



Berner Fachhochschule
Haute école spécialisée bernoise
Bern University of Applied Sciences

Hardwarenahe Softwareentwicklung

Bussysteme und Speicher

V5.1, ©2023 roger.weber@bfh.ch

Lernziele

Sie sind in der Lage:

- ▶ Einfache Schaltpläne für Microcontroller, Speicher und Peripherie zu analysieren.
- ▶ Eine Memory Map zu erstellen.



Inhaltsverzeichnis

1. Kommunikation zwischen CPU, Speicher und Peripherie

- Bus-Systeme

- Memory-Map

2. Programm- und Daten-Speicher

- Speicher-Technologien

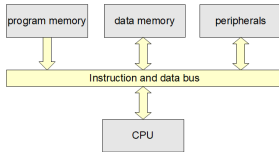
- Speicher-Architekturen

- Speicher-Hierarchie

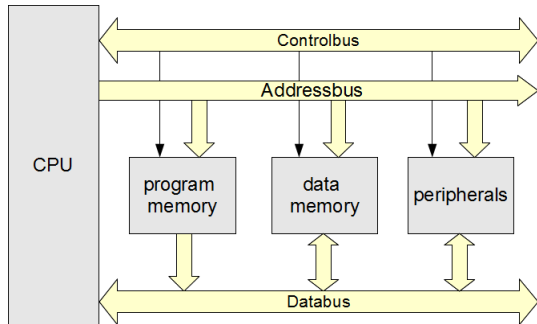
Kommunikation zwischen CPU, Speicher und Peripherie

Bus-Systeme

- ▶ Die Funktionseinheiten in einem Computersystem werden über Busse miteinander verbunden.
- ▶ Busse übertragen Daten, Adressen und Steuersignale.
→ Datenbus, Adressbus und Controlbus



Von-Neumann Architektur



Bus-Systeme

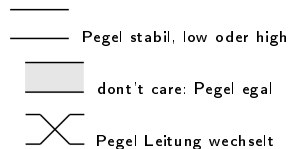
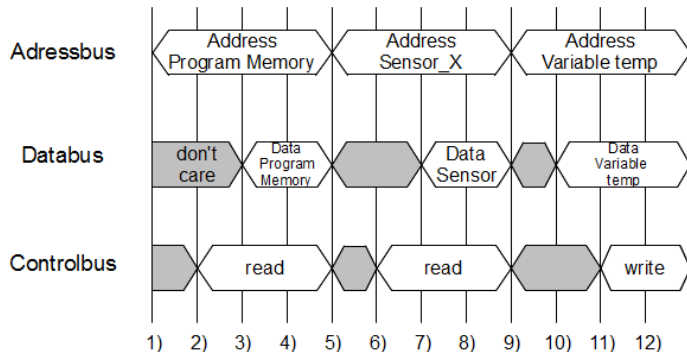
- ▶ **Adressbus:** Adressinformationen für Speicher oder Peripherie. Adressen werden vom Steuerwerk der CPU generiert.
→ mit wem?
- ▶ **Datenbus:** Programminstruktionen und Benutzerdaten.
→ was?
- ▶ **Controlbus** (Steuerbus): Steuerinformationen (Read/Write, Reset, Interrupts, Requests und Acknowledge, Handshake ...).
→ wie?



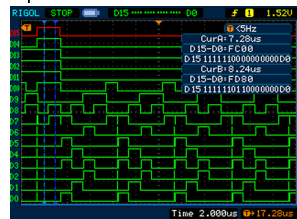
Bus-Systeme

Beispiel C-Code: `int temp = Temperatur_Sensor_x;`

Welche Signale können mit einem Logic-Analyzer auf dem Bus gemessen werden?



