

---

# Handbuch LC 80



# Handbuch Lerncomputer LC 80

(auszugsweise, nur die den LC-80 betreffenden Kapitel)

PDF-Version: Volker Pohlens, 2003

**veb mikroelektronik > karl marx < erfurt**  
**Stammbetrieb**

DDR- 5010 Erfurt, Rudolfstraße 47 Telefon 580, Telex 061306

Änderungen, insbesondere solche, die durch den technischen Fortschritt bedingt sind, vorbehalten.

## Inhaltsübersicht

	Seite
0. Einleitung	6
1. Grundbegriffe der Informationsdarstellung in Mikrocomputern	8
1.1. Binärelemente	8
1.2. Bit, Byte und Wort	9
1.3. Digitale Codes	10
1.4. Zahlensysteme	11
1.4.1. Das reine Binärsystem	11
1.4.2. Das Hexadezimalsystem	15
1.5. Darstellungsarten von Zahlen im Byte	17
1.5.1. Darstellungsart positive Binärzahl	17
1.5.2. Darstellungsart Zweierkomplementzahl	17
1.5.3. Darstellungsart gepacktes BCD-Format	18
2. Beschreibung des Mikroprozessorsystems U 880	20
2.1. Übersicht über das Mikroprozessorsystem U 880	20
2.2. Aufbau und Arbeitsweise des Mikroprozessors U 880	25
2.2.1. Registerstruktur	29
2.2.2. Zeitverhalten der CPU U 880	32
2.2.3. Interruptsystem	42
2.3. Aufbau und Arbeitsweise des PIO U 855 52	52
2.3.1. Schaltkreisbeschreibung	52
2.3.2. Erläuterung der einzelnen Betriebsarten	59
2.3.2.1. Betriebsart Byte-Ausgabe (Mode 0)	59
2.3.2.2. Betriebsart Byte-Eingabe (Mode 1)	60
2.3.2.3. Betriebsart Byte-Ein-/Ausgabe (bidirektional) (Mode 2)	61
2.3.2.4. Bit-Ein-/Ausgabe (Mode 3)	62
2.3.3. Interrupt-Bearbeitung	63
2.3.4. Programmierung des PIO's	64
2.3.4.1. Wahl der Betriebsart	64
2.3.4.2. Ein- bzw. Ausgabedefinition bei Betriebsart Bit-Ein-/Ausgabe	64

2.3.4.3.	Laden des Interrupt-Vektors	65
2.3.4.4.	Interrupt-Steuerung	65
2.4.	Aufbau und Arbeitsweise des CTC U 857	67
2.4.1.	Schaltkreisbeschreibung	67
2.4.2.	Arbeitsweise des Schaltkreises	73
2.4.2.1.	Schreibzyklus	73
2.4.2.2.	Lesezyklus	74
2.4.2.3.	Interrupt-Quittungs-Zyklus	75
2.4.2.4.	Prioritätskaskadierung	77
2.4.2.5.	Zähl- und Zeitgeber-Vorgang	79
2.4.3.	Programmierung des CTC	79
2.4.3.1.	Laden des Interrupt-Vektors	80
2.4.3.2.	Format des Kanalsteuerwortes	81
2.4.3.3.	Format des Zeitkonstantensteuerwortes	82
2.5.	Speicherschaltkreise	83
2.5.1.	Statischer Schreib-Lesespeicher U 214 (RAM)	83
2.5.2.	Statischer Festwertspeicher U 505	84
2.6.	Randelektronik des LC 80	85
2.6.1.	Decoderschaltkreis DS 8205	85
2.6.2.	Schmitt-Trigger Inverterschaltkreis DL 014	86
2.6.3.	NAND-Gatter Schaltkreis DL 000	86
3.	Lerncomputer LC 80/Schaltungskonfiguration und Wirkungsweise	87
3.1.	Grundkonzept des Gerätes	87
3.2.	Speicherbereiche und Adreßdecodierung	88
3.3.	Tastatureingabe und Displaysteuerung	94
3.4.	Adressierung der Peripherieschaltkreise (PIO/D 206, 207 und CTC/D 208)	98
3.5.	Prioritätskaskadierung der Peripherieschaltkreise	101
3.6.	RESET-Erzeugung für die PIO's	102
3.7.	Magnetbandinterface	103
3.8.	Takterzeugung	104
3.9.	Betriebespannungserzeugung und Bemerkungen zur zusätzlichen Stromentnahme	105

4.	Befehlsbeschreibung des U 880	107
4.1.	Befehlsstruktur	107
4.2.	Syntax der Assemblersprache	109
4.3.	Adressierungsarten	110
4.4.	Flag-Bit-Technik	112
4.5.	Befehlssatz	115
4.5.1.	Ladebefehle	116
4.5.1.1.	8-Bit-Ladebefehle	116
4.5.1.2.	16-Bit-Ladebefehle	126
4.5.2.	Registertauschbefehle	124
4.5.3.	Blocktransfer- und Suchbefehls	127
4.5.4.	Arithmetikbefehle	132
4.5.4.1	8-Bit-Arithmetikbefehle	132
4.5.4.2.	16-Bit-Arithmetikbefehle	134
4.5.5.	Sprungbefehle	135
4.5.6.	1-Byte-Logik-Befehle	137
4.5.7.	Rotations- und Schiebebefehle	139
4.5.8.	Bitmanipulationsbefehle	145
4.5.9.	Spezielle Akkumulator- und Flagbefehle	146
4.5.10.	Unterprogrammaufruf- und Rücksprungbefehle	147
4.5.11.	Allgemeine Steuerbefehle	150
4.5.12.	Ein- und Ausgabebefehle	151
4 5.13.	Pseudo-Assembler Anweisungen	153
5.	Programmierung der Peripherieschaltkreise des LC 80	155
5.1.	Programmatische Organisation einer Interrupt-ServiceRoutine (ISR)	155
5.2.	PIO-Programmierung	158
5.3.	CTC-Programmierung	162
6.	Befehlsliste U 880	165
Anhang:	Sachwörterläuterungen	181
Anlage:	Bestückungsplan LC 80	

Es folgen nur die Kapitel 0, 3, 5 und Anlage. Allgemeine Informationen zur Programmierung des U 880 können der Literatur entnommen werden. (vp)

## **0. Einleitung**

Mit dem Einzug der Mikroelektronik in alle Zweige der Volkswirtschaft entwickelt sich bei vielen Menschen der Wunsch, näheres über dieses interessante technische Gebiet zu erfahren und sich auch entsprechend selbst zu betätigen. Erzeugnisse, die früher ohne Elektronik auskamen, beinhalten heute die Mikroelektronik als deren miniaturisierte Form, z. B.

- Fotoapparate mit automatischer Einstellfunktion
- Werkzeugmaschinen mit mikroelektronischer Steuerung der Funktionen
- Landwirtschaftsmaschinen mit mikroelektronischer Steuerung der Mähhöhe, Pflugtiefe usw.
- Haushaltgeräte mit Programmablaufspeicherung.

Dazu kommen völlig neue Geräte und Einrichtungen, die nur mit Mikroelektronik möglich sind, z. B.

- Rechner verschiedenster Art
- Fahrkartenautomaten
- mikroelektronische Spiele.

Aus den genannten Beispielen wird deutlich, daß sich auch solche Berufsgruppen mit der Mikroelektronik beschäftigen, deren Haupttätigkeitsfeld der Maschinenbau, die Landwirtschaft, die Optik usw. ist.

Der Lerncomputer LC 80 ist dafür gedacht, der Bevölkerung im allgemeinen und Schüler, Studenten und Berufstätigen in der Aus- und Weiterbildung das modernste Gebiet der Mikroelektronik, die Mikroprozessortechnik, näher zu bringen.

Der Ursprung der Mikroprozessortechnik liegt im Bestreben der Menschen, so effektiv und preiswert wie möglich elektronische Bauelemente zu produzieren. Die Stückzahlen solcher Bauelemente (Schaltkreise) sind ein entscheidendes Kriterium.

