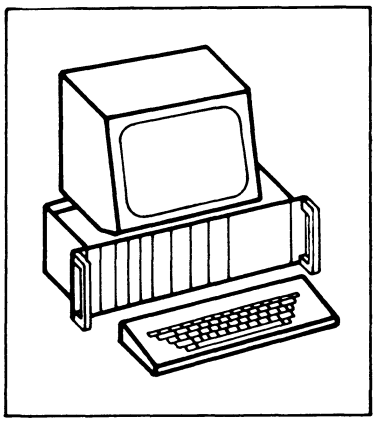
FACHPRAKTISCHE ÜBUNG MIKROCOMPUTER-TECHNIK



MAT 85

BFZ/MFA 7.1.

Diese Übung ist Bestandteil eines Mediensystems, das im Rahmen eines vom Bundesminister für Bildung und Wissenschaft, vom Bundesminister für Forschung und Technologie sowie der Bundesanstalt für Arbeit geförderten Modellversuches zum Einsatz der "Mikrocomputer-Technik in der Facharbeiterausbildung" vom BFZ-Essen e.V. entwickelt wurde.

Inhaltsverzeichnis Seite

System-Informationen (Überblick)

Gebrauch der Kommandos (Ausführl. Kommando-Beschreibung)

Anhang (Techn. Daten, Arbeitsblätter, Progr. Beispiele)

1. Einleitung, Speicherbelegung

Das Betriebsprogramm MAT 85*) gestattet mit Hilfe von 14 Kommandos das Ein- und Ausgeben, das Testen und das Verfolgen der Wirkungsweise von Anwenderprogrammen.
Das Betriebsprogramm ist in vier 2-KByte-EPROM's vom Typ 2716 gespeichert und belegt den Adreßraum ab Adresse 0000 bis 1FFFH. An Schreib-Lesespeicher benötigt es 1 KByte, so daß dem Anwender bei einer Bestückung der RAM-Karte mit einem 2-KByte-RAM-Baustein ein Speicherbereich von 1 KByte zur Verfügung steht. Der Schreib-Lesespeicher muß am Ende des adressierbaren Speicherbereiches liegen. Die erforderliche Speicherbelegung ist in Bild 1 dargestellt.