Beispiel Databus



Beispiel Bustiming



Diskutieren Sie die Signale von Adress-, Daten- und Controlbus für folgendes C-Code Beispiel:

```
/* defines for dip-switch and rgb addresses */
#define DIPSWITCH *((volatile unsigned short *) 0xC8000020)
#define RGBLINE  *((volatile unsigned short *) 0XC8000400)

int main(void)
{
    /* local variable to store dip-switch value */
    register uint16_t switch_value;

    /* Initialize Hardware */
    ....

    /* Main loop */
    for (;;) {
        switch_value = DIPSWITCH; /* read dip-switch state */
        RGBLINE = switch_value;   /* write to rgb-line */
    }
    return (0);
}
```


Memory-Map

- ▶ Die Memory-Map ist ein Speicherbelegungsplan. Sie gibt an, auf welchen Adressen Speicher und Peripherie liegen.
- ▶ Beispiel für einen 8-Bit Microcontroller mit 16-Bit Adressbus:

	Address	A15	A14	A13	A12	A11	A0
Periphery	0xFFFF	1	1	1	1		
	0xE000	1	1	1	0		
Not used							
RAM2 (8 kByte)	0xBFFF	1	0	1	1		
	0xA000	1	0	1	0		
RAM1 (8 kByte)	0x9FFF	1	0	0	1		
	0x8000	1	0	0	0		
Flash (32 kByte)	0x7FFF	0	1	1	1		
	0	0	0	0	0		



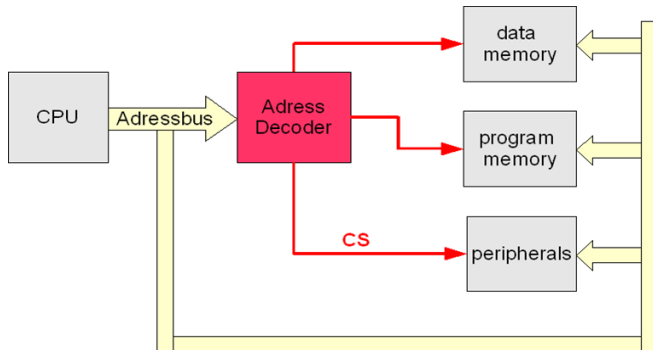
- ▶ Welches Bauteil liegt auf Adresse 0x9800?
- ▶ Welches ist die End-Adresse des Flash?
- ▶ Wie viele Pins hat RAM2?

Memory-Map

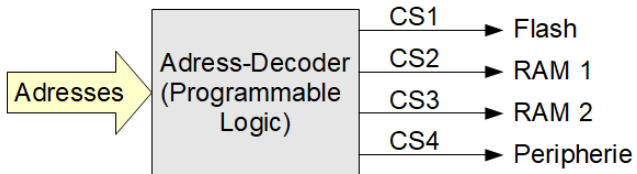
- ▶ 8-Bit Microcontroller: 16-Bit Adressbus (A0 bis A15)
→ total 64 kB Speicher und Peripherie
- ▶ 32-Bit Microcontroller: 32-Bit Adressbus (A0 bis A31)
→ 4 GB adressierbar
- ▶ Grösse der verwendeten Speicher:
 - ▶ 8-Bit Microcontroller: einige KB (kein Betriebssystem, nur einfache Applikationen).
 - ▶ 32-Bit Microcontroller: bis zu einigen MB, teilweise mit Betriebssystem wie FreeRTOS.

Adressdecoder

- ▶ Adressdecoder selektieren einzelne Bausteine oder Baugruppen.
- ▶ CS-Signal werden vom Adressdecoder generiert.
- ▶ Aufbau eines Adressdecoders:
 - ▶ diskrete Logikbausteine (AND, OR, 1 of n Decoder)
 - ▶ Programmierbare Logikbausteine (EE-PLD, CPLD, FPGA, ...)



