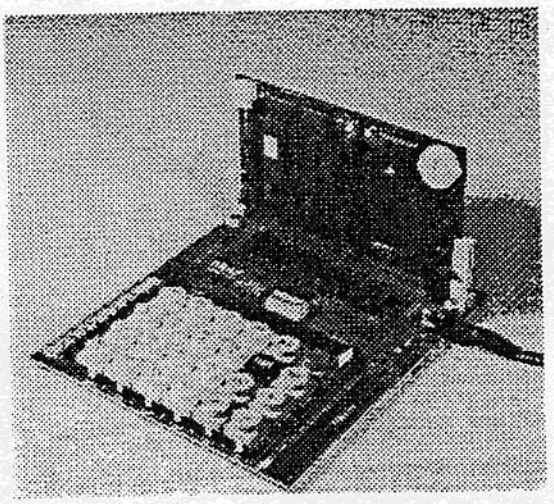
Der NDR-Computer

Kurzbeschreibung

zum

Einsteigersystem



Inhalt

1. Einführung

Mit dem Einsteigerpaket haben sie einen Rechner erworben, dessen Mikroprozessor sowohl als Vorläufer der modernen Personal Computer (PC's) betrachtet werden kann, der aber auch heute noch, vor allem in der Steuerungs und Regelungstechnik, eingesetzt wird.

Mit dem Verständnis dieses "relativ einfachen" Rechners und dem Verständnis der CPU Z80 wird es dem interessierten Leser sicherlich nicht schwerfallen auch moderne Rechner der 80x86 Familie zu verstehen.

Es sei allerdings darauf hingewiesen, daß dieses Handbuch nur eine Kurzbeschreibung darstellt, die lediglich den Einstieg in die Materie erleichtern soll. Bei einem tieferen Einstieg in die Arbeitsweise und die Programmierung von Mikrocomputern ist daher die Anschaffung von zusätzlicher Fachliteratur unvermeidlich. Zum Thema Mikroprozessor Z80 und Z80-Programmierung sind noch erhältlich:

Rolf-Dieter Klein,
Mikrocomputer selbstgebaut und programmiert
ISBN 3-7723-7162-0
Franzis-Verlag, München

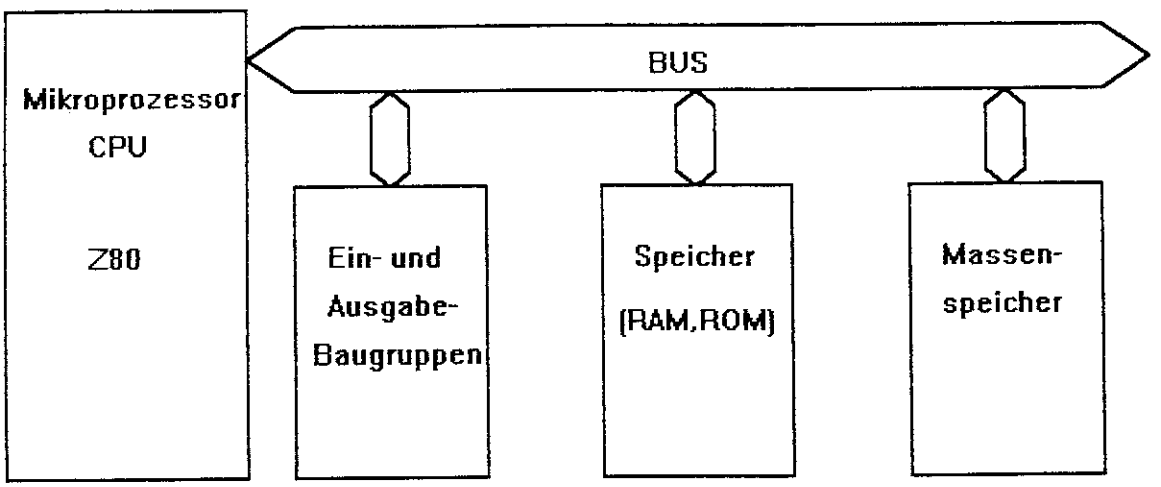
Peter Immerz,
Programmieren in Maschinensprache Z80
ISBN 3-921682-62-2
Hofacker, W

Poock-Haffmans, Peter /Burghardt, Kurt,
NDR-Computer Handbuch; Das Z80-System
Band I: ISBN 3-89171-003-8
Band II: ISBN 3-89171-004-6
Percomp

2. Grundsätzlicher Aufbau eines Mikrocomputers

Ein Mikrocomputer besteht grundsätzlich aus verschiedenen Einzelbaugruppen. Der Kern eines jeden Mikrocomputers ist der Mikroprozessor, in unserem Fall der Z80 Prozessor.
Der Mikroprozessor ist über Leitungen mit den anderen Baugruppen des Mikrocomputers verbunden. Diese Leitungsverbindungen bezeichnet man als ein Bussystem. Auf dieses Bussystem wird noch in einem gesonderten Kapitel eingegangen. Die Baugruppen, mit denen der Prozessor verbunden ist, sind:

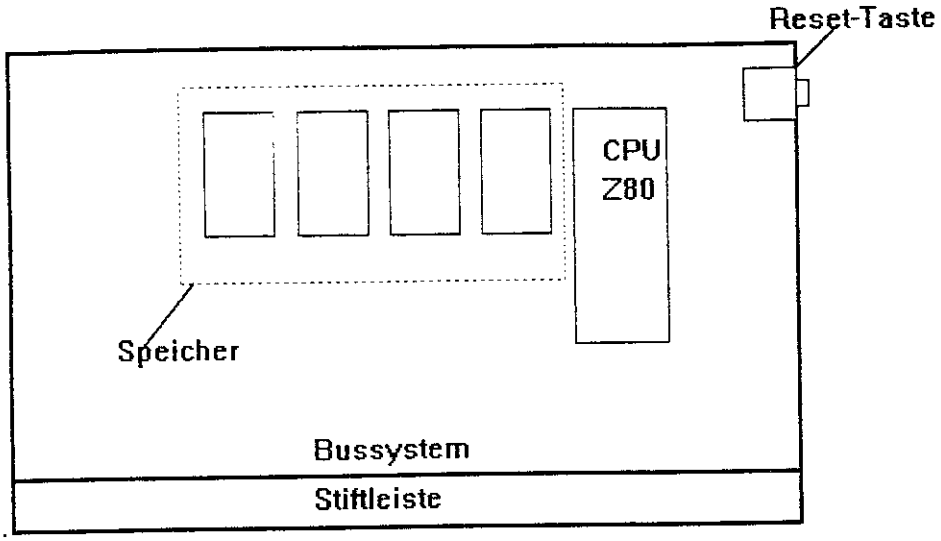
- Die Ein- und Ausgabeeinheiten (Tastatur, Monitor, hier: 7-Segment-Anzeigen)
- Speicher (RAM, ROM)
- Massenspeicher (Diskettenlaufwerk, Festplatte, Magnetband), gehören nicht zum Einsteigerpaket



Im folgenden wird nun auf die einzelnen Baugruppen ihres Einsteigerpakets näher eingegangen.

2.1 Die SBC3

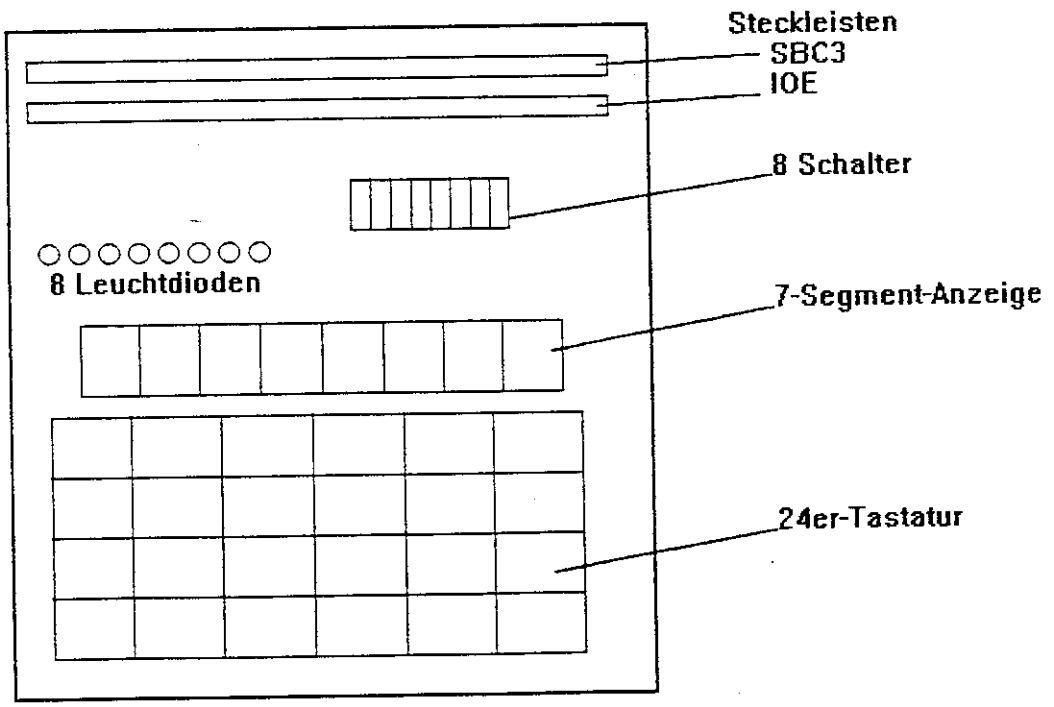
Zu Ihrem Einsteigerpaket gehört die Baugruppe SBC3, wobei SBC für Single-Bord-Computer (Einplatinenrechner) steht, d.h. die wichtigsten Funktionseinheiten des Mikrocomputers befinden sich auf einer Platine. In unserem Fall sind dies die CPU, Speicherbausteine und das Bussystem. Das Bussystem wird auf der Stiftleiste herausgeführt. Hiermit wird die SBC3 mit der HEXIO2 verbunden. Die Ein- Ausgabebaugruppen befinden sich nicht auf der SBC3. In der folgenden Skizze sehen Sie wo sich die einzelnen Bauteile auf der SBC3 befinden.