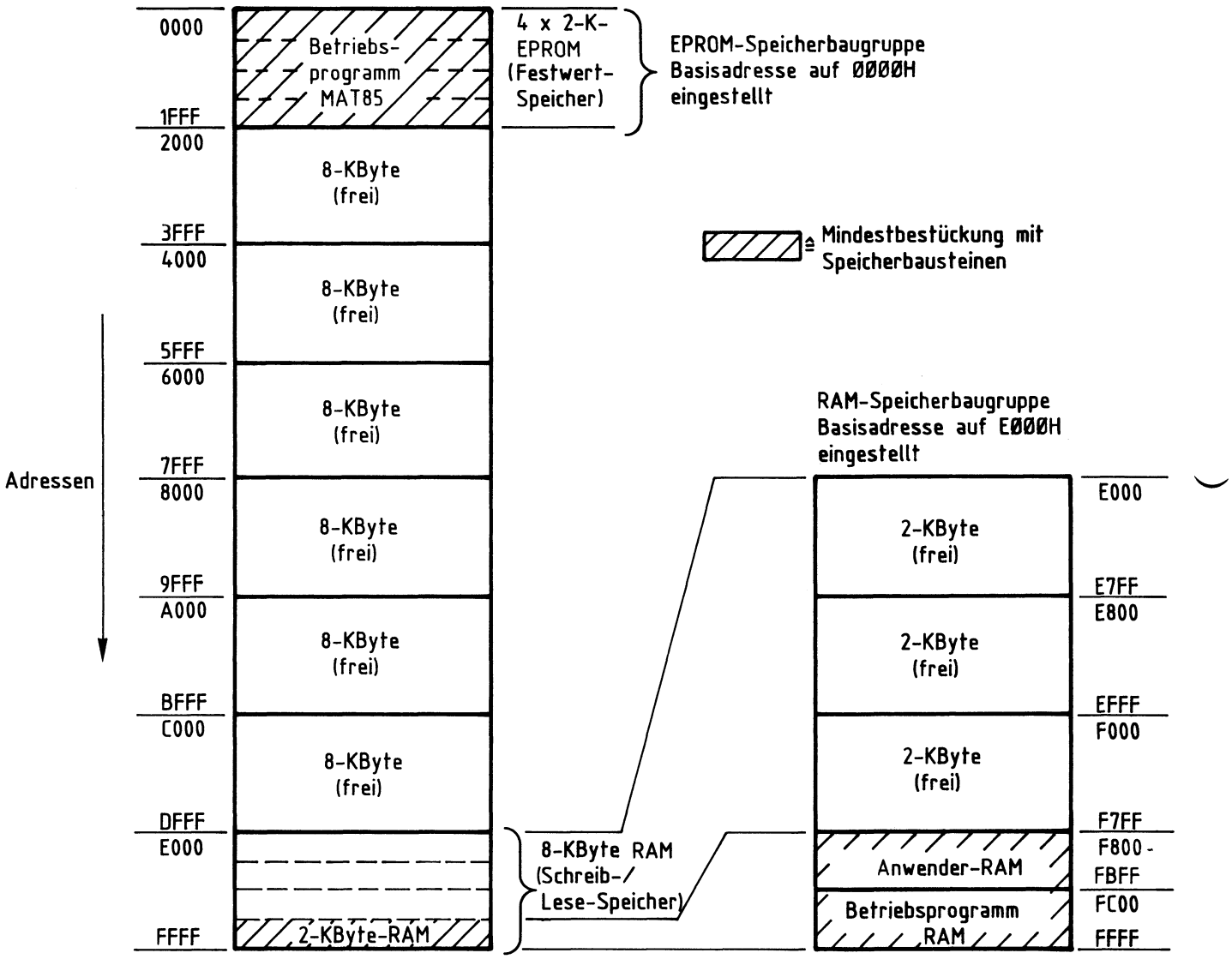


Bild 1: Speicher-Belegung

*) MAT 85 = Abkürzung für Monitor-Assembler-Tracer für Prozessor-Baugruppe 8085.

2. Aufbau des Systems

Für den Aufbau des Systems benötigen Sie die folgenden Baugruppen:

- Baugruppenträger mit Busverdrahtung BFZ/MFA 0.1.
- Busabschluß BFZ/MFA 0.2.
- Trafo-Einschub BFZ/MFA 1.1.
- Spannungsregelung BFZ/MFA 1.2.
- Prozessor 8085 BFZ/MFA 2.1.
- 8-K-RAM/EPROM BFZ/MFA 3.1. bestückt mit MAT 85
- 8-K-RAM/EPROM BFZ/MFA 3.1. bestückt mit mind. 2-K-RAM
- Video-Interface BFZ/MFA 8.2.
- ASCII-Tastatur BFZ/MFA 8.1. } Datensichtstation
- Monitor mit Cinch-Anschluß

In Bild 2 ist der Aufbau des Mikrocomputers aus diesen Baugruppen dargestellt.