Da die verschiedenen Anwender jeweils verschiedene Schaltkreise bei unökonomischen Stückzahlen, jedoch recht billig, verlangten, wurden die Techniker zu akzeptablen Lösungen angehalten. Sie schufen ein System von Schaltkreisen, die elektrisch zu einer



Schaltung verbunden werden müssen. Dies ist ein Mikroprozessorsystem, dessen Kernstück ("Gehirn") ein Mikroprozessor (CPU) ist. Solche Zusammenschaltung eines Systems von Schaltkreisen nennt man "Hardware", was gleichbedeutend ist mit "körperlich vorhanden". Die Hardware allein löst keine der in den Beispielen genannten Anwendungsfälle. Man muß dem Mikroprozessorsystem erst "eingeben" welche Aufgabe es erledigen soll. Das macht man mit der "Software", einem Programm.

Jedes der genannten Beispiele besteht also aus einer konkreten Hardware und Software, Während die Hardware in ihrem Kern sehr ähnlich ist (immer ist ein Mikroprozessor, z. B. U 880, eingesetzt), hat natürlich ein Fahrkartenautomat eine andere Software als eine Werkzeugmaschine.

Um auf den Ausgangspunkt zurückzukommen: Bei großen Produktionsstückzahlen der Bauelemente können umfangreiche Anwendungsgebiete mit den gleichen Bauelementen befriedigt werden.

Bei Mikroprozessorsystemen erwirbt der Anwender also verschiedene Schaltkreise, schaltet sie zusammen und "sagt" diesem Gebilde mit einer Software, welche Funktion es zu erfüllen hat. Deshalb wird gelegentlich vom Mikroprozessor als "Alleskönner" gesprochen. Was er aber kann übernimmt er jedoch in einer geeigneten Sprache vom Menschen. Hierfür wurde der Begriff Maschinensprache geprägt, um den qualitativen Unterschied zur menschlichen Sprache zu verdeutlichen.

Der Lerncomputer macht nicht nur mit dem Mikroprozessorsystem als Hardware bekannt. Die Sprache des Mikroprozessors und seine Arbeitsweise wird in diesem Handbuch beschrieben und man kann sogleich am Lerncomputer sein erworbenes Wissen überprüfen. So wird es möglich, verschiedene Programme auszuprobieren und im fortgeschrittenen Stadium dem "Alleskönner" Funktionen zu übertragen. Selbstverständlich handelt es sich beim Lerncomputer LC 80 um einen relativ kleinen Umfang der "Hardware".

Wer sich intensiv mit der Mikroprozessortechnik beschäftigen will, wird die im Handbuch genannten Erweiterungsmöglichkeiten nutzen oder sogar auf einer extra Leiterplatte eine spezielle Variante eines Mikroprozessorsystems aufbauen.

Mit der Maschinensprache wird der Mikroprozessor "direkt" angesprochen. Um diese zu erlernen, muß man sich Kenntnisse über die Eigenschaften der Bauelemente aneignen und kann sie dann maximal ausnutzen. Das Erlernen der Maschinensprache qualifiziert zum Programmierer, sowohl den Schüler als auch den Werk tätigen unterschiedlichen Alters und Berufes. Diese werden nur im Grad der Beherrschung der "Kniffe" und speziellen Fähigkeiten der Mikroprozessoren unterschieden.

Für solche Anwender, die mit Hilfe der Mikroelektronik "nur" bestimmte Probleme der Mathematik, Biologie, Statistik usw. lösen wollen, wurden "problemorientierte" Sprachen geschaffen, die in einem Befehl mehrere Maschinenbefehle realisieren, wie "Basic", "Pascal" usw.

Die Übersetzung in die Maschinensprache übernimmt in diesem Fall das Mikroprozessorsystem selbst. Auch diese Aufgabe wurde ihm vom Menschen mit einer entsprechenden "Software" vorgegeben. Moderne Heimcomputer sind vorwiegend für Basic vorgesehen.

Das Charakteristische der Maschinensprache ist, daß sowohl alle Informationen als auch alle Befehle in Zahlenform codiert werden müssen. Wie jede andere Sprache erfordert dies ein fleißiges Lernen der Vokabeln. Dem gegenüber steht die einfache "Grammatik" und "Syntax".

Wer also die erste Hürde genommen hat, wird mit Schwung, nach und nach die weiteren nehmen, um dann zum immer größer werdenden Kreis der "Mikroprozessor-Fan's" zu gehören.

### **3. Lerncomputer LC 80 / Schaltungskonfiguration und Wirkungsweise**

Nun sollen die bis hierher erworbenen Kenntnisse über die Mikroprozessortechnik des U 880-Systems anhand der Schaltungskonfiguration des LC 80 vertieft und das Gerät von seinem Aufbau her beschrieben werden.

Diese Kenntnisse werden später eine wertvolle Hilfe bei der Umsetzung eigener Anwendungsaufgaben in lauffähige Mikrorechnerprogramme sowie für das Gesamtverständnis des Zusammenwirkens Hardware - Software sein.

Die folgenden Ausführungen sind im Zusammenhang mit dem Stromlaufplan zu betrachten.

#### **3.1. Grundkonzept des Gerätes**

Der LC 80 enthält eine Minimalkonfiguration eines Mikrorechners, bestehend aus der CPU U 880 (D 201), den PIO's U 855 (D 206, 207), dem CTC U 857 (D208), dem 1 KByte-RAM mit zwei U 214 (D 204, 205) sowie 2 KByte ROM mit zwei U 505 (D 203, 202).

(Eine mögliche Speichererweiterung wird im nachfolgenden Abschnitt erklärt. Weiterhin ist Abschnitt 3.9. zu beachten).

Über sogenannte "Bus"-Strukturen sind die einzelnen Schaltkreise miteinander verbunden. Als "Bus" bezeichnet man die Zusammenfassung gleichartiger Signalleitungen und unterteilt in Daten-, Adreß- und Steuerbus. Die entsprechenden Schaltkreisein- bzw. -ausgänge liegen parallel am jeweiligen Bus an (so sind beispielsweise die einzelnen Datenleitungen D0 - D7 aller o. g. Schaltkreise jeweils zusammengeschaltet und bilden den Datenbus).

Die Beschaltung des Adreß- und Steuerbusses mit den einzelnen Schaltkreisen ist abhängig von der jeweils zu realisierenden Funktion und wird später noch genauer erläutert.

### 3.2. Speicherbereiche und Adreßdecodierung

Von der Busstruktur her liegen sämtliche Speicherschaltkreise parallel, d. h. die einzelnen Datenleitungen D0 - D7 und Adreßleitungen A0 ... A9 sind jeweils verbunden. Während eines Speicherzugriffes durch die CPU soll jedoch stets nur ein bestimmter Speicherplatz im ROM- oder RAM-Bereich angesprochen werden. Dies wird mit Hilfe zweier Decodierschaltkreise DS 8205 (D 209, 210) realisiert, die in Abhängigkeit von der jeweiligen Adresse und dem /MREQ-Signal der CPU stets nur einen bestimmten Speicherbereich durch Aktivierung der Freigabeeingänge (/CS=L) auswählen. Auf Grund der binären Arbeitsweise eines Mikrorechners verkörpert jeder Adreßwert eine bestimmte Binärzahl als Summe der positionsabhängigen Wertigkeit (vgl. Pkt. 1.4. Zahlensysteme) mit  $A_0 = 2^0$ ,  $A_1 = 2^1$ , ...  $A_{15} = 2^{15}$ .

Die in den Speicherschaltkreisen direkt über die Adreßeingänge A0 - A9 (bei 2 KByte-Speicher A0 - A10) auswählbaren Speicherplätze (je 8 Bit = 1 Byte-Breite) umfassen somit einen Bereich von

(A10)	A9	A8	...	....				A1	A0	
0	0	0	0	0	0	0	0	0	0	= 000H
.	.	.	.	.	.	.	.	0	0	1
.	.	.	.	.	.	.	.	1	0	
.	.	.	.	.	.	.	.	0	0	1
.	.	.	.	.	.	.	.	.	.	.
.	.	.	.	.	.	.	.	.	.	.
(1)	1	1	1	1	1	1	1	1	1	=3FFH (7FFH)

Das entspricht 1024 Byte = 1 KByte (bei 2 KByte-Speicher von 0 - 7FFH = 2048 Byte).