Adress-Decoder mit programmierbarer Logik



Beispiel:

$CS1 = \neg A15$

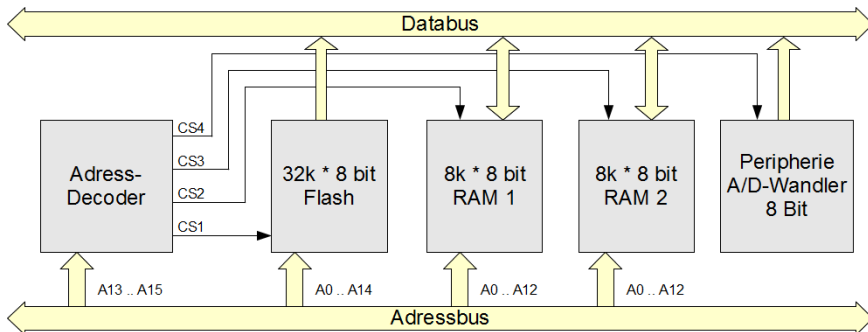
$CS2 = A15 * \neg A14 * \neg A13$

$CS3 = A15 * \neg A14 * A13$

$CS4 = A15 * A14 * A13$

Blockschaltbild mit Adressdecoder

Beispiel: Adressdecoder, Speicher und Peripherie für
32k Flash, 2 * 8k RAM sowie A/D-Wandler mit 1 Byte:



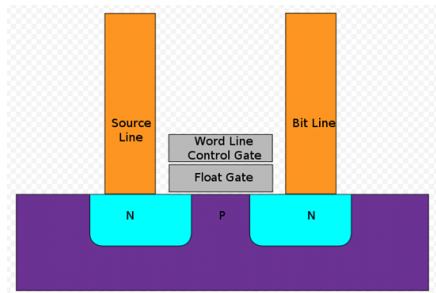
Programm- und Daten-Speicher

Speicher-Technologien, nichtflüchtige Speicher

- ▶ Der Inhalt bleibt auch nach dem Ausschalten der Versorgungsspannung erhalten.
- ▶ Bezeichnung: ROM (Read Only Memory)
- ▶ Speichertypen dieser Kategorie sind:
 - ▶ **ROM**
 - ▶ **EPROM** (Erasable Programmable ROM)
 - ▶ **OTP** (oder OTP ROM, One Time Programmable ROM)
 - ▶ **EEPROM** (Electrically Erasable Programmable ROM)
 - ▶ **FLASH** (FLASH EEPROM)
 - ▶ **FRAM** (Ferroelectric Random Access Memory)
- ▶ Verwendung: Programmspeicher, Konfigurations- und Produktionsdaten

Flash

- ▶ Kann elektrisch neu programmiert werden.
- ▶ Ideal für den Download neuer SW-Versionen.
- ▶ Information wird in einem Floating-Gate gespeichert.
- ▶ NOR flash: Standard Address- / Databus, kann andere ROM-Bausteine ersetzen.
- ▶ NAND flash: Braucht zusätzlichen Memory Controller oder Software. Grössere Speichermenge und billiger als NOR-flash.

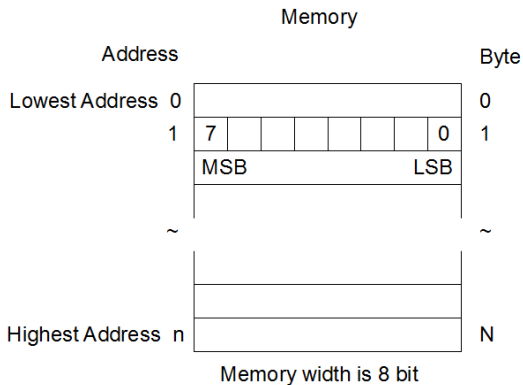


Speicher-Technologien, flüchtige Speicher

- ▶ Der Inhalt geht nach dem Ausschalten der Versorgungsspannung verloren.
- ▶ Bezeichnung: **RAM** (Random Access Memory)
- ▶ Speichertypen dieser Kategorie sind:
 - ▶ **SRAM** (Static RAM)
 - ▶ **PSRAM** (Pseudo Static RAM)
 - ▶ **DRAM** (Dynamic RAM)
 - ▶ **SDRAM** (Synchronous Dynamic RAM, getaktete Variante der DRAMs)
 - ▶ **DDR-SDRAM** (Double Data Rate SDRAM)
- ▶ Verwendung: Datenspeicher

