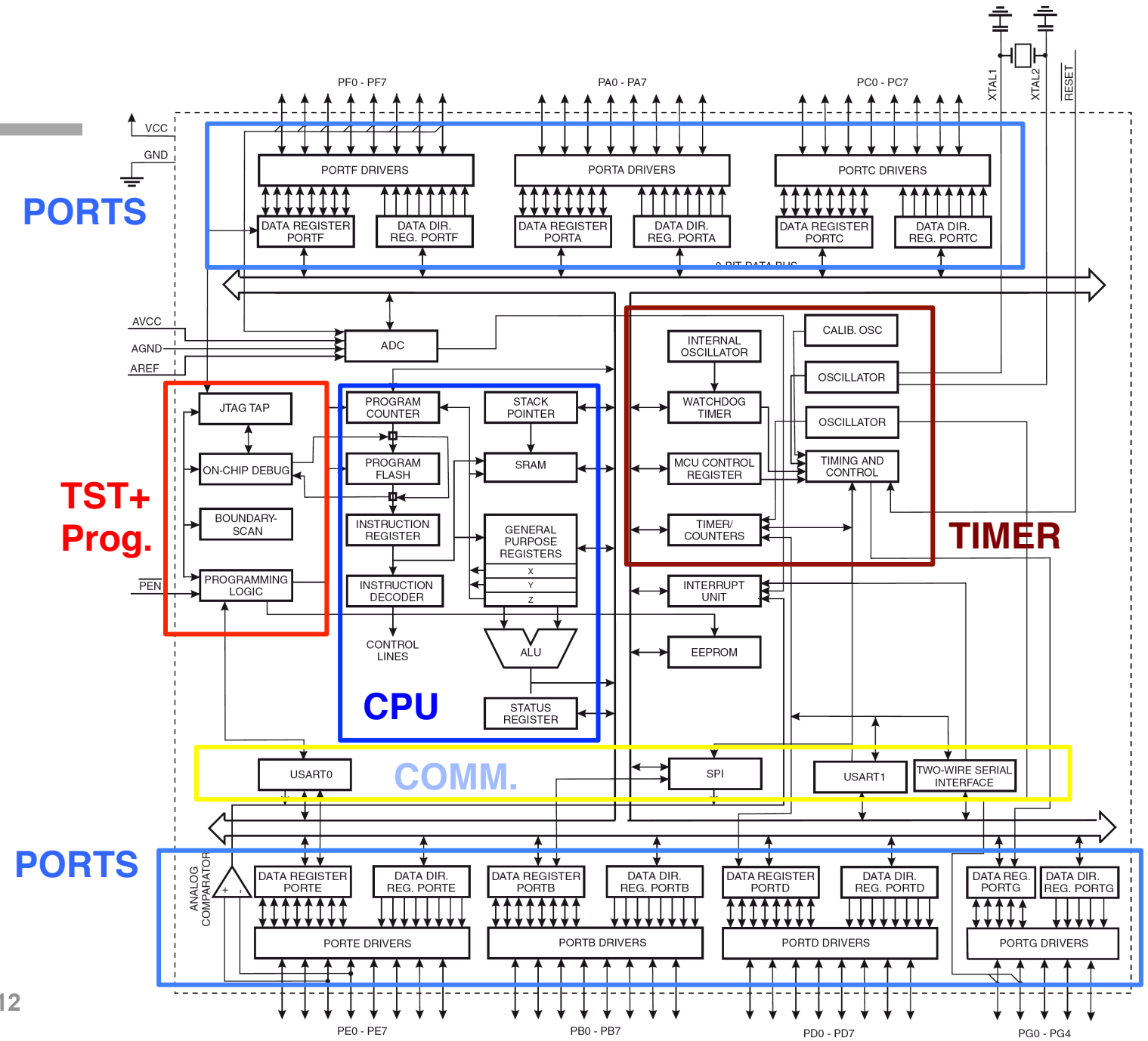
ATMEL

AVR 8-bit Microcontroller with 128K Bytes In-System Programmable Flash
ATmega128
ATmega128L

Data Sheet:
Rev. 2467M–AVR–11/04



ATmega 128 Block Schaltbild

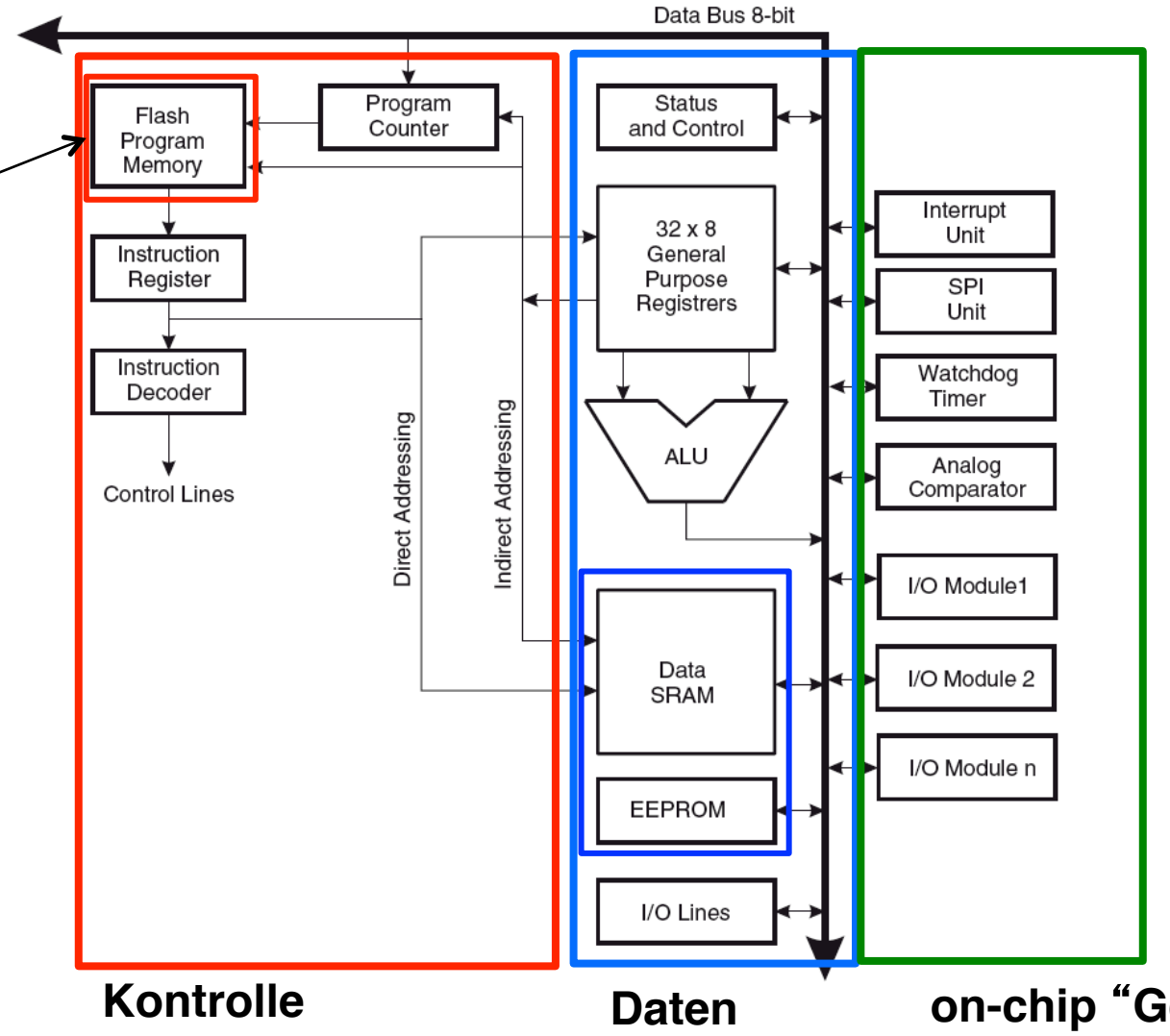


Data Sheet:
Rev. 2467M–AVR–11/04

Blockschaltbild der AVR CPU

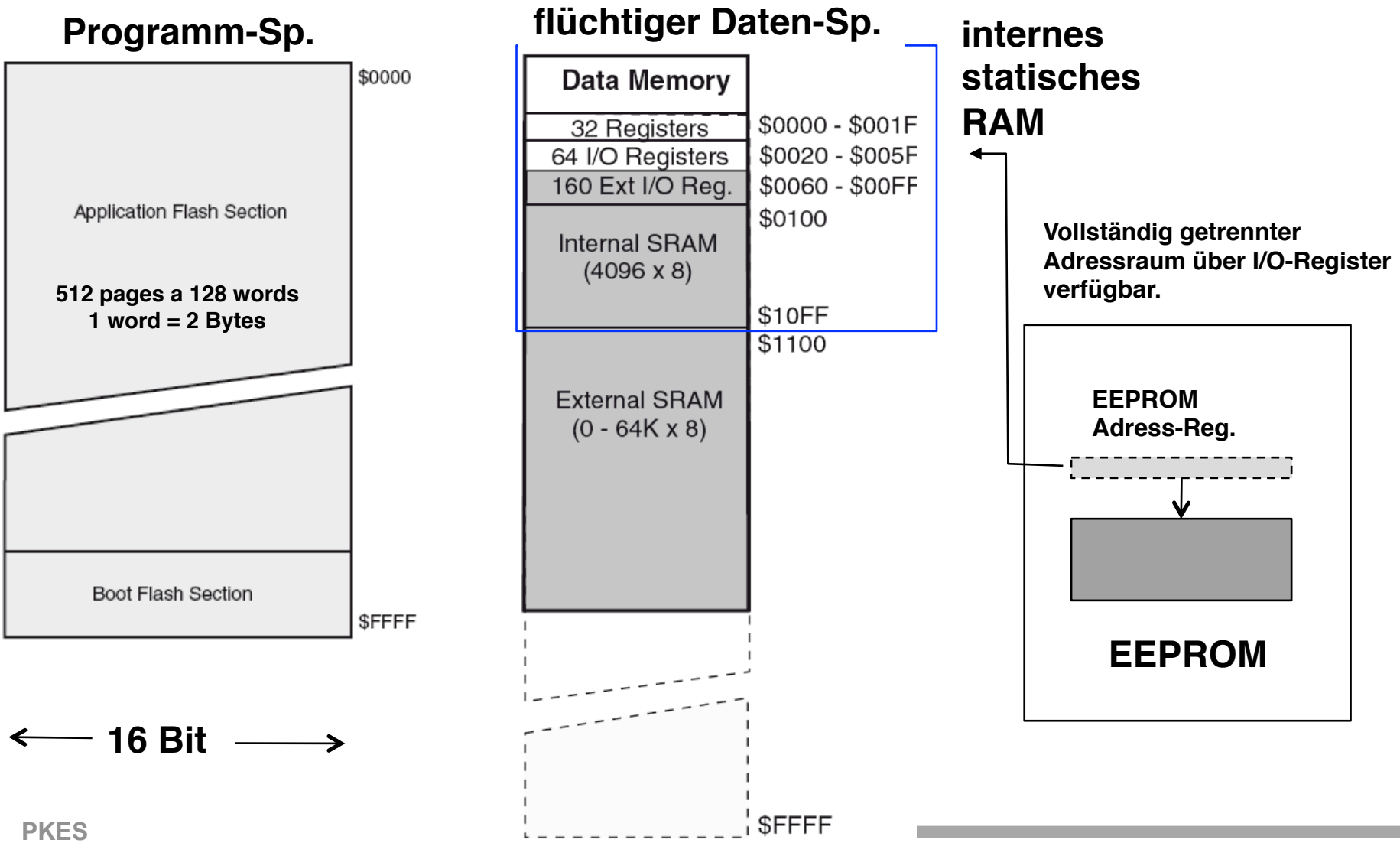
Programm-Speicher

Harvard Architektur

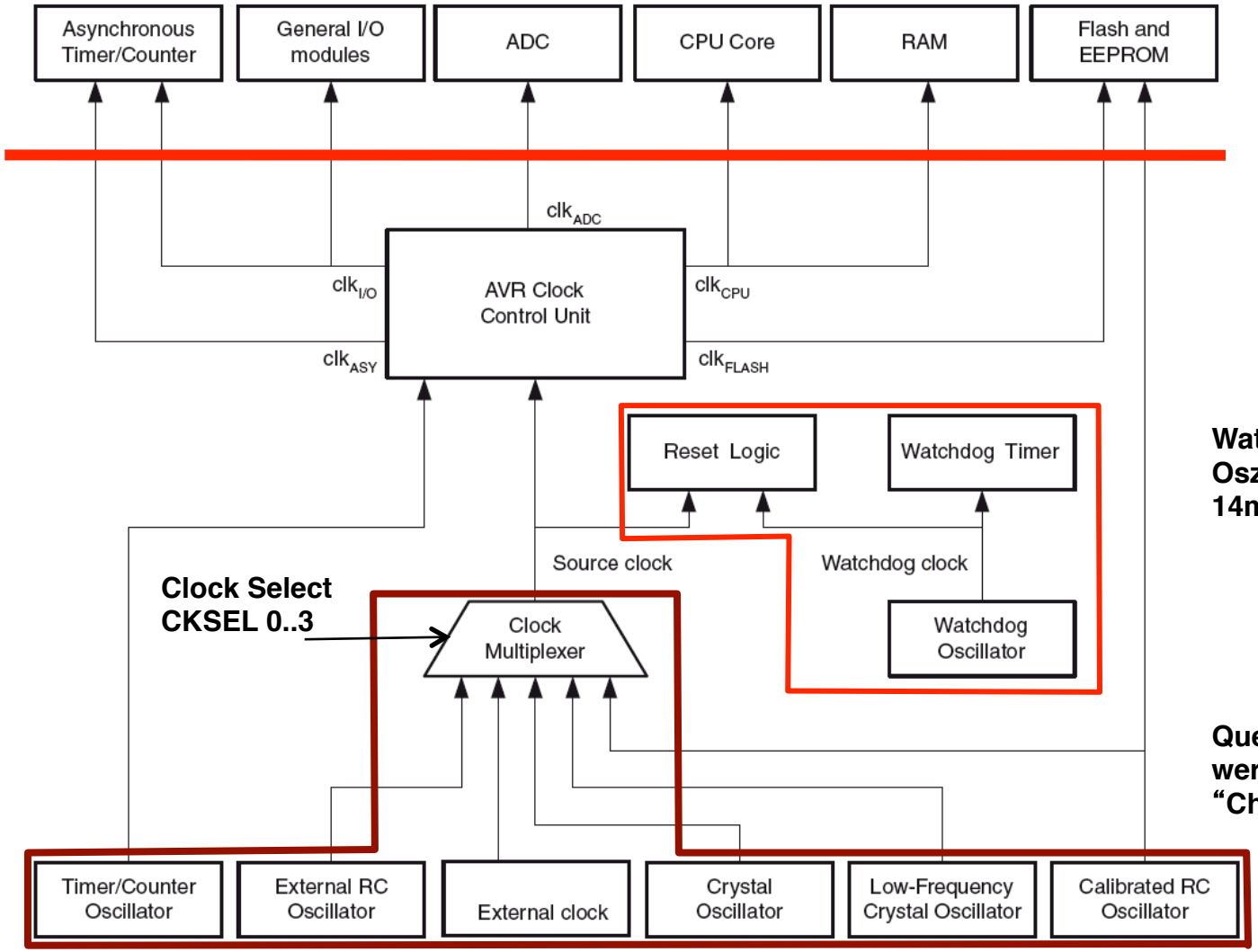


Data Sheet:
Rev. 2467M–AVR–11/04

Speicherorganisation des AVR (ATmega128)