Damit wird also durch den unteren Adreßbereich A0 - A9 (A0 - A10) je ein 8 Bit-Speicherplatz innerhalb der Speicherschaltkreise bestimmt.

Durch die teilweise Auswertung der höherwertigen Adressen A11 ... A15 mittels der Decodierschaltkreise DS 8205 wird der jeweils gewünschte Speicherschaltkreis ausgewählt.

### ROM-Auswahl (durch D 209)

Hierbei werden die Decodiereingänge A, B, C und /E1, /E2, E3 wie folgt beschaltet:

Eingang A	A11
" B	A12
" C	H-Pegel <sup>x)</sup>
" /E1	A13
" /E2	/MREQ
" E3	/MEDI

Die Decodierausgänge bilden die einzelnen Speicherauswahlsignale /CS (CS = chip select) der ROM-Schaltkreise:

Ausgang 0	ROM 4
" 1	ROM 5
" 2, 3, 7	frei
" 4	ROM 1
" 5	ROM 2
" 6	ROM 3

Damit ergeben sich die in Tabelle 3.1. aufgeführten logischen Bedingungen zur Aktivierung der einzelnen ROM-CS-Signale (/CS = L).

(Vgl. auch Schaltkreisbeschreibung DS 8205).

<sup>x)</sup> An der Stelle B1 (siehe Stromlaufplan) kann ein Kleinschalter KSD 11 eingefügt werden und damit der Logikpegel an Eingang D 209/C von H nach L (Massepotential) geschaltet werden. (Gegebenenfalls kann auch nur eine Drahtbrücke eingefügt werden.) Wie aus Tabelle 3.1. ersichtlich, ist somit ein Umschalten zwischen ROM 1 und 4 sowie ROM 2 und 5 möglich (Eingang C = H: ROM 1, 2 und 3 programmäßig ansprechbar, ROM 4 und 5 abgeschaltet; Eingang C = L: ROM 4 und 5 programmäßig ansprechbar, ROM 1, 2 und 3 abgeschaltet.)

### RAM-Auswahl (durch D 210)

Die Beschaltung der Decodiereingänge erfolgt hierbei wie folgt:

Eingang A	A10
" B	A11
" C	/MEDI
" /E1	L-Pegel
" /E2	/MREQ
" E3	A13

Die Decodierausgänge bilden die einzelnen /CS-Signale der RAM-Schaltkreise:

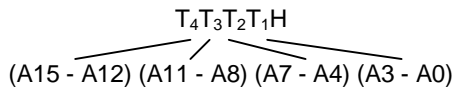
Ausgang 0 ... 3	frei
" 4	RAM 1
" 5	RAM 2
" 6	RAM 3
" 7	RAM 4

Damit ergeben sich die ebenfalls in Tabelle 3.1. aufgeführten logischen Bedingungen zur Auswahl der einzelnen RAM's.

Tabelle 3.1.: (x = beliebiger Logikpegel; besitzt keinen Einfluß auf die logische Verknüpfung)

	A15	A14	A13	A12	A11	A10	/MREQ	/MEDI	D209/C
ROM 1	x	x	0	0	0	x	0	1	1
ROM 2	x	x	0	0	1	x	0	1	1
ROM 3	x	x	0	1	0	x	0	1	1
ROM 4	x	x	0	0	0	x	0	1	0
ROM 5	x	x	0	0	1	x	0	1	0
RAM 1	x	x	1	x	0	0	0	1	x
RAM 2	x	x	1	x	0	1	0	1	x
RAM 3	x	x	1	x	1	0	0	1	x
RAM 4	x	x	1	x	1	1	0	1	x

Entsprechend der Tetradschreibweise mit



sind die logischen Bedingungen der /CS-Signale für folgende Adreßbereiche erfüllt:

Tabelle 3.2.:

Adreßbereich	Eing. D209/C	ausgewählter Speicherschaltkreis (/CS=L)
0000H ... 07FFH	1	ROM 1
	0	ROM 4
0800H ... 0FFFH	1	ROM 2
	0	ROM 5
1000H ... 17FFH	1	ROM 3
	0	-
2000H ... 23FFH	x	RAM 1
2400H ... 27FFH	x	RAM 2
2800H ... 2BFFH	x	RAM 3
2000H ... 2FFFH	x	RAM 4

Diese Adressen entsprechen dem Grundbereich mit A15 = A14 = 0 und außerdem dem RAM-Bereich mit A12 = 0. Da diese zwei bzw. drei Adressen von den Decodierschaltkreisen nicht ausgewertet werden (in Tabelle 3.1. mit "x" dargestellt) sind die einzelnen Speicherschaltkreise innerhalb des gesamten Adreßbereiches mehrfach ansprechbar. So sind die Adreßbereiche im Grundbereich bezüglich Speicherauswahl mit folgenden höheren Adreßbereich äquivalent:

Tabelle 3.3.:

Die Speicherschaltkreise im Grundbereich:	sind ebenfalls ansprechbar im Adreßbereich:
0xxxH	4xxxH 8xxxH BxxxH
1xxxH	5xxxH 9xxxH CxxxH
2xxxH	3xxxH 6xxxH 7xxxH AxxxH DxxxH ExxxH FxxxH

Das bedeutet beispielsweise, daß  
Unter den Adressen

0946H  
4946H  
8946H  
B946H

jeweils die gleiche Speicherzelle im ROM 2 auslesbar ist. (Es ist aber zu beachten, daß über die Tasteneingabe bei LC 80 ein unmittelbares Beschreiben von RAM-Speicherplätzen nur innerhalb des Grundbereiches möglich ist.)

Aus Tabelle 3.1. wird zugleich die Funktion des über Steckverbinder X2 (CPU-Bus) erreichbaren Signaleinganges /MEDI (memory dissable) deutlich:



Wird diese Signalleitung durch äußere Beschaltung bzw. Funktionseinheiten auf L-Pegel gezogen, so ist der gesamte LC 80-interne Speicherbereich abgeschaltet und von außen her können danach Speicher mit beliebiger Adreßaufteilung wirksam werden (Verwendung des gesamten 64 KByte-Adreßbereiches möglich). Der Bestückungsplan gibt nochmals einen Überblick über die mögliche Speichererweiterung. Mit der Erweiterung durch einzelne Speicherschaltkreise wird gleichzeitig die Bestückung mit je einem zugehörigen Stützkondensator 10 nF empfohlen. Damit besteht also die Möglichkeit, bestimmte Programme oder Daten bis zu 2 KByte als PROM auf Pos. "ROM 3" zu bestücken und diese Programme mit Hilfe der Monitorfunktion des LC 80 zu starten (analog einem Programm im RAM, hierbei nur entsprechende Startadresse im Bereich 1000H ... 17FFH).

Außerdem können Programme oder Daten bis zu 4 KByte als zwei PROM's auf Pos. "ROM 4" und "ROM 5" bestückt werden.

Gleichzeitig ist dann der bereits vorn erwähnte Kleinschalter auf Pos. B19 einzufügen (oder Drahtbrücke U-M).

In Schalterstellung "U" (User) wird dann das Anwenderprogramm nach jedem RESET ab der Adresse 0000H abgearbeitet. Nach Umschalten auf "M" ist wieder das Monitorprogramm und gegebenenfalls ROM 3 wirksam.

Zum besseren Verständnis des Zusammenwirkens Hardware - Software soll noch einmal in kurzer Form die prinzipielle Wirkung einer CPU-Befehlsabarbeitung auf den Speicherbereich erläutert werden: Ausgangspunkt soll beispielsweise sein, daß die CPU

- a) den Befehl LD A,(0974H) und
- b) den Befehl LD (2100H),A auszuführen hat.

Das heißt, der jeweilige Befehl wurde in Form des zugehörigen Maschinencodes zuvor im Rahmen der Programmabarbeitung aus dem Speicher gelesen, innerhalb der CPU decodiert und damit "erkannt", daß

- a) ein Speicher-Lesezyklus und
- b) ein Speicher-Schreibzyklus durchzuführen ist.