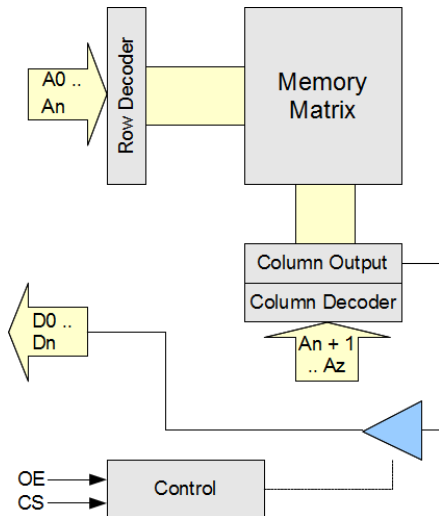
Speicher-Organisation

- ▶ Daten werden als Array von einzelnen Speicherwörtern organisiert:
- ▶ 1, 8 oder 16 Bit breit.
- ▶ Mit "m" Adressleitungen können 2^m Speicherwörter adressiert werden.



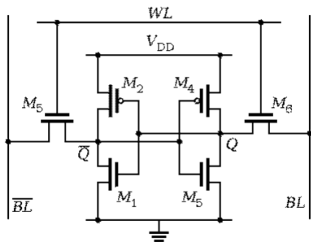
Architektur von ROM-Bausteinen

- ▶ Information in einem Array von Speicherzellen (Memory-Matrix).
- ▶ **CS** (Chip Select) zur Selektion des Bausteins (Adressdecoder).
- ▶ **OE** (Output Enable, read) zum aktivieren der Ausgangstreiber.
- ▶ Zusätzliche Steuerleitungen für die Programmierung.

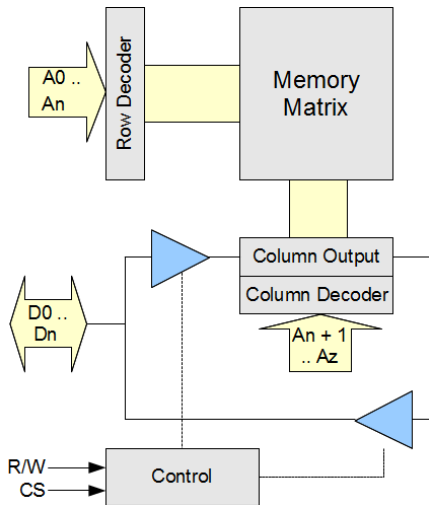


Architektur von SRAM-Bausteinen

- ▶ Ähnlich wie ROM
- ▶ Datenbus ist bidirektional (lesen und schreiben).
- ▶ R/W-Signal (Read/Write) als Steuerleitung.

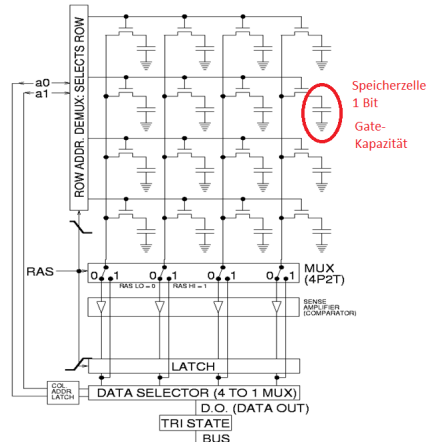
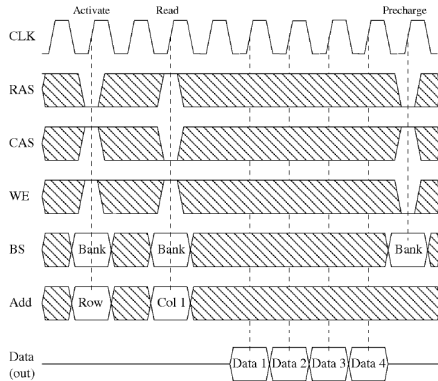


