Institut für Betriebssysteme
und Rechnerverbund

Programmierung verteilter eingebetteter Systeme

Teamprojekt

Stephan Rottmann, Ulf Kulau, Felix Büsching

Winter Term 2013/14

Ablauf

- Organisatorisches
- Grundlagen 1
- Praktikumshardware
- Aufgabe 1

Teilnehmerinnen und Teilnehmer

23 Anmeldungen

- Gruppe 1: Samuel, Andreas, Dirk
- Gruppe 2: Thiemo, Tobias
- Gruppe 3: Jan, Matthias, Malte
- Gruppe 4: Alex, Jasmin, Karsten
- Gruppe 5: Jannik, Mohammed, Lars
- Gruppe 6: Oliver, Welf
- Gruppe 7: Pablo, Carlos
- Gruppe 8: Fabian, Christoph
- Gruppe 9: Nico L., Foued, Nico B.

Raumnutzung und Belegung: Raum 148 (hier)

Praktikumsraum

- 3 Arbeitsplätze für je 3 Personen
  - Nicht essen -> macht eklige Tastaturen
  - Möglichst nichts verschütten
- Dieses Praktikum
  - Hiwis anwesend
  - Gruppen 1-3: Freitags, 11:30 – 14:30
  - Gruppen 4-6: Montags, 16:45 – 19:45
  - Gruppen 7-9: Freitags, 15:00 – 18:00
- Anderes Praktikum
- Freie Zeit bei Bedarf
  - Ggf. im Miclab fragen, ob man euch in den Raum lässt

Uhrzeit	Mo	Di	Mi	Do	Fr
8:00-9:30					
9:45-11:15					
11:30-13:00					1-3
13:15-14:45					1-3
15:00-16:30					7-9
16:45-18:15	4-6				7-9
18:30-20:00	4-6				

Raumnutzung Raum 147 (→) & Werkstatt (Keller)

Mikroprozessorlabor

- 2 Lötarbeitsplätze
- Umfangreiche Sammlung an Bauteilen für eigene Ideen
- Menschen mit Erfahrung

Werkstatt

- Platinen belichten, entwickeln, ätzen und bohren
- Mechanische Bearbeitung von allem möglichen

Betreuung

- Dennis Reimers, Freitag ab 11:30
- Karsten Hinz, Montag ab 16:45
- Jost Stolze, Freitag ab 15:00

- Ulf Kulau, IZ Raum 111
- Stephan Rottman, IZ Raum 118
- Felix Büsching, IZ Raum 132

Accounts und Zugänge

Rechner-Login

- Gruppenaccounts (lokal an den Rechnern, daher immer den gleichen PC nutzen)

Benutzer-Accounts

- http://www.ibr.cs.tu-bs.de/passwd/rz.html
- Y-Nummer registrieren

SVN-Zugang

- https://svn.ibr.cs.tu-bs.de/course-cm-ws1314-esys/
- Alles ist im SVN zu speichern, hierfür y-Account nutzen!

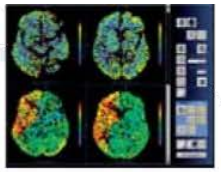
Wiki

- https://trac.ibr.cs.tu-bs.de/course-cm-ws1314-esys/wiki/
- Dokumentation, Kollaboration, Diskussion
- y-Nummer bitte **jetzt** eintragen

Ablauf

- Organisatorisches
- Grundlagen 1
- Praktikumshardware
- Aufgabe 1

Eingebettete Computer sind allgegenwärtig



Verkehr

- Haus- und Gebäudetechnik
- Industrieelektronik
- Sicherheitstechnik
- Raumfahrttechnik
- ...

Medizintechnik

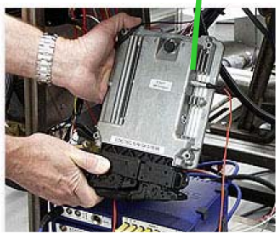
Unterhaltungs-elektronik

Kommunikation

Finanz-wesen

Quellen: Siemens, Toyota, Sony, u.a.

Eingebettete Computer mit verschiedenen Eigenschaften



robuste Computer
(Motorsteuergerät)

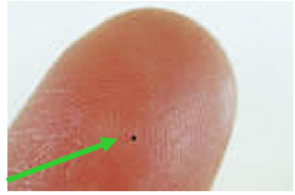
stromsparende
Computer
(Mobilkommunikation)

fehlertolerante Computer
(Raumfahrt)

leistungsfähige Computer
(z.B. Medizintechnik,
Graphik)

zugriffsgeschützte
Computer
(SmartCard)

ganz kleine
Computer
(RFID)



Microcontroller

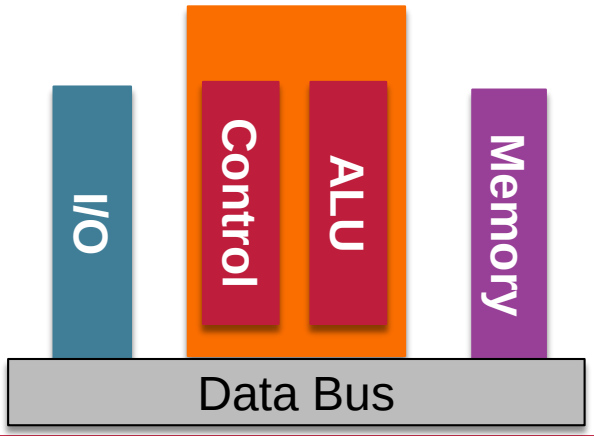
Microcontroller = Microprocessor + Peripherals

- Memory
  - Main-, program- and data-memory
    - SD-RAM, Flash, EEPROM
- Bus-controller
  - USART, UART, I²C, SPI, DMA, USB, CAN, Ethernet
- GPIO
- ADC / Comparators, DAC
- Timer

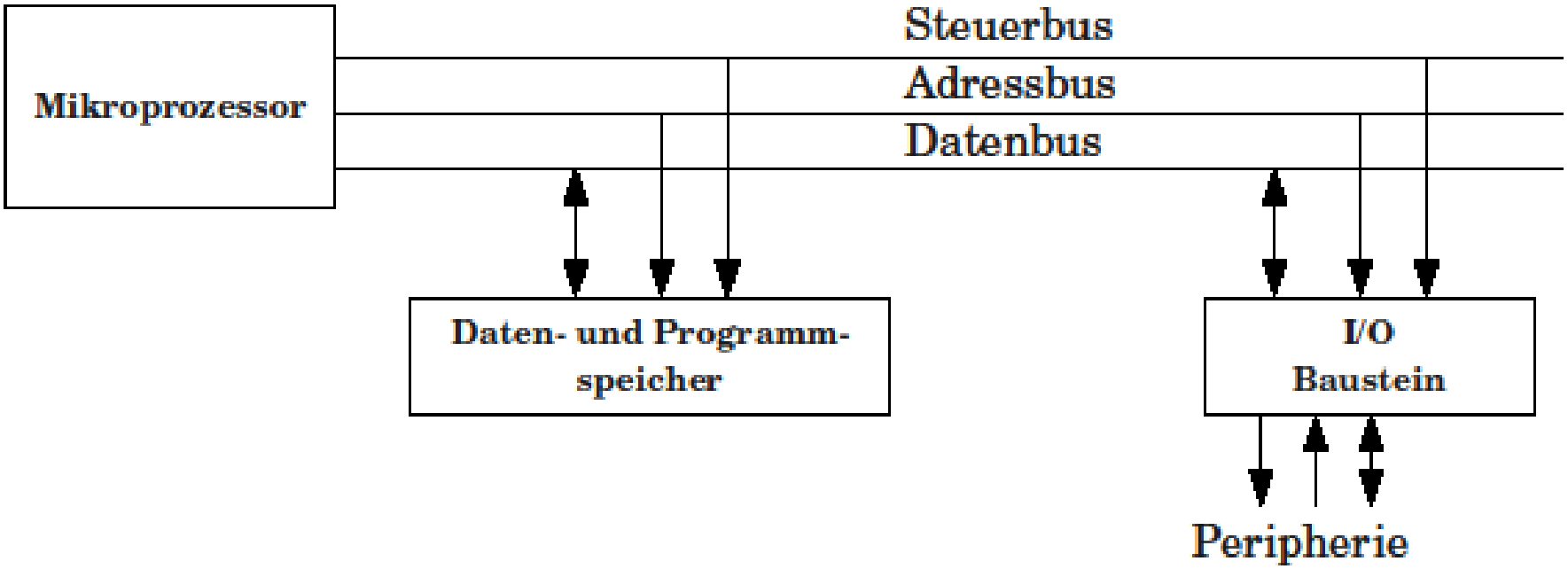
Architectures: Von Neumann vs. Harvard

Von Neumann Architecture

- 1945 developed by John von Neumann
- Simple
- Sequential (strict)
- Same memory for data and instructions
- Shared bus between data memory and program memory
  - CPU either reading instruction or reading/writing data from/to memory
  - “Von Neumann Bottleneck”
  - No race-conditions