AVR Clock-System



System-Module

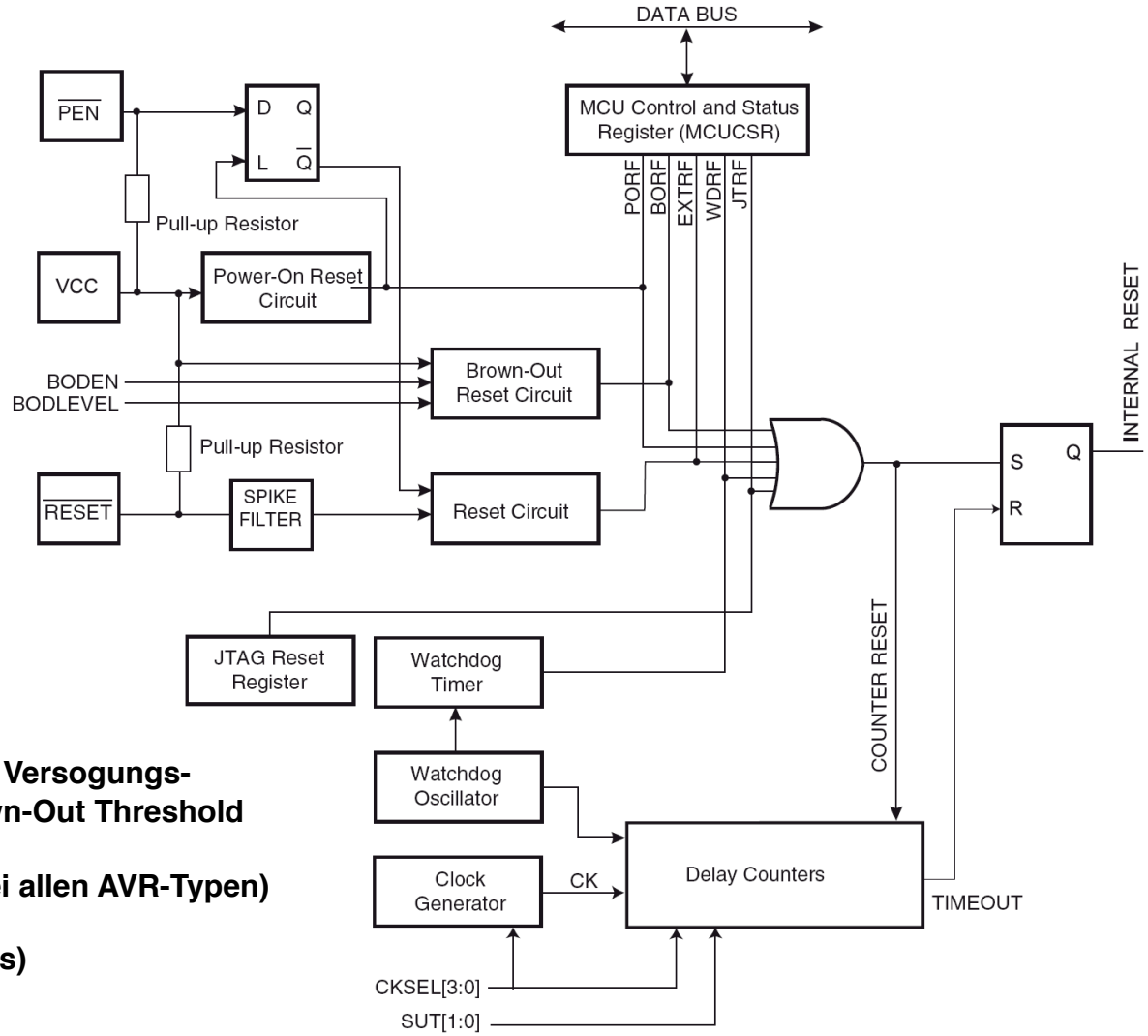
Watchdog hat separaten on-Chip Oszillator. Periode kann zwischen 14ms und 1,9 Sek. eingestellt werden.

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: Programing Enable
BODEN: Brown-Out Enable
Brown-Out detektiert, dass die Versogungs-spannung 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 Characterictics

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}	$\overline{\text{RESET}}$ Pin Threshold Voltage		0.2 V_{CC}		0.85 V_{CC}	V
t_{RST}	Pulse width on $\overline{\text{RESET}}$ 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 clk_{Flash}). Andere Komponenten, wie ADC, SPI, USART, Watchdog (WTD), Timer/Counter laufen weiter.

ADC Rauschunterdrückung: Hält CPU und I/O an (clk_{cpu}, clk_{Flash} und $clk_{I/O}$). 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.

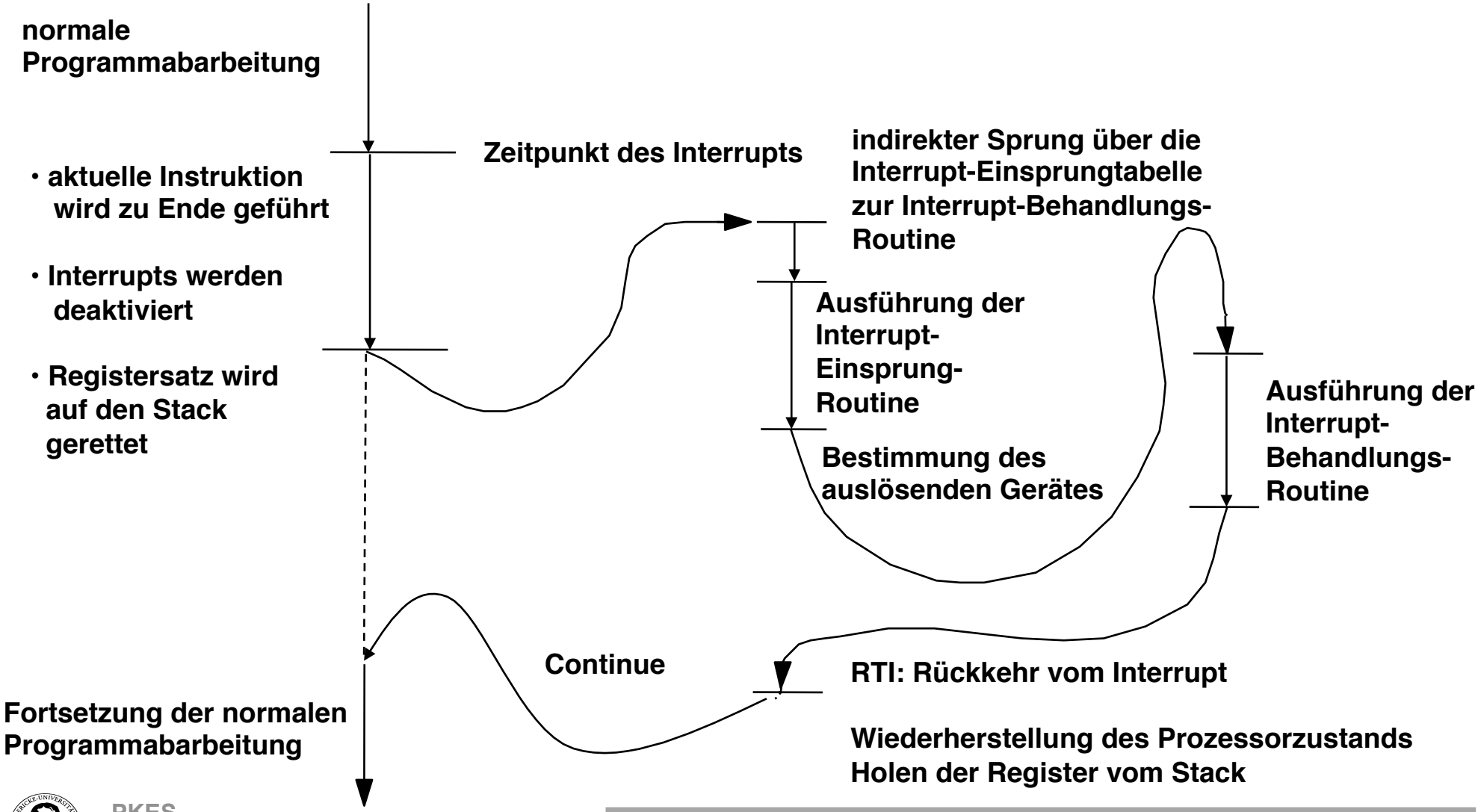
Reduktion des Stromverbrauch durch “Schlaf-Modi”

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

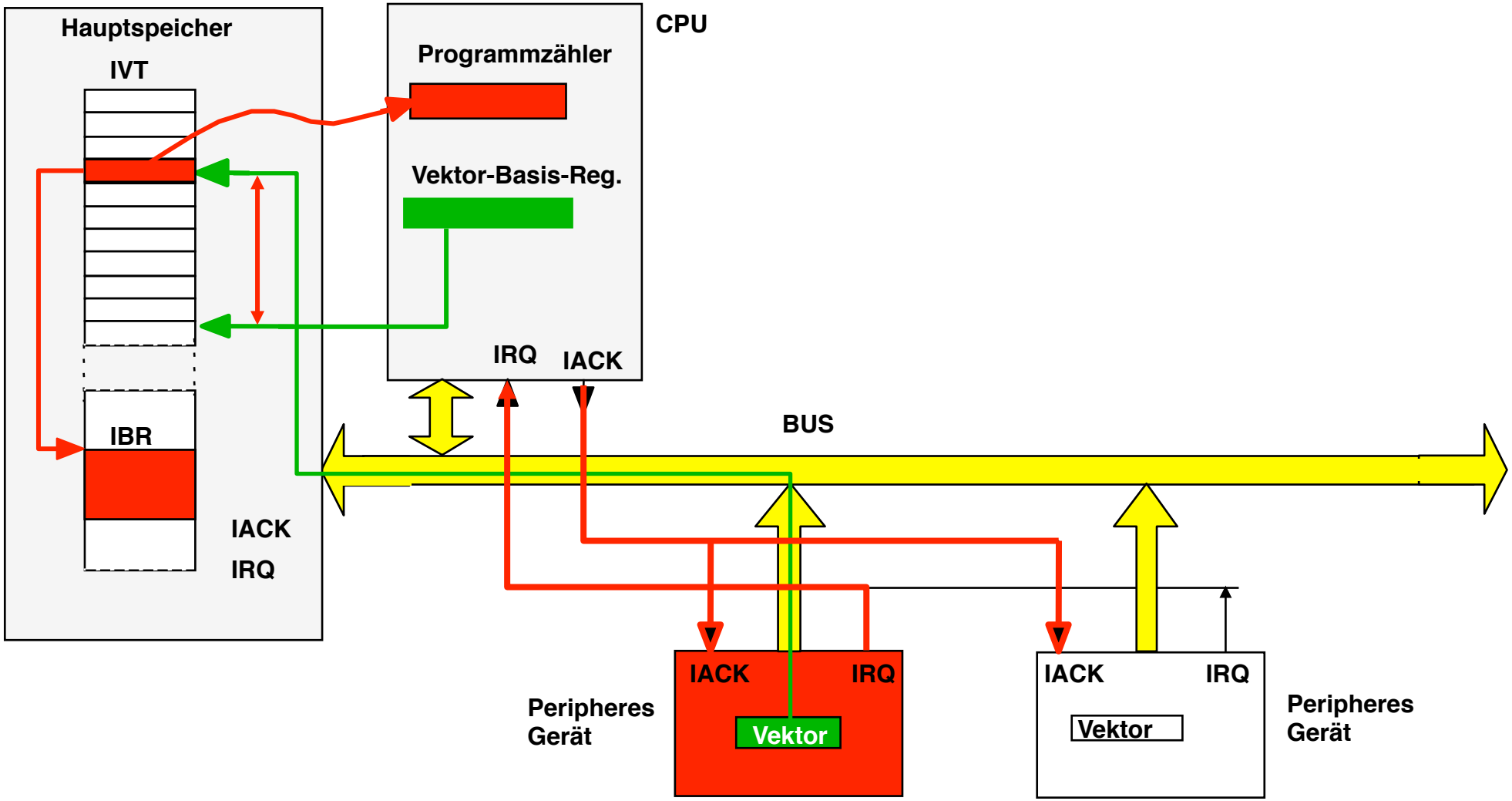
Einsparungen bis ca. 1:1000 möglich

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.

