

## 4 Speicherorganisation und Datentransfer

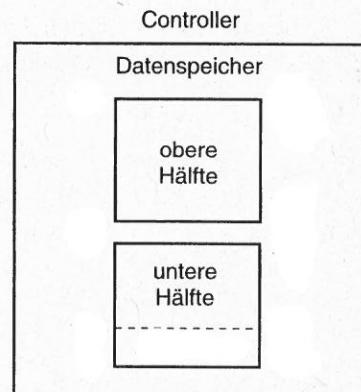
Es sollen die für die Programmbehandlung verwendbaren Programm- und Datenspeicher aufgezeigt werden: ihre Lage, ihr Adressbereich und die Assemblerbefehle, mit denen sie zu erreichen sind.

### 4.1 Speicherorganisation

Der Controller 80535 enthält einen internen Datenspeicher. Das Programm steht in einem externen Speicher, der über ein externes Bussystem angeschlossen ist. An das externe Bussystem lässt sich auch ein zusätzlicher externer Datenspeicher anschließen.

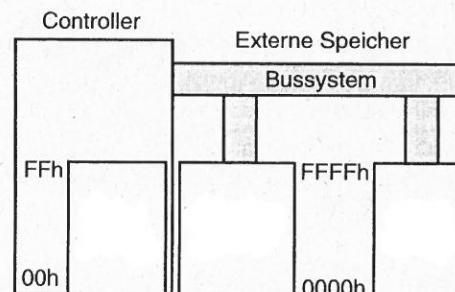
#### Datenspeicher im Controller

Der Controller besitzt einen internen Datenspeicher von 256 Byte. Er ist unterteilt in eine untere und eine obere Hälfte von je 128 Byte.



#### Externer Programm- und Datenspeicher

Der Controller kann einen externen Programmspeicher und einen externen Datenspeicher von je 64 KByte ansprechen. Die Speicher sind an ein externes Bussystem anzuschließen.



Nach einem Reset liest der Controller das Programm aus dem externen Programmspeicher. Dazu gibt er die Adresse und das Steuersignal PSEN (Programm Store Enable) aus. Der Speicher legt den Inhalt der Adresse auf den Datenbus. Der Controller liest zuerst den Befehl, dann führt er ihn aus. Bei der Ausführung von Transferbefehlen greift er auf die verschiedenen Speicher zu.

Die untere Hälfte des Datenspeichers im Controller lässt sich direkt adressieren, die obere Hälfte ist indirekt adressierbar. Als Zeiger dienen die Register R0 oder R1.

Die externen Speicher sind nur indirekt adressierbar. Für beide Speicher lässt sich das Doppelregister DPTR (Datapointer) als Zeiger verwenden. Das Übertragungs-Register für beide ist der Akku.

### 4.2 Assemblerbefehle zum Datentransfer

Die Transferbefehle für den internen Speicher und die internen Register beginnen mit MOV. Die Transportbefehle zwischen Akku und externem Datenspeicher beginnen mit MOVX, die zwischen Akku und externem Programmspeicher mit MOVC.

#### Befehle zum Datentransfer (Auszug aus dem Datenbuch: Instruktion Set Summary)

Mnemonic	Description	Byte	Cycle
<b>Data Transfer</b>			
MOV A, Rn	Move register to accumulator	1	1
MOV A, direct	Move direct byte to accumulator	2	1
MOV A, @Ri	Move indirect RAM to accumulator	1	1
MOV A, # data	Move immediate data to accumulator	2	1
MOV Rn, A	Move accumulator to register	1	1
MOV Rn, direct	Move direct byte to register	2	2
MOV Rn, # data	Move immediate data to register	2	1
MOV direct, A	Move accumulator to direct byte	2	1
MOV direct, Rn	Move register to direct byte	2	2
MOV direct, direct	Move direct byte to direct byte	3	2
MOV direct, @Ri	Move indirect RAM to direct byte	2	2
MOV direct, # data	Move immediate data to direct byte	3	2
MOV @Ri, A	Move accumulator to indirect RAM	1	1
MOV @Ri, direct	Move direct byte to indirect RAM	2	2
MOV @Ri, # data	Move immediate data to indirect RAM	2	1
MOV DPTR, # data 16	Load data pointer with a 16-bit constant	3	2
MOVC A, @A+DPTR	Move code byte relative to DPTR to accumulator	1	2
MOVC A, @A+PC	Move code byte relative to PC to accumulator	1	2
MOVX A, @Ri	Move external RAM (8-bit addr.) to A	1	2
MOVX A, @DPTR	Move external RAM (16-bit addr.) to A	1	2
MOVX @Ri, A	Move A to external RAM (8-bit addr.)	1	2
MOVX @DPTR, A	Move A to external RAM (16-bit addr.)	1	2