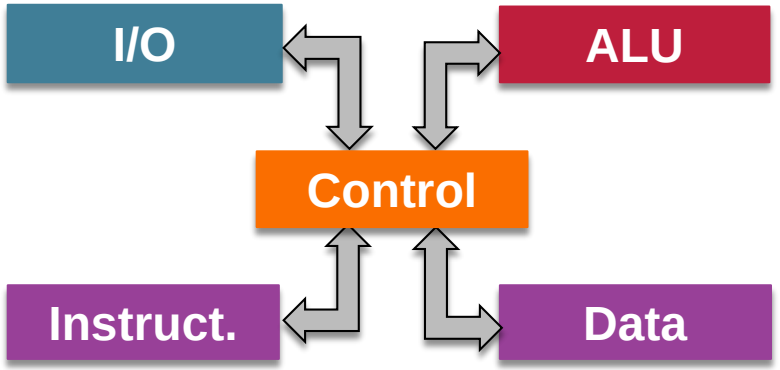
Von-Neumann-Architektur



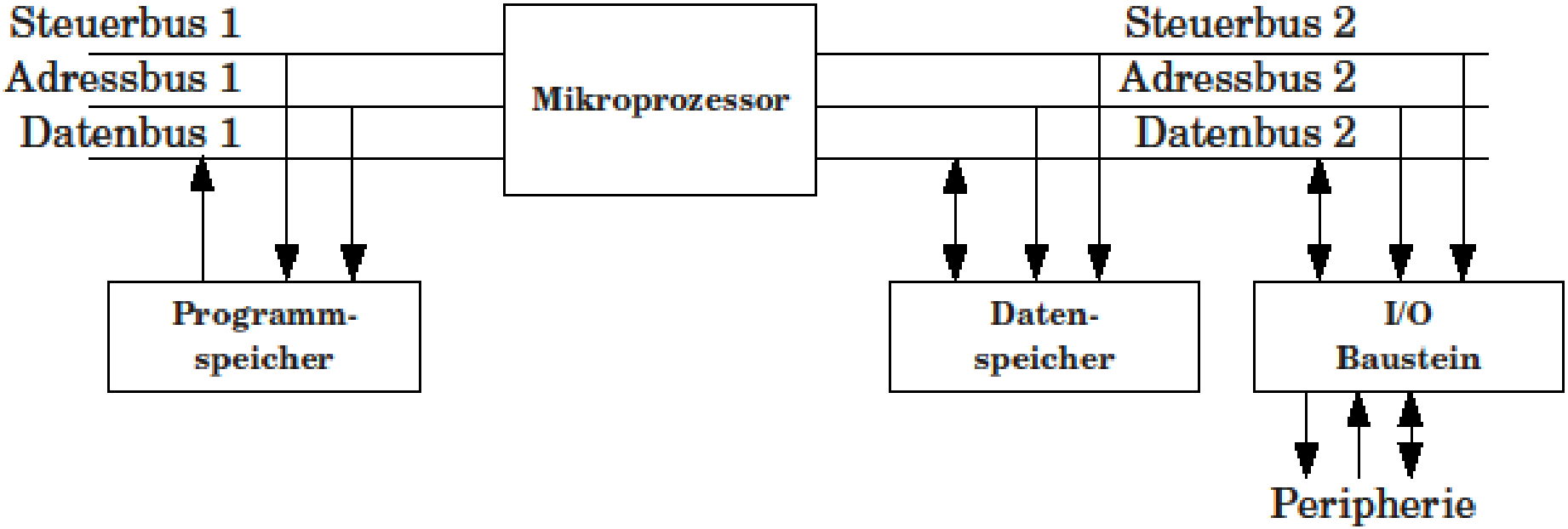
Architectures: Von Neumann vs. Harvard

Harvard Architecture

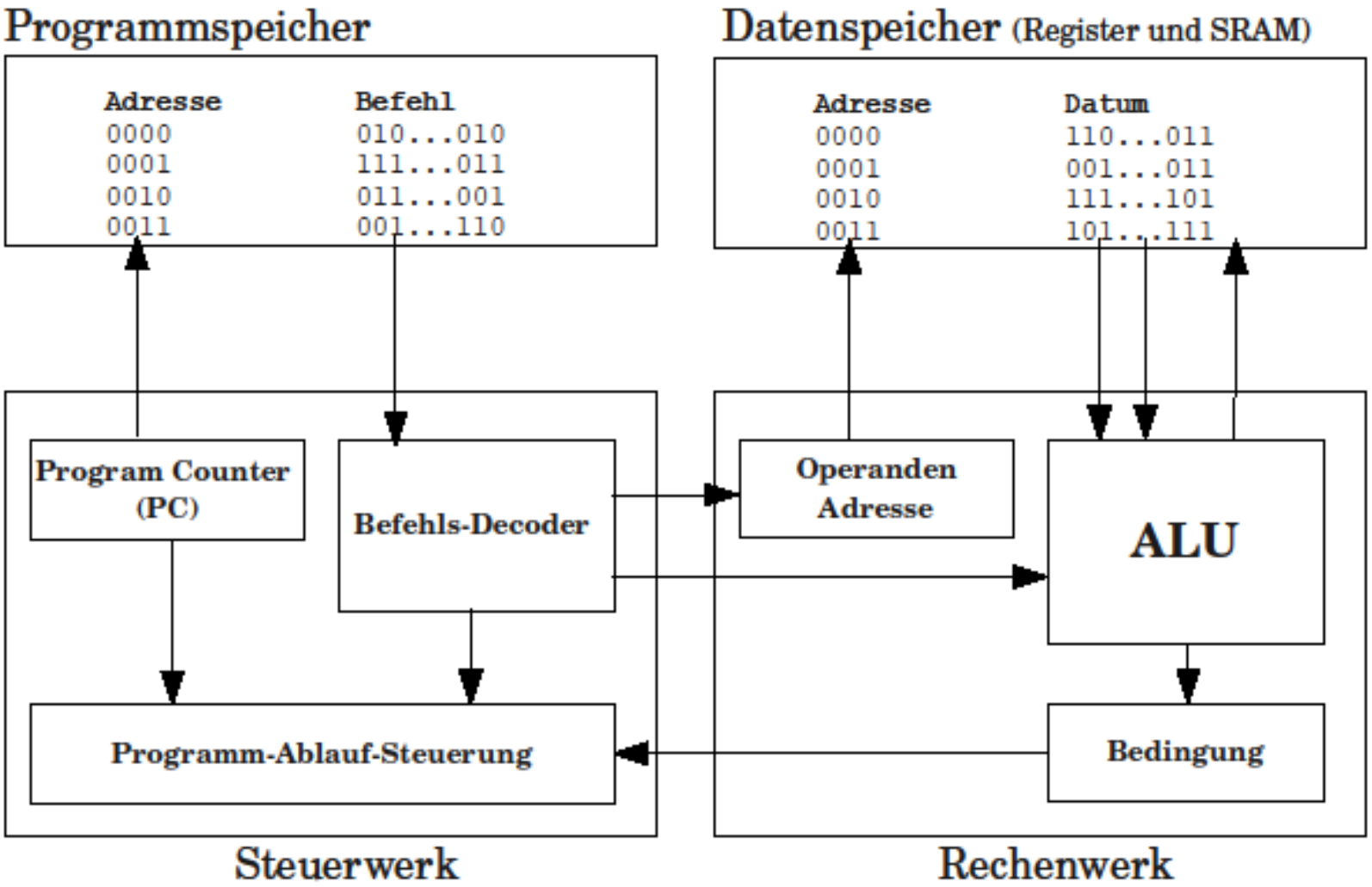
- Named after Harvard Mark I (1944)
  - IBM Automatic Sequence Controlled Calculator
- Separated storage and signal pathways
  - Physically – for instructions and data
  - Read instruction and perform data memory access at same time
- Modifications:
  - Pathway between the instruction memory and the CPU (e.g. for constants)
  - 2+ Memory Busses:
    - Relaxed separation of data / instructions