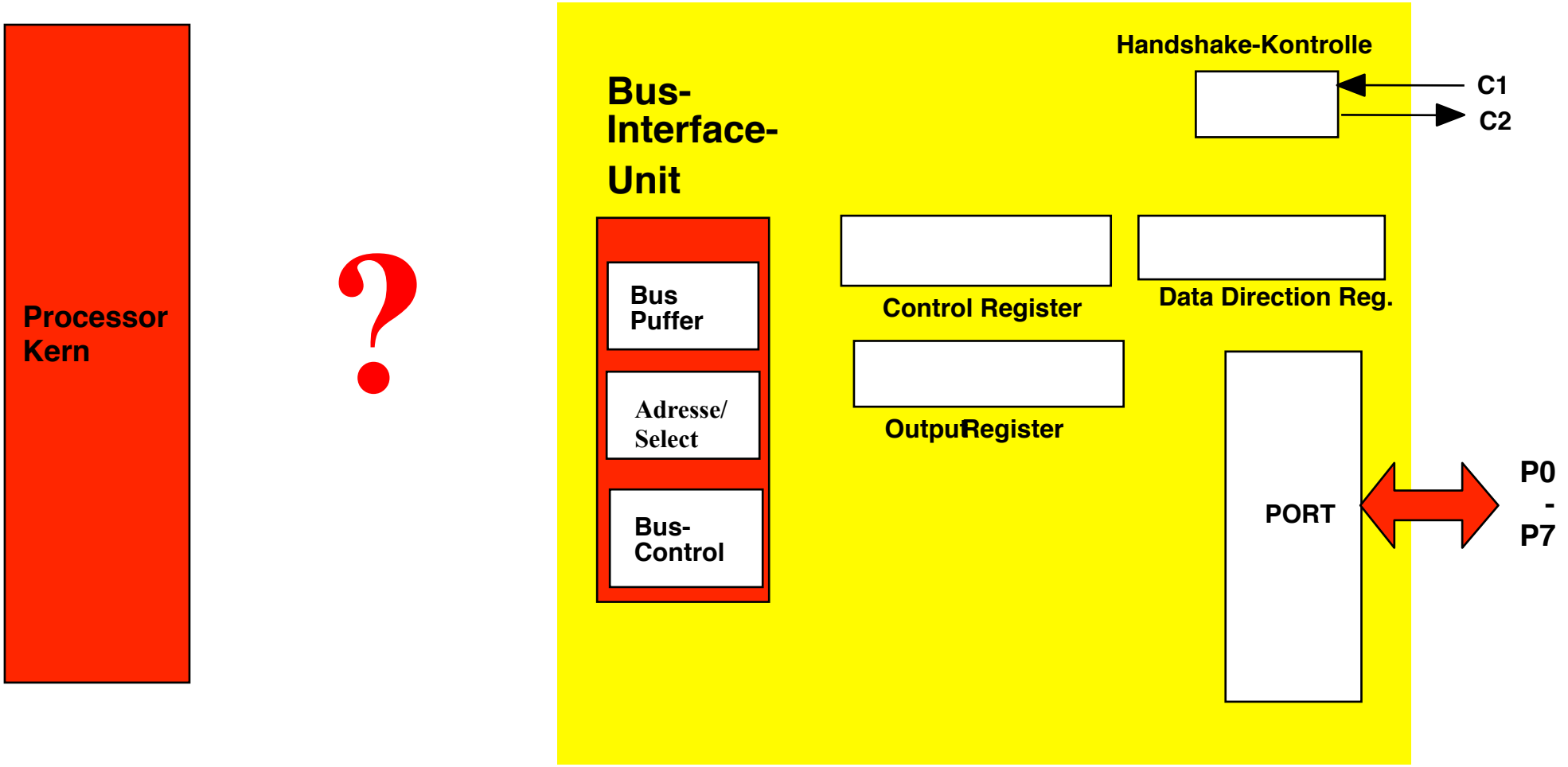
Prioritäten:
Reset
ext.Interrupts 0-7
Timer (hohe Prio)
Kommunikation
ADC
Analog-Komparator
Timer (niedrigere Prio)
Kommunikation (niedrigere Prio)

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

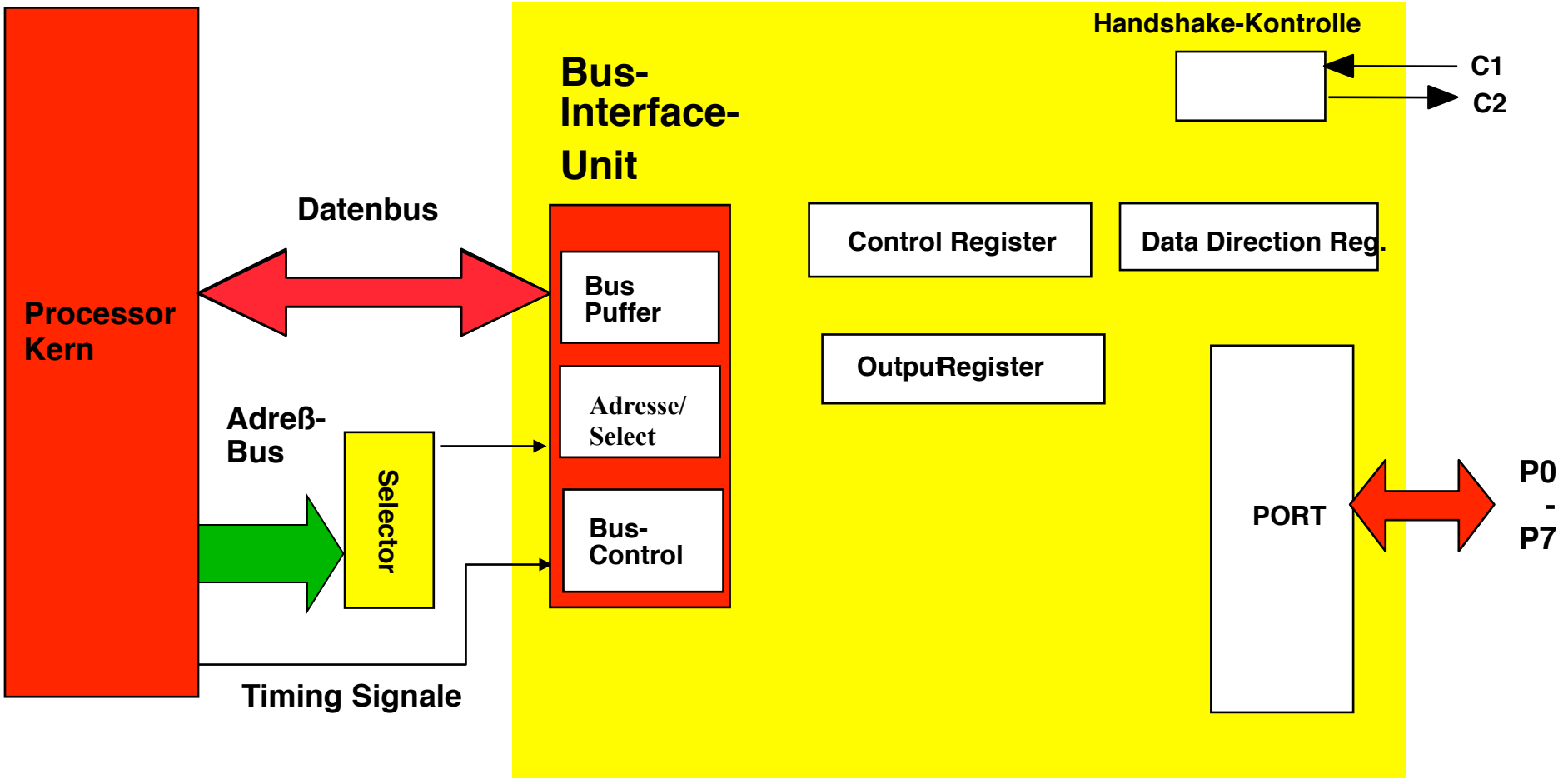
```
Addr.  LabelsCode    Comments
$0000 jmp RESET ;    Reset Handler
$0002 jmp EXT_INT0 ; IRQ0 Handler
$0004 jmp EXT_INT1 ; IRQ1 Handler
$0006 jmp EXT_INT2 ; IRQ2 Handler
$0008 jmp EXT_INT3 ; IRQ3 Handler
$000A jmp EXT_INT4 ; IRQ4 Handler
$000C jmp EXT_INT5 ; IRQ5 Handler
$000E jmp EXT_INT6 ; IRQ6 Handler
$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_COMPC; 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_COMPC; 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
```