2.2 Die HEXIO2

HEXIO steht für HEX-Input-Output und dies heißt übersetzt hexadezimale Ein- Ausgabe. Hier finden sich also diejenigen Bestandteile des Mikrocomputers, die auf der SBC3 noch fehlen, nämlich die Ein- und Ausgabeeinheiten. Auf der HEXIO2 befinden sich je zwei Ein- und Ausgabeeinheiten. Zum einen eine 24er-Tastatur und eine darüberliegende 7-Segment-Anzeige, zum anderen acht Schalter und acht Leuchtdioden (LED) für binäre Ein- und Ausgaben. Die Verbindung zwischen der HEXIO2 und der SBC3 erfolgt über die Stiftleiste, die das Bussystem enthält.

Auch hier eine Skizze zur Verdeutlichung:



2.3 IOE (Input-Output-Einheit)

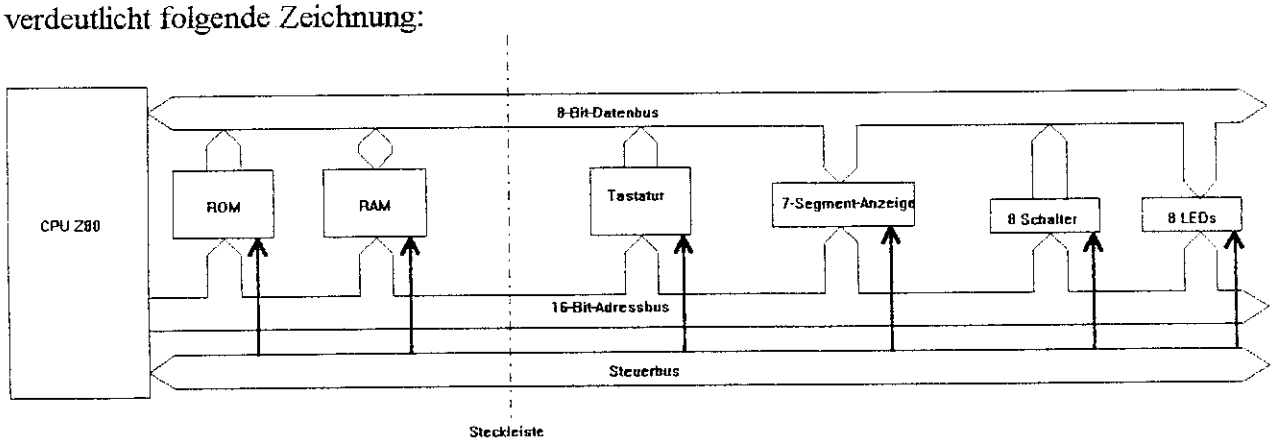
Die IOE-Baugruppe dient als Schnittstelle zur Außenwelt. Über die IOE-Baugruppe läßt sich z.B. eine Modelleisenbahn oder eine Heizungsanlage steuern und regeln. Die Karte verfügt über je zwei 8-Bit breite Ein- und Ausgabeports.

Den Aufbau Ihrer IOE-Karte entnehmen Sie bitte dem Handbuch zur IOE. Hier finden Sie auch, ab Seite 13, eine kurze Schaltungsbeschreibung. Außerdem wird in Kapitel 4 anhand eines Beispielprogramms noch näher auf die IOE-Karte eingegangen.

3. Die Arbeitsweise und Funktion der Einzelbausteine

3.1 Das Bussystem

In Kapitel 1 wurde bereits kurz auf das Bussystem eingegangen. Dort wurde es als Leitungen beschrieben, die die einzelnen Baugruppen miteinander verbinden. Auf dieses Bussystem soll nun näher eingegangen werden. Das es weit mehr darstellt, als ein Bündel Leitungen, verdeutlicht folgende Zeichnung:



Adreßbus

Der Adreßbus besteht aus 16 Leitungen (= 16 Bit), die vom Prozessor aus in das System gerichtet sind. Adressen kann nur der Prozessor erzeugen und auf den Adreßbus geben. Durch die ausgegebene Adresse wird ein bestimmter Speicherplatz ausgewählt mit dem dann Daten ausgetauscht werden können. Außer den Speicherplätzen adressiert der Prozessor auch die Ein-Ausgabeeinheiten. Für diese Adressierung stehen aber nur 8-Bit (das Low-Byte) zur Verfügung.

Das Low-Byte (niederwertiges Byte) beinhaltet die Adreßleitungen A0 bis A7, das High-Byte (höherwertiges Byte) beinhaltet die Adreßleitungen A8 bis A15.

Steuerbus

Der Steuerbus wird auch als Kontrollbus bezeichnet. Über den Steuerbus kann der Prozessor andere Funktionseinheiten für bestimmte Aufgaben anwählen. So wird z.B. beim Lesevorgang ein Signal ausgegeben, mit dem der Prozessor dem angesprochenen Baustein "mitteilt", daß er Daten lesen will (READ-Signal). Das gleiche gilt für Port-Zugriffe oder Schreiboperationen.

Datenbus

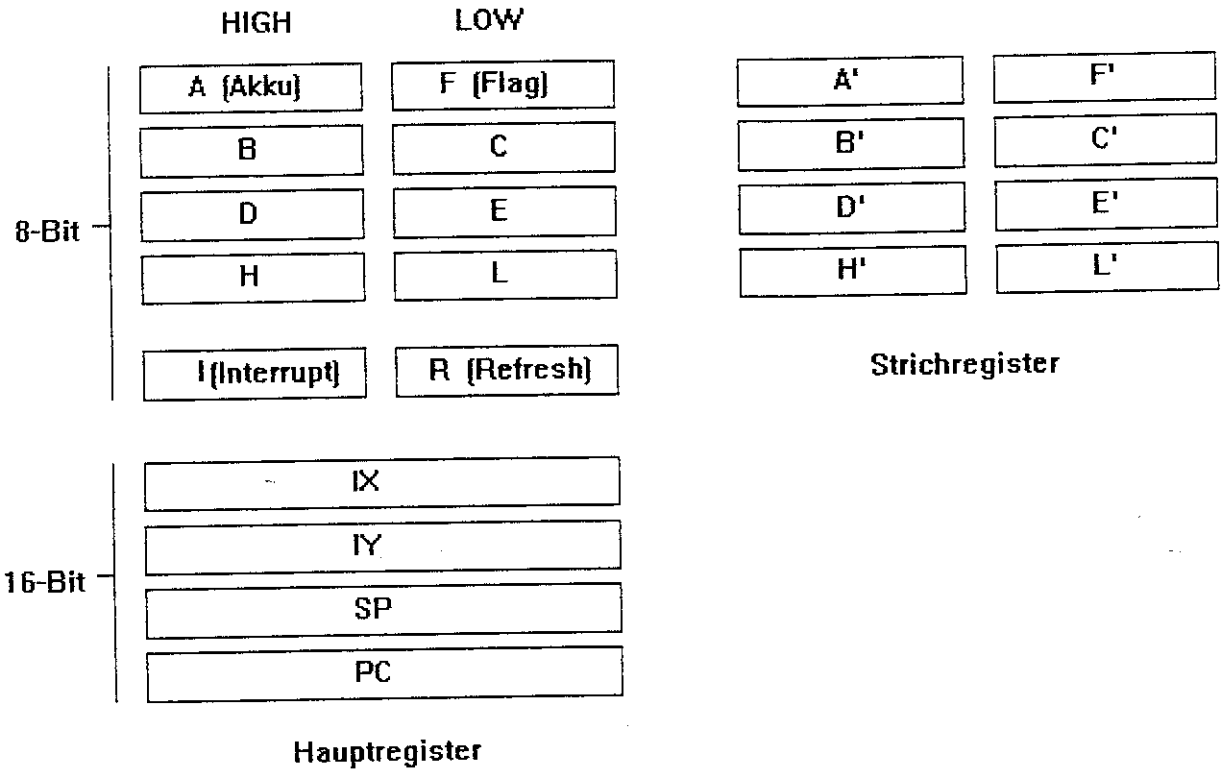
Der Datenbus besteht aus acht Leitungen (8 Bit). Über den Datenbus werden Daten ausgetauscht. Es können Daten aus dem Speicher oder von einem Port eingelesen werden oder auch in den Speicher geschrieben werden. Der Datenbus ist bidirektional, d.h. es können Daten in zwei Richtungen übertragen werden, sowohl von anderen Funktionseinheiten zum Prozessor, als auch vom Prozessor zu anderen Funktionseinheiten (Speicher, Ports, etc.).

3.2 Der Mikroprozessor Z80

Der Prozessor verfügt intern über Speichermöglichkeiten. Diese werden als Register bezeichnet. Ihr Aufbau gliedert sich wie folgt:

Register

Die Register sind die internen Speicher der Mikroprozessors. Der Z80 Prozessor verfügt sowohl über 8-Bit als auch über 16-Bit breite Register.
Diese werden wie folgt bezeichnet:



Auf die Strichregister soll im folgenden nicht näher eingegangen werden, da dieser zusätzliche Registersatz eine Besonderheit des Z80 darstellt, die hier nicht benötigt wird.

Die Register A, F, B, C, D, E, H, L werden für bestimmte Aufgaben, wie bereits in der Zeichnung angedeutet, zu den Registerpaaren AF, BC, DE, HL zusammengefaßt. Jedes Registerpaar verfügt dann über ein High- und ein Low- Register.
Es folgt nun eine kurze Beschreibung der wichtigsten Register.