Diese laufen nach einem festen Schema, wie es im Abschnitt "Zeitverhalten der CPU U 880" dargestellt ist, ab (vgl. Bild 2.6.). In Fall a) gibt damit die CPU die Adresse 0974H, /MREQ = 0, /RD = 0 und /WR = 1 aus.

Nach Tab. 3.1. sind somit die logischen Bedingungen für /CS = 0 von ROM 2 erfüllt und ROM 2 ist damit aktiv.

Die interne Adreßdecodierung im Speicher ROM 2 wertet über ihre Adreßeingänge den unteren Adreßbereich A0 - A9 aus (der Adresse 0974H entspricht also eine interne Adresse von 174H) und gibt den Inhalt der durch diese interne Adresse festgelegten Speicherzelle (8 Bit) auf den Datenbus (D0 - D7).

Diese Information wird nun über die Dateneingänge der CPU übernommen und intern gemäß abzuarbeitendem Befehl in das A-Register übertragen.

Im Fall b) gibt die CPU /MREQ = 0, /RD = 1, /WR = 0, über den Adreßbus die Adresse 2100H und über den Datenbus den Inhalt des A-Registers aus.

Nach Tab. 3.1. sind somit die logischen Bedingungen für /CS = 0 von RAM 1 erfüllt und dieser damit aktiv.

Da über den RAM-Eingang /WR die Datenrichtung bestimmt wird (/WR = 0 Einschreiben der anliegenden Daten in die über A0 - A9 ausgewählte Speicherzelle; /WR = 1 Auslesen dieser Speicherzelle), erfolgt somit ein Abspeichern des A-Registerinhaltes in der angewählten RAM-Zelle (Dabei entspricht dieser Adresse 2100H die interne RAM-Adresse 100H).

### **3.3. Tastatureingabe und Displaysteuerung**

Das Kernstück dieser Funktionsgruppe sind die beiden PIO's/D 206 und D 207, die, gesteuert durch das Monitorprogramm, die Tastaturabfrage und parallel dazu die Ansteuerung der Siebensegmentanzeigen (3 x VQE 23; gemeinsame Katode) H 322 - 324 realisieren.

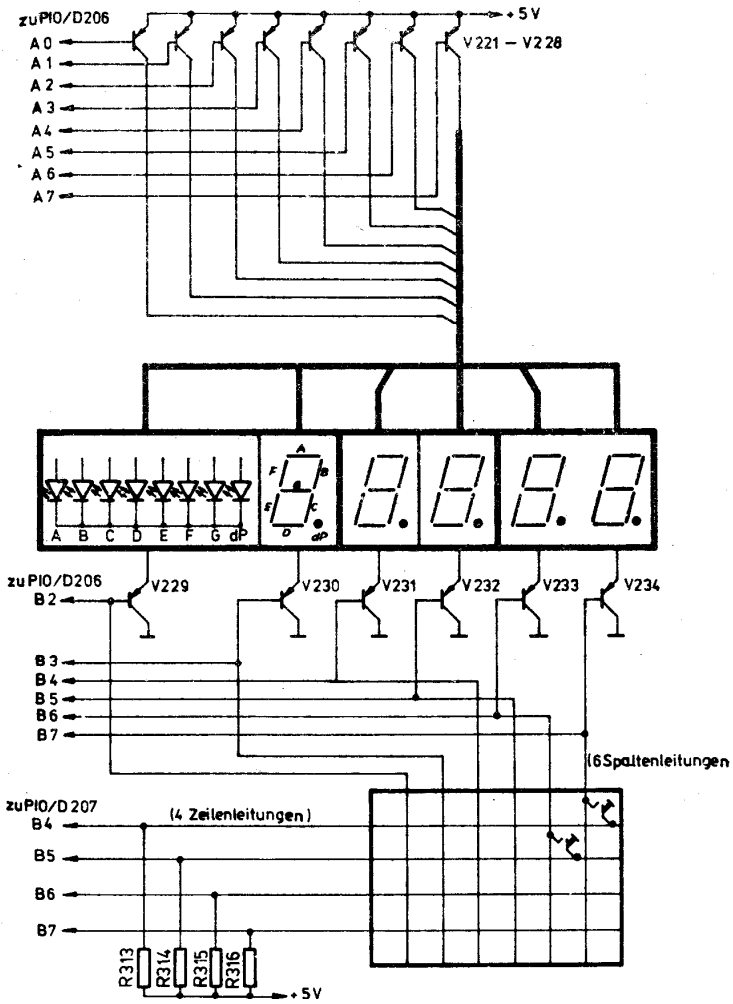


Bild 3.1.: Prinzipdarstellung zur Funktionserläuterung

Im Bild 3.1. ist das Prinzip der Displayansteuerung und Tastaturbeschaltung dargestellt.

Durch das Monitorprogramm werden die hierfür verwendeten PIO-Ports entsprechend der zu erfüllenden Aufgaben programmiert:

PIO/D 206	A-Port:	Einzelbitsteuerung (Bit-Ein/Ausgabe) Ausgänge
PIO/D 206	B-Port:	Einzelbitsteuerung B1 - B7 Ausgänge, B0 Eingang
PIO/D 207	B-Port	Einzelbitsteuerung Eingänge (B0 - B3 über "User-Bus" zugänglich)

#### Displayansteuerung:

Die Ansteuerung der sechs Siebensegmentstellen erfolgt im Multiplexbetrieb, d. h. die einzelnen Displaystellen (Digit's) werden zeitlich nacheinander aktiviert und zwar in einer für das menschliche Auge nicht mehr wahrnehmbaren Geschwindigkeit. Somit erscheint es uns, als ob alle sechs Stellen gleichzeitig leuchten.

Dieses Verfahren bietet den Vorteil, alle Segmente A-dP der sechs Digits jeweils parallel zu schalten und somit den Schaltungsaufwand stark zu reduzieren.

Die Segmentinformation (Segment aus/leuchtet) liegt also gleichzeitig an jedem Digit an, jedoch können nur jene Segmente leuchten, deren Strompfad durch den als Schalter wirkenden Transistor zwischen den Segmentkatoden und Massepotential geschlossen ist. Durch die verwendete Beschaltung sind folgende logische Pegel zur Displayansteuerung erforderlich:

Tabelle 3.4.:

<u>PIO-Port Aufgabe</u>	<u>log. Pegel</u>	<u>Wirkung</u>
D 206/A Ansteuerung der einzelnen Segmente A-dP mittels der als Schalter wirkenden Transistoren V 221 ... 228	0	jeweiliges Segment aktiv (Transistor durchgesteuert, Strompfad nach +5 V hin geschlossen)
	1	jeweiliges Segment inaktiv (Transistor gesperrt, Strompfad unterbrochen)
D 206/B Ansteuerung der einzelnen Digits 1 - 6 mittels der als Schalter wirkenden Transistoren V 229 - 234	0	jeweiliges Digit aktiv, alle aktivierten Segmente dieses Digits leuchten (Transistor durchgesteuert, Strompfad nach Masse hin geschlossen)
	1	jeweiliges Digit inaktiv, kein zugehöriges Segment leuchtet (Transistor gesperrt, Strompfad unterbrochen)

Zum Multiplexbetrieb der Anzeige innerhalb des Monitorprogrammes wird nun zyklisch nacheinanderfolgend über je einen Ausgang B2 - B7/D 206 "Low"- und über die restlichen Ausgänge "High"-Pegel, sowie über A0 - A7/D 206 die darzustellende Segmentinformation ausgegeben.

#### Tastaturabfrage:

Die einzelnen Tasten der LC 80-Tastatur sind in einer sogenannten Matrix angeordnet (außer RES und NMI), so daß man in Anlehnung an den mathematischen Begriff von Zeilen- und Spaltenleitungen spricht. Die einzelnen Tasten verbinden bei Betätigung je

eine Zeilen- mit einer Spaltenleitung.

Die sechs Spaltenleitungen sind mit den Ausgängen B2 - B7 von PIO/D 206 verbunden.

Die Tastenabfrage erfolgt innerhalb des Monitorprogrammes stets in Verbindung mit der Displayansteuerung.