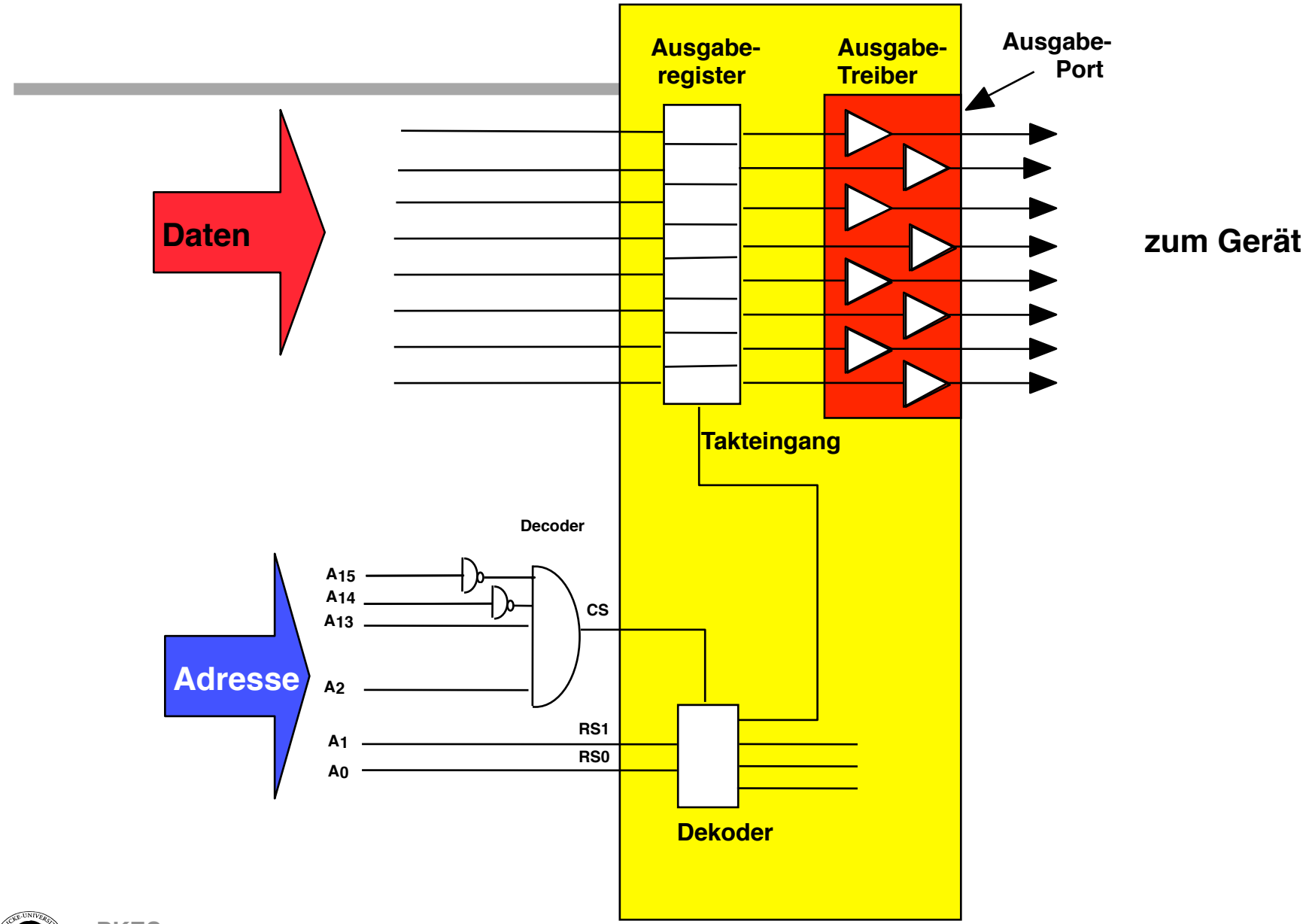
Ports, die Schnittstelle zur Peripherie

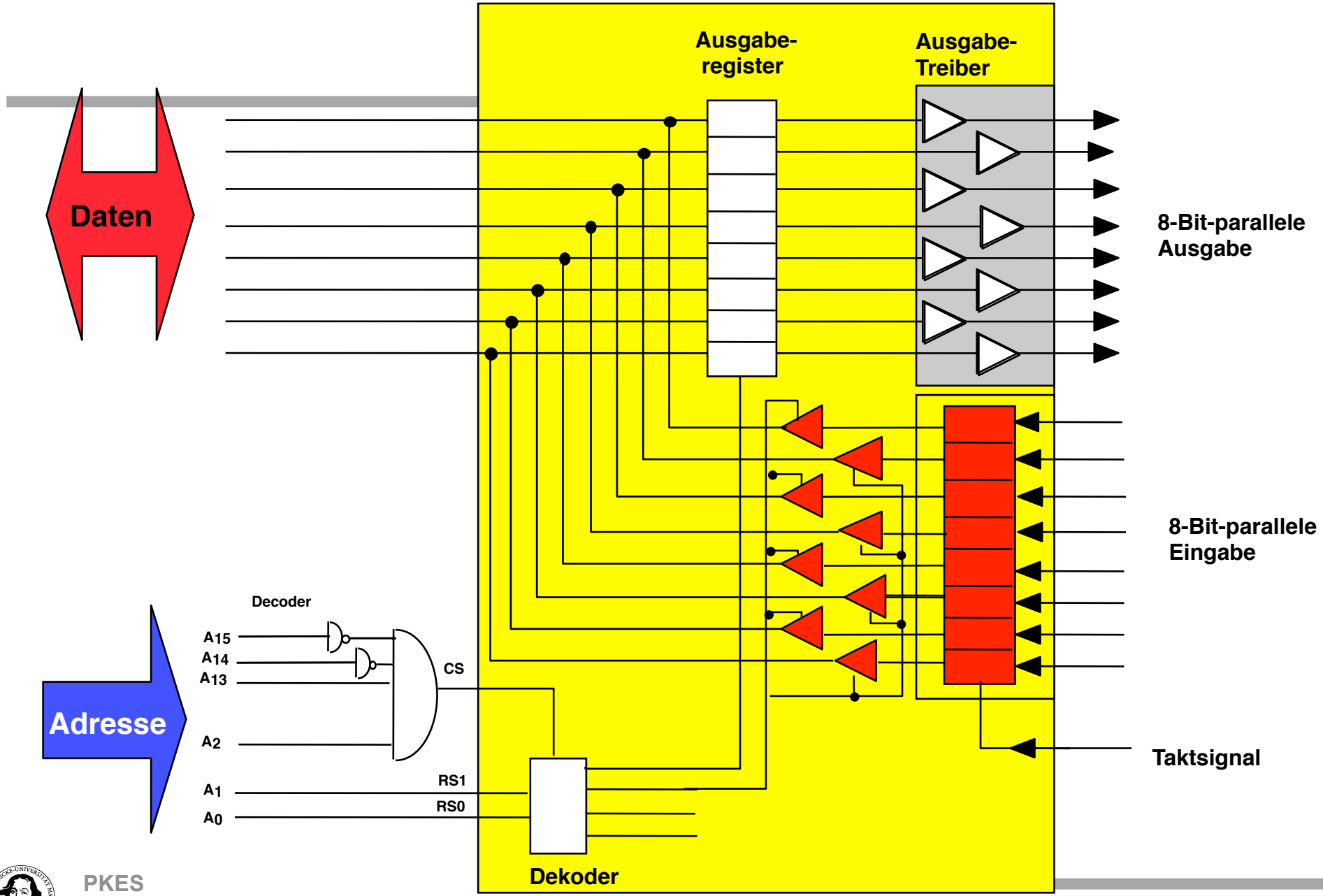
Eine Byte-parallele Geräteschnittstelle

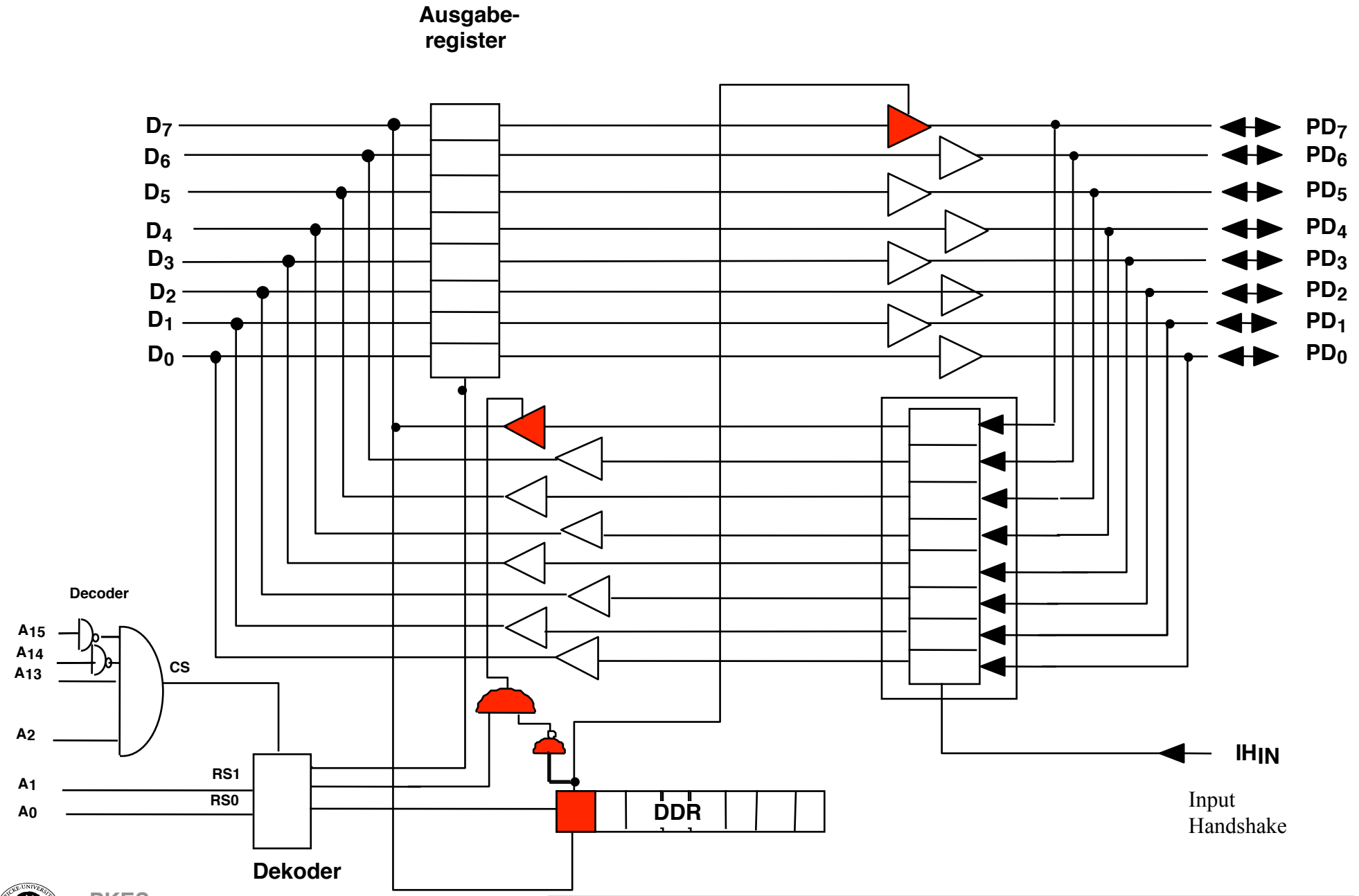


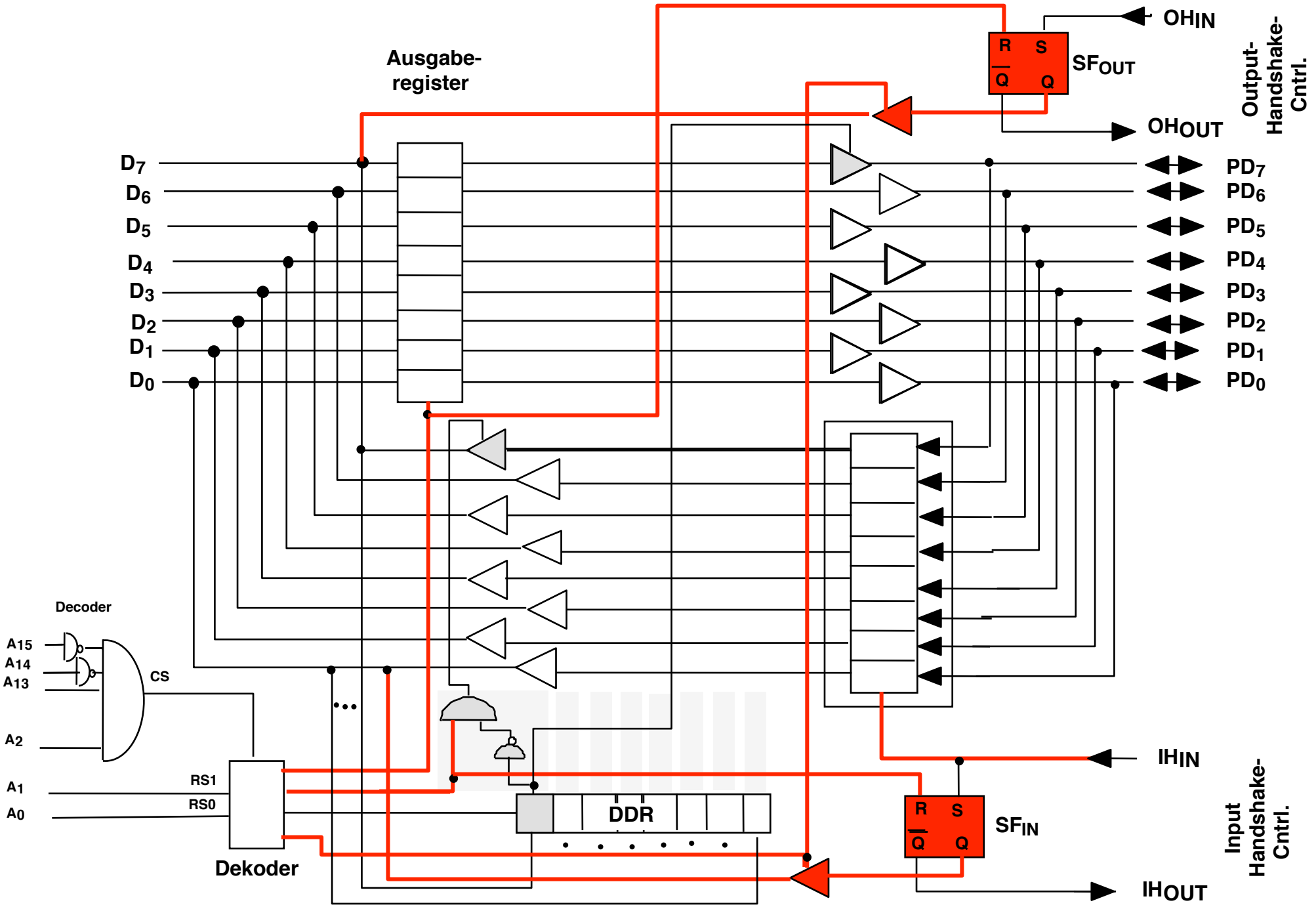
Eine Byte-parallele Geräteschnittstelle




Ausgabe-
register
Ausgabe-
Treiber
Ausgabe-
Port
Daten
zum Gerät
Takteingang
Decoder
A15
A14
A13
CS
Adresse
A2
RS1
A1
RS0
A0
Dekoder


Ausgabe-
register
Ausgabe-
Treiber
Daten
8-Bit-parallele
Ausgabe
8-Bit-parallele
Eingabe
Taktsignal
Decoder
A15
A14
A13
CS
A2
A1
A0
RS1
RS0
Adresse
Dekoder


Ausgabe-
register
D7
D6
D5
D4
D3
D2
D1
D0
PD7
PD6
PD5
PD4
PD3
PD2
PD1
PD0
Decoder
A15
A14
A13
CS
A2
A1
A0
RS1
RS0
Dekoder
DDR
IHIN
Input
Handshake


Ausgabe-
register
OHIN
R
S
Q
Q
SFOUT
OHOUT
Output-
Handshake-
Cntrl.
D7
D6
D5
D4
D3
D2
D1
D0
PD7
PD6
PD5
PD4
PD3
PD2
PD1
PD0
Decoder
A15
A14
A13
CS
A2
A1
A0
RS1
RS0
Dekoder
DDR
IHIN
R
S
Q
Q
SFIN
IHOUT
Input
Handshake-
Cntrl.