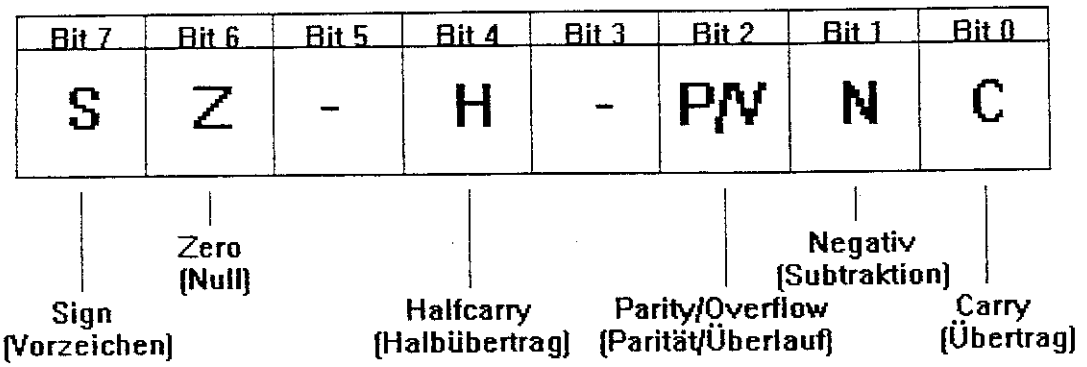
8-Bit-Register:

A-Register (Akku)

Der Akku ist das wichtigste und am häufigsten benötigte Register des Prozessors. Die meisten logischen, arithmetischen und Ein- Ausgabe- Operationen werden über den Akku ausgeführt.

F-Register (Flag)

Im Flag-Register werden abhängig von der letzten Operation bestimmte Bits gesetzt. Man kann die Flags auch als "Merker" bezeichnen. So wird z. B. wenn das Ergebnis der letzten Operation Null ist, das Zero-Flag gesetzt. Das Flag-Register ist für den Programmierer wichtig, da sich mit Hilfe der gesetzten Flags Programmverzweigungen (bedingte Sprünge) programmieren lassen. Vom Setzten eines Flags spricht man, wenn das entsprechende Bit eine logische 1 annimmt, andernfalls spricht man vom (zu)rücksetzten eines Flags.
Die Bits des Flag-Registers sind wie folgt belegt:



Sign-Flag:

Im Sign-Flag steht bei arithmetischen Operationen das Vorzeichen des Operanden. Das Sign-Flag ist 1 bei einer positiven und 0 bei einer negativen Zahl.

Zero-Flag:

Wenn der Inhalt eines Registers nach einer Operation zu Null wird, wird das Zero-Flag 1, bei allen anderen Werten wird das Zero-Flag 0. Das Zero-Flag wird z. B. zur Programmierung von Schleifen verwendet.

Halfcarry-Flag:

Halfcarry = Halbübertrag; Wenn sich bei einer Addition oder Subtraktion ein Übertrag von Bit 3 nach Bit 4 ergibt, wird das Halfcarry-Flag gesetzt.

Parity/Overflow-Flag:

Bei logischen Operationen stellt das Bit 2 des Flag-Registers das Parity-Flag und bei arithmetischen Operationen das Overflow-Flag dar.

Parity-Flag:
Ist die Anzahl der gesetzten Bits nach einer Operation gerade, wird das Parity-Flag gesetzt (d.h.: logisch 1) ansonsten wird es zurückgesetzt (logisch 0).

Overflow-Flag:
Tritt beim Arbeiten mit vorzeichenbehafteten Zahlen ein Überlauf auf, wird das Overflow-Flag gesetzt, ansonsten zurückgesetzt.

Negativ-Flag:

War die letzte Operation eine Subtraktion, so wird das N-Flag gesetzt, falls das Ergebnis negativ ist, nach einer Addition wird das N-Flag zurückgesetzt.

Carry-Flag:

Carry = Übertrag; Dieses Flag ist bei arithmetischen Operationen von Bedeutung. Tritt bei einer arithmetischen Operation ein Ergebnis > 255d auf, ergibt sich ein Übertrag von Bit 7 auf ein (im Arbeitsregister nicht vorhandenes) Bit 8. Kommt es nun zu einem solchen Übertrag, wird das Carry-Flag gesetzt.

Das Carry-Flag wird 1, wenn ein Übertrag vom Bit 2^7 in die nächsthöhere Potenz erfolgt.

B-, C-, D-, E-, H-, L-Register

Diese Register bieten dem Programmierer weitere Speichermöglichkeiten. Es können auch arithmetische und logische Operationen mit diesen Registern vorgenommen werden.

I-Register (Interrupt)

Interrupt heißt Unterbrechung. Beim Z80 besteht die Möglichkeit, spezielle Bausteine für Programmunterbrechungen anzuschließen. Der NDR-Computer arbeitet ohne Interrupt-Bausteine, wir wollen daher nicht näher auf das Interrupt-Register eingehen.

R-Register (Refresh)

Refresh heißt auffrischen, und aufgefrischt werden dynamische Speicher, die nach wenigen Millisekunden ihre Informationen verlieren würden. Der NDR-Computer arbeitet nicht mit dynamischen Speichern, daher wird hier nicht weiter auf das R-Register eingegangen.

16-Bit-Register

PC-Register

PC = Program Counter (Programmzähler); Der Programmzähler beinhaltet immer die 16-Bit breite Adresse des nächsten zu bearbeitenden Befehls, d.h. der Programmzähler enthält die im Programmablauf aktuelle Adresse. Diese Adresse kann vom Programm aus per Befehl verändert werden. Geschieht dies, setzt der Prozessor die Programmabarbeitung ab dieser neuen, geänderten Adresse fort.

SP-Register

SP = Stack Pointer (Stapelzeiger). Der Stack (Stapel) ist ein Speicherbereich in dem der Prozessor Daten sichern kann. Dies geschieht vor allem bei der Verwendung von Unterprogrammen, bei Registerinhalte auf den Stapel "gerettet" werden müssen. Außerdem enthält der Stapel die Rücksprungadresse bei Unterprogrammaufrufen. Der Stapelzeiger als Register enthält nun immer die Adresse des Stapels, in der zuletzt Daten abgelegt wurden.

IX- und IY-Register

Indexregister; Sie ermöglichen dem Anwender eine weitere Art der Adressierung von Speicherplätzten, die insbesondere bei der Dateibearbeitung von Interesse ist.

3.3 Die Speicherbausteine

Wie bereits bekannt befinden sich auf der SBC3 auch Speicherbausteine. Es soll nun darauf eingegangen werden wie der Prozessor mit diesen Speichern arbeitet. Zunächst unterscheidet man Speicher die nur einen Lesezugriff erlauben (ROM) und Speicher deren Inhalt verändert werden kann, d.h. in die der Prozessor auch schreiben kann (RAM).
Die Speicherbausteine sind über das Bussystem mit dem Prozessor verbunden. Das Bussystem besteht aus Daten- Adress- und Steuerbus.

Der Prozessor gibt auf dem Adressbus die Adresse derjenigen Speicherstelle aus, die er ansprechen möchte.

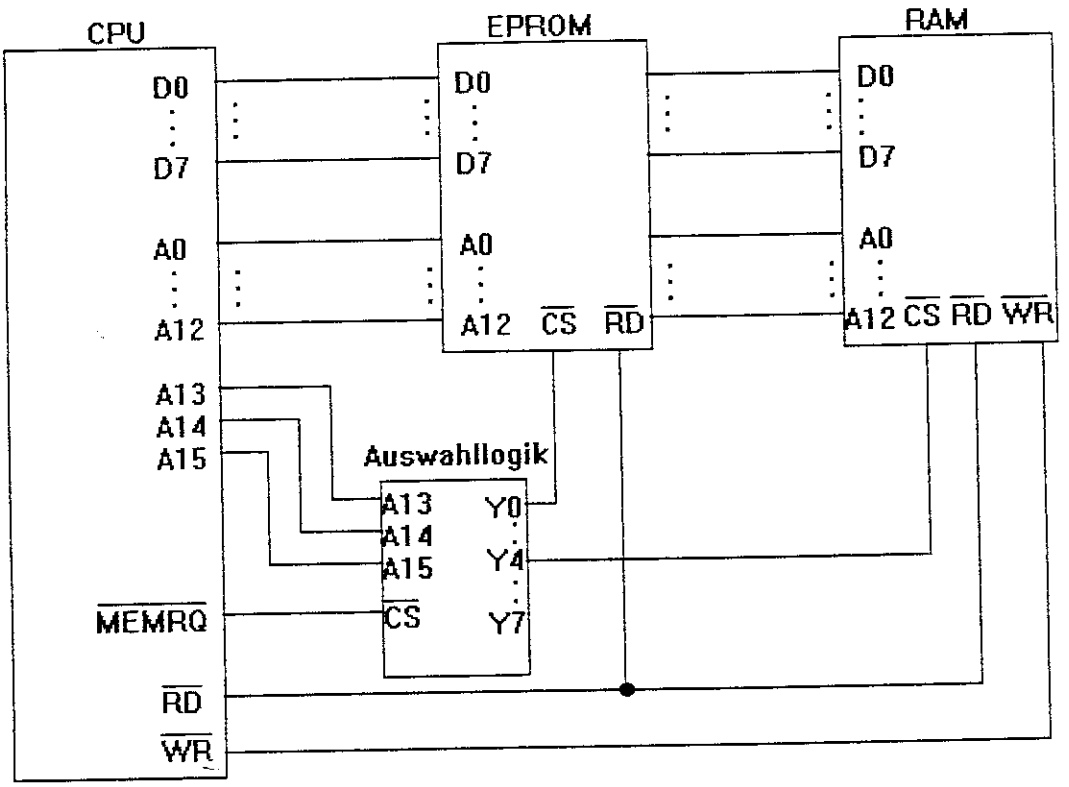
Mit dem Steuerbus gibt der Prozessor an, ob er einen Lese- oder Schreibzugriff durchführen möchte und daß dieser Zugriff auf den Speicher erfolgen soll. Hierfür werden folgende Signale verwendet:

- $\overline{MEMRQ}$ = **Memory request**: Dieses Signal gibt der Prozessor immer aus, wenn auf einen Speicherbaustein zugegriffen werden soll.
- $\overline{RD}$ = **Read**: Dieses Signal wird bei einem Lesezugriff ausgegeben.
- $\overline{WR}$ = **Write**: Das WR-Signal wird bei einem Schreibzugriff aktiv

Auf dem Datenbus werden die Daten, je nach Zugriff, vom Prozessor zum Speicherbaustein oder vom Speicherbaustein zum Prozessor transportiert.