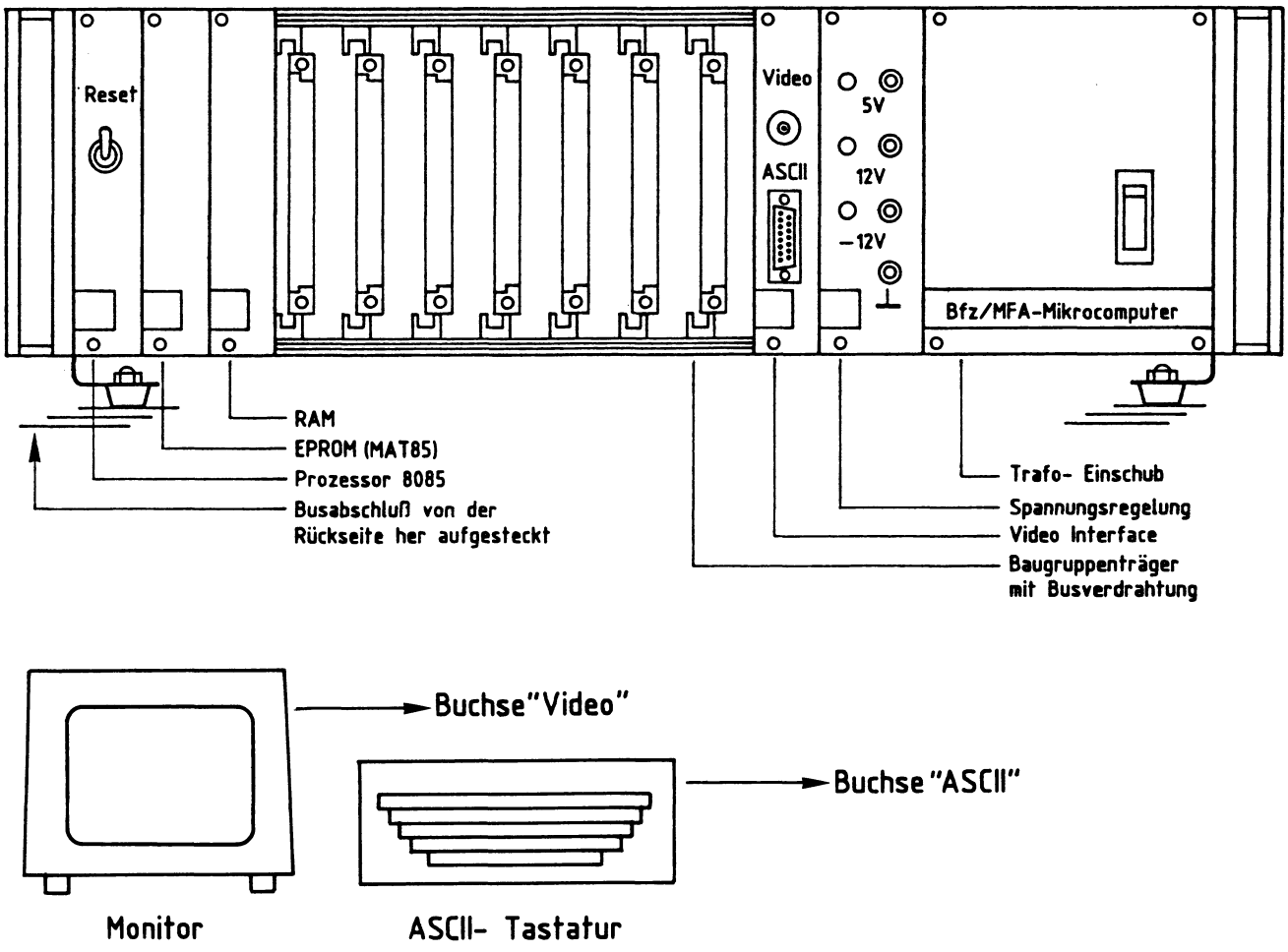


Bild 2: Aufbau des Mikrocomputers

Soll anstelle der dargestellten Datensichtstation eine andere oder ein Fernschreiber (Teletype, TTY) verwendet werden, so müssen zunächst die Anschlüsse für diese Geräte vorbereitet werden. Hinweise hierzu finden Sie im Anhang.

3. Arbeitsweise des Betriebsprogramms

3.1. Kaltstart

Mit dem Einschalten der Betriebsspannung (Kaltstart) wird das Betriebsprogramm gestartet und die Übertragungsgeschwindigkeit (Baud-Rate) des angeschlossenen Dialoggerätes (Datensichtstation bzw. TTY) erfaßt, um die eigene Übertragungsgeschwindigkeit an die des Dialoggerätes anzupassen. Dazu ist es erforderlich, daß ein bestimmtes Zeichen vom Dialoggerät an den Mikrocomputer gesendet wird.