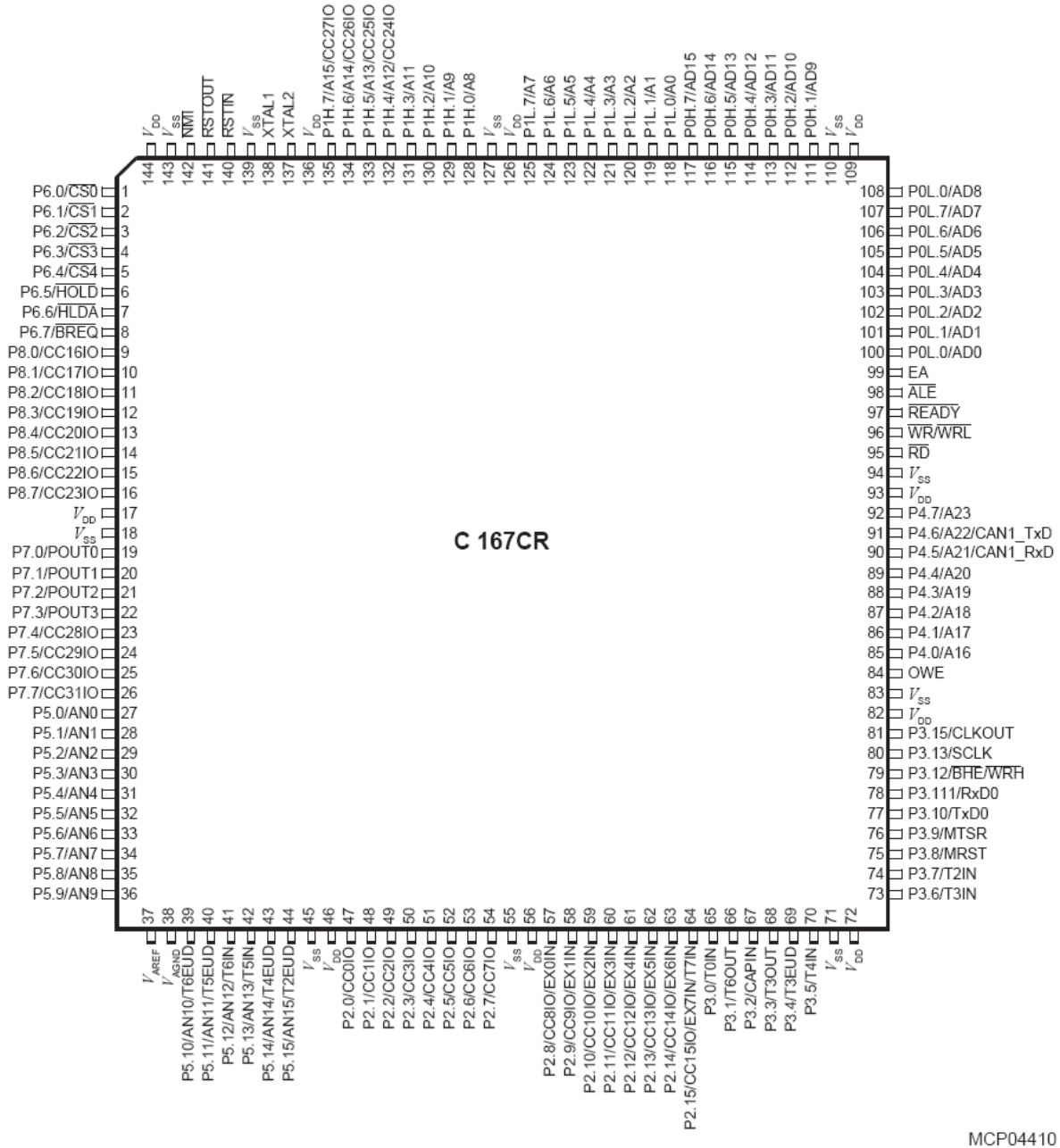
Infineon C 167 I/O-Ports

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



MCP04410

Infineon C 167 I/O-Ports

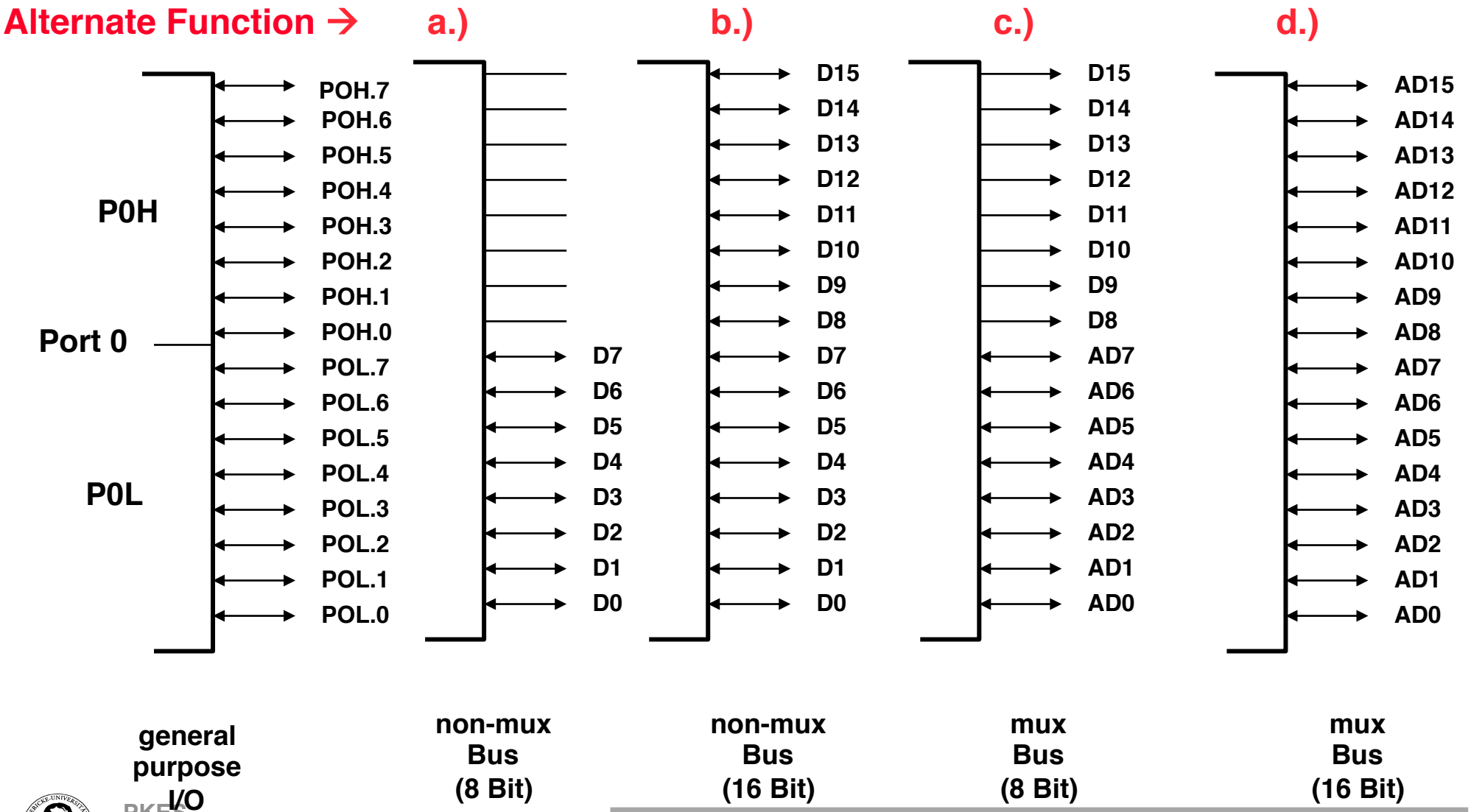
Bedeutung für Port-Pins für Port 0:

- **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)**

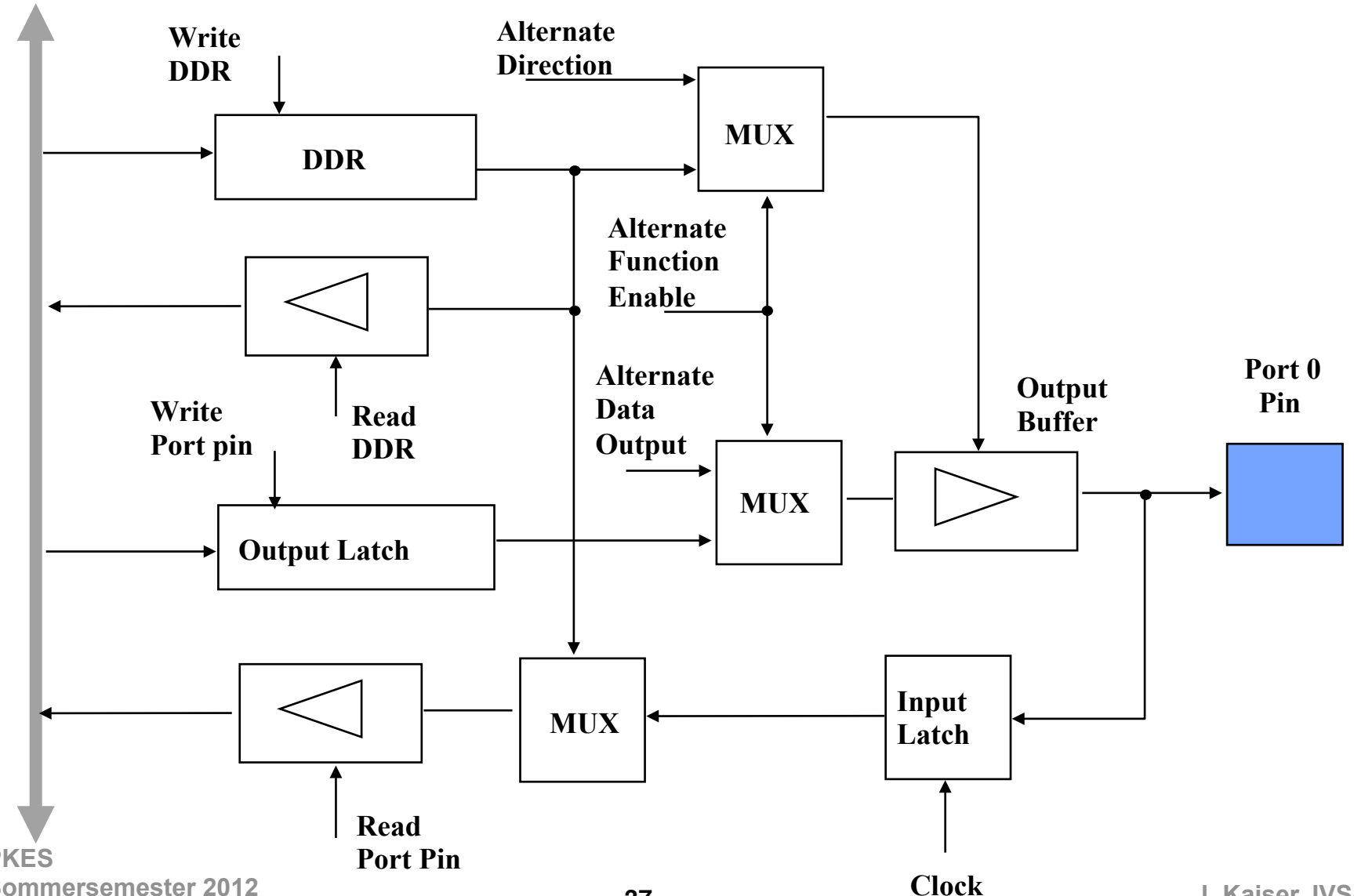
Spezialfunktionen der übrigen Ports (Alternate Functions) :