Harvard-Architektur



Steuer und Rechenwerk



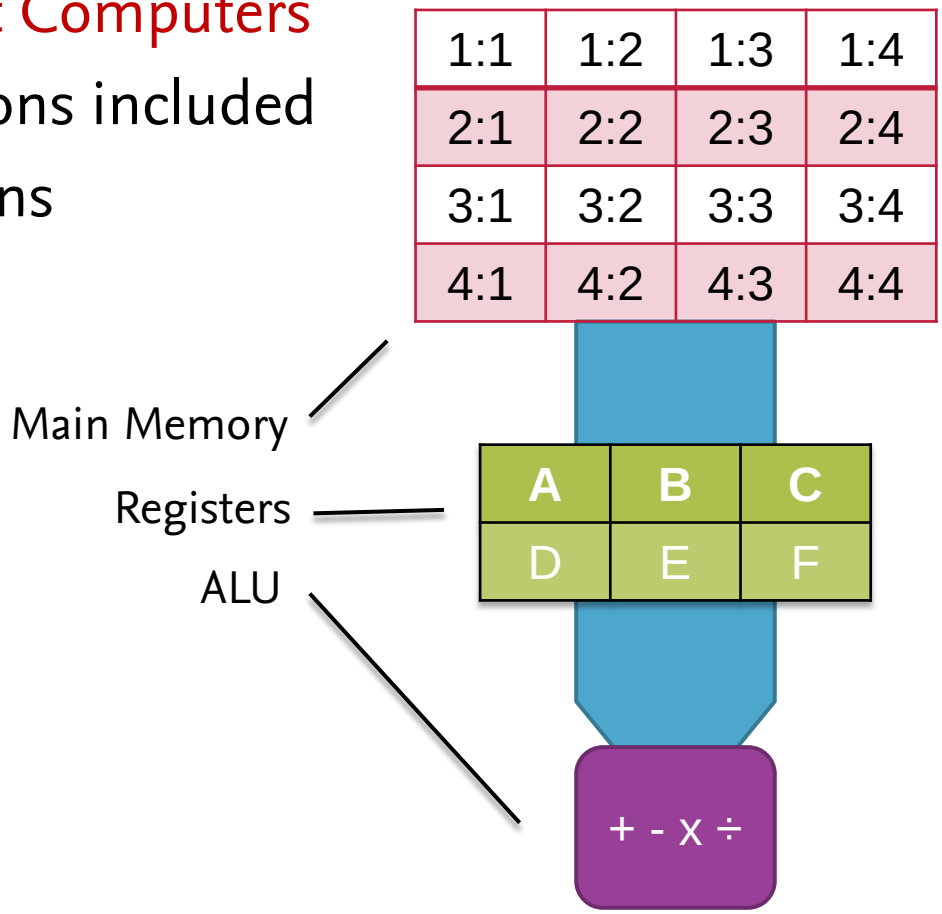
Ausführung eines binär codierten Befehls

- Befehlsadresse aus dem PC an den Programmspeicher senden
- Befehl an dieser Adresse im Programmspeicher auslesen und dekodieren
- Operationscode (was soll gemacht werden) an die ALU übergeben
- Operanden aus dem Datenspeicher holen und Operation ausführen
- Ergebnis an den Datenspeicher übergeben und
- den nächsten Befehl vorbereiten

Microprocessors and Architectures: CISC

CISC = Complex Instruction Set Computers

- Multi-clock complex instructions included
- Memory-to-Memory operations
  - LOAD and STORE inherent
- Smaller machine code
- High cycles per second
  - To achieve comparability to RISC
- Emphasis on hardware



Multiply 2 numbers

- MULT 1:3, 2:3

Microprocessors and Architectures: RISC

RISC = Reduced Instruction Set Computer

- Single-clock, only reduced instructions
- Register to register
  - LOAD and STORE independent instructions
- Larger code sizes
- Low cycles per second
- Emphasis on Software
- Pipelining allows 2-4 times faster performance than CISC
  - At similar clock rate

Multiply 2 numbers

- LOAD A, 1:3
- LOAD B, 2:3
- PROD A, B
- STORE 3:3

RISC can process more instructions per clock cycle than CISC

Microprocessors and Architectures: RISC vs. CISC

Performance

$$\frac{time}{program} = \frac{time}{cycle} * \frac{cycles}{instruction} * \frac{instructions}{program}$$