Wie oben beschrieben, erfolgt dabei ein zyklisches Durchschieben eines Low-Pegels von B2 - B7 des PIO/D 206. Nach jeder dieser Ausgabeoperationen werden über B4 - B7 von PIO/D 207, die mit den Zeilenleitungen verbunden sind, die jeweiligen Zeileninformationen abgefragt. Im Ausgangszustand liegen die Zeilenleitungen über die Widerstände R 313 - 316 auf High-Pegel. Wird eine Taste betätigt und verbindet somit die jeweilige Zeilen- und Spaltenleitung, so ändert sich der logische Pegel dieser Zeilenleitung von High -> Low, sobald die entsprechende Spaltenleitung Low-Pegel besitzt.

Durch Auswertung der Nummer der Low-Pegel-führenden Zeilen- und Spaltenleitung wird dann die betätigte Taste ermittelt. Die Tasten "RES" und "NMI" wirken in Verbindung mit je einer RC-Beschaltung zur Entprellung (unsichere Kontaktgabe im Schaltmoment kann zu Mehrfachauslösung führen; bei restlichen Tasten programmäßige Entprellung durchgeführt) und zwei Negatoren direkt auf die entsprechenden CPU-Eingänge.

Nach Zuschalten der Versorgungsspannung des LC 80 muß der Kondensator C 243 erst über R 271 aufgeladen werden, so daß eine gewisse Zeit  $/RES = L$  wirksam ist.

Dieses als "power on" Reset bezeichnete Rücksetzen bringt den Computer nach jedem Einschalten in einen definierten Grundzustand.

### **3.4. Adressierung der Peripherieschaltkreise (PIO/D 206, 207 und, CTC/D 208)**

Zur Ausführung der gewünschten PIO- bzw. CTC-Betriebsarten sind diese durch entsprechende CPU-Befehle zu programmieren sowie zwischen ihnen und der CPU entsprechende Informationen auszutauschen. Hierfür verfügt die CPU über zahlreiche IN- und OUT-Befehle. Bei der Betrachtung der symbolischen Operationen der

Ein-Ausgabebefehlsgruppe (siehe Befehlsliste) sowie des zugehörigen Zeitverhaltens dieser E/A-Lesezyklen (IN) bzw. E/A-Schreibzyklen (OUT) (siehe Abschn. Zeitverhalten CPU U 880) ist folgendes Grundschema zu erkennen:

- z. B. IN A,(FAH) Durch die CPU wird hierbei über AO - A7 die Adresse FAH ausgegeben sowie /IORQ = 0 zur Kennzeichnung des E/A-Zyklus und /RD = 0, /WR = 1 zur Kennzeichnung der Datenrichtung (Lesezyklus)
- OUT (FAH),A Im Unterschied zum IN-Befehl ist hierbei /RD = 1, /WR = 0 zur Kennzeichnung des Schreibzyklus.

Hiervon ausgehend folgt die Betrachtung der Beschaltung der PIO's/D 206, 207 und des CTC/D 208 im LC 80 (dazu sollte man sich nochmals die Bedeutung der einzelnen Anschlüsse der Peripherieschaltkreise vergegenwärtigen):

Tabelle 3.5.:

	A7	A6	A5	A4	A3	A2	A1	A0
PIO/D 206	x	x	x	1 <sup>x)</sup>	/CS	1 <sup>x)</sup>	C//D	B//A
D 207	x	x	x	1 <sup>x)</sup>	1 <sup>x)</sup>	/CS	C//D	B//A
CTC/D 208	x	x	x	/CE	1 <sup>x)</sup>	1 <sup>x)</sup>	CS1	CS0

(x: beliebiger logischer Pegel; wird durch den Schaltkreis nicht ausgewertet.)

- <sup>x)</sup> Leitung wird durch den Schaltkreis nicht direkt ausgewertet, sie muß jedoch H-Pegel führen, da sonst gleichzeitig ein weiterer Schaltkreis ausgewählt werden würde (so werden beispielsweise mit der Adresse F0H beide PIO's/AD angesprochen).

Es ist zu erkennen, daß ähnlich der Speicherauswahl durch Ausgabe bestimmter Adreßwerte sowie Steuersignale, jeweils eine bestimmte Peripherieeinheit (z. B. ein bestimmter CTC-Kanal oder PIO-Port) angesprochen bzw. ausgewählt wird. Unter Beachtung der Bedeutung dieser einzelnen Schaltkreisanschlüsse ergeben sich folgende Zugriffsadressen:

Tabelle 3.6.

	interne Anwahl	verwendbare Adresse (B)	Grund- adresse (H)
PIO/D 206	Port A/Daten (AD)	x x x 1 0 1 0 0	F4
	Port B/Daten (BD)	x x x 1 0 1 0 1	F5
	Port A/Steuer- wort (AC)	x x x 1 0 1 1 0	F6
	Port B/Steuer- wort (BC)	x x x 1 0 1 1 1	F7
PIO/D 207	Port A/Daten	x x x 1 1 0 0 0	F8
	Port B/Daten	x x x 1 1 0 0 1	F9
	Port A/Steuerwort	x x x 1 1 0 1 0	FA
	Port B/Steuerwort	x x x 1 1 0 1 1	FB
CTC/D 208	Kanal 0	x x x 0 1 1 0 0	EC
	1	x x x 0 1 1 0 1	ED
	2	x x x 0 1 1 1 0	EE
	3	x x x 0 1 1 1 1	EF

Bei den hierbei als Grundadresse bezeichneten Worten wurde allen nicht ausgewerteten Adreßleitungen H-Pegel zugeordnet. Damit ergibt sich bei einer evtl. Peripherieerweiterung durch zusätzliche Anwender-PIO's (über Steckverbinder CPU-Bus) die einfache Möglichkeit, für deren /CS-Anschlüsse A7 - A5 zu nutzen.

Ansonsten ist beispielsweise PIO D 207/AD über die Adressen F8H, 18H, 58H, 78H, 98H, B8H, D8H erreichbar.

Kehren wir nochmals kurz zu den o. g. Ein/Ausgabebefehlen zurück. Es soll angenommen werden, daß die CPU mit der Ausführung



- a) des Befehls    OUT (ECH),A    (A besitze den Wert 01H) und  
b)                IN (A),F8H        beginnt und somit gemäß

Abschnitt "Zeitverhalten der CPU U 880"

- a) einen E/A-Schreibzyklus und  
b) einen E/A-Lesezyklus nach Bild 2.7.

durchfährt.

Im Fall a) gibt damit die CPU die Adresse 01ECH, /IORQ = 0, /RD = 1, /WR = 0 sowie als Daten 01H aus.

Nach Tab. 3.6. ist somit der CTC-Kanal 0 angesprochen. Durch /RD = 1 (nicht lesen = ^ Schreibzyklus) übernimmt der CTC die anliegenden Daten D0 - D7 in Kanal 0, wo sie in diesem Falle (01H, vgl. Programmierung CTC) als Kanalsteuerwert interpretiert werden.

Im Fall b) gibt die CPU die Adresse 01F8H, /IORQ = 0, /RD = 0, /WR = 1 aus. Damit wird PIO D 207/AD angesprochen. Durch /RD = 0 (Lesezyklus) wird der PIO veranlaßt, die am Port A (A0 - A7) anliegenden Daten auf den Datenbus (D0 - D7) auszugeben, wo sie dann von der CPU übernommen und intern in das Register A geladen werden.

### **3.5. Prioritätskaskadierung der Peripherieschaltkreise**

Wie bereits erwähnt, verfügen die verwendeten Schaltkreise U 855 und U 857 über ein komfortables Interruptsystem zur Unterbrechung der laufenden CPU-Programmabarbeitung bei bestimmten , programmierbaren Ereignissen (z. B. Nulldurchgang eines Zählerkanales oder bestimmter logischer Zustand an einem PIO-Port). Über eine sogenannte Prioritätskaskadierung wird eine bestimmte Vorrangordnung bei der Abarbeitung von Interruptanforderungen durch Peripherieschaltkreise festgelegt. Dies geschieht mittels der IEI-(Interrupt-Freigabeeingang, 1 = INT möglich) und IEO-Anschlüsse (Interrupt-Freigabeausgang, 1 = es wird z. Z. kein Interrupt dieses oder eines höher priorisierten Schaltkreises

abgearbeitet).

