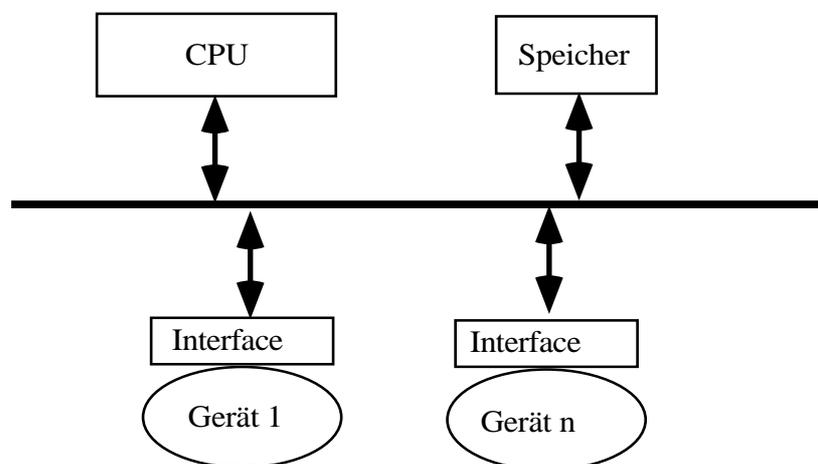


Kapitel 5 Ein- Ausgabeeinheiten

Am gemeinsamen Bus einer CPU hängt neben dem Hauptspeicher die Peripherie des Rechners: d. h. sein Massenspeicher und die Ein- Ausgabeeinheiten. Diese externen Geräte sind mit dem Bus über Interfaceschaltungen verbunden.



5.1. (Externer) Bus

Ein Bus besteht aus mehreren Leitungsbündeln, die verschiedene Funktionen abdecken.

- Datenleitungen (8, 16, 32, 64 Bit)
- Adressleitungen (16, 32, 48 Bit)
(Bei kleinen Rechnern ggf. mit den Datenleitungen gemultiplext:
z. B. D0 ... D7 = A0 ... A7. Adressleitungen A8 ... A15 extra Leitungen)
- Funktionsleitungen
 - lesen/schreiben
 - Daten/Adressen (falls Daten und Adressen gemultiplext)
 - Halbwort / Ganzwort - Zugriff
- Synchronisationsleitungen
Taktleitungen und Leitungen zur Angabe der Gültigkeit von Daten/Adressen
z. B. AS = address strobe : Adresse ist gültig
DR = data ready: Daten stehen bereit
DTACK = data transfer acknowledge: Quittung
- Steuerleitungen (z. B. reset, ...)

- Busvergabeleitungen (für DMA, ...)
- Interruptleitungen (signalisieren asynchrone Ereignisse von außen)

5.2. Einfaches Interface

Es ist ein passives Gerät, adressiert von der CPU, die Kommandos, Statusabfragen und Daten Ein- Ausgabe veranlaßt. Es kann ggf. von sich aus Interrupts erzeugen.

Die CPU muß fast alles selber machen (in Form eines Treiberprogramms für das angeschlossene Gerät). Typische Anwendungen: Lesen einer Tastatur, Lesen einer Maus.

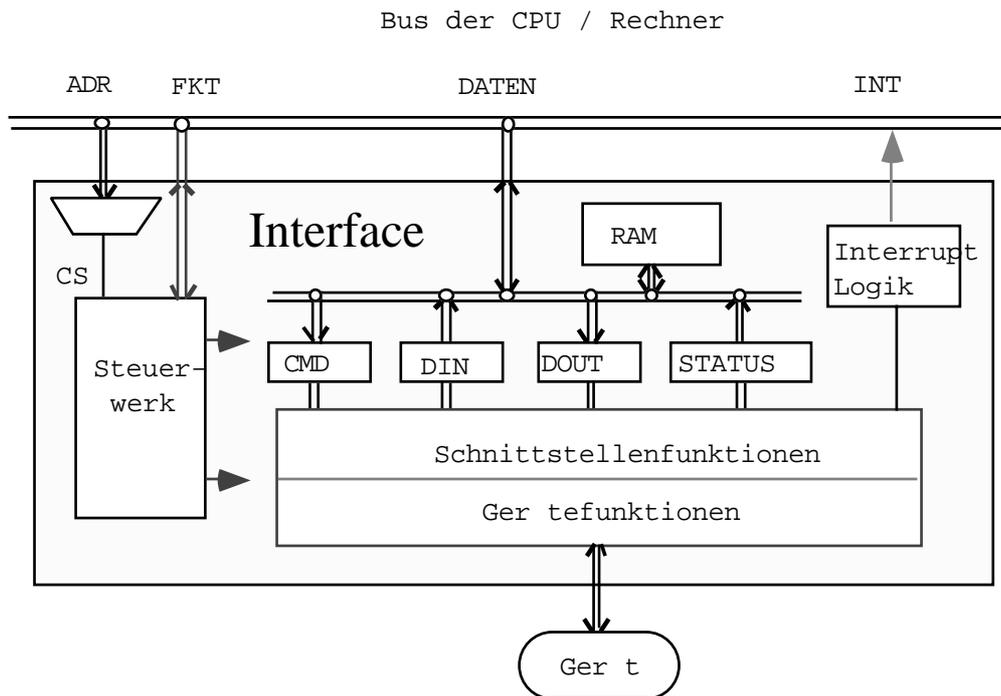
Die einzelnen Aufgaben lassen sich unterteilen in