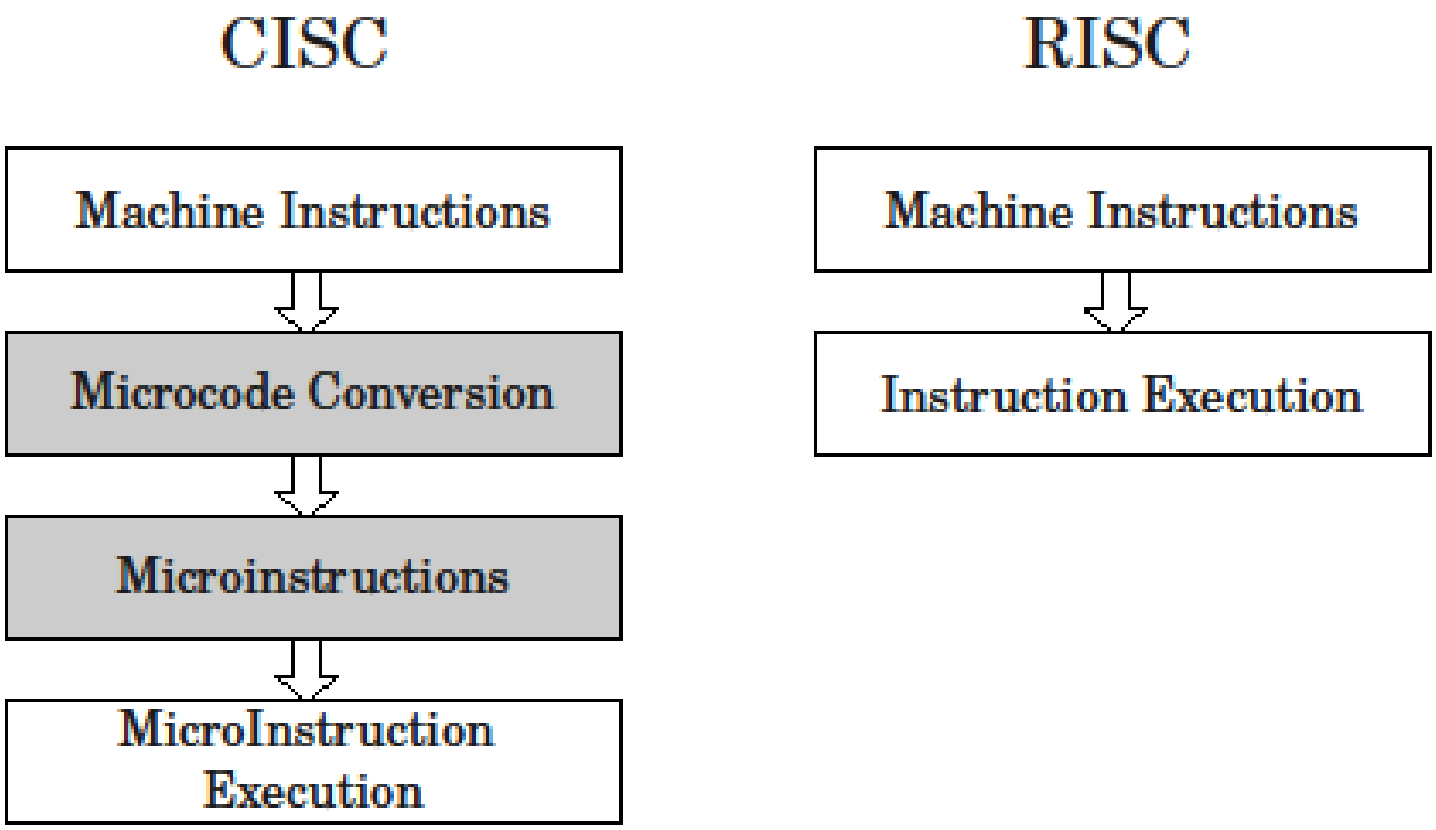
↓RISC ↓CISC

- CISC minimizes instructions / program
  - Less memory needed
  - More cycles per instructions needed
- RISC minimizes cycles / instruction
  - More memory needed
  - Less cycles per instruction needed

Memory is cheap – nowadays even in microcontrollers!

High clock rates → more energy needed

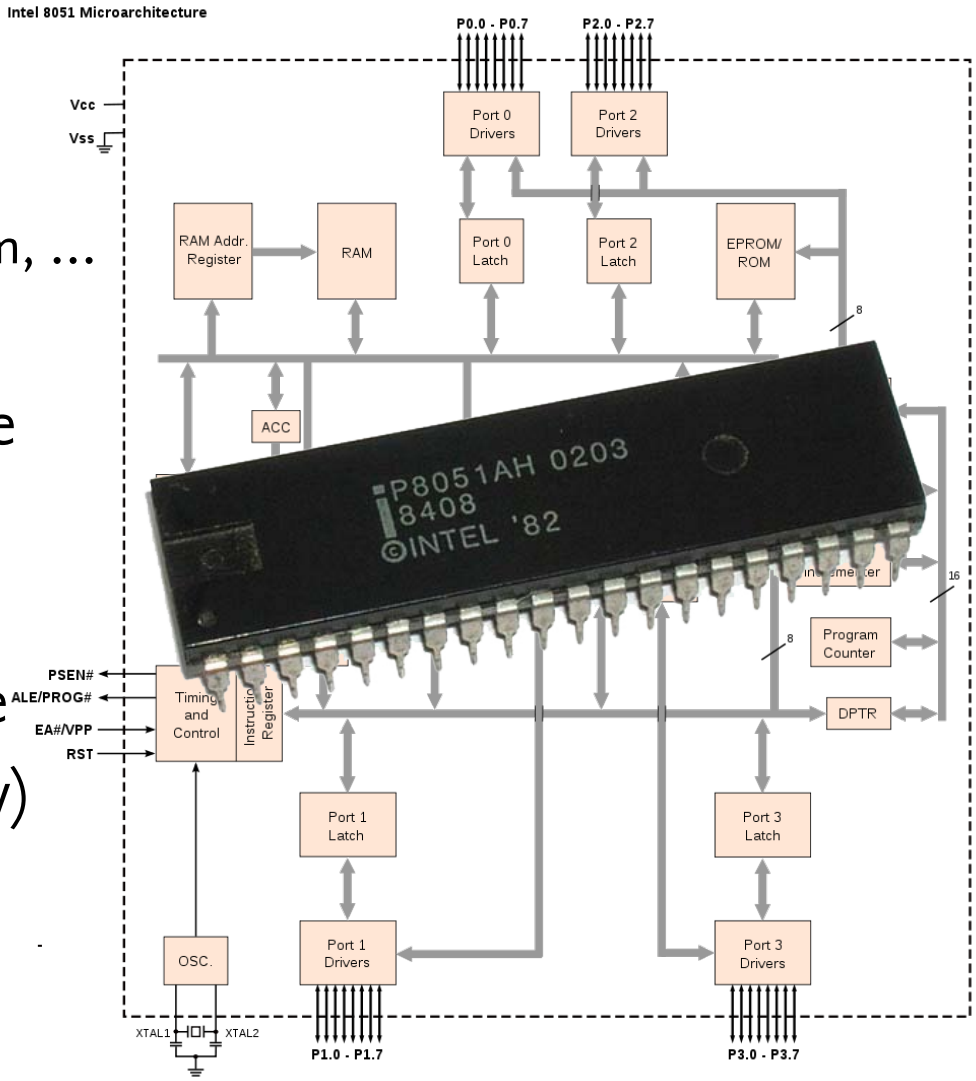
CISC und RISC



Intel 8051 – “the” microcontroller

MSC-51 (8051, 8031, 8751)

- 1980 presented by Intel
  - Built by AD, Atmel, Infineon, Maxim, …
- 8-bit, CISC
- (modified) Harvard architecture
- ≥ 128 Byte internal RAM
- External ROM, RAM
- 16 bit address bus → ≤64 Kbyte
- 4 8-bit I/O-Ports (2 for memory)
- UART (Full-Duplex)
- Timers



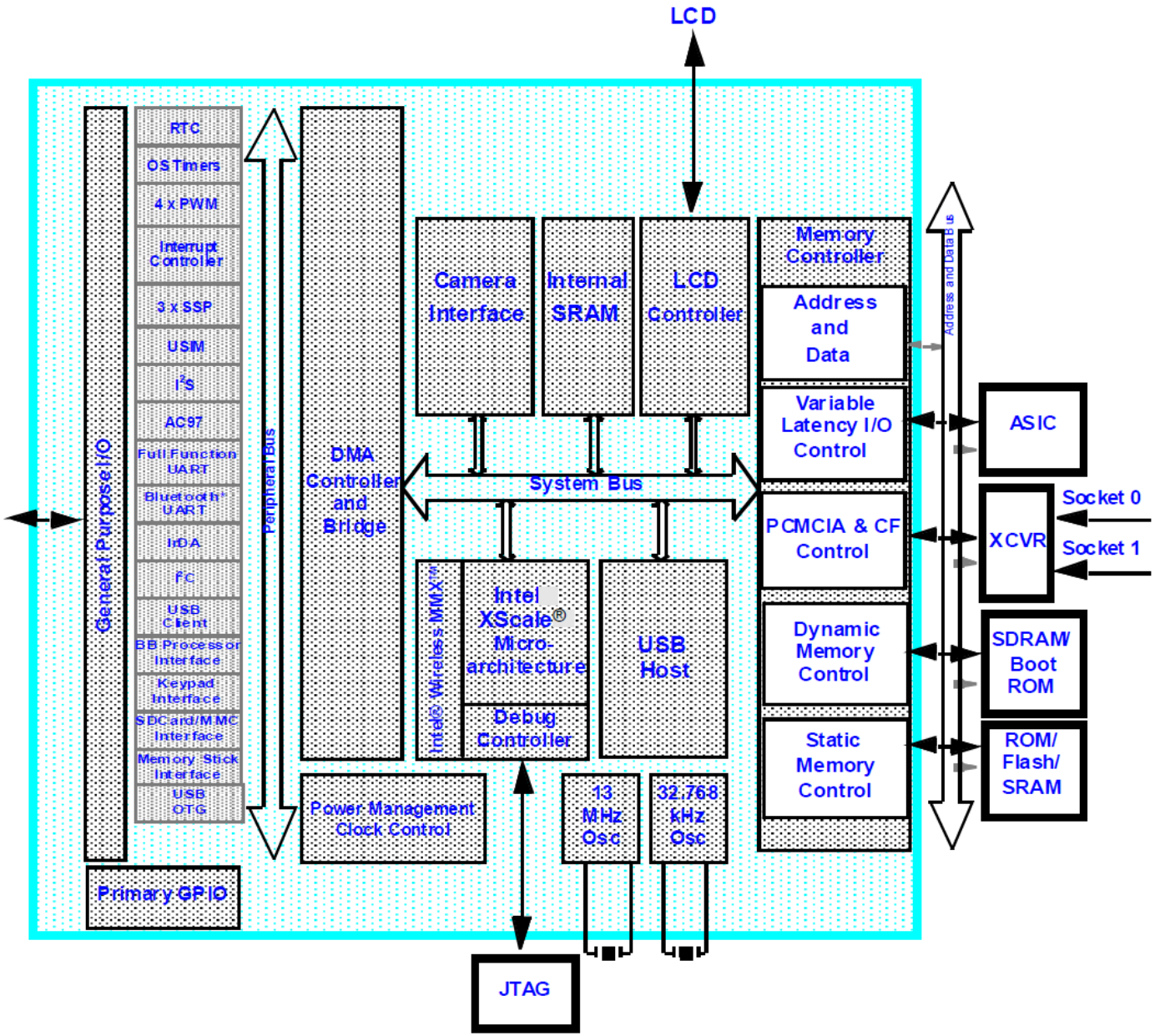
Advanced RISC Machine (ARM)