Daher muß die SPACE-Taste kurz betätigt werden, worauf sich das Betriebsprogramm mit der Versionsnummer und dem Ausdruck aller zur Verfügung stehenden Bediener-Kommandos meldet (Bild 3). Die Überschrift mit der Versionsnummer wird auf dem Bildschirm nur kurzzeitig angezeigt.

```
ASSEMBLER
BREAKPOINT
DISASSEMBLER
GO
HELP
IN
LOAD TAPE
MEMORY
NEXT INSTRUCTION
OUT
PRINT
REGISTER
SAVE
TRACE INTERVAL

KMD >_
```

Bild 3: Ausdruck der verfügbaren Monitor-Kommandos nach einem Kaltstart

3.2. Kommando-Eingabe

Die Bereitschaft zur Annahme eines Kommandos vom Bediener wird durch den Ausdruck "KMD>_" angezeigt (Kommando-Modus). Jedes der aufgelisteten Kommandos kann durch Eingabe seines ersten Buchstabens und durch anschließendes Betätigen der Taste "RETURN" bzw. "CR" (Wagenrücklauf) oder "SPACE" (Leertaste) aufgerufen werden. Daraufhin druckt das Betriebsprogramm den vollständigen Kommandonamen aus und fordert eventuell zusätzlich erforderliche Informationen an. Soll das Kommando abgebrochen werden, so muß die Taste "ESC" (Escape = flüchten) betätigt werden. Das Betriebsprogramm quittiert diese Eingabe durch ein akustisches Signal und fordert durch den Ausdruck "KMD>_" ein neues Kommando an.

3.3. Reset-Betätigung, Warmstart

Im Gegensatz zum Kaltstart erfolgt nach Betätigung der RESET-Taste (Warmstart oder warmer RESET) keine Erfassung der Übertragungsgeschwindigkeit und auch kein Auflisten der Bediener-Kommandos, sondern die Ausgabe

```
*** RESET ***
```

und die Aufforderung zur Kommando-Eingabe "KMD>_".

3.4. Bildschirm-Modus

Das Betriebsprogramm unterscheidet je nach gemessener Übertragungsgeschwindigkeit zwischen einem Bildschirm- und einem Drucker-Modus (siehe Anhang Kapitel 2.)

Im Bildschirm-Modus können falsch eingegebene Zeichen (Kommandos, usw.) durch Betätigung der Taste "DEL" (Delete = streichen) oder "RUBOUT" (ausradieren) gelöscht werden.
Bei längeren Protokollen (z.B. beim PRINT-Kommando) wird nach jeder Bildschirmseite (16 Zeilen, zu je maximal 64 Zeichen) der Ausdruck gestoppt und der Text "==> SPACE" ausgegeben. Der Bediener erhält damit die Möglichkeit, die Protokollierung auch bei hohen Übertragungsgeschwindigkeiten zu verfolgen. Der Ausdruck wird fortgesetzt, wenn die SPACE-Taste kurz betätigt wird.

3.5. Bedienerführung

Unabhängig vom Bildschirm- bzw. Drucker-Modus wird der System-Bediener vom Betriebsprogramm geführt, indem es eventuell zusätzliche Informationen für die Kommando-Ausführung (z.B. Adressen usw.) anfordert. Dabei erfolgt sofort eine Kontrolle, ob die Eingabedaten dem notwendigen Format entsprechen (SYNTAX-Prüfung). Ist dies nicht der Fall, wird der Bediener durch ein akustisches Signal auf seinen Fehler aufmerksam gemacht. Solch ein Signal ertönt z.B. dann, wenn das Betriebsprogramm eine Adresse angefordert hat und das eingegebene Zeichen kein Hex-Zeichen ist.
Im Bildschirm-Modus wird das falsch eingegebene Zeichen angezeigt, indem der CURSOR (Schreibstellen-Zeiger, Schreibmarke auf dem Bildschirm) auf dieses Zeichen zeigt. Im Drucker-Modus werden falsche Zeichen vom Betriebsprogramm ignoriert.

4. Struktur des Betriebsprogramms

Das Betriebsprogramm MAT 85 ist in drei Programmblöcke unterteilt. Jedem dieser Blöcke ist eine bestimmte Aufgabe und ein Teil der Kommandos zugeordnet. Bild 4 zeigt diese Struktur.