Speicherzelle 1 Bit



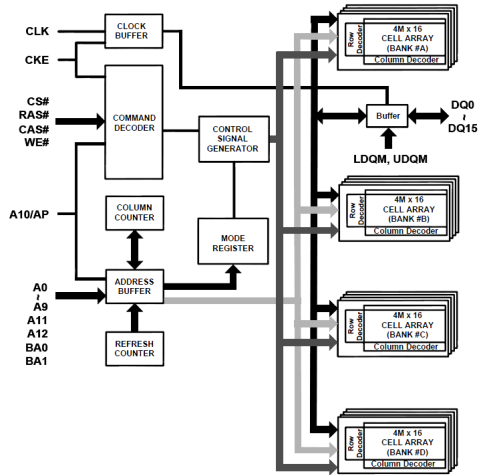
Architektur von SDRAM-Bausteinen

- ▶ Adressierung im Multiplexverfahren: Row-Address und Column-Address.
- ▶ zusätzliche Steuersignale:
 - ▶ Row-Address-Strobe (RAS)
 - ▶ Column-Address-Strobe (CAS)



Blockdiagramm eines SDRAM-Bausteines

AS4C16M16SA von Alliance Memory (4M * 16 Bit * 4 Banks, 256Mb)



- ▶ Speichergrösse (in Bit oder Byte)
- ▶ Pinbelegung
- ▶ Speicherorganisation (Bit, Byte oder Word)
- ▶ DC-Charakteristiken (Spannungen, Stromverbrauch ...)
- ▶ AC-Charakteristiken (Zugriffszeiten...)
- ▶ Wahrheitstabellen (Steuerlogik)

Features

- Fast access time from clock: 5 ns
- Fast clock rate: 166 MHz
- Fully synchronous operation
- AEC-Q100 Compliant
- Internal pipelined architecture
- 4M word x 16-bit x 4-bank
- Programmable Mode registers
 - CAS Latency: 2 or 3
 - Burst Length: 1, 2, 4, 8, or full page
 - Burst Type: Sequential or Interleaved
 - Burst stop function
- Auto Refresh and Self Refresh
- 8192 refresh cycles/32ms
- CKE power down mode
- Automotive Ambient Temperature: -40~105°C
- Single +3.3V ±0.3V power supply
- Interface: LVTTTL
- 54-pin 400 mil plastic TSOP II package
 - Pb free and Halogen free
- 54-ball 8.0 x 8.0 x 1.2mm (max) FBGA package
 - Pb free and Halogen free

Overview

The 256Mb SDRAM is a high-speed CMOS synchronous DRAM containing 256 Mbits. It is internally configured as 4 Banks of 4M word x 16 DRAM with a synchronous interface (all signals are registered on the positive edge of the clock signal, CLK). Read and write accesses to the SDRAM are burst oriented; accesses start at a selected location and continue for a programmed number of locations in a programmed sequence. Accesses begin with the registration of a BankActivate command which is then followed by a Read or Write command.

The SDRAM provides for programmable Read or Write burst lengths of 1, 2, 4, 8, or full page, with a burst termination option. An auto precharge function may be enabled to provide a self-timed row precharge that is initiated at the end of the burst sequence. The refresh functions, either Auto or Self Refresh are easy to use.

By having a programmable mode register, the system can choose the most suitable modes to maximize its performance. These devices are well suited for applications requiring high memory bandwidth and particularly well suited to high performance PC applications.