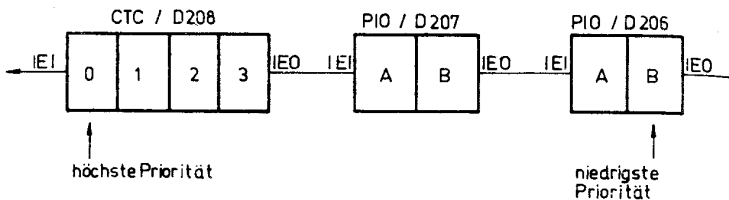
Dabei wird IEO des jeweils "vorrangigen" IS mit IEI des nachfolgend "niederrangigeren" IS verbunden. Innerhalb der IS ist folgende Priorität (vorrangig -> niederrangig) festgelegt:

CTC Kanal 0 -> 3

PIO Port A -> B

Im LC 80 ist folgende Prioritätskette realisiert:

Bild 3.2.



Wird nun innerhalb der Programmabarbeitung eine Interruptanforderung eines bestimmten Kanals durch die CPU bestätigt (vgl. "Zeitverhalten der CPU bei Interruptanforderung" sowie "Interruptsystem"), so sperrt dieser Kanal eine mögliche Interruptanmeldung niederer Kanäle zum einen intern sowie extern durch  $IEO = 0$ .

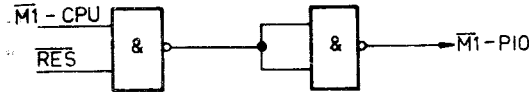
### **3.6. RESET-Erzeugung für die PIO's**

Im Gegensatz zur CPU und CTC besitzt der PIO keinen separaten Rücksetzeingang. Jedoch wird durch eine interne Logik bei folgender Signalkombination an der PIO ein internes RESET-Signal erzeugt (ebenso beim Zuschalten der Spannungsversorgung):

$/(PIO - RES) = /M1 * RD * IORQ$ , d. h., wenn  $/M1 = 0$ ,  $/RD = 1$  und  $/IORQ = 1$  gleichzeitig anliegen (diese Kombination tritt bei normaler CPU-Programmabarbeitung nicht auf).

Dies wird mit zwei NAND-Gattern/D 212 erreicht:

Bild 3.3.



Es ist zu beachten, daß jedoch der RESET-Zustand der PIO erst mit dem Übergang L - H vom M1 - PIO eingenommen wird.

Das bedeutet in diesem Fall, daß erst mit dem Loslassen der RES-Taste die PIO's rückgesetzt werden.

### **3.7. Magnetbandinterface**

Wie bereits bei der Bedienungsanleitung zum LC 80 dargelegt wurde, werden die einzelnen Digitalinformationen jedes Bytes bit-seriell, also nacheinander in Form einer Kombination zweier Frequenzen (1 kHz und 2 kHz) auf Magnetband aufgezeichnet. Außer dem eigentlichen zu übertragenden Speicherinhalt fügt das, den gesamten Ablauf steuernde Monitorprogramm bestimmte Zusatzinformationen bei, um ein späteres zuverlässiges Rückspeichern in den Computer zu sichern.

Diese Gesamtheit der zu übertragenden Signale, als Datensatz bezeichnet, enthält damit noch die einzelnen Kennfrequenzen zur internen Synchronisation beim Rückladen, einen vom Anwender festgelegten Programmnamen, die Start- und Endadresse des Speicherbereiches aus dem die Daten entstanden sowie eine Prüfsumme als Summe aller auf Band abgespeicherten Daten dieses Bereiches.

Beim Rückladen in den Computer werden alle empfangenen Daten addiert und diese Summe mit der jeweiligen Prüfsumme verglichen. Tritt dabei eine Differenz auf und liegt somit ein Übertragungsfehler vor, so wird dies in Form einer Error-Anzeige dargestellt. Die Ausgabe zu übertragender Signale bzw. die Eingabe empfangener

Signale durch den Rechner wird über zwei PIO-Leitungen B1 und B0 vom PIO D 206 realisiert.

Hardwaremäßig werden diese Signale durch je eine Zwischenstufe den erforderlichen bzw. vorhandenen Pegelverhältnissen eines Magnetbandgerätes angepaßt:

Ausgabe: Das von B1 des vom PIO D 202 (B1 als Ausgang programmiert, Mode 3) kommende Ausgangssignal wird mittels Tiefpaß und nachfolgendem Spannungsteiler auf die Größe des Eingangspegels eines Magnetbandgerätes verringert.

Eingabe: Das vom Magnetbandgerät kommende Signal wird mit dem als nichtinvertierenden Verstärker beschalteten Operationsverstärker N 239 auf TTL-Pegel erhöht und B0 von PIO D 206 (als Eingang programmiert, Mode 3) zugeleitet.

Parallel zum PIO-Ausgang B1/D 206 liegt noch der als Schalter wirkende Transistor V 220, der sowohl die LED "OUT" B 236 als auch die Hörkapsel B 237 steuert.

Damit signalisiert die LED "OUT" den jeweiligen Ausgangspegel von B1/D 206:

LED leuchtet	B1 = 0
LED aus	B1 = 1

Periodische Pegeländerungen innerhalb eines bestimmten Zeitbereiches an B1 werden über die Hörkapsel als Töne wahrnehmbar.

### **3.8. Takterzeugung**

Zur Takterzeugung wurde ein Generator mit einer Taktfrequenz von ca. 900 kHz verwendet.

Die hiermit erzeugte Taktfrequenz besitzt nicht die Konstanz einer Quarzfrequenz.

Dies soll beispielsweise bei der Realisierung eines Zeitgeberprogrammes (Uhr o. ä.) beachtet werden.

### **3.9. Betriebsspannungserzeugung und Bemerkungen zur zusätzlichen Stromentnahme**

Die Schaltkreise des LC 80 benötigen eine Betriebsspannung von +5 V mit einer zulässigen Toleranz von  $\pm 0,25$  V.

Diese Spannung wird aus einem von außen anzuschließenden Netzteil ( $\sim 9 - 12$  V) mit Hilfe des Stabilisierungsschaltkreises MA 7805/Pos. N 238 gewonnen. Die Eingangsspannung wird durch die Dioden V 213 - 216, die als sogenannte Graetz-Brücke geschaltet sind, gleichgerichtet. Aus diesem Grunde braucht auch bei Anwendung einer Eingangsgleichspannung nicht auf eine bestimmte Polung der Anschlußdrähte geachtet zu werden.

Die beiden HF-Drosseln L 319 - 320 sowie der Kondensator C 245 dienen der Funkentstörung des Computers.

Die maximale Stromaufnahme des LC 80 beträgt  $\leq 600$  mA bei DC und  $\leq 1000$  mA bei AC. Aus entwicklungstechnischen Gründen mußte dieser Wert auch für die Dimensionierung des Kühlkörpers zugrundegelegt werden (als Schutzgüteforderung an das Erzeugnis darf beispielsweise bei maximal zulässiger Umgebungstemperatur und Eingangsspannung eine bestimmte Kühlkörpertemperatur, die niedriger als die technisch mögliche ist, nicht überschritten werden).

Daraus resultiert zugleich, daß der Hersteller eine zusätzliche Stromentnahme durch den Anwender (z. B. für Zusatzschaltungen oder Zusatzbestückung zur Speichererweiterung) unter Sicherung der Kennwerte des LC 80 nicht gewähren kann.

Möchte der Anwender jedoch trotzdem (eigenverantwortlich!) einen zusätzlichen Strom entnehmen, so seien folgende Hinweise gegeben.