Die Speicherbausteine ihrerseits sind über den Adressbus mit den Adressleitungen A0-A12 des Prozessors verbunden. Der Datenbus mit den Datenleitungen D0-D7 liegt ebenfalls an den Speicherbausteinen. Jeder Speicherbaustein benötigt, um aktiv zu werden, noch zusätzliche Signale vom Steuerbus. Zunächst das $\overline{CS}$ = Chip select Signal. Mit diesem Signal wird der Speicherbaustein aktiviert, d.h. erst wenn das $\overline{CS}$-Signal aktiv ist, kann auf den Baustein zugegriffen werden. Außerdem muß dem Speicherbaustein mitgeteilt werden, ob ein Lese- oder Schreibzugriff erfolgt. Hierfür haben die Speicherbausteine ihrerseits $\overline{RD}$ - bzw. $\overline{WR}$ -Eingänge.

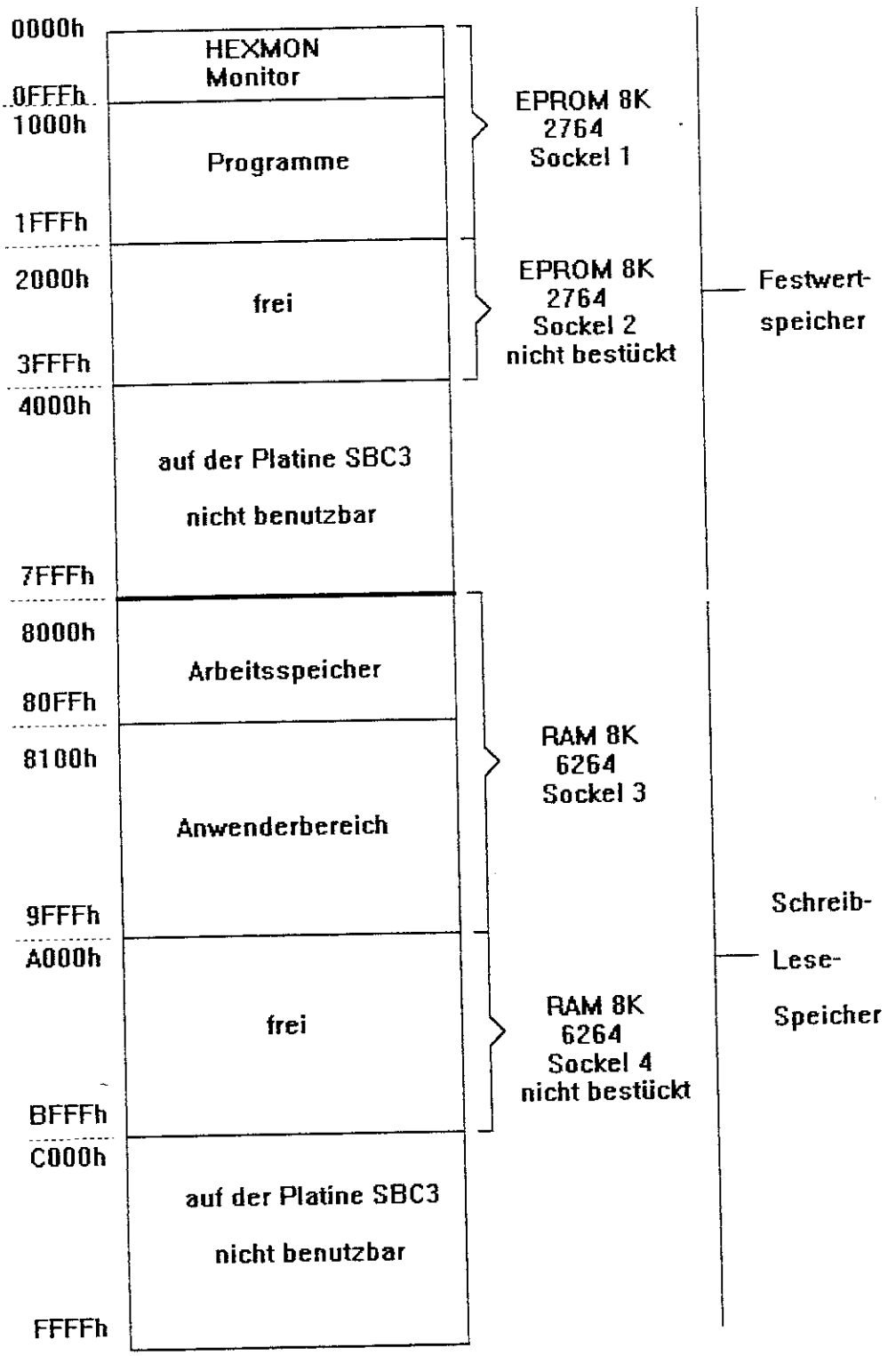
Um Konflikte auf dem Datenbus zu vermeiden, muß gewährleistet werden, daß nie mehr als ein Baustein aktiv ist. Dies kann aber nur garantiert werden, wenn für jeden Baustein je nach Adresse, Schreib- oder Lesezugriff die Signale $\overline{CS}$, $\overline{RD}$ oder $\overline{WR}$ "individuell" generiert werden. Dies geschieht mittels einer Auswahllogik. Die Funktion dieser Schaltung verdeutlicht folgende Skizze:



An jedem Speicherbaustein liegen nun die Adreßleitungen A0 bis A12 des Prozessors an. Damit ergibt sich eine Adressierungsmöglichkeit von 2^{13} = 8192 Byte pro Baustein.
Die Adreßleitungen A13-A15 liegen zusammen mit dem $\overline{MEMRQ}$ -Signal an einer Auswahllogik, die je nach Eingangssignalen (A13-A15) die Ausgangssignale Y0-Y7 generiert, die dann als Freigabesignale für die Speicherbausteine benutzt werden.

Von den Y-Leitungen kann, je nach dem welche Teiladresse" an den Anschlüssen A13-A15 liegt, immer nur eine Leitung aktiv werden.

Mit diesem Wissen ist nun die Speicherbelegung des NDR-Computers von Interesse:



Das 8k Eprom belegt den Speicherbereich von 0000h bis 1FFFh. Im Bereich von 1000h bis 1FFFh sind Programme gespeichert. Diese Programme (z.B. Lottozahlen) mit ihren Listings, Startadressen und Funktionen finden Sie in ihrem Handbuch zum Einsteigersystem ab Seite 10. Die hier als frei bezeichneten Speicherbereiche sind im Einsteigersystem noch nicht bestückt, können aber nachträglich mit einem EPROM bzw. RAM Baustein aufgerüstet werden (Siehe Handbuch zur SBC3 S. 14 ff). Zur Verdeutlichung der Adressierung der Speicherbausteine, sei hier noch die binäre Darstellung der Zustände der Adressleitungen für das Eprom angegeben:

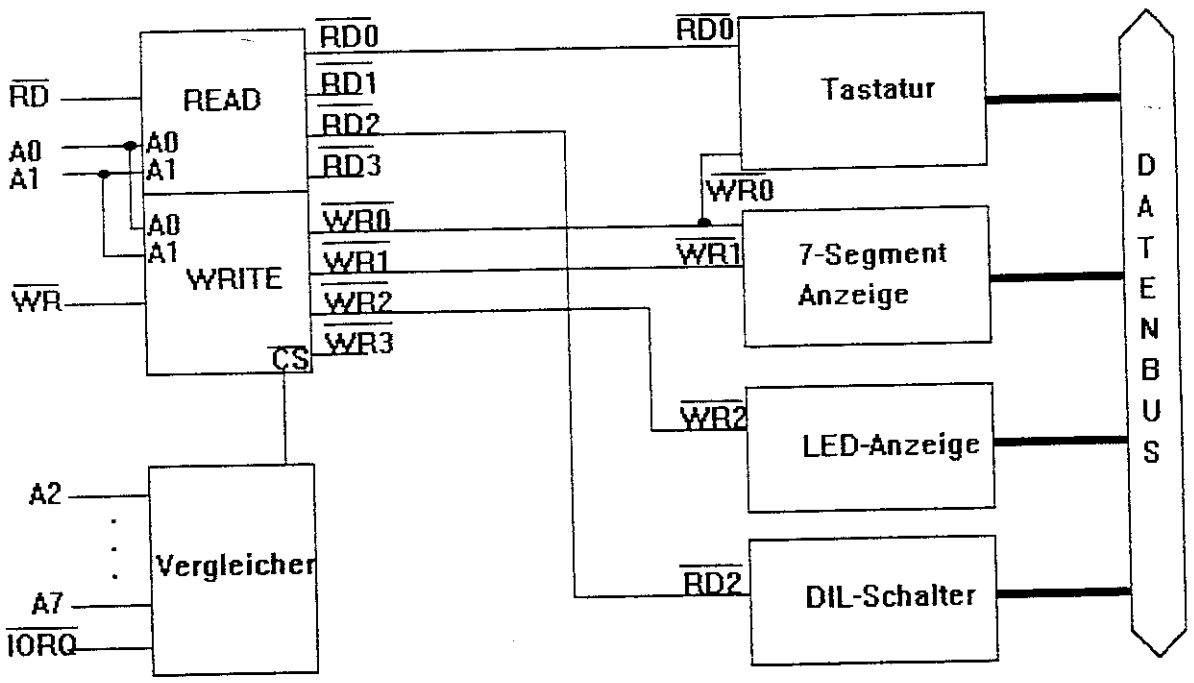
Adresse hex	Adreßleitungen														
	A15	A14	A13	A12	A11	A10	A9	A8	A7	A6	A5	A4	A3	A2	A1	A0
0000h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0
0001h	0	0	0	0	0	0	0	0	0	0	0	0	0	0	0	1
.																.
.																.
000Ah	0	0	0	0	0	0	0	0	0	0	0	0	1	0	1	0
.																.
.																.
1FFFh	0	0	0	1	1	1	1	1	1	1	1	1	1	1	1	1

bleiben unverändert,
erzeugen mit $\overline{MEMRQ}$
das Y0-Signal

Das 8k RAM belegt den Speicherbereich von 8000h bis 9FFFh. Hiervon wird der Bereich von 8000h bis 80FFh vom System benötigt, weshalb wir unsere Programme im Speicherbereich von 8100h bis 9FFFh ablegen.