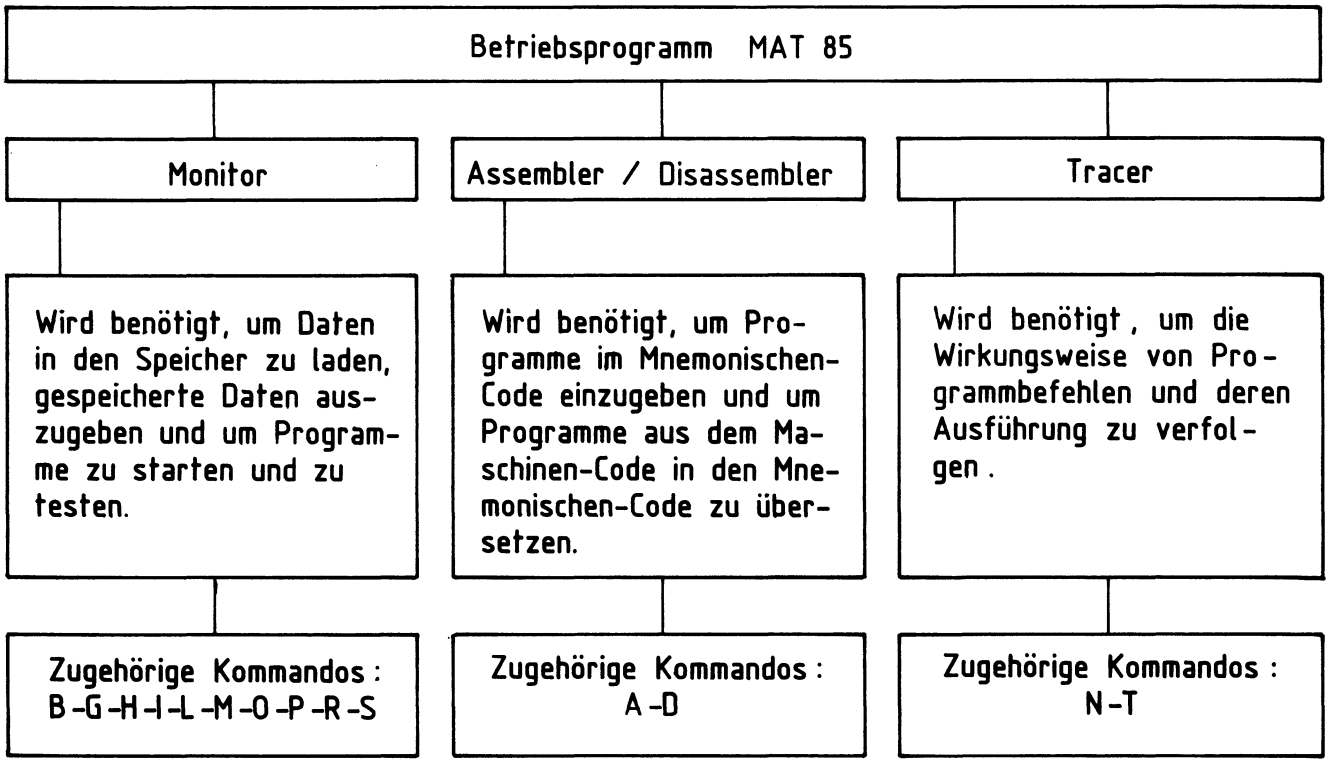


Bild 4: Struktur des Betriebsprogramms MAT 85

4.1. Kommando-Kurzbeschreibung

4.1.1. Monitor-Kommandos

BREAKPOINT_____: Dieses Kommando ermöglicht es, mit dem GO-Kommando Unterbrechungspunkte einzugeben. Unterbrechungspunkte (engl. Breakpoints) sind Adressen aus dem Speicherbereich des Anwenderprogramms, an denen die Programmabarbeitung unterbrochen werden soll.
Nach der Unterbrechung werden die Inhalte der CPU-Register angezeigt.

GO_____: Mit diesem Kommando können eingegebene Programme gestartet werden.

HELP_____: Dient dazu, alle verfügbaren Kommandos des Betriebsprogramms anzuzeigen.

IN_____: Dieses Kommando dient dazu, Daten von Eingabe-Ports zu lesen und anzuzeigen.

LOAD TAPE_____: Lädt Daten von einer Magnetband-Kassette in den Speicher des Mikrocomputers. Hierzu wird das Kassetten-Interface BFZ/MFA 4.4.a benötigt.

