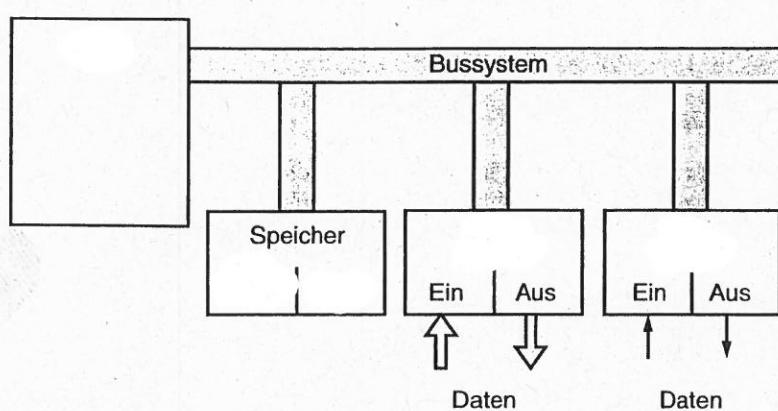


1 Der Mikrocomputer

In diesem Kapitel werden der grundsätzliche Aufbau und die Arbeitsweise eines Mikrocomputers beschrieben. Damit soll die Grundlage für das Verständnis der darauf folgenden Inhalte gelegt werden.

1.1 Der Aufbau eines Mikrocomputers



Ein Mikrocomputer besteht aus einer zentralen Prozess-Einheit CPU, die über ein Bussystem mit dem Speicher und den Eingabe- und Ausgabeeinheiten verbunden ist.

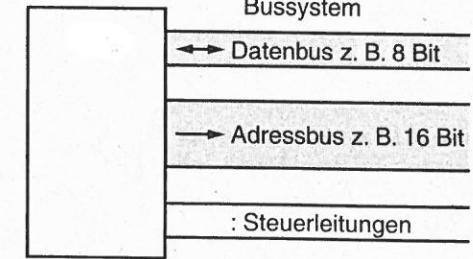
Die CPU wird über das Programm gesteuert. Sie verarbeitet die Daten, die sie aus dem Speicher oder den Eingabe-Ports liest. Die Ergebnisse schreibt sie wieder in den Speicher oder in die Ausgabe-Ports. Der Transport der Daten erfolgt über eine Anzahl paralleler Leitungen, die als Datenbus bezeichnet werden. Der Datenbus ist ein Teil des Bussystems.

Programm und Daten liegen im Speicher in bestimmten Adressen. Auch die Ports werden durch Adressen voneinander unterschieden. Die CPU muss die Adressen ausgeben, aus denen sie Programm oder Daten lesen will oder in die sie Daten schreiben will. Die Adresse wird von der CPU als Dualzahl über den Adressbus ausgegeben. Der Adressbus ist ein weiterer Teil des Bussystems.

Da z. B. beim Speicher Daten in die gleiche Adresse geschrieben oder aus ihr gelesen werden, ist ihm mitzuteilen, ob es sich um einen Lese- oder Schreibvorgang handelt. Das gleiche gilt für die Ein- und Ausgabeports, falls sie gleiche Adressen haben. Zur Veranlassung dieser und ähnlicher Schaltvorgänge hat die CPU noch einige Steuerleitungen. Sie werden unter der Bezeichnung Steuerbus zusammengefasst. Der Steuerbus ist der dritte Teil des Bussystems. Während sich die Leitungen von Daten- und Adressbus immer in ihrer Gesamtheit betrachten lassen, da sie zum gleichen Zeitpunkt in einen neuen Zustand übergehen, muss beim Steuerbus jede Leitung im Einzelnen betrachtet werden. Die Steuerleitungen geben die Zeitpunkte vor, zu denen die Schaltvorgänge im System stattfinden.

Insgesamt besteht das Bussystem, über das die CPU mit den angeschlossenen Baugruppen korrespondiert, also aus dem Datenbus, dem Adressbus und dem Steuerbus.

An das Bussystem sind die Baugruppen angeschlossen, die die CPU für ihre Arbeit benötigt.