- der maximale Gesamteingangsgleichstrom darf in keinem Fall 900 mA überschreiten.
- entweder ist der Kühlkörper unter Vermeidung eines Wärmestaus gegen Berührung zu sichern (durch den zusätzlichen Strom kann es somit unter Grenzbedingungen zur Überschreitung der schutzgütemäßig festgelegten Maximaltemperatur des Kühlkörpers kommen), oder ist der Wert für die Eingangsspannung in der Nähe des unteren Bereiches (9 V ... 10 V) zu gewährleisten.
- zur Stromentnahme über Steckverbinder X1 bzw. X2 können die noch freien Steckverbinderkontakte (vgl. Stromlaufplan) verwendet werden, in dem eine Drahtbrücke zwischen den oberhalb der Steckverbinder befindlichen Lötungen "X1/A1" und "+5V" sowie "X1/A29" und "+5V" eingefügt wird.
- der verwendete Stabilisierungsschaltkreis besitzt einen internen Schutz vor Zerstörung bei Ausgangskurzschluß sowie thermischer Überlastung.

Achtung!

Bei allen Veränderungen an der LP sind die Garantiebestimmungen des LC 80 zu beachten!

## **5. Programmierung der Peripherieschaltkreise des LC 80**

Nachfolgend soll anhand einiger Beispiele die Programmierung von PIO und CTC erläutert werden. Diese Programmierung erfolgt auf der Grundlage der in den Abschnitten 2.3.4. und 2.4.3. aufgeführten schaltkreisspezifischen Programmiervorschriften.

Es wurde hierbei nicht angestrebt, alle möglichen Varianten zu erfassen, sondern eher den prinzipiellen Aufbau dieser Programmteile zu verdeutlichen.

Die Lösung eigenständiger Anwendungsaufgaben ist dann unter Nutzung o. g. Abschnitte leicht möglich.

Zuerst soll nochmals kurz auf die programmäßig notwendigen Aktivitäten zur Organisation einer Interruptserviceroutine (vgl. Abschnitt 2.2.3.) in Verbindung mit CTC und PIO eingegangen werden.

### **5.1. Programmäßige Organisation einer Interruptserviceroutine**

Gemäß des im Abschnitt 2.2.3. dargelegten Ablaufes für den hierbei betrachteten Interruptmode 2 (IM 2) sind folgende Aktivitäten erforderlich:

- Im Grundzustand nach RESET ist die Interruptannahme (INT) gesperrt und muß mittels EI-Befehl erst freigegeben werden. Gleiches gilt nach einer erfolgten Interruptannahme.
- Prinzipiell ist ein Interrupt-Aufruf in Mode 2 mit einem Unterprogrammaufruf mittels CALL-Befehl vergleichbar.

In der Regel wird es notwendig sein, die in der ISR verwendeten Register entweder durch Arbeit mit dem Alternativregistersatz (EXX, EXAF) oder durch PUSH-Befehle zu retten. Am Ende der ISR ist der Rücktausch (EXAF, EXX) bzw. die Rückkellerung der Register (POP) notwendig.

- Das im Rahmen einer ISR abzuarbeitende Unterprogramm kann an beliebiger Stelle im RAM abgespeichert werden und muß zur Kennzeichnung des programmäßigen Endes dieser ISR mit dem Befehl RETI (bei NMI : RETN) abgeschlossen werden.
- Die Startadresse dieses Programme kann ebenfalls an beliebiger Stelle im RAM abgespeichert werden. Die hierfür gewählte RAM-Adresse bestimmt den Inhalt des zu ladenden I-Registers (oberer Adreßteil) sowie des Interruptvektors (unterer Adreßteil) des interruptauslösenden Peripheriekanals.

Ein kurzes Prinzipbeispiel soll die erforderliche Programmstruktur verdeutlichen:

Drei Peripheriekanäle sollen interruptfähig sein. Die jeweiligen Interrupt-Unterprogramme seien im RAM nacheinanderfolgend abgespeichert.

(Die verwendeten Adressen wurden beliebig gewählt.)

```

      .
      .
      .
LD A,23H      oberer Adreßteil der Startadressentafel
LD I,A       der INT-Unterprogramme in I-Register laden
IM 2

      .
      .
      .
OUT (Peripheriekanal 1, Steueradresse),00H   Laden der
                                              jeweiligen
                                              Interruptvektoren
                                              im Rahmen der Ini-
OUT (Peripheriekanal 2, Steueradresse),02H   tialisierung der
                                              Peripherieschalt-
                                              kreise (Beachte
                                              spezielle
OUT (Peripheriekanal 3, Steueradresse),04H   Interruptvek-
                                              torerzeugung der
                                              CTC-Kanäle 1 - 3)

```



```

.
.
.
Schleife: EI
        HALT
        JR Schleife
.
.
.
2060H: . . .
        (INT-Unterprogramm 1)
.
.
.
        RETI
2073H: . . . .
        (INT-Unterprogramm 2)
.
.
.
        RETI
20A9H: . . .
        (INT-Unterprogramm 3)
.
.
.
        RETI

```

```

2300H:DEFW 2060H
2302H:DEFW 2073H
2304H:DEFW 20A9H

```

Startadressentafel der  
INT- Unterprogramme

Das eigentliche Hauptprogramm dieses Prinzipbeispiels besteht aus einer einfachen Schleife mit HALT-Befehl, d. h. nach jeder ISR kehrt die CPU in den HALT-Zustand zurück und erwartet eine erneute INT-Anmeldung. Zuvor muß dabei jeweils mittels EI-Befehl die INT-Freigabe erteilt werden.

Soll beispielsweise Kanal 2 eine laufende ISR des Kanals 3 unterbrechen dürfen (vorausgesetzt Kanal 2 besitzt hardwaremäßig höhere Priorität, vgl. Abschnitt 2.4.2.4.), so muß innerhalb des INT-Unterprogrammes 3 ebenfalls eine INT-Freigabe mittels EI-Befehl erfolgen.

## **5.2. PIO-Programmierung (vgl. hierzu Abschnitt 2.3.4. und 3.4.)**

Beispiel 1: PIO D207/Port A soll in Byte-Ausgabe arbeiten und nicht interruptfähig sein. Anschließend soll der Registerinhalt von A über Port A ausgegeben werden:

LD A,0FH	Betriebsarten-Steuerwort: <u>0 0 0 0</u> <u>1 1 1 1</u>
OUT (FAH),A	Ausgabe Betriebsarten-Steuerwort auf D207/Port A, Steueradresse FAH
LD A,03H	INT-Steuerwort: <u>0 0 0 0</u> <u>0 0 1 1</u>
OUT (FAH),A	INT gesperrt      Identifikation als INT-Steuerwort

.  
.  
.  
LD A, ...

OUT (F8H),A      Ausgabe des A-Registerinhaltes auf Port A,  
                            Datenadresse FBH

Beispiel 2:      PIO D207/Port A soll in Byte-Eingabe arbeiten und  
                            interruptfähig sein (INT-Vektor = 50H gewählt, INT  
                            Auslösung über /ASTB):

.  
.  
.  
LD A,50H      INT-Vektor      0101    0 0 0 0  
  |  
  Identifikation als INT-Vektor

OUT (FAH),A

LD A,4FH      Betriebsarten-Steuerwort 0 1 0 0 1 1 1 1  
  |  
  Identifikation als  
  Betriebsarten-Steuerwort  
  Byte-Eingabe

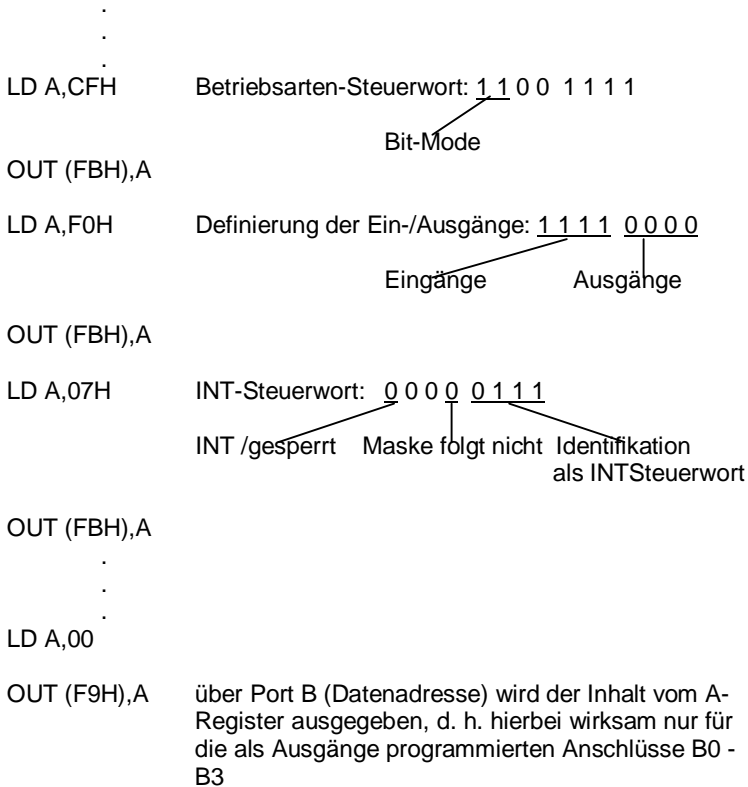
OUT (FAH),A  
LD A,83H      INT-Steuerwort: 1 0 0 0 0 0 1 1  
  |  
  INT freigegeben  
  |  
  Identifikation als  
  INT-Steuerwort

OUT (FAH),A

.  
.  
.