Table 1. Key Specifications

AS4C16M16SA-Automotive		-6
tCK3	Clock Cycle time(min.)	6 ns
tAC3	Access time from CLK (max.)	5 ns
tRAS	Row Active time(min.)	42 ns
tRC	Row Cycle time(min.)	60 ns

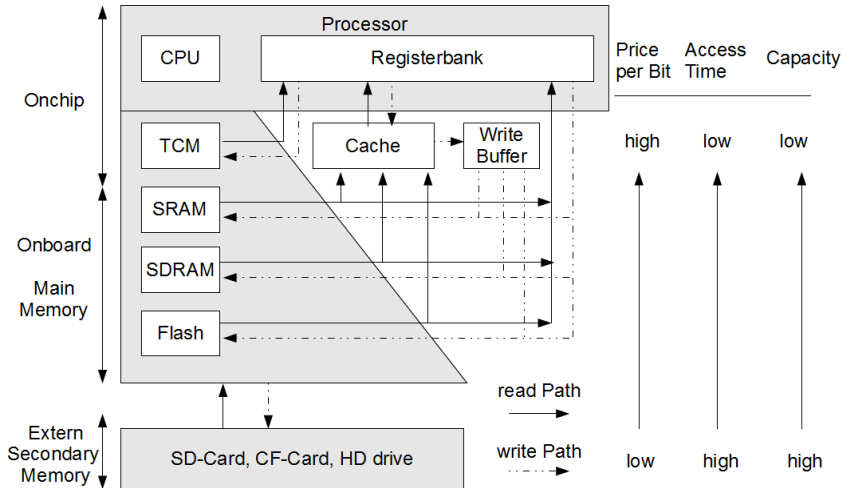
Table 2. Ordering Information

Part Number	Frequency	Package	Temperature	Temp Range
AS4C16M16SA-68AN	166MHz	54-Ball FBGA	Automotive	-40~105°C
AS4C16M16SA-6TAN	166MHz	54-Pin TSOPII	Automotive	-40~105°C

B: indicates FBGA package

T: indicates TSOP II package

Speicher-Hierarchie



Speicher-Hierarchie

- ▶ **Register**

Schnellster Speicher im System. Dient als Speicher für Operanden und Resultate der ALU.

- ▶ **Cache**

Schneller Zwischenspeicher zwischen Registern und Arbeitsspeicher.

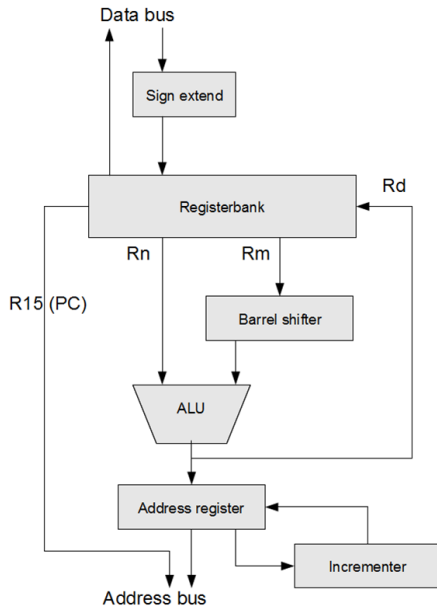
- ▶ **Arbeitsspeicher**

- ▶ Speicher für Programmcode (Flash) und Daten (SRAM, SDRAM).
- ▶ PC-Welt: Programmcode wird vom Massenspeicher ins RAM kopiert und aus dem RAM ausgeführt.
- ▶ Embedded-Welt: Programmcode wird aus dem Flash ausgeführt oder beim Bootvorgang vom Flash ins RAM kopiert.

- ▶ **Massenspeicher**

Festplatten, CF-Cards, SD-Cards oder Memory-Sticks.

Register



Fragen ?



1. Welche der folgenden Memory-Technologien ist die schnellste?
 - Flash
 - SDRAM
 - SRAM
 - EPROM
2. Wie viele Adressen können mit einem 32-bit Adressbus angesprochen werden?
3. Wie weiss der Linker auf welchen Adressen sich RAM und Flash befinden?