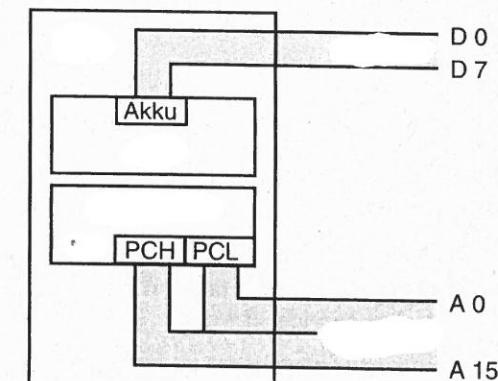


Die CPU-Baugruppe

Das zentrale Register in der CPU ist für den Programmierer der Akkumulator. Der Akku ist das Eingabe- und Ausgaberegister der Arithmetik-Logik-Einheit (Arithmetical Logical Unit) ALU. Zu Beginn einer arithmetischen oder logischen Operation steht im Akku einer der beiden Operanden. Am Ende der Operation steht darin das Ergebnis.

Der Operand muss durch Befehle des Programms vor der Operation in den Akku transportiert und danach wieder in den Datenspeicher zurückgebracht werden. Damit ist der Akku auch die Übertragestation für die Daten, die in die CPU hinein oder heraus transportiert werden.

Die CPU gibt über den Adressbus die Adressen der Speicherstellen oder der Ports aus, deren Inhalte sie lesen oder überschreiben will. Diese Adressen werden in der CPU in einem Doppelregister, dem Programmzähler (Programm Counter) PC vorbereitet. Im PC steht immer die Adresse, die als nächstes ausgegeben wird.



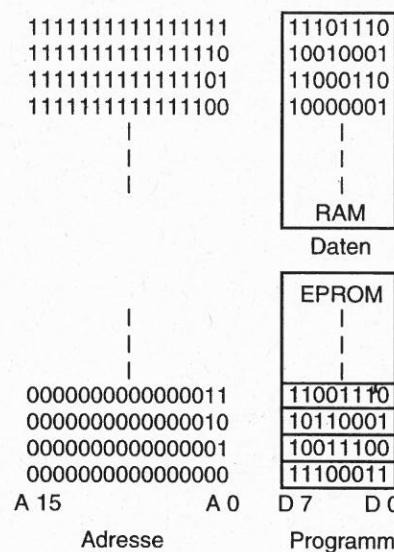
Der Speicher

Im Speicher stehen das Programm und die zu verarbeitenden Daten. Die Daten können entweder konstante oder variable Werte sein. Die Variablen werden über die Ports eingelesen oder ausgegeben.

Das Programm beinhaltet die Arbeitsanweisungen, die die CPU der Reihe nach liest und ausführt.

Die CPU beginnt die Programmbearbeitung nach einem Reset ab der Adresse 0000h. Ab dieser Adresse müssen die Befehle gespeichert sein. Als Speicherbaustein für das Anfangsprogramm lässt sich nur ein EPROM verwenden, damit nach dem Einschalten der Stromversorgung mit anschließendem Reset sofort das Programm vorhanden ist.

Die variablen Daten lassen sich nur in einem RAM-Baustein speichern.

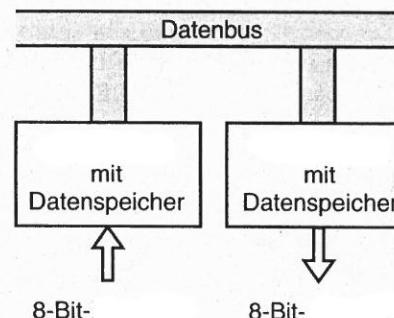


Die parallelen Ports

Über die parallelen Ports werden die Daten zu je acht Bit eingelesen oder ausgegeben.

Die CPU schreibt die Ausgabedaten über den 8-Bit-Datenbus in die Ausgabeports. Darin werden die Daten bis zum nächsten Überschreiben gespeichert.