Beispiel 3:

PIO D207/Port B soll in Bit-Mode arbeiten, dabei sollen B0 - B3 Ausgänge und B4 - B7 Eingänge sein, Interrupt gesperrt. Anschließend soll ein Datenschreibe- und -lesezyklus erfolgen:



IN A,(F9H) Einlesen von Port B (Datenadresse) in das A-Register der CPU. Dabei werden die an B4 - B7 (Eingänge) anliegenden Daten sowie die im Ausgaberegister von B0 - B3 (Ausgänge) enthaltenen Werte (hierbei B0 - B3 = 0) gelesen.

Beispiel 4: PIO D207/Port B soll in Bit-Mode arbeiten, B0 - B7 sollen Eingänge sein, wobei ein Interrupt unter folgender Bedingung ausgelöst werden soll:

$/B0 * /B1 * /B3$  (Low-Pegel an B0, B1 und B3 -> UND-Verknüpfung) (INT-Vektor = 70H gewählt)

```

      .
      .
      .
LD A,70H   INT-Vektor:  0 1 1 1  0 0 0 0
                                |
                                Identifikation als INT-Vektor

OUT (FBH),A

LD A,CFH   Betriebsarten-Steuerwort

OUT (FBH),A

LD A,FFH   Definition der Ein-/Ausgänge

OUT (FBH),A

LD A,DFH   INT-Steuerwort:  1 1 0 1  0 1 1 1
                        |         |
                        INT-Freigabe UND-Verknüpfung Low-aktiv Maske folgt

OUT (FBH),A

```

LD A,F4H

Maskierung:

1 1 1 1 0 1 0 0

B3, B1, B0 für INT ausgewertet

OUT (FBH),A

·  
·  
·

### **5.3. CTC-Programmierung (vgl. hierzu Abschnitt 2.4.3.)**

#### Beispiel 1:

CTC D208/Kanal 1 soll als Zeitgeber arbeiten, wobei der Systemtakt (Bezugsbasis 900 kHz) durch den Vorteilerfaktor 256 geteilt wird.

Die Zeitkonstante soll 09H betragen. Die Zeitgeberoperation soll mit der steigenden Systemtaktflanke des CPU-Maschinenzyklusses, der auf das Laden der Zeitkonstante folgt (Software-Triggerung), beginnen.

INT freigegeben, INT-Vektor Kanal 0 = 50H gewählt.

·  
·  
·

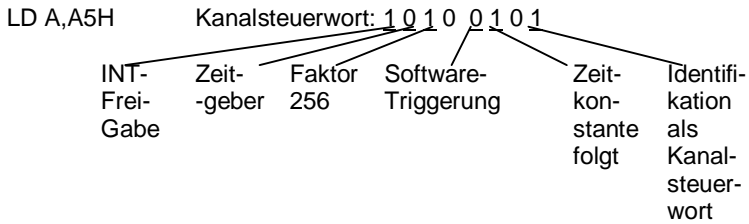
LD A,50H

INT-Vektor: 0 1 0 1 0 0 0 0

Identifikation als INT-Vektor

OUT (ECH),A

INT-Vektor in Kanal 0 (!) einschreiben (gemäß Abschnitt 2.4.3. wird damit bei INT-Annahme von Kanal 1 der INT-Vektor = 52H gesendet!)



OUT (EDH),A      Kanalsteuerwort in Kanal 1 einschreiben

LD A,09H      Zeitkonstante 09H

OUT (EDH),A  
Beginn der Zeitgeberoperation

.  
. .

Die Dauer bis zur INT-Auslösung (Nulldurchgang des Rückwärtszählers) beträgt damit gemäß Abschnitt 2.4.3.2.

$$t = 1/900 \text{ ms} * 256 * 9 = 2,56 \text{ ms}$$

Beispiel 2: Kanal 3 soll als Zähler arbeiten, wobei die am CTC-Eingang C/TRG3 ankommenden Impulse mit der positiven Flanke (L -> H) gezählt werden. Beim Nulldurchgang des Rückwärtszählers soll ein Interrupt erzeugt werden (INT-Vektor Kanal 0 = 50H gewählt).

Als Zeitkonstante soll der maximale Wert = 0 verwendet werden (entspricht einer Zeitkonstante von 256, d. h. nach 256 Eingangsimpulsen erfolgt INT-Anmeldung).

·  
·  
·  
LD A, 50H

OUT (ECH),A INT-Vektor in Kanal 0 einschreiben (damit wird bei  
INT-Annahme von Kanal 3 der INT-Vektor = 56H  
gesendet)

LD A,D5H Kanalsteuerwort: 1 1 0 1 0 1 0 1

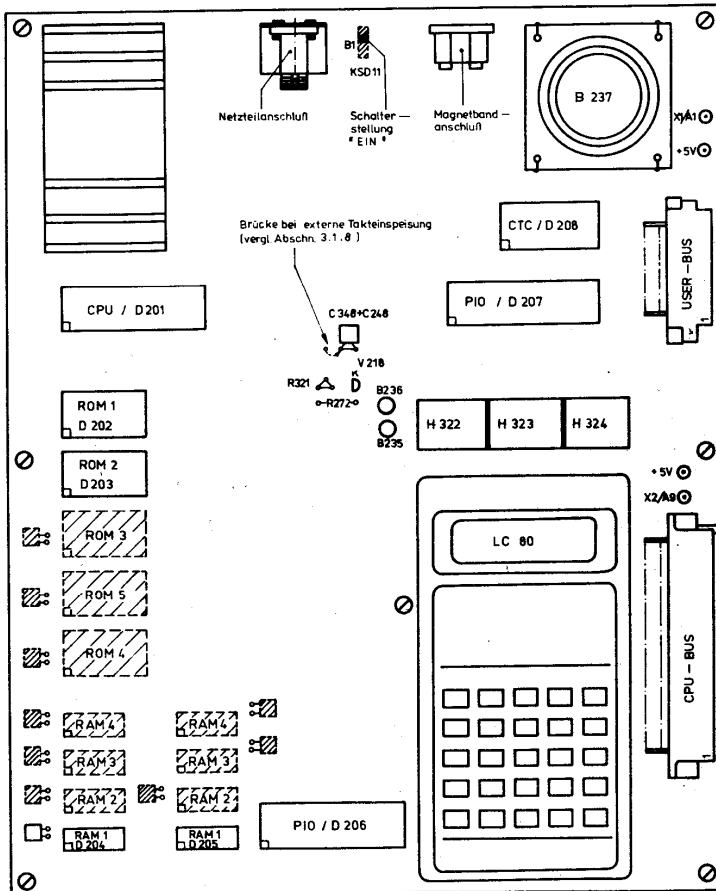
INT- Frei- Gabe	Zähler	positive Trigger- flanke	Zeit- kon- stante	Identifikation als Kanal- steuerwort
-----------------------	--------	--------------------------------	-------------------------	--

OUT (EFH),A Kanalsteuerwort in Kanal 3 einschreiben

LD A,00

OUT (EFH),A Zeitkonstante 0: nach 256 Eingangsimpulsen erfolgt  
INT-Anmeldung durch CTC





Anlage: Bestückungsplan LC 80