3.4. Die Ein- Ausgabeeinheiten

Wie Sie bereits erfahren haben verfügt die HEXIO2 über zwei verschiedene Möglichkeiten der Ein- und Ausgabe von Daten. Zum einen die 24er-Tastatur und zum anderen die Schalter als Eingabemöglichkeiten. Die Leuchtdioden sowie die 7-Segment-Anzeige dienen als Ausgabemöglichkeiten. Ähnlich wie bei den Speicherbausteinen wird auch hier eine Auswahllogik benötigt, die den gewünschten Port "aktiviert".
Hier stehen nun vom Prozessor die Adressleitungen A0 bis A7 sowie die Signale $\overline{RD}$, $\overline{WR}$ und $\overline{IORQ}$ zur Verfügung (IORQ = Input / Output Request). Mit diesem Signal signalisiert der Prozessor einen Zugriff auf einen Portbaustein, ähnlich wie mit $\overline{MEMRQ}$ ein Speicherzugriff signalisiert wird. Die Signalerzeugung für die Portbausteine sieht prinzipiell wie folgt aus:



Der Vollständigkeit halber ist hier noch anzumerken, daß die mit einem Strich über der Signalbezeichnung, z.B. $\overline{MEMRQ}$, dargestellten Signale LOW-aktiv sind, d.h. eine aktive Leitung führt eine logische 0. Dies entspricht nahezu 0 Volt (positive Logikzuordnung).

Für die Ports wurden hier die Adressen von 00h bis 03h vergeben, d.h. bei einem Portzugriff führen die Adreßleitungen A2 bis A7 eine logische 0, ebenso wie die $\overline{IORQ}$-Leitung. In der Zeichnung sehen sie, daß diese Leitungen alle auf einen Vergleicher geführt sind. Liegt nun auf allen diesen Leitungen eine logische 0 an, so gibt der Vergleicher ein Signal aus, das Chip-select Signal für die Auswahllogik. Damit wird nun die Auswahllogik aktiviert und generiert, je nach logischem Zustand der Adreßleitungen A0 und A1 sowie der Signale $\overline{RD}$ und $\overline{WR}$, die Auswahlsignale für die einzelnen Portbausteine. Auf die Frage, warum manche Bausteine mehrere Auswahlsignale benötigen, soll hier nicht weiter eingegangen werden. Hierüber können sie sich in der entsprechenden Fachliteratur informieren.

In der folgenden Tabelle sind noch einmal alle für Lese- und Schreiboperationen mit Portbausteinen notwendigen Signale zusammengefaßt:

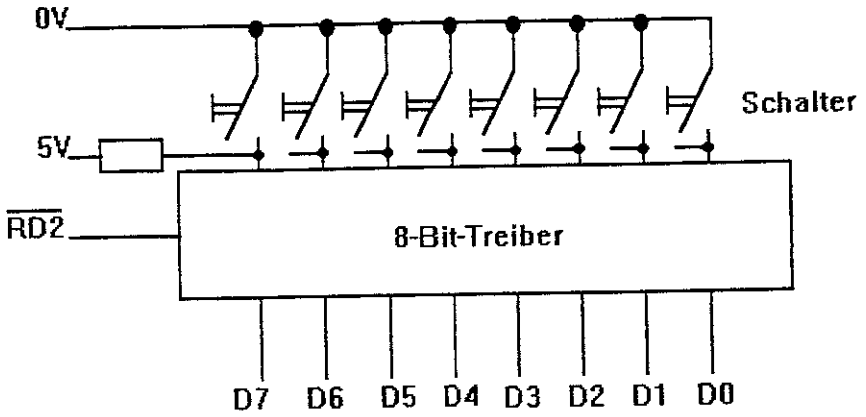
Operation	Adreßleitungen								Port-adresse	Select-leitung	Port
	A7	A6	A5	A4	A3	A2	A1	A0			
READ	0	0	0	0	0	0	0	0	00h	$\overline{RD}0$	Tastatur
	0	0	0	0	0	0	0	1	01h	$\overline{RD}1$	nicht belegt
$\overline{RD} = 0$	0	0	0	0	0	0	1	0	02h	$\overline{RD}2$	DIL-Schalter
$\overline{IORQ} = 0$	0	0	0	0	0	0	1	1	03h	$\overline{RD}3$	nicht belegt
WRITE	0	0	0	0	0	0	0	0	00h	$\overline{WR}0$	Tastatur / 7-Segment
	0	0	0	0	0	0	0	1	01h	$\overline{WR}1$	7-Segment-Anzeige
$\overline{WR} = 0$	0	0	0	0	0	0	1	0	02h	$\overline{WR}2$	LED-Anzeige
$\overline{IORQ} = 0$	0	0	0	0	0	0	1	1	03h	$\overline{WR}3$	nicht belegt

Wie sind nun die Ports aufgebaut ? Es soll nun anhand der DIL-Schalter und der Leuchtdioden der prinzipielle Aufbau erläutert werden. Die genaue Schaltungsbeschreibung aller Ports finden Sie in ihrem HEXIO-Handbuch ab Seite 16.

3.4.1 DIL-Schalter

Über die DIL-Schalter lassen sich binäre Signale in den NDR-Computer eingeben. Die **Portadresse ist 02h.**

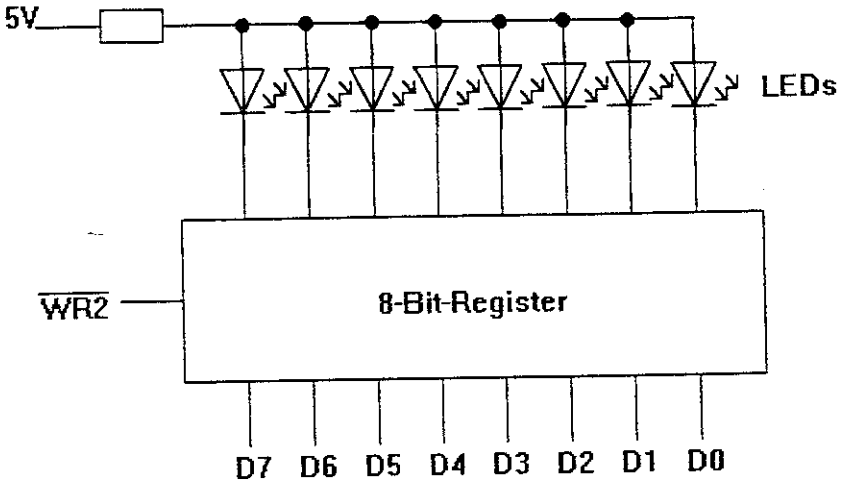
Prinzipschaltbild:



Wird nun einer der Schalter geschlossen und ist das $\overline{RD2}$ Signal aktiv, dann wird eine logische Null auf die entsprechende Leitung des Datenbusses gelegt. Ist ein Schalter geöffnet, liegt eine logische 1 auf der entsprechenden Leitung. Die Schalterstellung kann über den entsprechenden Befehl als Bitmuster in den Akku geladen werden.

3.4.2 LED-Anzeige

Auch hier zunächst eine Prinzipskizze:



Hier kann nun ein Datenwort ausgegeben werden (1Wort = 8 Bit). Liegt ein Datenwort auf dem Datenbus und ist das $\overline{WR2}$ Signal aktiv, dann wird das Wort auf die LEDs gegeben. Aus der Schaltung ist ersichtlich, daß eine logische 0 auf einer Datenleitung zu einem Aufleuchten der entsprechenden LED führt.

Soweit die Kurzbeschreibung der Hardware. Im nächsten Teil wird anhand zweier Beispiele die Programmierung des Systems erläutert. Auch dieser Teil stellt lediglich eine kurze Einleitung dar, ein Studium der entsprechenden Fachliteratur ist auch hier unvermeidlich.

4. Programmierung des Einsteigersystems

4.1 Einfaches Programmbeispiel

Das folgende Programm dient dem Einstieg in die Programmierung. Es ist keineswegs optimal oder so kurz wie möglich gehalten. Es wurde vielmehr wert darauf gelegt, daß die grundlegendsten Befehle und Techniken sowie die Unterprogramme von HEXMON demonstriert werden.

Beginnen Sie mit der Eingabe des Programms. Die Tastenbelegung und ihre Funktion finden Sie im Handbuch zum Einsteigerpaket ab Seite 47. Wir wollen Sie aber trotzdem mit Hilfe dieses Programms ein wenig in die Bedienung des Systems einführen.

Wenn Sie Ihr System einschalten meldet es sich mit der Meldung "HALLO-1.1". Um nun das Programm einzugeben müssen Sie die Speicherinhalte verändern. Drücken Sie hierfür die Taste "SPE". Das System meldet nun "Adr 8100", wobei 8100h die erste Speicheradresse (als hexadezimale Zahl) darstellt, ab der wir unsere Programme ablegen können (vgl. 3.3). Mittels der Tastatur können Sie nun den Speicherbereich eingeben auf den Sie zugreifen wollen. Bestätigen Sie die Eingabe dann mit der "CR" Taste. Wenn Sie dies tun, sehen Sie nun links die Adresse und rechts den Inhalt der Speicherstelle. Dies könnte z.B. so aussehen: "8100 3b". Mit den Tasten "+" und "-" können Sie nun durch den Speicher "blättern". Sehen Sie sich ruhig einige Speicherstellen an. Nun wollen wir aber mit der Eingabe des Programms beginnen. Wählen Sie, wie beschrieben, die Adresse 8100h aus. Geben Sie jetzt über die Hexadezimaltastatur den ersten Wert ein. In diesem Fall ist das CDh. Steht der richtige Wert in der Anzeige ? Dann bestätigen Sie bitte mit der "CR" Taste. Jetzt erscheint die nächste Speicherstelle in der Anzeige. Geben Sie auch hier wieder den entsprechenden Wert ein. Der zweite Wert bei diesem Programm ist 33h. Bestätigen Sie auch hier wieder mit der "CR"-Taste. So verfahren Sie jetzt bitte bis zur Speicherstelle 8121h. Geben Sie die 81h ein und bestätigen Sie mit der "CR"-Taste. Jetzt erscheint die Speicherstelle 8122h in der Anzeige.