MEMORY_____: Mit diesem Kommando lassen sich die Inhalte von Speicherzeilen in verschiedenen Formaten ausdrucken und ändern.

OUT_____: Dient dazu, Daten an Ausgabe-Ports zu senden.

PRINT_____: Mit diesem Kommando können die Inhalte von Speicherzeilen in verschiedenen Formaten (Binär, Hexadezimal, Dezimal, ASCII) formatiert (pro Zeile max. 8 Inhalte) ausgedruckt werden.

REGISTER_____: Mit diesem Kommando können die Anfangswerte der CPU-Register, z.B. vor einem Testlauf des Anwenderprogramms, vorgegeben werden.

SAVE_____: Dient dazu, Daten auf einem Kassetten-Recorder zu speichern. Hierzu wird das Kassetten-Interface BFZ/MFA 4.4.a benötigt.

4.1.2. Assembler/Disassembler-Kommandos

ASSEMBLER_____: Mit diesem Kommando wird ein Programm aufgerufen, das es ermöglicht, Anwendungsprogramme im Mnemo-Code (8085-Intel-Format) einzugeben. Der eingegebene Code wird Zeile für Zeile in den zugehörigen Maschinen-Code übersetzt und im RAM-Speicher abgelegt.

DISASSEMBLER____: Mit diesem Kommando können Programme, die im Maschinen-Code gespeichert sind, in den Assembler-Code übersetzt werden.

4.1.3. Tracer-Kommandos

NEXT INSTRUCTION_: Mit diesem Kommando wird ein Tracer (Verfolger) aktiviert, der es ermöglicht, die Ausführung und Wirkungsweise einer vorgegebenen Anzahl von Programmbefehlen zu verfolgen. Dazu wird nach jedem Befehl (engl. Instruction) die Programmbearbeitung kurz unterbrochen und die Inhalte aller CPU-Register werden protokolliert.

TRACE INTERVAL___: Dieses Kommando bewirkt eine Protokollierung der Registerinhalte immer dann, wenn diejenigen Programmbefehle abgearbeitet werden, die in einem vorher zu bestimmenden Speicherbereich liegen.

4.1.4. Ordnung der Kommandos nach Einsatzgebieten

Die in Bild 5 dargestellte Grafik zeigt die beschriebenen Kommandos nach Einsatzgebieten geordnet.