- Introduced 1987
- 32 bit, RISC
  - XNU-Kernel (Mac OS X/iOS)
  - Linux-Kernel
  - Windows NT Kernel

XScale PXA271 (ARMv5 architecture)

- Released 2004
- 13 - 416 MHz
- 32 MB Flash, 32 MB RAM
- USB (Client & Host), AC'97 Audio
- 13 MHz Active Power = 44.2 mW

XScale PXA271 (ARMv5 architecture)



Atmel AVR Series – ATMega 128A

AVR Series

- Developed in 1996
- 8-bit, RISC
- (modified) Harvard architecture

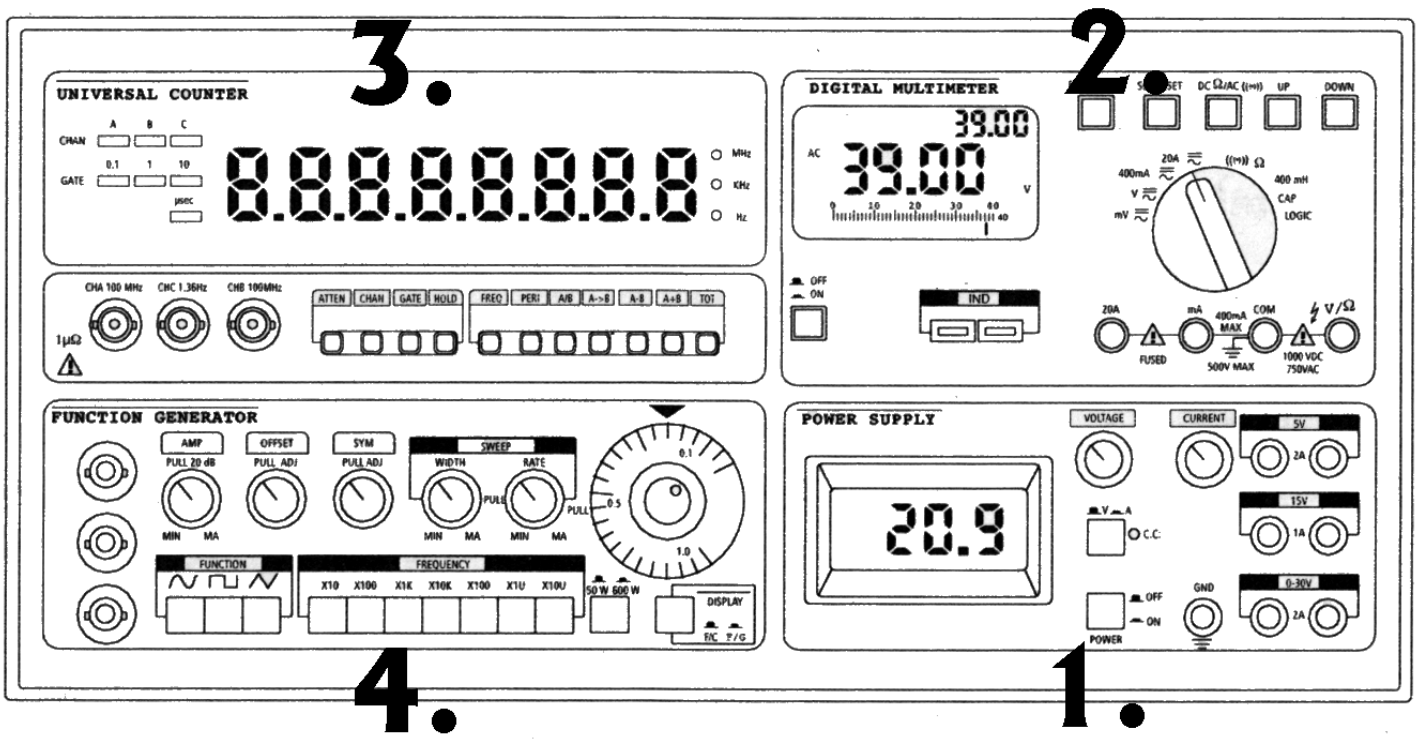
ATMega 128A

- 128 kB Flash, 4 kB SRAM, 4 kB EEPROM
- 2 USARTs, separated buses for SPI, I^2C
- 8 channel 10-bit ADC