Wie Sie dem Programmlisting entnehmen können, benutzen wir die folgenden Speicherplätze bis 8130h nicht. Nachdem Sie bestimmt nicht die "+"-Taste solange drücken wollen bis Sie bei 8130h angelangt sind, drücken Sie nun die "BEF"-Taste. In der Anzeige erscheint nun "-bef- ". Damit haben sie den Eingabemodus verlassen und können nun eine andere Funktion auswählen. Im Moment wollen wir aber mit der Eingabe fortfahren. Wie oben beschrieben wählen Sie nun den Speicherbereich ab 8130h über die "SPE"-Funktion für die Eingabe aus. Kehren Sie bei der Adresse 8153h erneut in den Befehlsmodus zurück und fahren Sie an der Speicherstelle 8500h mit der Eingabe des Unterprogramms fort. Vergessen Sie bitte nicht, die Zeichentabelle ab Adresse 9000h ebenfalls einzugeben.

Wenn Sie das Programm vollständig eingegeben haben, dann können Sie es jetzt starten. Dazu begeben Sie sich bitte zunächst durch drücken der "BEF"-Taste in den Befehlsmodus. Drücken Sie jetzt die "START"-Taste. Auf der Anzeige erscheint wiederum die Adresse 8100h. Dies ist für unser Programm die Startadresse. Auch hier können Sie mittels der Tastatur die Adresse verändern. Die richtige Startdresse bestätigen Sie dann bitte durch Drücken der "CR"-Taste. Damit starten Sie ihr Programm. Wenn Sie alles richtig eingegeben haben, fordert Sie ihr Rechner zum Drücken einer Taste auf. Folgen Sie dieser Anweisung, können Sie anhand der Leuchtdioden verfolgen, wie ihr Rechner binär von 0 bis 255d zählt.
Danach erscheint wieder die "HALLO-1.1" - Meldung auf der Anzeige und Sie können das Programm erneut starten.

Sollte es nicht geklappt haben, drücken Sie bitte die RESET-Taste und vergleichen Sie die Speicherinhalte nochmals mit dem Programmlisting.

Hat alles funktioniert, so sollten Sie noch einige weitere Funktionen der HEXIO2 ausprobieren. Die Funktionen der einzelnen Tasten finden sie ab Seite 49 in Ihrem Handbuch zum Einsteigersystem. Je nach Bedarf oder Ihren Wünschen können Sie sich mit diesen Funktionen vertraut machen.

Den Befehlssatz und die Programmierung des Z80 Prozessors zu erklären ist **nicht** die Aufgabe dieser Kurzbeschreibung. Dazu existiert bereits genügend Fachliteratur. Dieses Programm ist in seinem Aufbau und den Erläuterungen so gehalten, daß sie es bereits mit wenig zusätzlichem Literaturstudium verstehen und auch verändern können. Sicherlich gibt es für Einzelfunktionen innerhalb des Programms bessere oder kürzere Lösungen. Versuchen Sie sich doch daran.

Hauptprogramm

Adresse	HEX-Code	Befehl	Erläuterung
8100	CD	CALL 0033h	Unterprogramm "Clear33" an der Speicherstelle 0033h (EPROM) aufrufen.
8101	33		
8102	00		
8103	CD	CALL 0009h	Unterprogramm "Anzeige9" an der Speicherstelle 0009h (EPROM) aufrufen
8104	09		
8105	00		
8106	21	LD HL, 9000h	HL-Registerpaar laden. 00h wird ins L-Register, 09h wird ins H-Register geladen
8107	00		
8108	90		
8109	01	LD BC, 8000h	BC-Registerpaar laden. 00h wird ins C-Register, 80h wird ins B-Register geladen.
810A	00		
810B	80		
810C	7E	LD A, (HL)	Lade den Akku mit dem Inhalt der durch das HL-Registerpaar bezeichneten Speicherstelle
810D	F6	OR 00	Der Akkuinhalt wird mit 00 logisch ODER-verknüpft. Diese Operation dient dem Setzen des Zero-Flags für die folgende Programmverzweigung
810E	00		
810F	CA	JP Z, 8520h	Wenn das Zero-Flag gesetzt ist, das Ergebnis der ODER-Verknüpfung also 0 war, dann erfolgt ein Sprung an die Adresse 8520h.
8110	20		
8111	85		
8112	57	LD D, A	Lade den Inhalt des Akkus in das D-Register; der Akkuinhalt bleibt unverändert.
8113	DE	SBC A, 99h	Subtrahiere 99h vom Akkuinhalt
8114	99		
8115	CA	JP Z, 8130h	Ist das Ergebnis der Subtraktion 0, dann erfolgt ein Sprung an die Adresse 8130h
8116	30		
8117	81		
8118	7A	LD A, D	Lade den Inhalt des D-Registers in den Akku
8119	02	LD (BC), A	Speichere den Inhalt des Akkus in der durch das BC-Registerpaar bezeichneten Speicherstelle
811A	CD	CALL 8500h	Aufruf des Unterprogramms Anzeige an der Speicherstelle 8500h
811B	00		
811C	85		

Achtung Fehler: Der in Speicherstelle 8110 angegebene Wert ist falsch, richtig muß es 15 heißen!

811D	03	INC B	Erhöhe den Inhalt des B-Registers um 1
811E	23	INC HL	Erhöhe den Inhalt des HL-Registerpaares um 1
811F	C3	JP 810Ch	Sprung nach 810Ch
8120	0C		
8121	81		
. . .			***Speicherbereich bis 8130h wird nicht verwendet. Hier ist die Trennstelle zwischen dem Programm zur Ansteuerung der 7-Segment-Anzeige und der Ansteuerung der Leuchtdioden.***
8130	CD	CALL 000Ch	Aufruf des Unterprogramms "Holetastec" an der Speicherstelle 000Ch (EPROM).
8131	0C		
8132	00		
8133	3E	LD A,FFh	Lade FFh in den Akku.
8134	FF		
8135	D3	OUT (02h), A	Gebe den Inhalt des Akkus auf den Port mit der Adresse 02h aus.
8136	02		
8137	57	LD D,A	Lade das D-Register mit dem Akkuinhalt.
8138	21	LD HL, 3A00h	Lade das HL-Registerpaar mit FFFFh. Anzahl der Schleifendurchläufe (Hier dürfen Sie die fettgedruckten Bytes variieren)
8139	**00**		
813A	**3A**		
813B	2B	DEC HL	Verringere den Inhalt des HL-Registerpaares um 1.
813C	7C	LD A,H	Lade den Inhalt des H-Registers in den Akku.
813D	F6	OR L	ODER-Verknüpfung mit Inhalt des L-Registers. (Nur Bei (H) = (L) = 0 wird das ZERO-Flag gesetzt).
813E	00		
813F	C2	JP NZ, 813Bh	Ist das ZERO-Flag nicht gesetzt, erfolgt ein Sprung nach 813Dh.
8140	3B		
8141	81		
8142	7A	LD A, D	Lade den Akku mit dem Inhalt des D-Registers.
8143	3D	DEC A	Verringere den Akkuinhalt um 1.
8144	C2	JP NZ, 8135h	Ist der Akkuinhalt noch nicht 0, dann erfolgt ein Sprung nach 8135h.
8145	35		
8146	81		
8147	D3	OUT (02h), A	Ausgabe des Akkuinhalts auf den Port mit der Adresse 02h.
8148	02		
8149	C3	JP 0000h	Sprung an die Adresse 0000h (EPROM), Startadresse des Systems. *
814A	00		
814B	00		

* Wenn Sie bei der Adresse 814Bh anstelle der von 00h, 81h eingeben, so beginnt Ihr Programm erneut. Ein Verlassen ist dann nur noch mit der RESET-Taste möglich.

Unterprogramm Anzeige

Adresse	HEX-Code	Befehl	Erläuterung
8500	F5	PUSH AF	Registerinhalte auf den Stapel retten.
8501	C5	PUSH BC	
8502	D5	PUSH DE	
8503	E5	PUSH HL	
8504	06	LD B, FFh	Lade FFh ins B-Register (Schleifenzähler für die Warteschleife).
8505	FF		
8506	CD	CALL 0009h	Aufruf des Unterprogramms "Anzeige9".
8507	09		
8508	00		
8509	05	DEC B	Verringere den Inhalt des B-Registers um 1.
850A	C2	JP NZ, 8506h	Ist der Inhalt des B-Registers <>0, dann erfolgt ein Sprung nach 8506h.
850B	06		
850C	85		
850D	E1	POP HL	Registerinhalte vom Stapel zurückholen.
850E	D1	POP DE	
850F	C1	POP BC	
8510	F1	POP AF	
8511	C9	RET	Rücksprung in das aufrufende Programm.

Einsprungstelle "00"
(An diese Ardesse wird gesprungen, wenn 00h aus der Zeichentabelle geladen wird.)

8515	CD	CALL 0033h	Aufruf des Unterprogramms "Clear33".
8516	33		
8517	00		
8518	CD	CALL 0009h	Aufruf des Unterprogramms "Anzeige9".
8519	09		
851A	00		
851B	23	INC HL	Erhöhe den Inhalt des HL-Registerpaares um 1.
851C	C3	JP 8109h	Sprung nach 8109h.
851D	09		
851E	81		

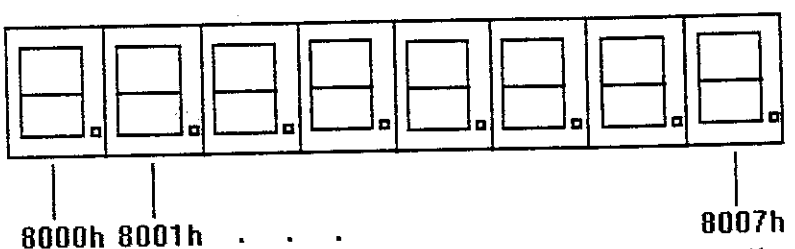
Zeichentabelle:

Adresse	HEX-Code	Zeichen
9000	83	B
9001	CF	i
9002	87	t
9003	87	t
9004	86	e
9005	00	
9006	86	e
9007	CF	i
9008	AB	n
9009	86	e

900A	00	
900B	87	T
900C	88	a
900D	92	s
900E	87	t
900F	86	e
9010	00	
9011	A1	d
9012	AF	r
9013	C1	u
9014	86	e
9015	C6	c
9016	85	k
9017	86	e
9018	AB	n
9019	00	
901A	99	wird hier als Zeichen für Abbruch verwendet.