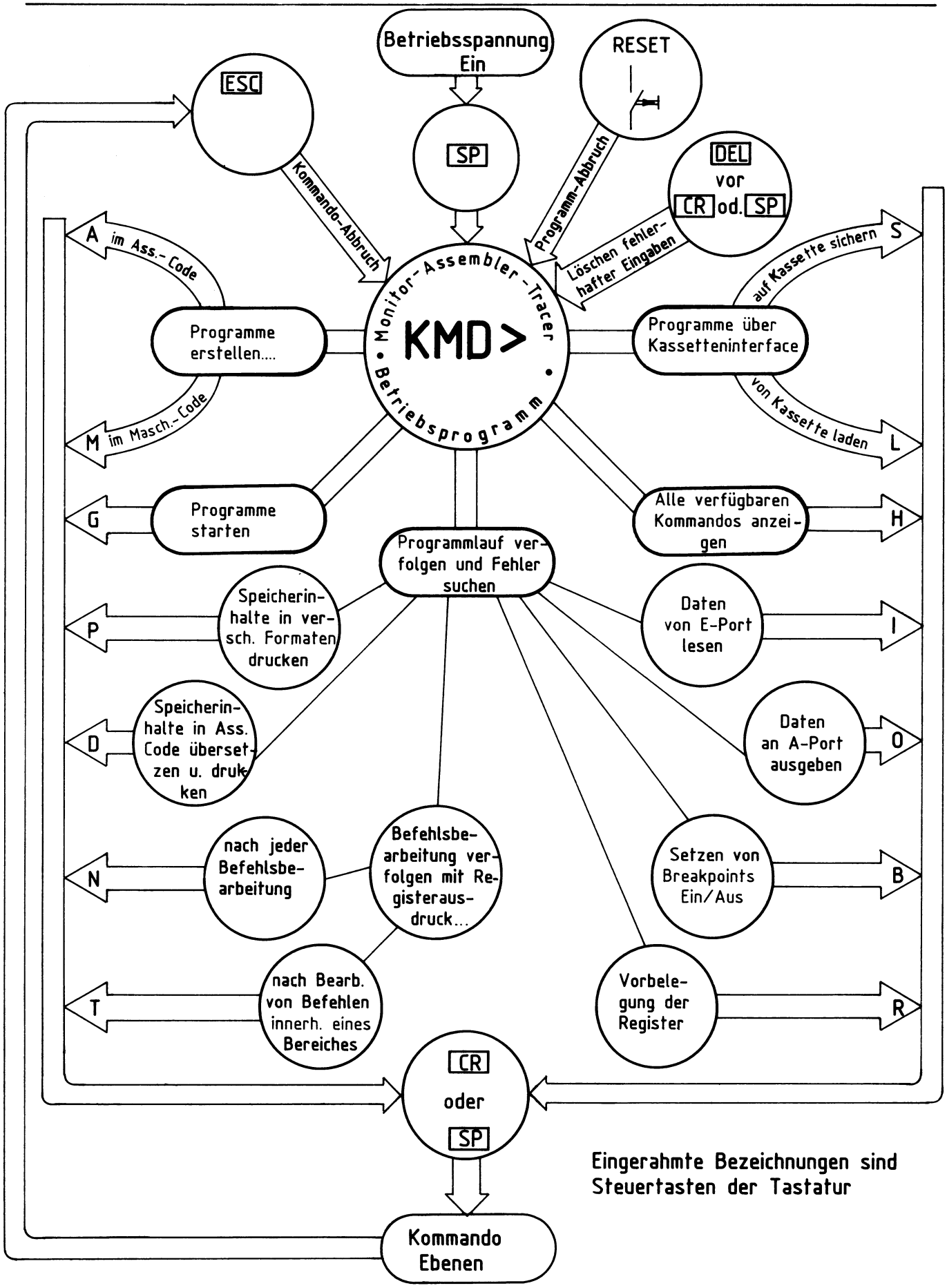


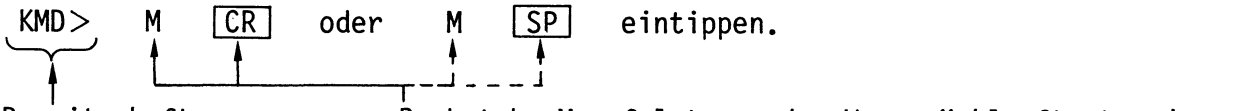
Bild 5: Zuordnung der Kommandos zu Einsatzgebieten

Inhaltsverzeichnis

— Hinweise zur Beschreibung der Kommandos

Unter Kapitel 3.2 der "System-Informationen" wurde kurz beschrieben, wie der Mikrocomputer seine Bereitschaft zur Annahme eines Kommandos anzeigt, wie ein Kommando aufgerufen wird und wie man ein Kommando abbricht.
Im folgenden werden Aufruf und Verwendung der einzelnen Kommandos ausführlich beschrieben. Anhand von Bildschirmausdrucken und Kommentaren kann die Anwendung eines jeden Kommandos nachvollzogen werden. Übungsaufgaben dienen dazu, das Erlernte zu vertiefen.
Um Tastatureingaben, Bildschirmausdrucke und die Kommentare dazu übersichtlich und allgemeingültig zu gestalten, werden einige Abkürzungen und Darstellungsweisen verwendet, die am Beispiel des MEMORY-Kommandos zunächst erklärt werden sollen:

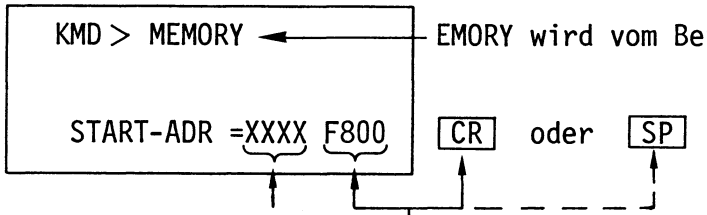
● Aufruf des MEMORY-Kommandos (wenn KMD >_ angezeigt wird):



Bereitschafts-meldung

Buchstabe M gefolgt von der Wagenrücklauftaste oder der Leertaste eintippen.
Eingerahmte Zeichen sind Tasten mit einer Steuerfunktion!