Die an den Ports anliegenden Eingabedaten werden von der CPU über den Datenbus gelesen.

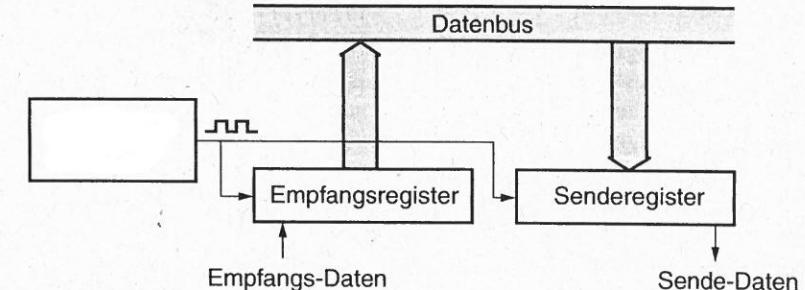


Die seriellen Ports

Serielle Ports senden oder empfangen die Daten nacheinander, Bit für Bit. Deshalb wird für das Senden oder das Empfangen der Daten nur je eine Leitung benötigt. Dazu kommt bei einem Minimalausbau nur noch das Bezugspotenzial 0 V. Der serielle Datenstrom wird eingerahmt von einem Startbit zu Beginn und einem Stopbit am Ende der Übertragung. In der Regel wird ein Byte seriell übertragen. Die Übertragung erfolgt mit einer bestimmten Geschwindigkeit, der Baudrate. Die Baudrate gibt an, wie viele Bits pro Sekunde übertragen werden. Bei einer Baudrate von z. B. 9600 werden 9600 Bits pro Sekunde übertragen. Die Übertragung wird getaktet von einem Baudratengenerator.

Für die CPU stellt sich der serielle Port genauso dar wie der parallele Port. Bei der Datenausgabe wird die Umwandlung in den seriellen Bitstrom einschließlich Start- und Stopbit in der seriellen Schnittstelle selbst vorgenommen.

Die entsprechende Serien-Parallel-Wandlung erfolgt auch bei den Empfangsdaten. Die interne Portschaltung stellt der CPU die Daten parallel zur Verfügung.

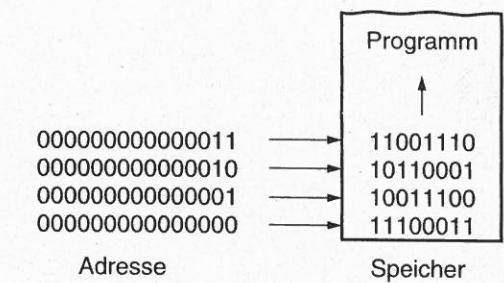


1.2 Die Arbeitsweise eines Mikrocomputers

Die CPU führt die Befehle aus, die ihr vom Programm vorgegeben werden. Dabei liest sie erst den Befehl, um ihn anschließend auszuführen. Auf diese Weise wird das Programm Befehl für Befehl abgearbeitet.

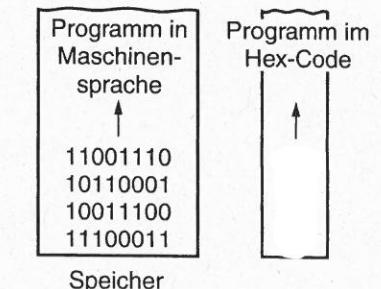
Das Programm

Das Programm ist eine Folge von Befehlen oder Instruktionen. Es liegt in aufeinander folgenden Adressen im Speicher. Jede Adresse markiert einen Speicherplatz mit acht Bit bzw. einem Byte Inhalt. Die Adressen sind z. B. 16-Bit-Dualzahlen.



Die Instruktionen des Programms sind in Bytes aufgeteilt. Jede Operation, die die CPU ausführen soll, wird durch ein bestimmtes Bitmuster in einem Byte dargestellt. Das so gespeicherte Programm, welches die CPU lesen und verstehen kann, wird Maschinenprogramm genannt.

Fasst man die Bitmuster zur leichteren Lesbarkeit zu je vier Bit zusammen, lassen sie sich als Hex-Zahlen darstellen.