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

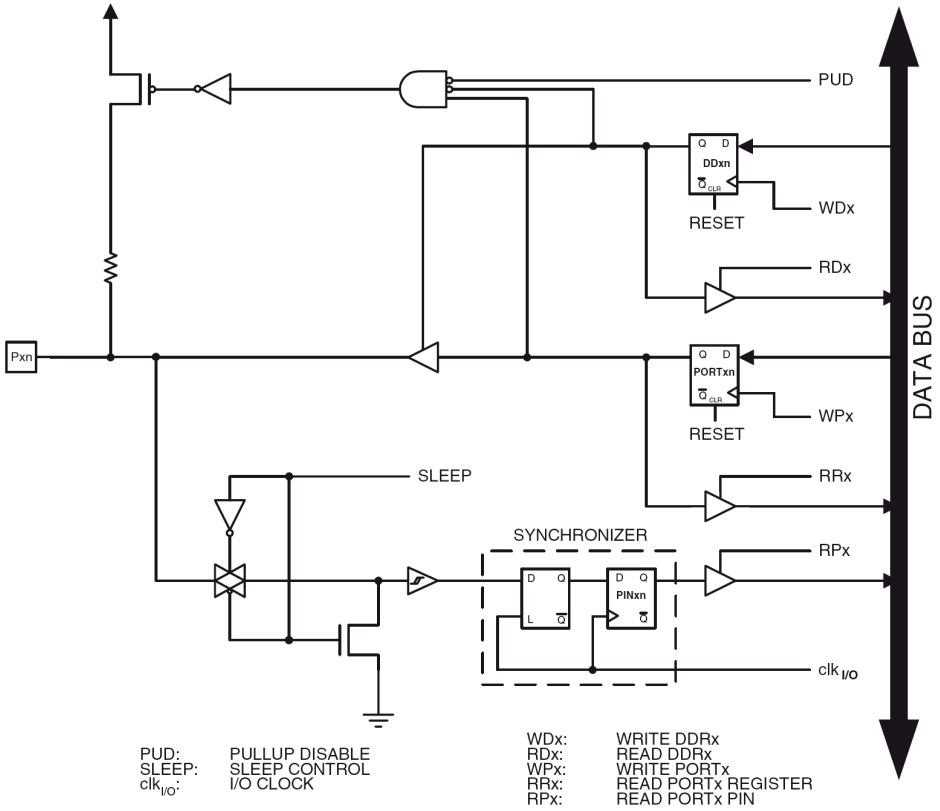
Infineon C 167 I/O-Ports

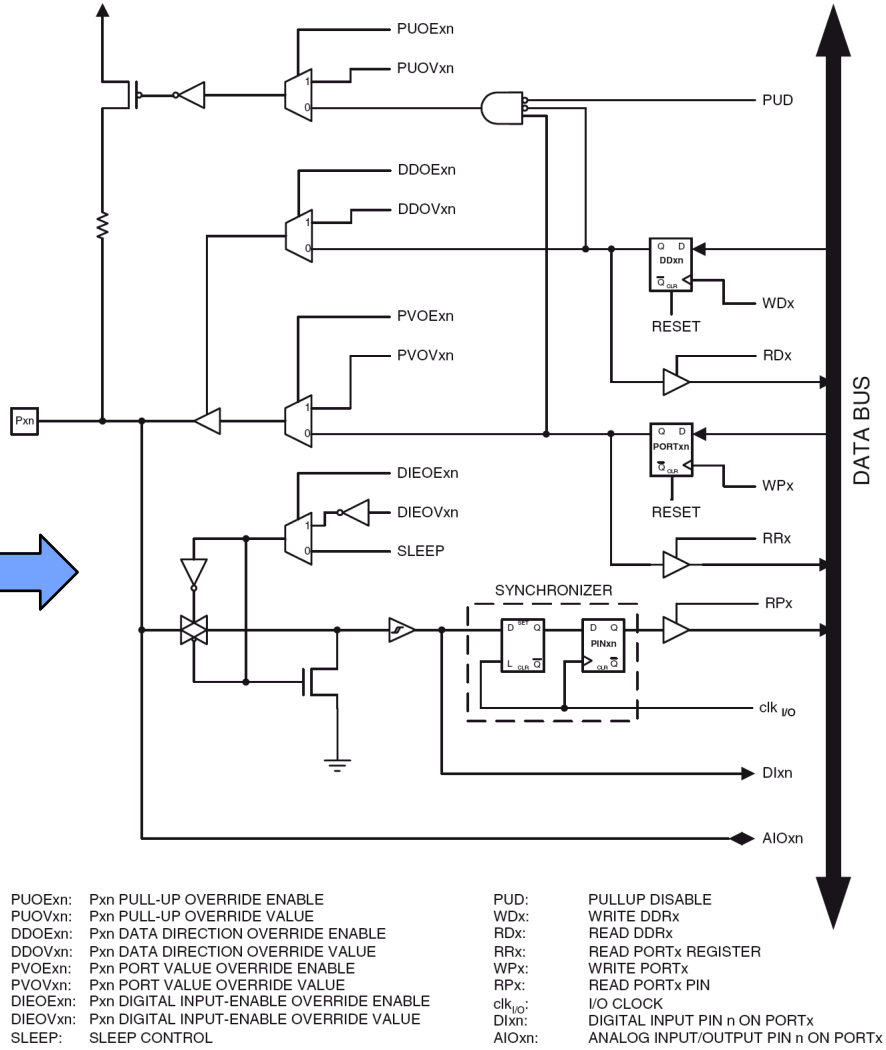


Beispiel für die Schaltung eines Port Pins



....Port Pins des AVR



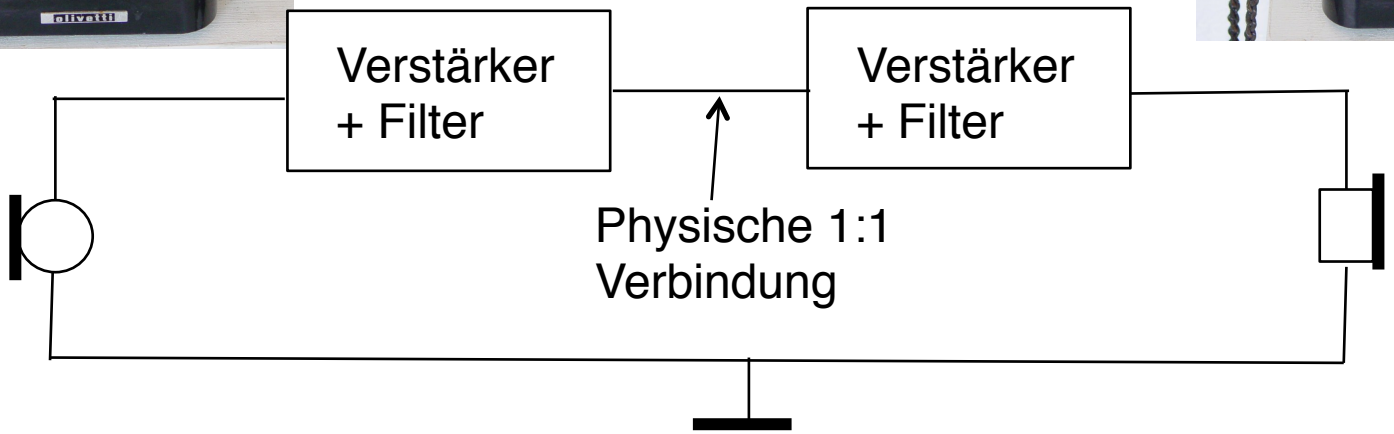


digital Standard

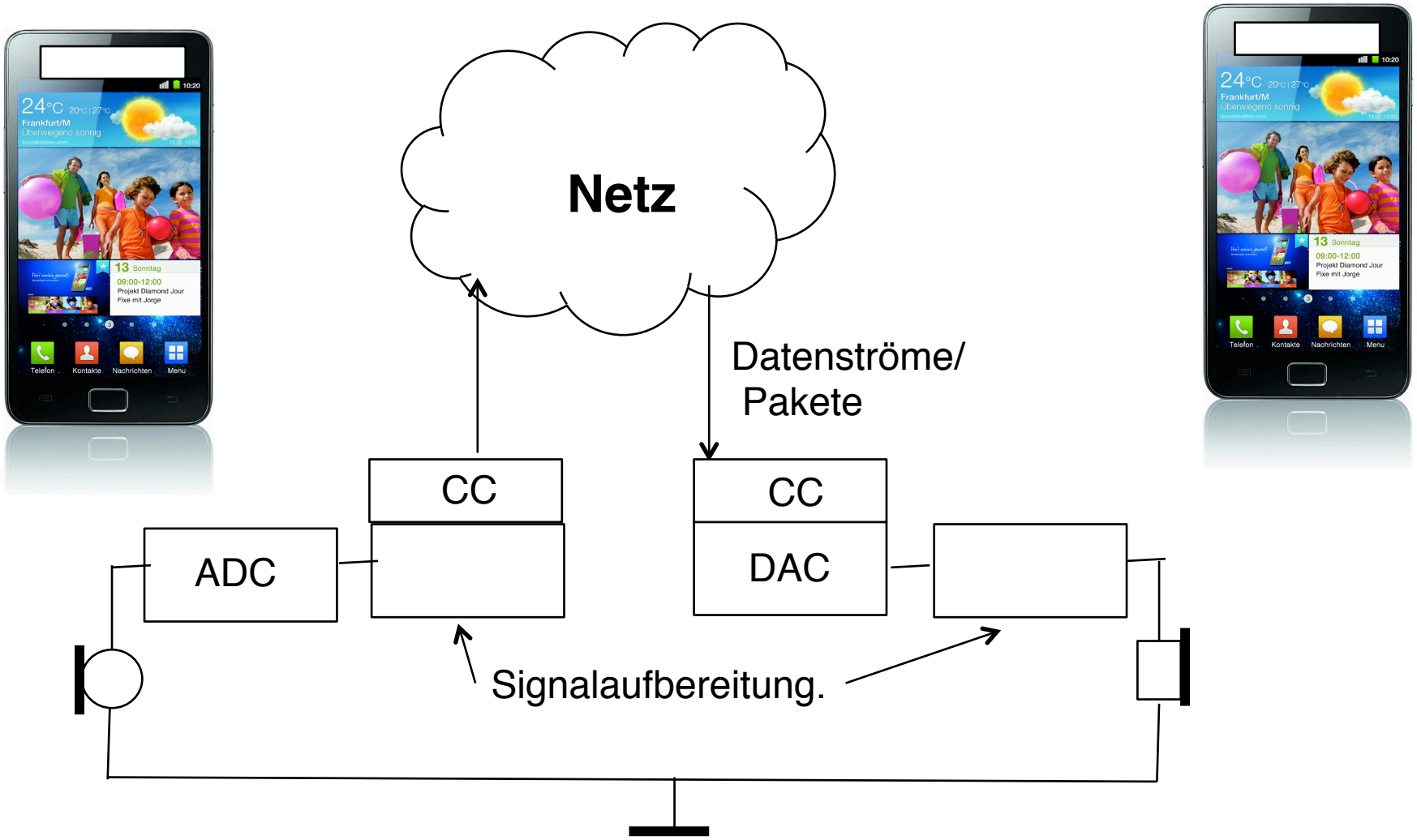
mit alternativen Fuktionen

Analoge Signalverarbeitung

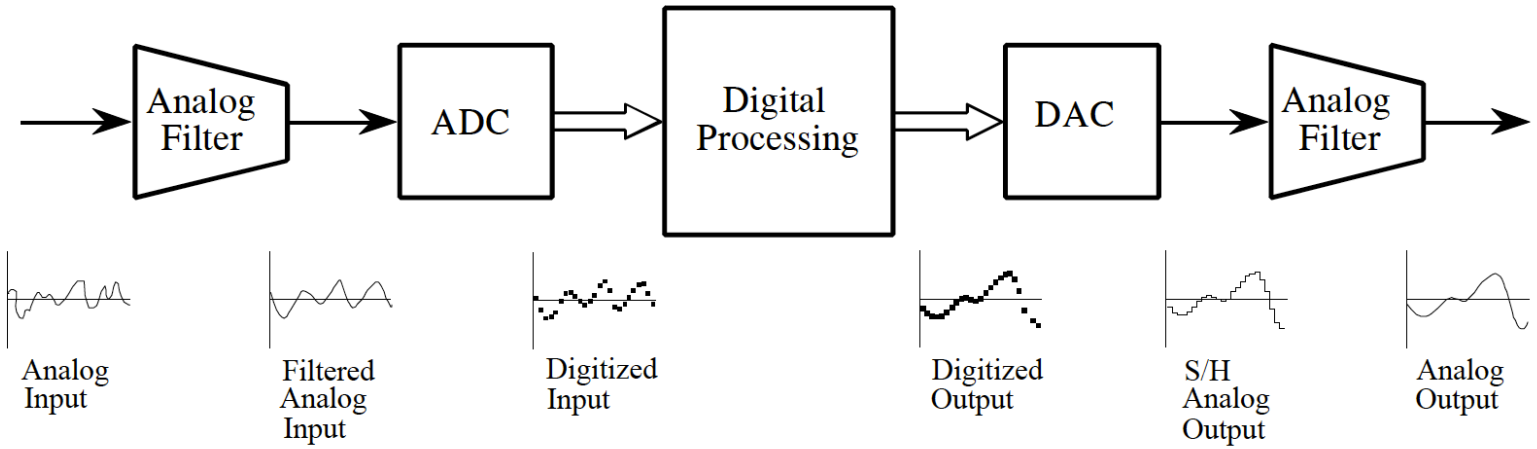
Spulen, Kondensatoren,
Widerstände, aktive Bauelemente



Digitale Signalverarbeitung



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:

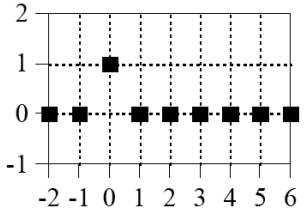
Digitale Filter
Spektralanalyse (Fourieranalyse)
Datenkompression
Signalmultiplexing

In den Anwendungsbereichen:

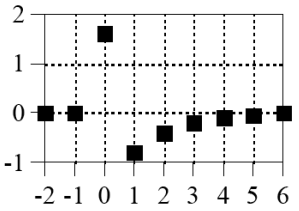
Radar
Sonar
Medizin: MRT, CT, Ultraschall,
EKG, EEG Analyse
Bildverarbeitung
Audio Verarbeitung

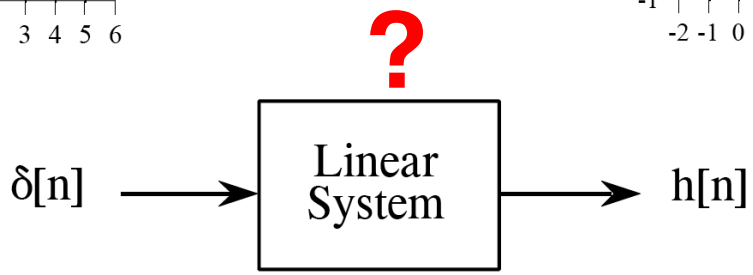
Digitale Filter

Einheitsimpuls (δ-Funktion)



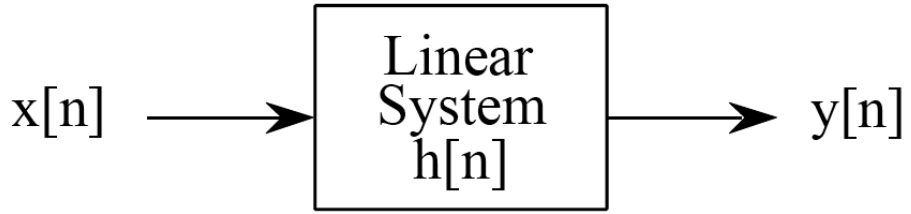
Ausgangssignal





Filter, beschrieben durch die Impulsantwort (Filterkern)

Eingangs-Signal



Ausgangs-Signal

$$x[n] * h[n] = y[n]$$

Faltung

Eigenschaften eines linearen Systems

1. Homogenität
2. Additivität
3. Verschiebungsinvarianz

Homogeneity

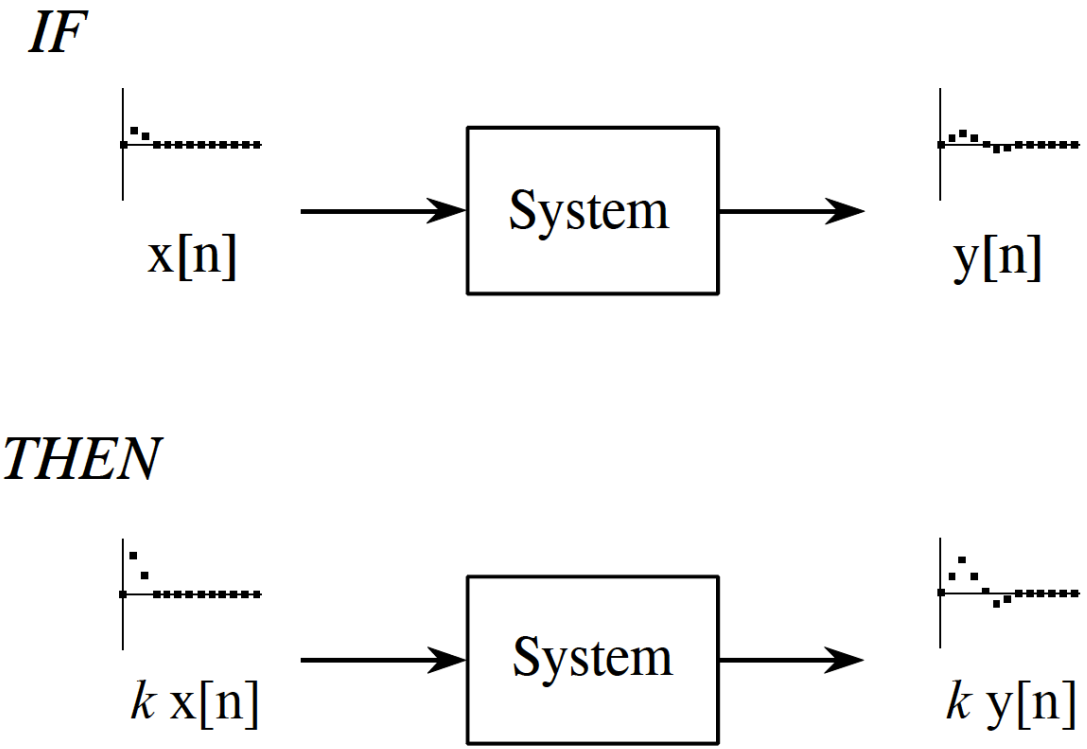


FIGURE 5-2
Definition of homogeneity. A system is said to be *homogeneous* if an amplitude change in the input results in an identical amplitude change in the output. That is, if $x[n]$ results in $y[n]$, then $kx[n]$ results in $ky[n]$, for any signal, $x[n]$, and any constant, k.