In diesem Beispiel werden häufig Unterprogramme im Speicherbereich von 0000h bis 1FFFh aufgerufen. Dieser Speicherbereich wird vom Eprom belegt. Bei diesen Unterprogrammen handelt es sich um Unterprogramme des Monitors HEXMON. Diese Unterprogramme dienen der Steuerung der 7-Segment-Anzeige. Ihre Funktionen sind im Einzelnen:

Anzeige9:
Aufruf des Unterprogramms "Anzeige9" werden die in den Speicherplätzen 8000h bis 8007h auf der 7-Segment-Anzeige ausgegeben. Die Zuordnung ist dabei wie folgt:



Das Unterprogramm "Anzeige9" finden Sie im Eprom ab der Adresse 0009h.

Clear33

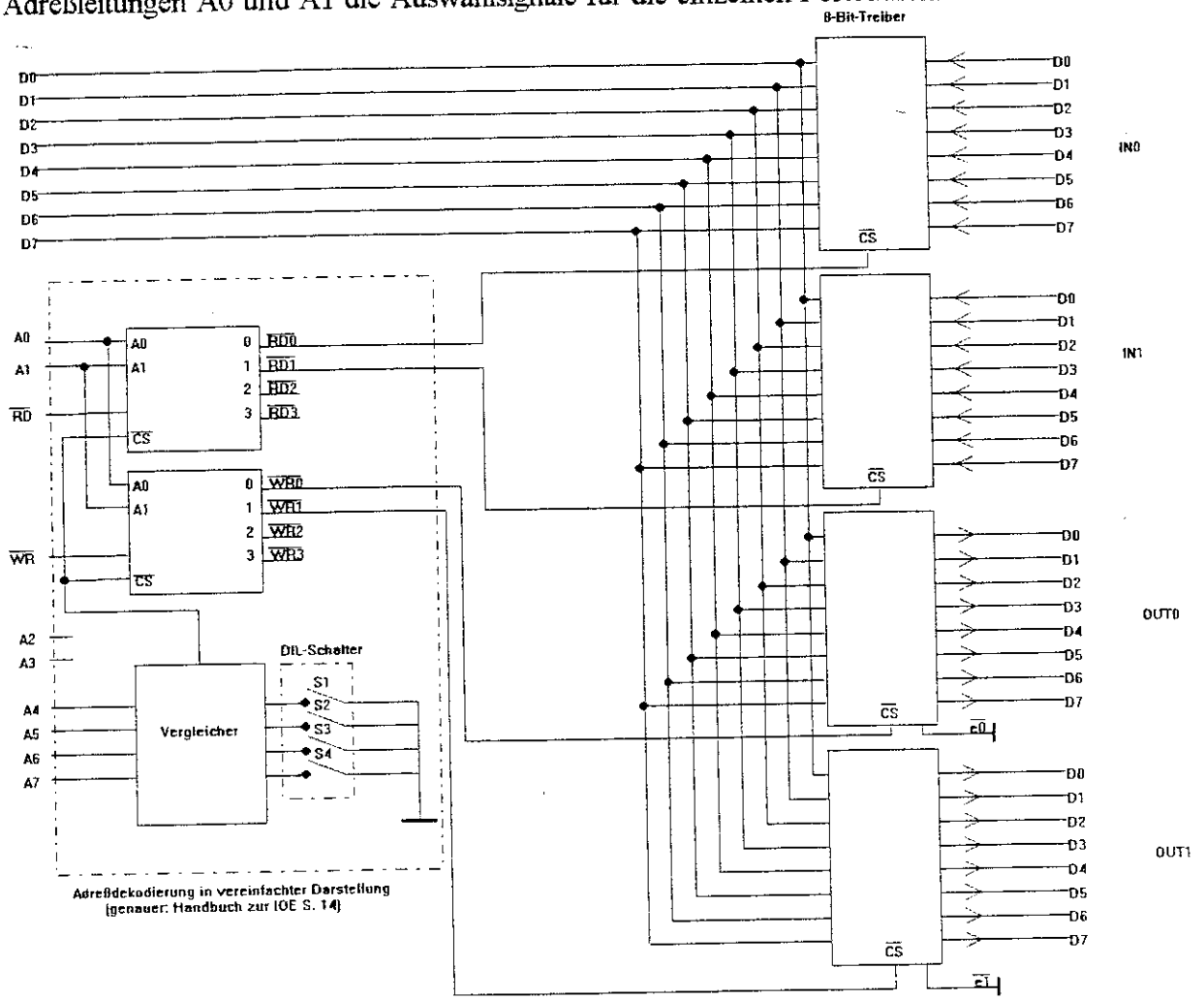
Mit dem Unterprogramm "Clear33" läßt sich der Speicherbereich von 8000h bis 8007h mit FFh füllen. FFh bedeutet Anzeige dunkel. Mit dem Aufruf von "Clear33" und "Anzeige9" kann die Anzeige gelöscht werden. Dieses Unterprogramm beginnt mit der Speicherstelle 0033h.

Holetastec

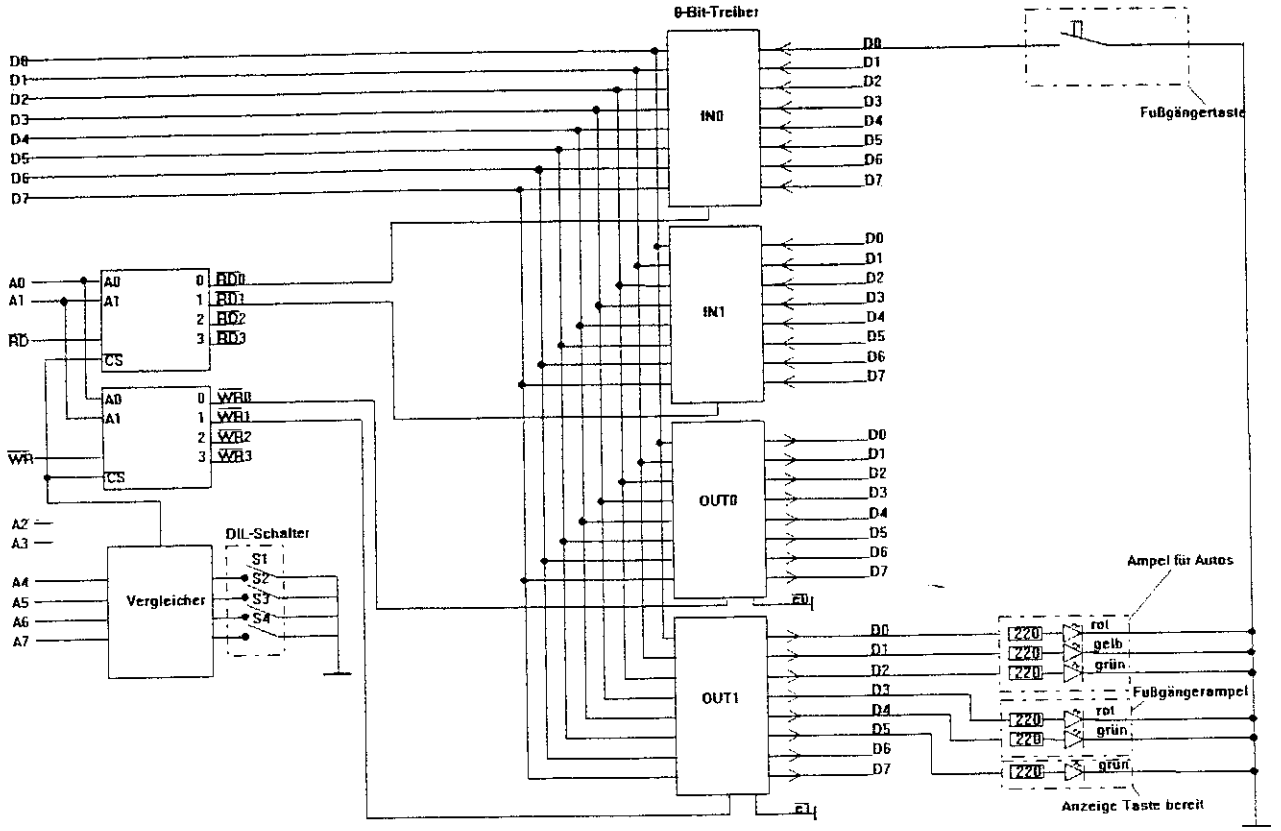
Das Unterprogramm "Holetastec" gibt die Anzeigecodes von den Speicherplätzen 8000h bis 8007h so lange an die Anzeige aus, bis eine Taste betätigt wird. Nach dem Tastendruck wird der Tastencode der betätigten Taste in den Akku geschrieben und ins aufrufende Programm zurückgesprungen. "Holetastec" belegt den Speicher ab der Adresse 000Ch.

4.2 Programmbeispiel mit IOE-Karte

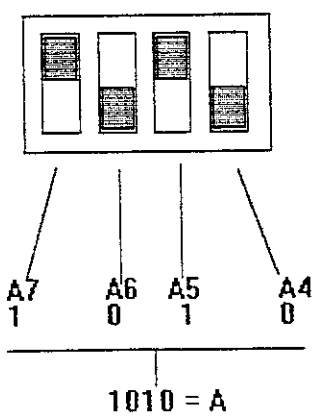
In der folgenden Skizze sehen Sie die Schaltung der IOE-Karte. Die Karte verfügt über je zwei 8-Bit breite Ein- und Ausgabeports. Um diese Ports ansteuern zu können, wird wiederum eine Adreßdekodierung benötigt. Wie diese funktioniert, ist Ihnen von den Ein- Ausgabeports auf der HEXIO2 bekannt. Hier werden zur Adressierung die Signale $\overline{RD}$, $\overline{WR}$, $\overline{IORQ}$ und das Low-Wort des Adreßbusses verwendet. Mittels der DIL-Schalter S1-S4 kann das höherwertige Nibble (1 Nibble = 4 Bit) der Portadresse eingestellt werden (dies war bei der HEXIO2 nicht möglich). Wird ein Schalter geschlossen, dann gelangt eine logische Null zum Eingang des Vergleichers. Der Vergleicher gibt eine logische 0 auf die $\overline{CS}$-Eingänge der Auswahllogik, wenn die Bitmuster auf beiden Seiten übereinstimmen, d.h. A4 = S1, A5 = S2, usw. Wird nun die Auswahllogik vom Vergleicher aktiviert, generiert Sie aus den Signalen $\overline{RD}$, $\overline{WR}$ und den Adreßleitungen A0 und A1 die Auswahlsignale für die einzelnen Portbausteine.