- Adressieren des Geräts durch den Rechner und Entschlüsseln der Adresse;
- dem Gerät sagen, was es tun soll.
 - Steuerleitungen aktivieren (z. B. reset)
 - Eine gerätespezifische Adresse benutzen, um dem Gerät ein Kommando zu geben. Das Wort muß in ein Kommandoregister eingeschrieben werden, um die unterschiedlichen Geschwindigkeiten auszugleichen.
 - Vom Gerät den Status erfragen. Hierzu ist ein Statusregister im Interface nötig, das von der CPU gelesen werden kann.
 - Daten vom/zum Gerät übertragen. Auch hier sind Pufferregister nötig.

Daneben realisiert das Interface

- Schnittstellenfunktionen (z. B. Daten wandeln seriell/parallel, ..., oder Pegel umwandeln)
- Gerätefunktionen Sie sind gerätespezifisch.

Vom Gerät zum Rechner hin kann i. allg. ein Interruptsignal erzeugt werden, mit dem das Gerät z. B. eine Fertigmeldung gibt oder eine Störung bei der Ausführung anzeigt. Die CPU wird das Statusregister lesen, in dem die Ursache des Interrupts angezeigt wird.

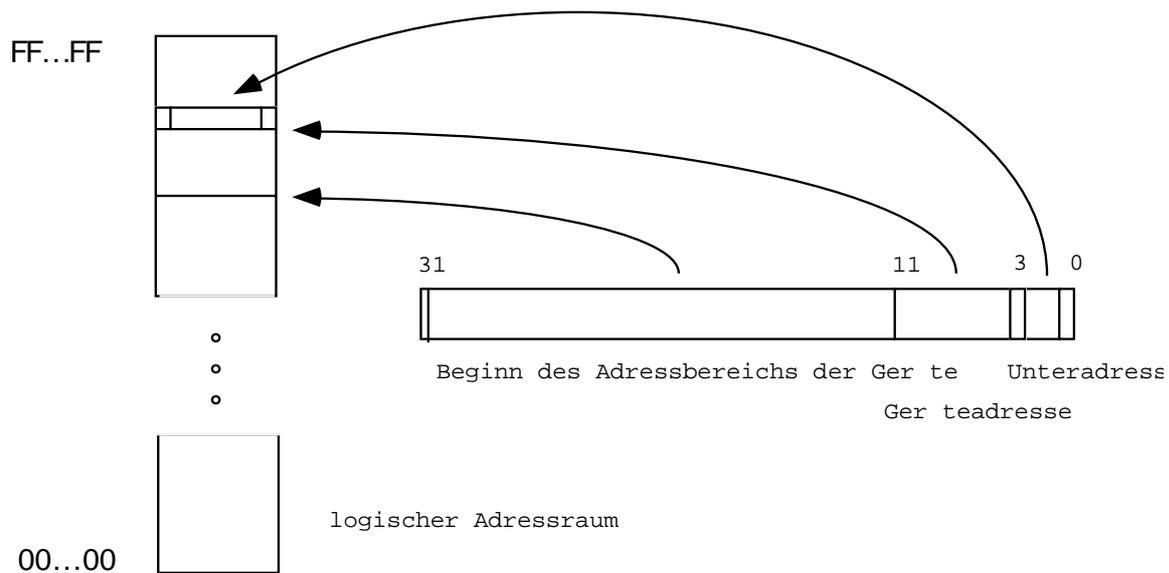


Die Adressierung des Geräts kann auf verschiedene Weisen erfolgen.

- a) eigene E/A-Befehle und E/A Adressleitungen (veraltet).
- b) eigene E/A-Ports auf der CPU; diese Art der Adressierung findet man bei Microcontrollern.
- c) Zentralbussystem
 Man hat einen homogenen Adressraum; Geräte werden angesprochen wie Speicherplätze auch.
 Es sind einige Adressen für Geräte reserviert; häufig am oberen Ende des logischen Adressraumes.
 Man reserviert für ein Gerät mehrere Adressen, z. B. 4, 8, 16, ...