Additivity

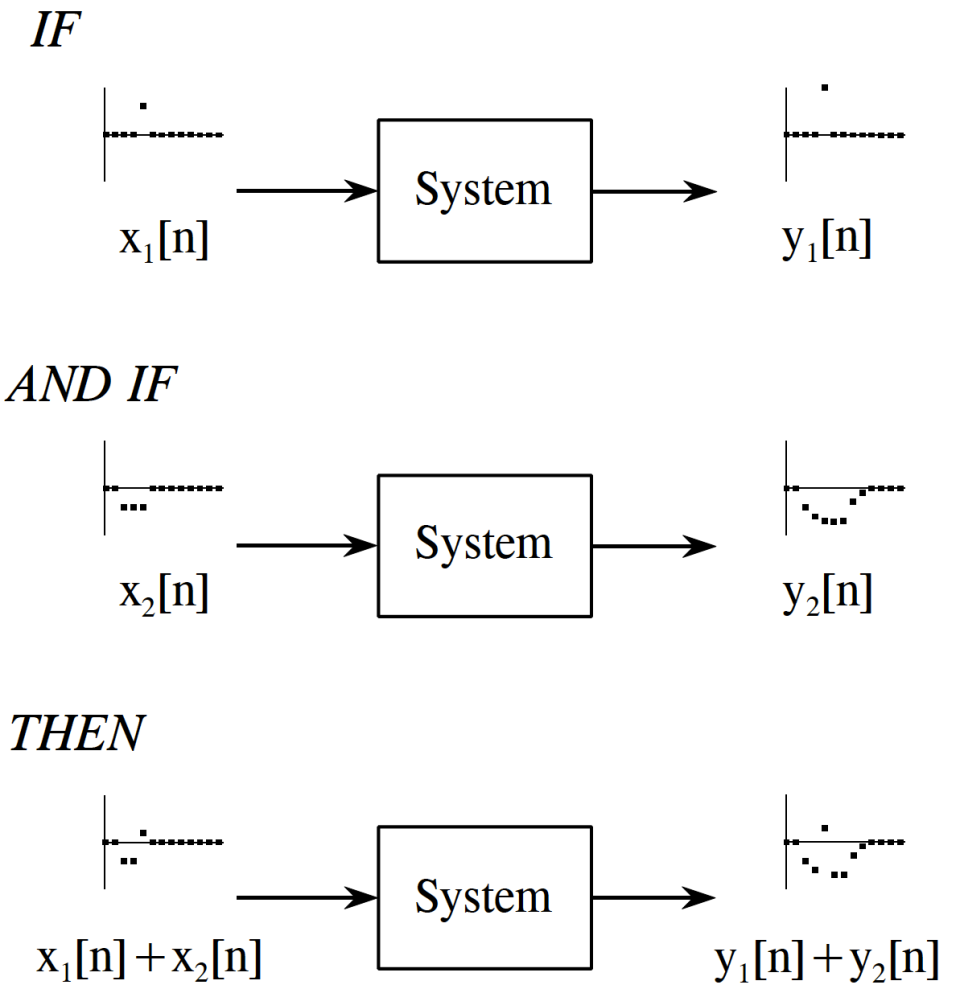


FIGURE 5-3
Definition of additivity. A system is said to be *additive* if added signals pass through it without interacting. Formally, if $x_1[n]$ results in $y_1[n]$, and if $x_2[n]$ results in $y_2[n]$, then $x_1[n]+x_2[n]$ results in $y_1[n]+y_2[n]$.

Shift Invariance

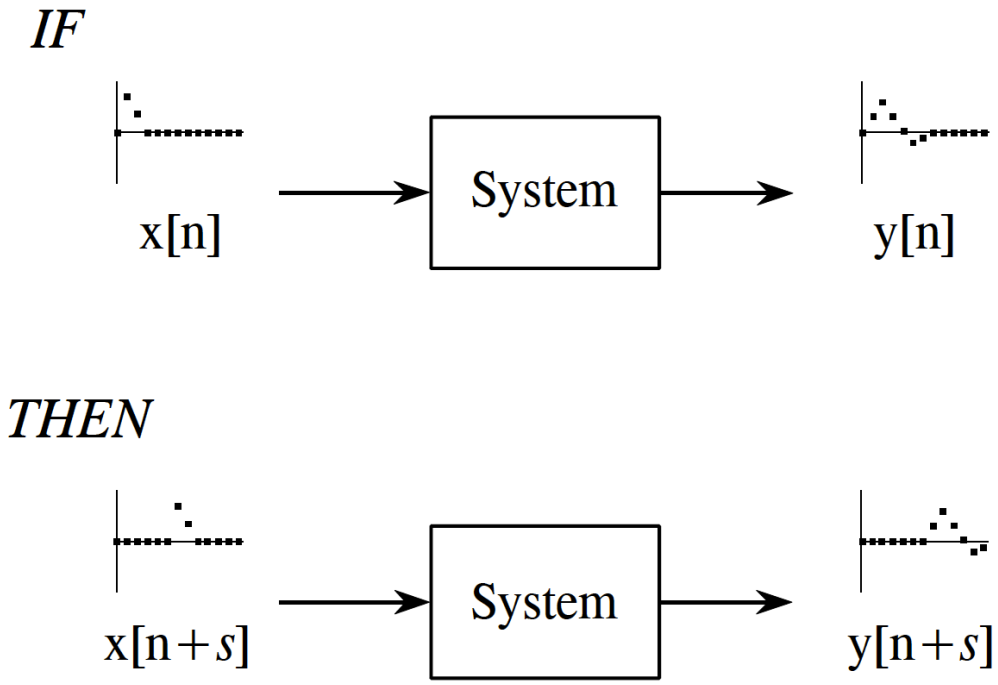


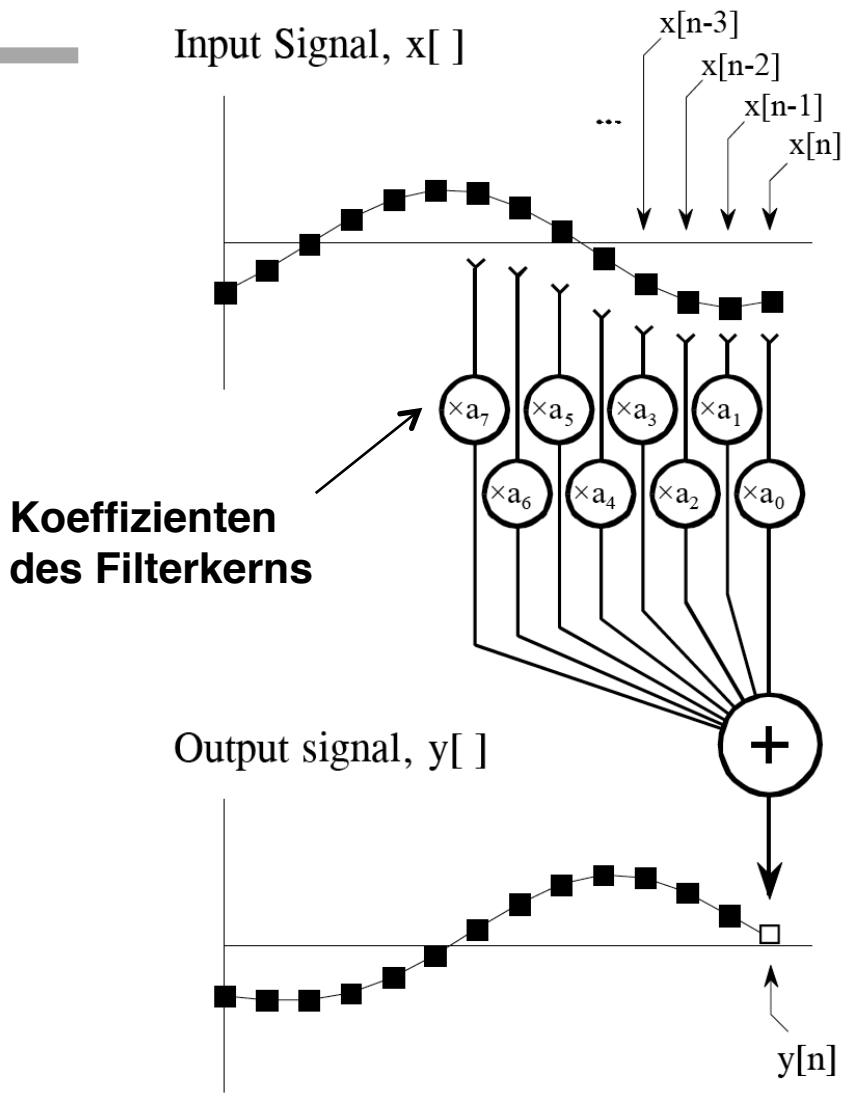
FIGURE 5-4
Definition of shift invariance. A system is said to be *shift invariant* if a shift in the input signal causes an identical shift in the output signal. In mathematical terms, if $x[n]$ produces $y[n]$, then $x[n+s]$ produces $y[n+s]$, for any signal, $x[n]$, and any constant, s.

Digitale Filter

Die Faltung des Signals mit dem Filterkern wird beschrieben durch:

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

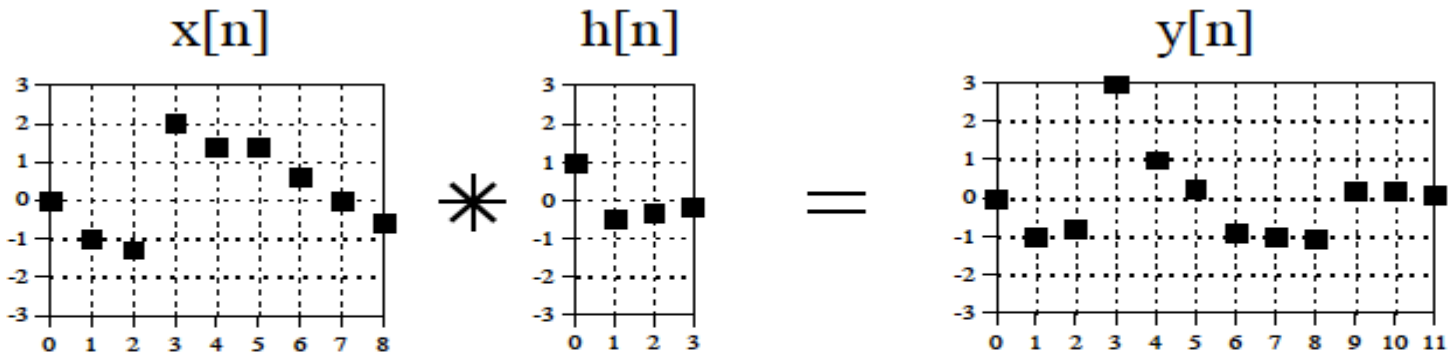
Das Ausgangssignal kann als gewichtete Summe über einer Folge von Eingangssignalen betrachtet werden.