Ein Programm in Maschinensprache lässt sich, selbst wenn es im Hexcode dargestellt wird, kaum lesen oder schreiben. Deshalb stellt man die Operationen symbolisch durch Abkürzungen dar.

Diese Abkürzungen werden Mnemonics genannt. Ein so geschriebenes Programm ist ein Assemblerprogramm; die Programmiersprache heißt Assembler.

Ein in Assemblersprache geschriebenes Programm wird von einem Übersetzerprogramm, Makro-Assembler genannt, in Maschinensprache übersetzt und dann in den Speicher des Mikrocomputers geladen.

Aufbau einer Instruktion

Eine Instruktion besteht prinzipiell aus der Operation und dem Operanden. Es gibt einfache und „mächtige“ Befehle. Entsprechend kurz oder lang sind die Instruktionen oder Befehle. Zur Unterbringung im Speicher müssen sie byteweise aufgeteilt werden. Einfache Befehle sind nur ein Byte lang, mächtige Befehle können drei Byte lang sein.

Bei 1-Byte-Befehlen sind Operation und Operand in acht Bit zusammengefasst. Dabei befindet sich der Operand in der Regel in einem Register in der CPU.

Bei 2-Byte-Befehlen steht die Operation im ersten und der Operand im zweiten Byte.

3-Byte-Befehle enthalten die Operation und in den nächsten beiden Bytes den Operanden mit Adressen oder Konstanten.

Instruktion:
"Transferiere: Zieladresse, Quelladresse"

Assemblersprache Maschinensprache

MOV 20 h, 30 h 10000101 MOV

MOV 20 h, 30 h	10000101 00100000 00110000	MOV Ziel Quell
----------------	----------------------------------	----------------------

Speicher

1-Byte-Befehl:

"Transferiere: Ziel, Quelle"
= Register A = Register R

MOV A, R1

11101001

Speicher

2-Byte-Befehl

"Transferiere: Ziel, Konstante
= Register A = h

MOV A # 1

01110100

000100

Speicher

3-Byte-Befehl

"Transferiere: Zieladresse, Konstante

— 74 —

— 11 —

011101

000010
0000110

Programmbearbeitung durch die CPU

Die CPU beginnt ihre Arbeit nach einem Reset bei der Adresse 0000h. An dieser Adresse erwartet sie den Operationscode der ersten Instruktion des Programms. Der Operationscode sagt der CPU, was sie machen soll und wie viel Bytes zu der Instruktion gehören. Die CPU liest dann die komplette Instruktion. Nach dem Lesen wird die Instruktion ausgeführt.

Die Ausführung kann, je nach Befehl, innerhalb der CPU erfolgen oder externe Baugruppen ansprechen. Erfolgt die Ausführung intern, ist sie auf dem externen Bus nicht sichtbar. Wird auf externe Baugruppen wie Speicher oder Ports zugegriffen, können die Signale auf den Busleitungen verfolgt werden. Ist die Instruktion ausgeführt, gibt die CPU die nächste Adresse auf dem Adressbus aus und das Steuersignal zum Lesen des Speicherinhaltes. Dieser Inhalt ist der Operationscode der nächsten Instruktion. Die Instruktion wird wieder komplett gelesen und dann ausgeführt.

So arbeitet die CPU Befehl für Befehl des Programms ab.

Übung 1.1

1. Aus welchen Leitungen besteht das Bussystem eines Mikrocomputers?
2. Was steht im Programm Counter?
3. Wie viele Adressen lassen sich mit 16 Bit darstellen?
4. Mit welcher Anzahl Leitungen kann eine serielle Datenübertragung arbeiten?
5. Erklären Sie den Begriff „Baudrate“.
6. Stellen Sie zwei Befehle in Assemblersprache und in Maschinensprache dar.
7. Was steht im ersten Byte jedes Befehls?
8. In welchem Speicherbaustein ist das Programm ab Adresse 0000h gespeichert?
9. In welchem Speicherbaustein werden variable Daten gespeichert?