Beispiel:

4096 Adressen sind für Geräte reserviert, aufgeteilt in max. 256 Geräte mit jeweils 16 Unteradressen. Aus der anliegenden Adresse A31 - A12 und der Geräteadresse A11 - A4 wird ein Chip-Select-Signal erzeugt; die Unteradressen dienen zum Ansteuern der Register und des Steuerwerks.
 Ein "address-valid" Steuersignal gibt an, wann die anliegende Adresse gültig ist.



5.3. Interface mit direktem Speicherzugriff (DMA)

Ein Interface mit DMA (direct memory access) ist ein aktives Gerät. Es erzeugt von sich aus Speicheradressen und transferiert Daten vom Speicher zur Peripherie. Typische Anwendungen: Platten-Controller, Drucker-Controller.

Die CPU sagt der DMA, was sie tun soll:

- Anfangsadresse im Speicher
- Blocklänge (wieviel übertragen werden soll)
- Anfangsadresse im Gerät
- Transferrichtung (lesen/schreiben)

Diese Daten werden in Register der DMA geschrieben.

Dann führt die DMA den Transfer wortsequentiell durch:

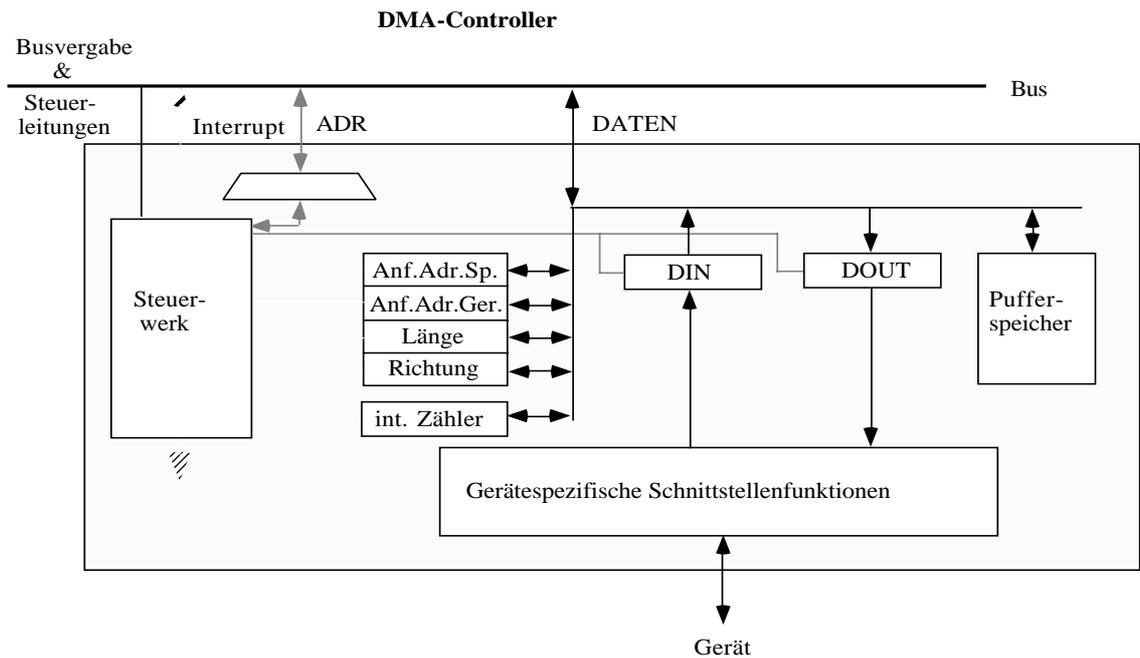
```
repeat
    Erzeuge Adresse für Speicher und Gerät;
    Übertrage ein Wort;
    Inkrementiere Adresse;
until    Blocklänge erreicht.
```

Die Fertigmeldung geschieht durch einen Interrupt aus der DMA.

CPU und DMA adressieren beide den Speicher.

Das führt zu Zugriffskonflikten, die durch Busvergabe-Mechanismen gelöst werden müssen.

Häufig werden DMA und CPU kaum aufeinander warten müssen:



Es sei kein Cache vorhanden und kein Befehlspipelining angewandt. Die CPU hat Priorität gegenüber der DMA. In mehr als der Hälfte aller Zugriffe der CPU auf den Speicher wird nur ein Befehl gelesen und die Daten intern verarbeitet. In die zeitliche Lücke zwischen dem Zugriff auf zwei Befehle kann die DMA einen Zugriff auf den Speicher machen. Die Zugriffe von DMA und CPU sind dann verzahnt; man spricht von "cycle stealing".

Sei in der CPU ein Cache vorhanden für Daten und Programm und Befehlspipelining angewandt. Die CPU greift nur selten auf den Bus zu, um den Cache nachzuladen. Dann kann in der Zwischenzeit die DMA unbehelligt Daten übertragen.



Um die Daten vom Gerät "en block", z. B. als Segment, übertragen zu können, sieht man in der DMA häufig einen Pufferspeicher für ein Segment vor.