Die Möglichkeiten der IOE-Karte sollen nun Anhand eines Beispiels erläutert werden. Hierzu wird eine Fußgängerampel simuliert. Die Schaltung besteht aus einer Ampel für die Fahrbahn (rot, gelb, grün), der Fußgängerampel (rot, grün), sowie einer Bereitschaftsanzeige für die Taste (grün). Bauen Sie dazu mit den Ihnen mitgelieferten Bauteilen folgende Schaltung auf:

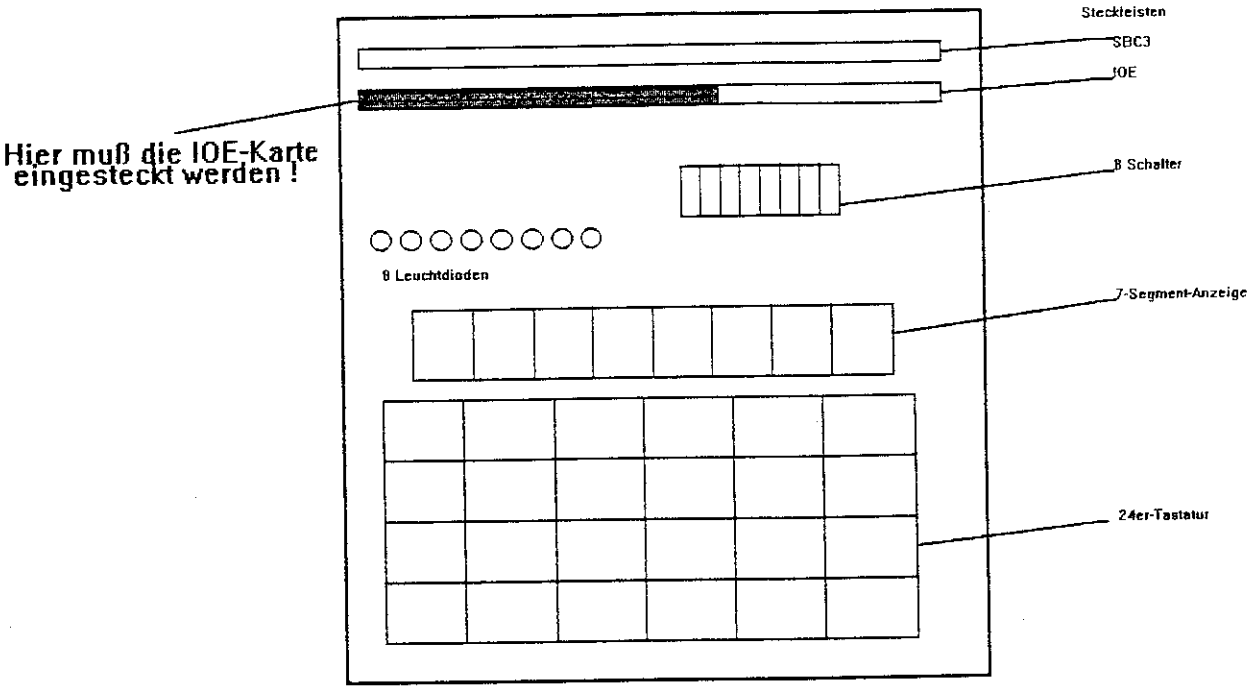


Sie benötigen hierzu die Belegung der Steckleiste auf der IOE-Karte. Diese finden Sie in Ihrem IOE-Handbuch auf Seite 17. Hier ist nur der Schaltplan gegeben. Wie Sie den Aufbau auf Ihrer IOE-Karte gestalten bleibt Ihnen überlassen. Ihr Portbaustein OUT1 (74LS374) kann natürlich nur die Leuchtdioden ansteuern, wenn sein Ausgang aktiviert wird. Dies wird mit der Verbindung (Lötbrücke zum benachbarten Lötpunkt) $\overline{e1}$ nach Masse durchgeführt (siehe Handbuch, Teil IOE Seite 4). Das folgende Programm benutzt für die IOE-Karte die Adressen A0h und A1h. Sie müssen mit den DIL-Schaltern auf der Karte das höherwertige Nibble, den Adreßteil A, einstellen. Die binäre Darstellung der Hexziffer A ist 1010. Die Stellung der Schalter ist demzufolge wie folgt:



So wird die Karte adressiert. Die 4 höherwertigen Bits stehen als Adreßbits fest, aus den Bits A0 und A1 werden die Aktivierungssignale für die Portbausteine generiert. In unserem Fall wird bei A0 = A1 = 0 der Port IN0 aktiviert und bei A0 =1 und A1 = 0 der Port OUT1. Die Ports IN1 und OUT0 werden nicht verwendet.

Wenn Sie die Karte auf Ihre HEXIO2-Einheit stecken, müssen Sie darauf achten, daß die IOE-Karte linksbündig eingesteckt werden muß. Ein falsches Einstecken kann zur Zerstörung von Bauteilen führen!



Sie können das Programm dann eingeben und testen. Wie bereits angesprochen ist diese Kurzbeschreibung nicht geeignet um das Programmieren zu lernen. Wir wünschen Ihnen aber trotzdem viel Spaß mit unseren Beispielen.

Programm zur Ampelsteuerung

Unterprogramm Zeit

Adresse	HEX-Code	Befehl	Erläuterung
9100	21	LD HL, 0000h	HL-Registerpaar für Zeitschleife setzen.
9101	00		
9102	00		
9103	11	LD DE, 0001h	DE-Registerpaar für Zeitschleife setzen.
9104	01		
9105	00		
9106	19	ADD HL, DE	Addiere den Inhalt von DE zum Inhalt von HL
9107	30	JR NC, FD	Springe, solange das Carry-Flag nicht gesetzt ist nach 9106 (genau: 3 Adressen zurück).
9108	FD		
9109	05	DEC B	Dekrementiere das B-Register.
910A	20	JR NZ, F4	Springe, solange der Inhalt des B-Registers nicht null ist, nach 9100 (12 Adressen zurück).
910B	F4		
910C	C9	RET	

Hauptprogramm

Adresse	HEX-Code	Befehl	Erläuterung
9000	3E	LD A, 2Ch	Akku mit dem Bitmuster für Ruhestellung laden.
9001	2C		
9002	D3	OUT (A1h), A	Akkuinhalt an Port A1 ausgeben.
9003	A1		
9004	DB	IN (A0h), A	Daten vom Port A0 (IN0) in den Akku lesen.
9005	A0		
9006	E6	AND 01h	Akkuinhalt mit 01h UND verknüpfen. (=Maskierung des benötigten Bits vom Taster)
9007	01		
9008	20	JR, NZ FA	Springe nach 9004 bis die Taste gedrückt wird (6 Adressen zurück).
9009	FA		
900A	3E	LD A, 0C	Akku mit dem Bitmuster "Tastendruck" laden.
900B	0C		
900C	D3	OUT (A1h), A	Bitmuster auf den Port A1 ausgeben.
900D	A1		
900E	06	LD B, 20h	20h als Parameter ins B-Register laden.
900F	20		
9010	CD	CALL 9100h	Aufruf des Unterprogramms an der Adresse 9100. Hier: "Zeit"
9011	00		
9012	91		
9013	3E	LD A, 0Ah	Bitmuster "Gelb für Autofahrer" in den Akku laden.
9014	0A		
9015	D3	OUT (A1h), A	Bitmuster an Port A1h ausgeben.
9016	A1		
9017	06	LD B, 05h	Parameter für Warteschleife ins B-Register laden.
9018	05		
9019	CD	CALL 9100h	Aufruf des Unterprogramms "Zeit" an der

901A	00		Speicherstelle 9100h.
901B	91		
901C	3E	LD A, 09h	Akku mit dem Bitmuster "Rot für Autofahrer" laden.
901D	09		
901E	D3	OUT (A1h), A	Akkuinhalt auf Port A1h ausgeben.
901F	A1		
9020	06	LD B, 03h	Parameter für Zeitschleife ins B-Register laden.
9021	03		
9022	CD	CALL 9100	Aufruf des Unterprogramms "Zeit".
9023	00		
9024	91		
9025	3E	LD A, 11h	Akku mit Bitmuster "Grün für Fußgänger" laden
9026	11		
9027	D3	OUT (A1h), A	Akkuinhalt auf Port A1h ausgeben.
9028	A1		
9029	06	LD B, 23h	Parameter für Zeitschleife ins B-Register laden.
902A	23		
902B	CD	CALL 9100	Aufruf des Unterprogramms "Zeit".
902C	00		
902D	91		
902E	3E	LD A, 09h	Akku mit Bitmuster "Rot für Fußgänger" laden.
902F	09		
9030	D3	OUT (A1h), A	Akkuinhalt auf Port A1h ausgeben.
9031	A1		
9032	06	LD B, 13h	Parameter für Zeitschleife ins B-Register laden.
9033	13		
9034	CD	CALL 9100	Aufruf des Unterprogramms "Zeit".
9035	00		
9036	91		
9037	3E	LD A, 0Bh	Akku mit dem Bitmuster "Rot/gelb für Autofahrer laden.
9038	0B		
9039	D3	OUT (A1h), A	Akkuinhalt auf Port A1h ausgeben.
903A	A1		
903B	06	LD B, 05h	Parameter für Zeitschleife ins B-Register laden.
903C	05		
903D	CD	CALL 9100	Aufruf des Unterprogramms "Zeit".
903E	00		
903F	91		
9040	C3	JP 9000	Sprung nach 9000h, Endlosschleife.
9041	00		
9042	90		