Koeffizienten des Filterkerns

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

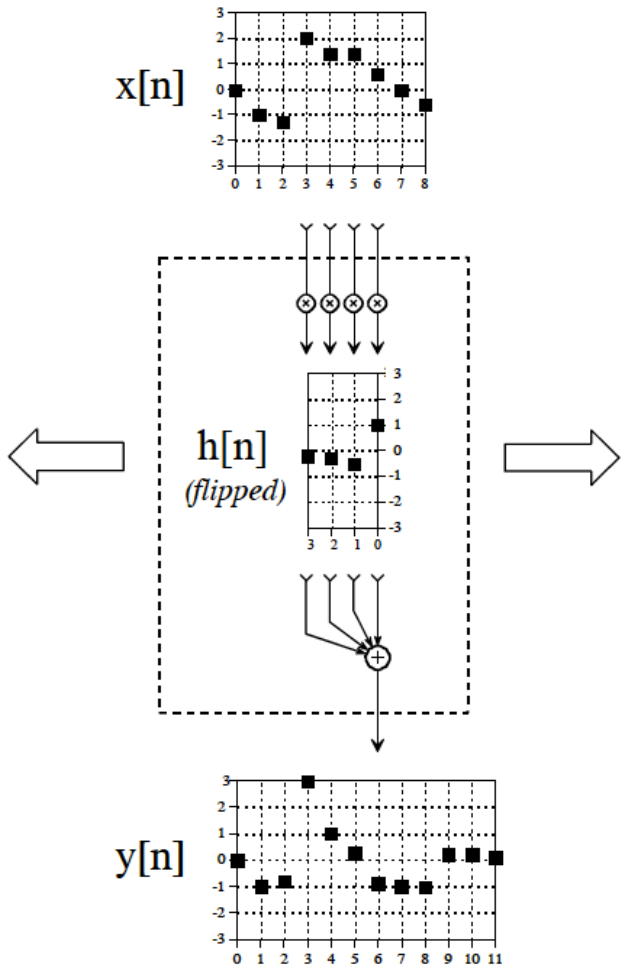
Digitale Filter

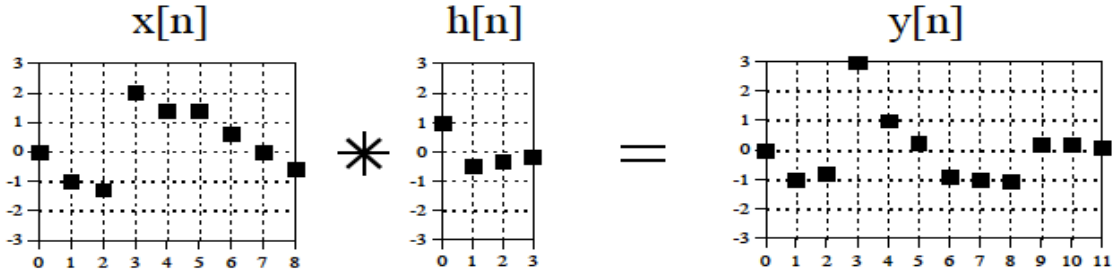


Beispiel der Faltung einer Signalfolge mit einem Filterkern

$y[n] = x[n] \ast h[n]$

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



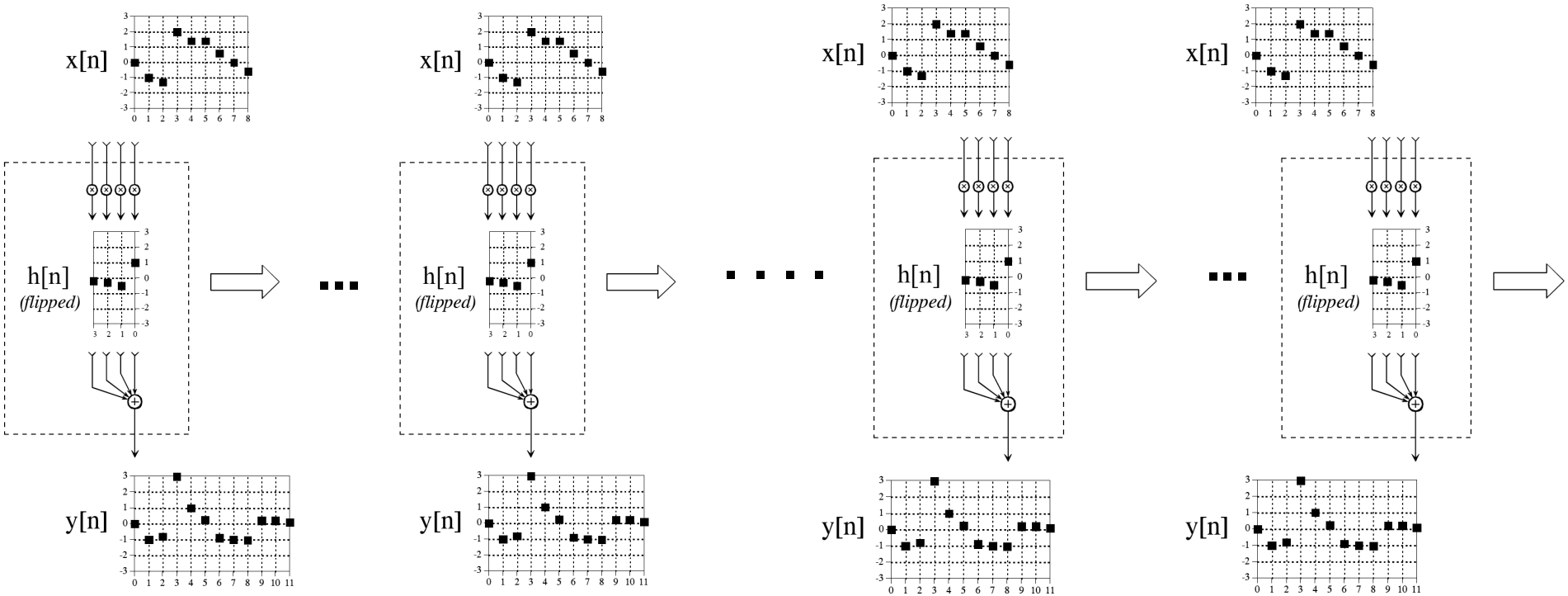


bei der mathematischen Darstellung sind die Elemente von h[n] von links nach rechts aufsteigend geordnet: h[0], h[1], h[2], h[3]

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

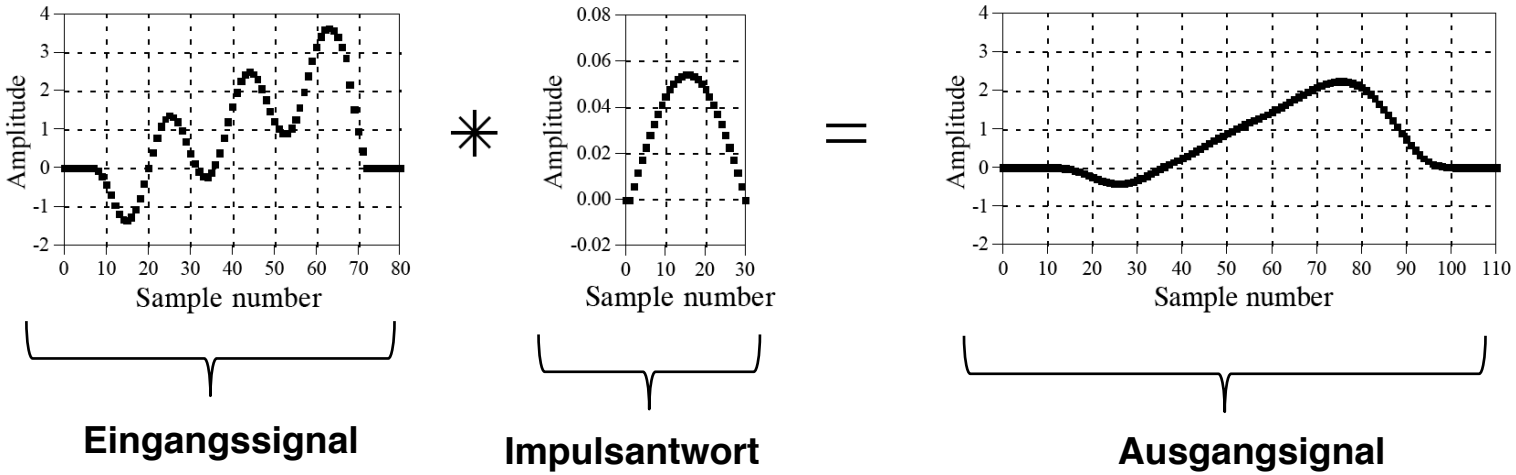
$$y[i] = \sum_{j=0}^{M-1} h[j]\, x[i-j]$$

Digitale Filter

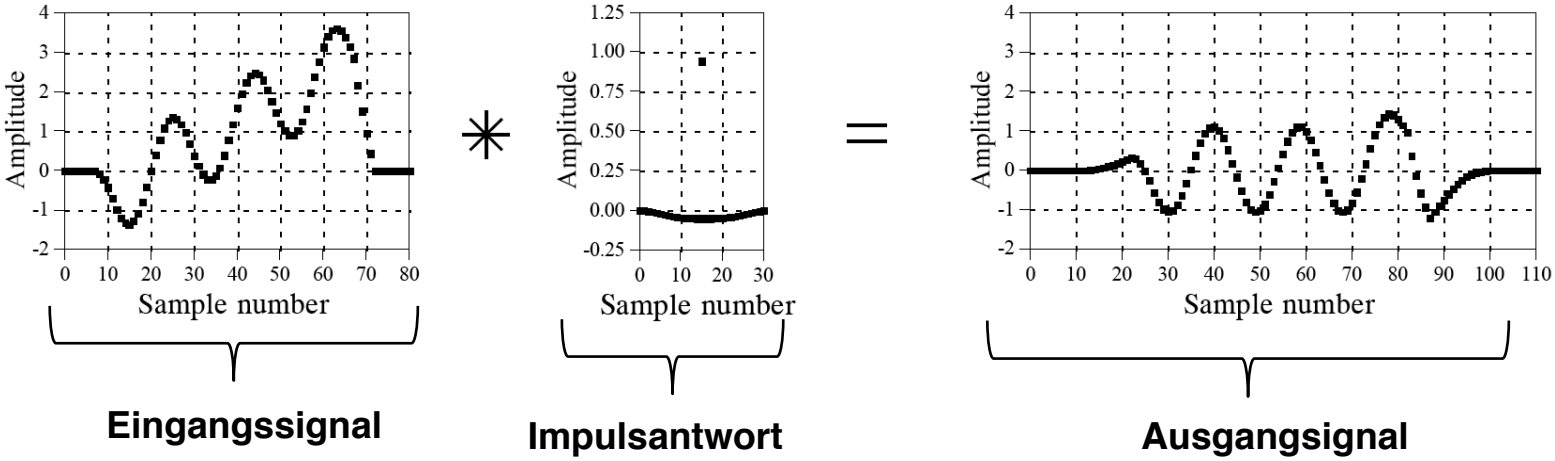


Digitale Filter

Tiefpass-Filter



Hochpass-Filter



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

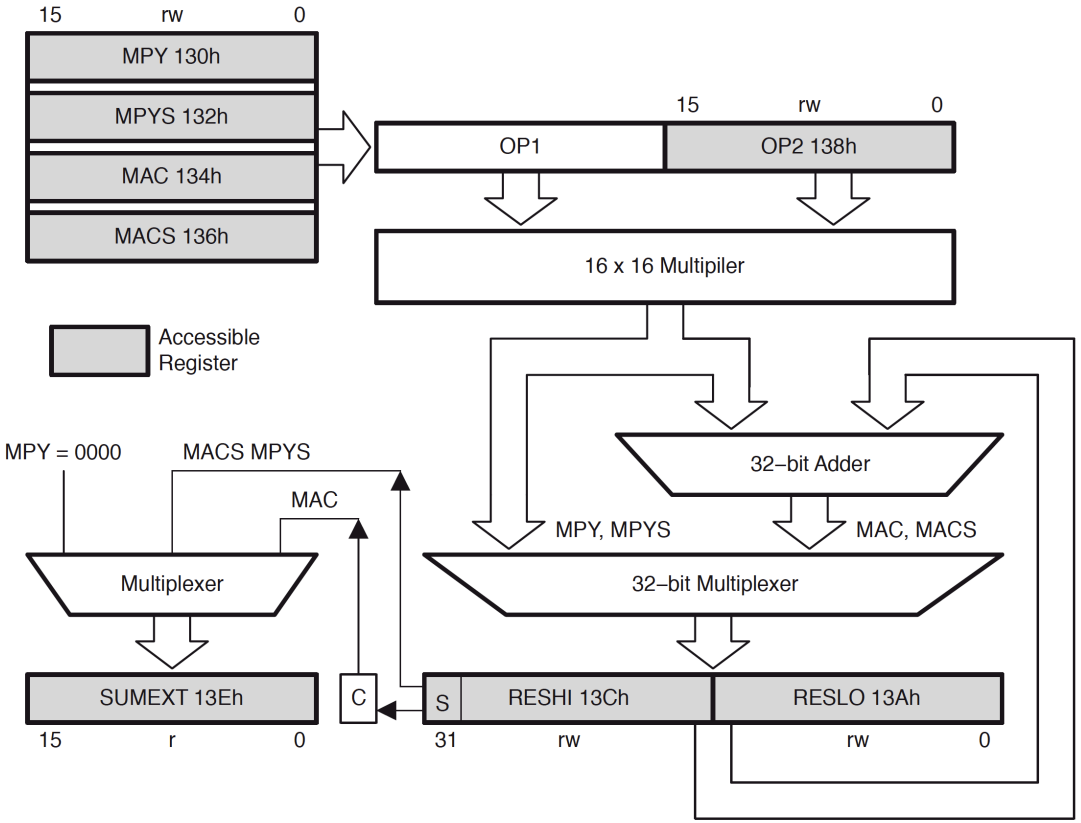
```
// convolution using output side algorithm

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

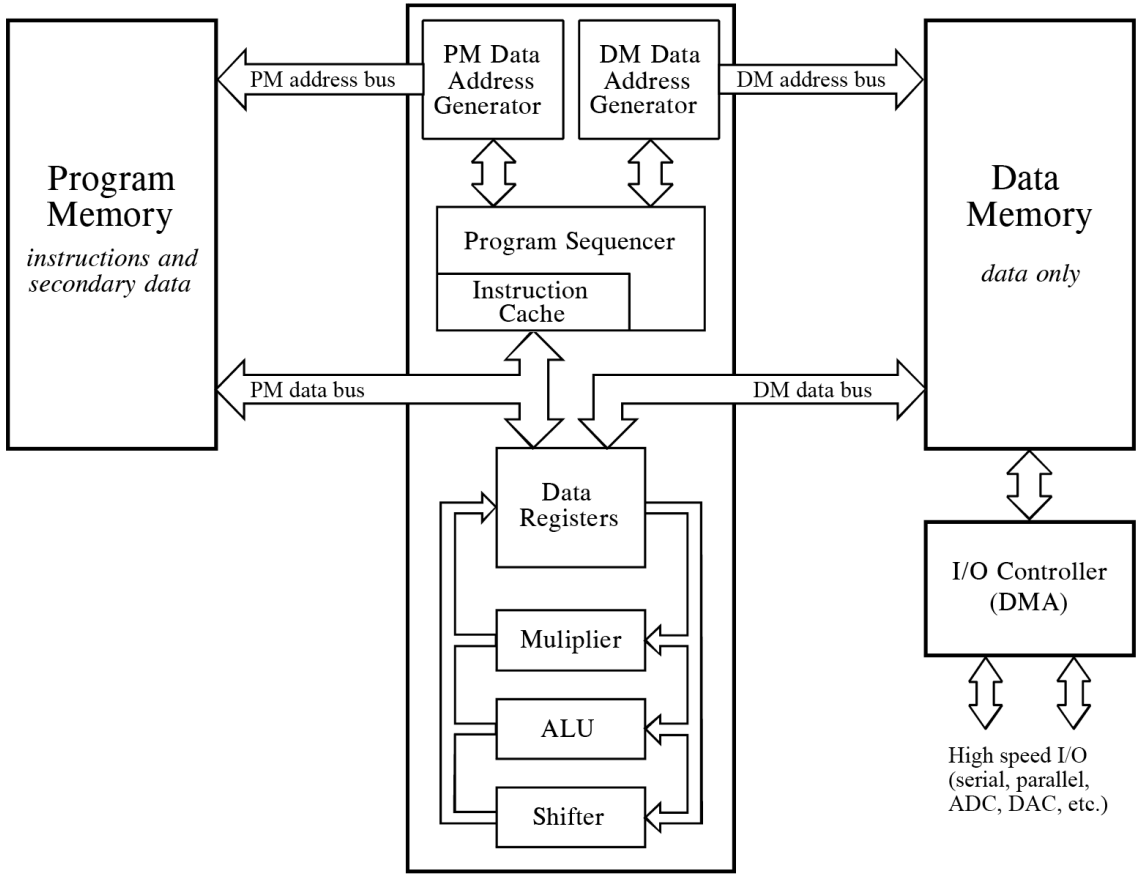
DSPs haben meist Optimierungen in folgenden Bereichen:

- **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**

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 Speicherschnitt-stelle zur Peripherie realisieren, stellen Mikro-Controller konfigurierbare Ports zur Verfügung, die spezielle Ein-Ausgabe-Funktionen unterstützen.