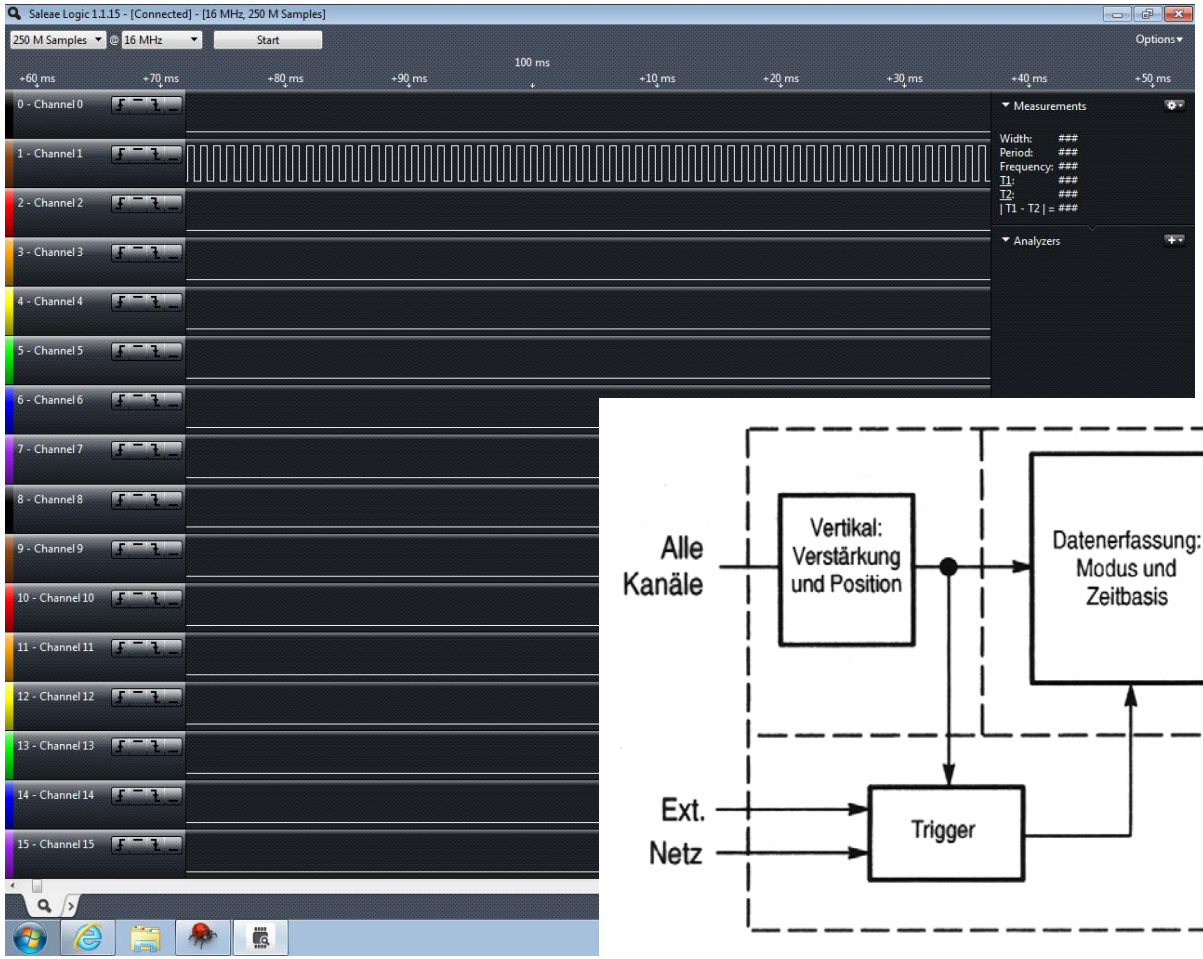
Ablauf

- Organisatorisches
- Grundlagen 1
- Praktikumshardware
- Aufgabe 1

Mess-Station

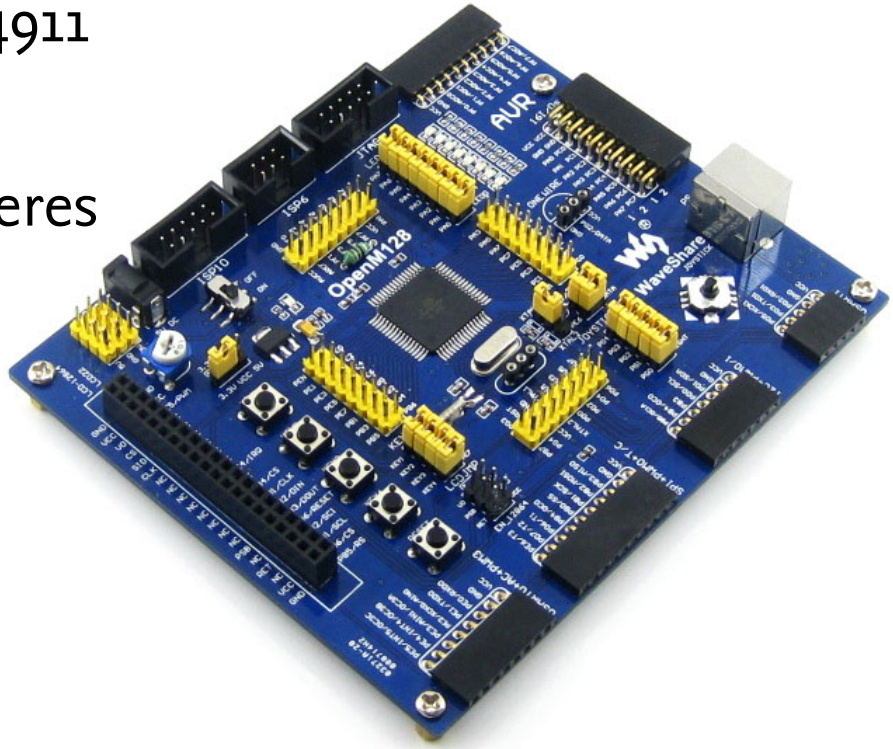


Logikanalysator & digitales Speicheroszilloskop

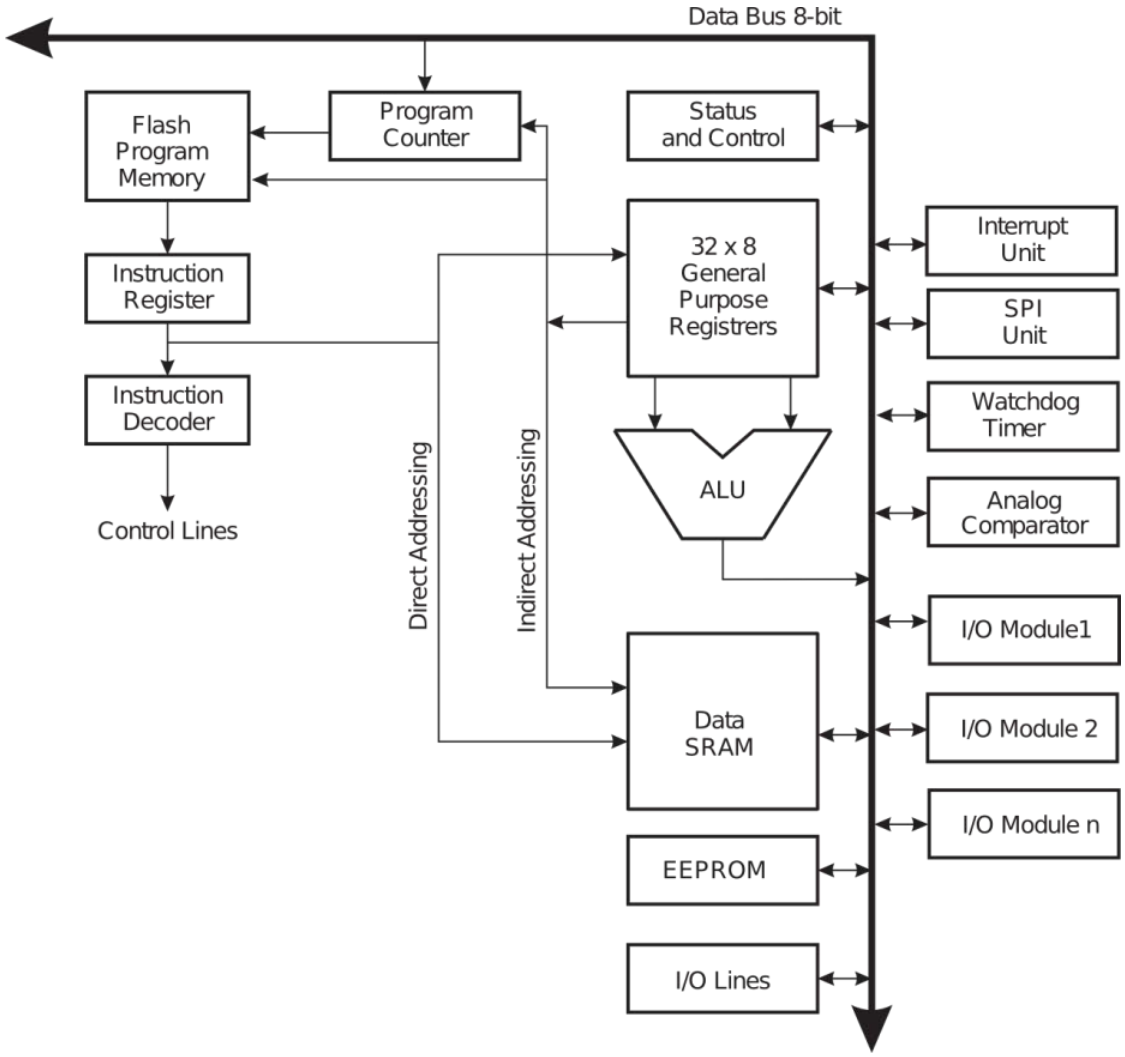


Entwicklungsboard

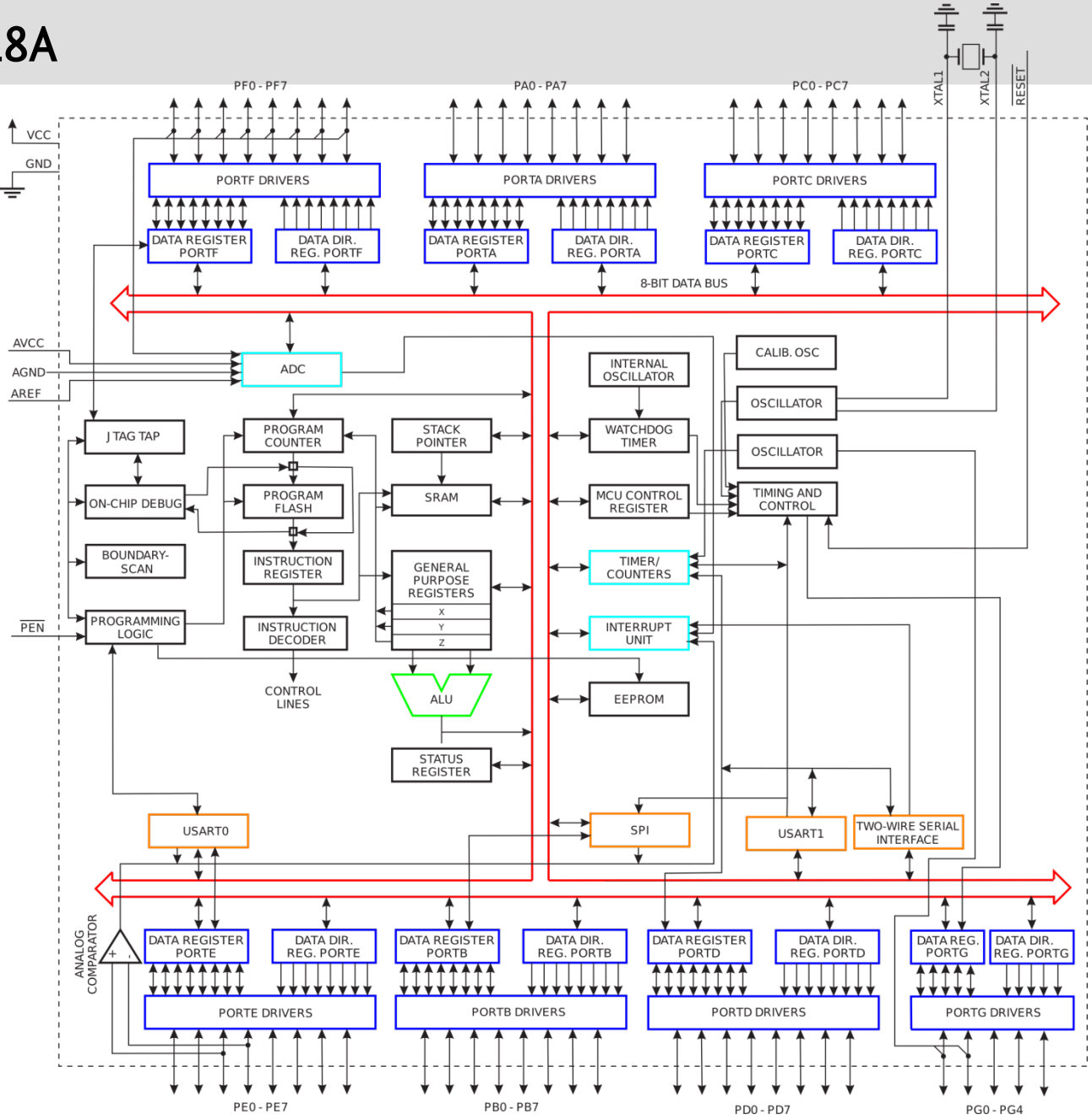
- Temperatursensor für I2C-Bus DS1621
- I2C-EEPROM AT24Cox
- SPI-Digital-Analogwandler MCP4911
- LC-Display
- Funkmodule: Bluetooth und anderes auf 2,4GHz
- weitere Speicher, Displays, Eingabemodule (Tastenfelder, ...)
  ... und vieles mehr



Architektur der Atmel AVR Serie



Atmega 128A

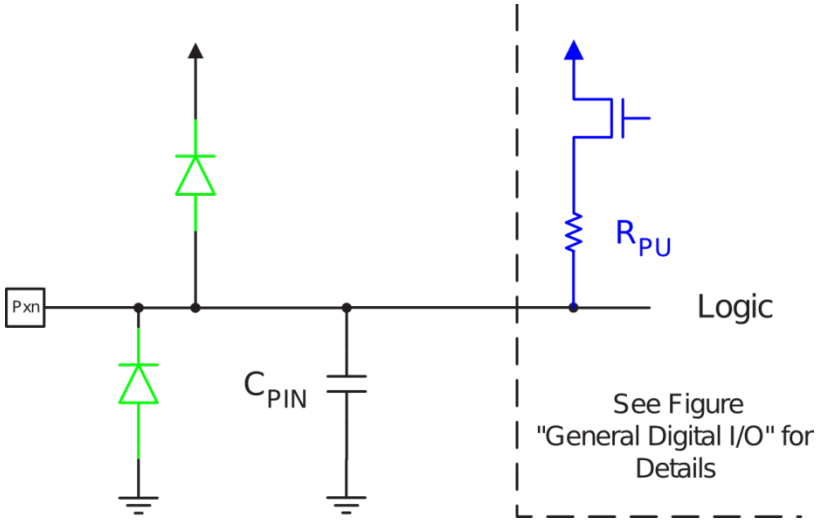

PF0 - PF7
PA0 - PA7
PC0 - PC7
XTAL1
XTAL2
RESET
VCC
GND
PORTF DRIVERS
PORTA DRIVERS
PORTC DRIVERS
DATA REGISTER PORTF
DATA DIR. REG. PORTF
DATA REGISTER PORTA
DATA DIR. REG. PORTA
DATA REGISTER PORTC
DATA DIR. REG. PORTC