Rn = R0 bis R7

Ri = R0 oder R1 = Zeiger für indirekte Adressierung intern

DPTR = Zeiger für indirekte Adressierung extern Vor die Zeiger muss das Zeichen @ gesetzt werden.

direkt = Adresse im direkt adressierbaren internen Speicher

data = konstanter Wert als Hex-Zahl.

Vor die Konstante muss das Zeichen # gesetzt werden.

Alle in den Befehlen einzusetzende Zahlen sind Hex-Zahlen.

### Transfer im Controller, untere Speicherhälfte

Beispiele:

Inhalt Akku nach Speicherstelle 25

MOV

Inhalt Speicherstelle 20 nach Akku

MOV

Inhalt Speicherstelle 70 nach 25

MOV

Konstante FF in Akku

MOV

Konstante 3A nach Speicherstelle 30

MOV

Inhalt Speicherstelle 40 nach Register R4

MOV

### Transfer im Controller, untere und obere Speicherhälfte

Die obere Hälfte lässt sich nur über die Register R0 oder R1 indirekt adressieren.

Beispiele:

Inhalt Akku nach Speicherstelle 80

MOV

;Zeiger

MOV

Konstante 3A nach Speicherstelle 90

MOV

;Zeiger

MOV

Inhalt Speicherstelle 20 nach Speicherstelle F0

MOV

;Zeiger

MOV

Inhalt Speicherstelle 85 nach Speicherstelle 95

MOV

;Zeiger 1

MOV

;Zeiger 2

MOV

MOV

### Übung 4.1

Schreiben Sie die Assemblerbefehle für folgende Datentransporte:

1. Inhalt Speicherstelle 31 nach Akku
2. Inhalt Speicherstelle 17 nach R5
3. Inhalt Speicherstelle 40 nach 45
4. Konstante 31 in Speicherstelle 12
5. Inhalt Akku nach Speicherstelle F1
6. Inhalt Speicherstelle F0 nach Speicherstelle FF
7. Konstante 55 nach Speicherstelle 95.

### Transfer zwischen Controller und externem Datenspeicher

Bei der Ausführung dieser Befehle aktiviert der Controller die Steuersignale RD und WR. Der externe Datenspeicher lässt sich nur indirekt über den Datapointer DPTR adressieren.

Beispiele:

Inhalt Akku nach Speicherstelle E000

;Zeiger

Inhalt Speicherstelle 0100 nach Akku

;Zeiger

### Transfer zwischen Controller und externem Programmspeicher

Bei der Ausführung dieser Befehle aktiviert der Controller das Steuersignal PSEN. Aus dem Programmspeicher können Daten nur gelesen werden. Daten im Programmspeicher können z. B. auszugebende konstante Texte sein.

Der externe Programmspeicher lässt sich nur indirekt adressieren. Als Zeiger dient die Summe der Inhalte von Akku und Datapointer DPTR oder Akku und Programcounter PC.

Beispiel:

Inhalt Speicherstelle A500 nach Akku

;Zeiger

### Übung 4.2

Schreiben Sie die Assemblerbefehle für folgende Datentransporte:

1. Inhalt Datenspeicher D000 nach Akku
2. Inhalt Akku nach Datenspeicher 0000
3. Inhalt Programmspeicher 0200 nach Akku
4. Inhalt interner Datenspeicher 10 nach externem Datenspeicher F000
5. Inhalt externer Datenspeicher F100 nach internem Datenspeicher 90.