8-BIT DATA BUS
AVCC
AGND
AREF
ADC
INTERNAL OSCILLATOR
CALIB. OSC
OSCILLATOR
J TAG TAP
PROGRAM COUNTER
STACK POINTER
WATCHDOG TIMER
OSCILLATOR
ON-CHIP DEBUG
PROGRAM FLASH
SRAM
MCU CONTROL REGISTER
TIMING AND CONTROL
BOUNDARY-SCAN
INSTRUCTION REGISTER
GENERAL PURPOSE REGISTERS
X
Y
Z
TIMER/ COUNTERS
PEN
PROGRAMMING LOGIC
INSTRUCTION DECODER
INTERRUPT UNIT
CONTROL LINES
ALU
EEPROM
STATUS REGISTER
USART0
SPI
USART1
TWO-WIRE SERIAL INTERFACE
ANALOG COMPARATOR
DATA REGISTER PORTE
DATA DIR. REG. PORTE
DATA REGISTER PORTB
DATA DIR. REG. PORTB
DATA REGISTER PORTD
DATA DIR. REG. PORTD
DATA REG. PORTG
DATA DIR. REG. PORTG
PORTE DRIVERS
PORTB DRIVERS
PORTD DRIVERS
PORTG DRIVERS
PE0 - PE7
PB0 - PB7
PD0 - PD7
PG0 - PG4

Peripherie des Controllers

- Digitale Ein- und Ausgänge
- Schnittstellen
  - U(S)ART
  - I²C
  - SPI
- Analoge Eingänge
  - Analog-Digital Wandler
  - Komparator
- PWM

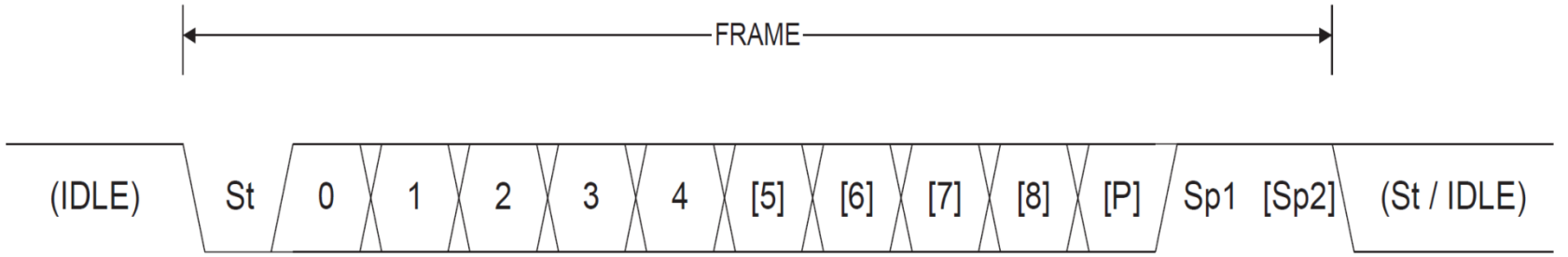
I/O-Ports

- I/O-Pins können als Ein- oder Ausgang verwendet werden
- Richtung wird mit *Data Direction Register* bestimmt
- Rudimentäre Schutzbeschaltung vorhanden
- Verwendung zum Beispiel für
  - LEDs
  - Taster
  - Steuerleitungen
  - …



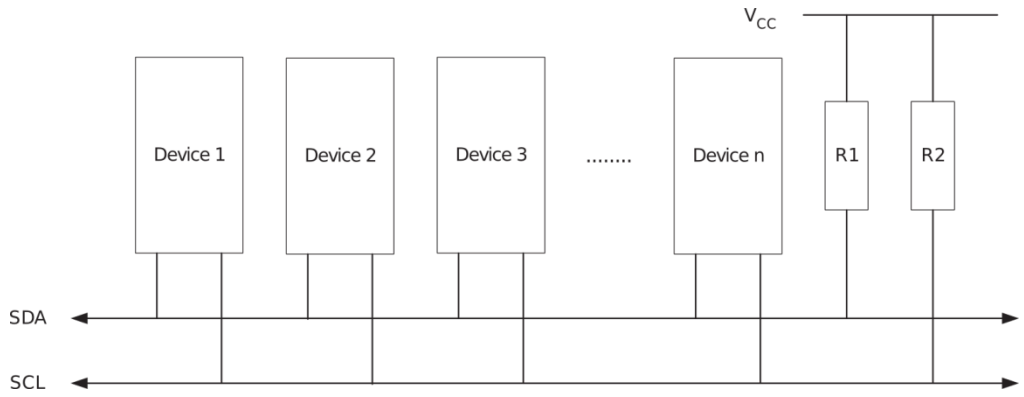
U(S)ART

- Seriell:
  - Datenbits werden *nacheinander* über eine Leitung gesendet
  - U(S)ART:
    - Universal synchronous/asynchronous Receiver/Transmitter, a.k.a. „COM-Port“
  - Asynchron: ohne Taktleitung, synchron: mit



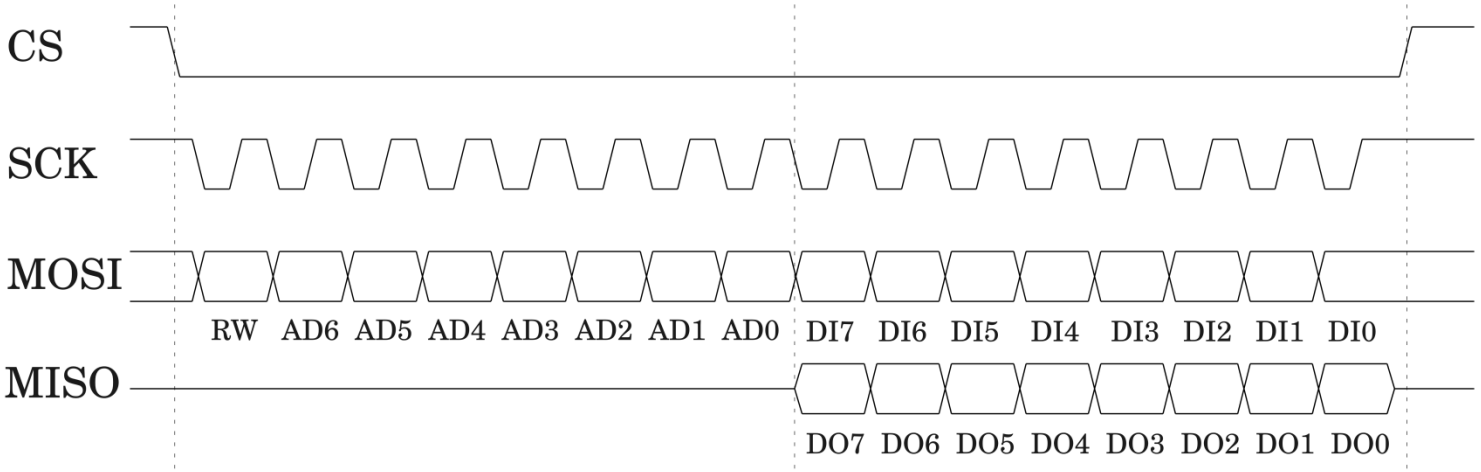
I²C

- Inter- Integrated Circuit bus
  - „Fernseher-Bus“
- Master/Slave-Bus:
  - Ein Master gibt Takt an
  - Viele Bausteine können adressiert werden
  - Pro Bus hat jeder Baustein eindeutige ID



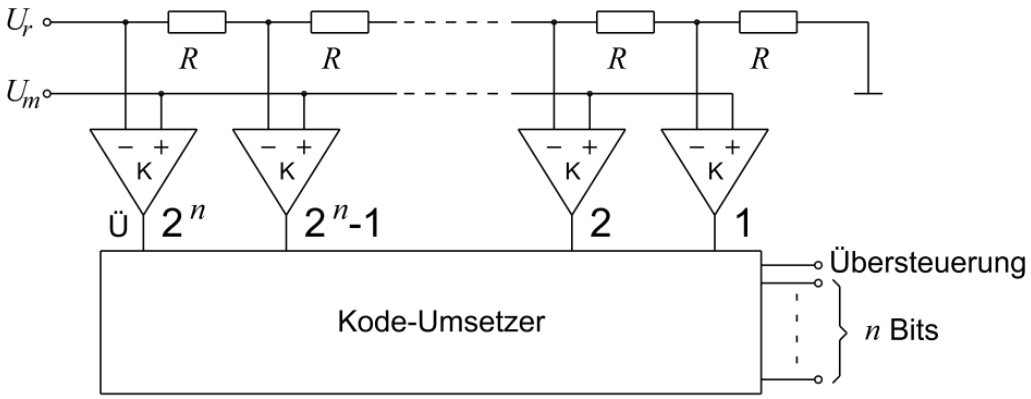
SPI

- Serial Peripheral Interface
- Master/Slave System
  - Master adressiert Slave über ChipSelect-Leitung
- 1-2 Datenleitungen, Clock-Leitung, CS



Analog-Digitalwandler

- Unterschiedliche Verfahren
  - Flash-Wandler
    - Hoher Hardwareaufwand
    - schnell
  - Sukzessive Approximation:
    - Ein Komparator
    - Vergleichsspannung wird nachgeregelt
    - Langsamer



Ablauf

- Organisatorisches
- Grundlagen 1
- Praktikumshardware
- Aufgabe 1