● Wirkung:



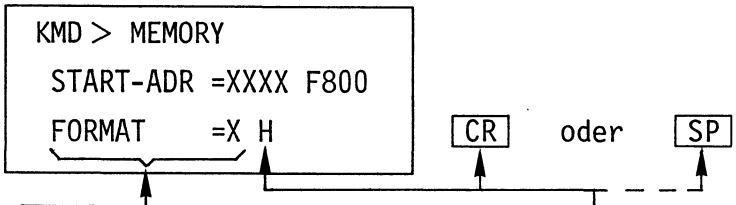
EMORY wird vom Betriebssystem ergänzt.

CR oder SP

Hier steht die Vorgabe-Adresse des Computers (Vorschlag). Wird ihr Wert akzeptiert, müssen Sie die CR - oder SP-Taste betätigen, wenn nicht,

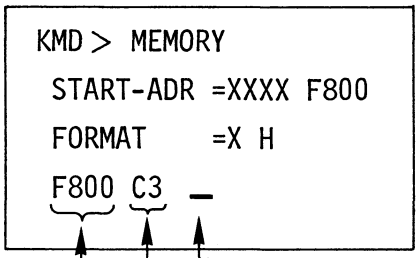
müssen Sie die hexadezimale Adresse derjenigen Speicherzeile eintippen, die als erste bearbeitet werden soll. Hier wird diese Adresse zu F800 gewählt,indem hintereinander die Zeichen F800 eingetippt werden, gefolgt von der Betätigung der CR- oder SP- Taste. Allgemeingültig wird diese Adresse "Neu-Adresse" genannt und durch "YYYY" gekennzeichnet.

● Wirkung:



Das Betriebsprogramm schlägt vor, die Daten in dem bei X angegebenen Format darzustellen bzw. zu bearbeiten. Wird die Vorgabe akzeptiert, müssen Sie CR oder SP betätigen. Wenn nicht, müssen Sie einen der Buchstaben A, B, D oder H gefolgt von CR oder SP eintippen. Hier wird H für Hexadezimale Darstellung der Daten eingegeben.
Allgemeingültig wird das Format "Neu-Format" genannt und durch "Y" gekennzeichnet. Die Bedeutung der verschiedenen Formate wird bei der Beschreibung des MEMORY-Kommandos gezeigt.

● Wirkung:



Die gewünschte Adresse wird angezeigt.

Der Cursor (Schreibmarke) erwartet weitere Eingaben, die unter der Beschreibung des MEMORY-Kommandos gezeigt werden.

Dies ist der Inhalt der Speicherzeile mit der angegebenen Adresse im gewünschten Format. Der hier ausgegebene Wert C3 ist vom Zufall abhängig, bei Ihrem Computer kann ein anderer Wert angezeigt werden.

In Bild 6 sind die oben beschriebenen Arbeitsschritte in gekürzter Form dargestellt. Diese Art der Darstellung wird bei der Beschreibung der Kommandos verwendet.

Schirmbild (oft Ausschnitt)	Tastatureingaben und Kommentare zum Schirmbild
KMD > MEMORY START-ADR =XXXX YYYY FORMAT =X Y	M [CR] oder M [SP] eintippen "EMORY" wird ergänzt XXXX = Vorgabe; Neu: YYYY [CR] oder [SP] Vorgabe: [CR] oder [SP] X = Vorgabe; Neu: Y[CR] oder [SP] Vorgabe: [CR] oder [SP] Y = A: ASCII (Druckbare Zeichen) B: Binär (0,1) D: Dezimal (0...9) H: Hexadezimal (0...F)

Bild 6: Kurzform der Darstellung des Schirmbildes, von Tastatureingaben und Kommentaren zum Schirmbild

- Alle weiteren vom Betriebsprogramm vorgegebenen, oder vom Benutzer zu verändernden Werte sind sinngemäß zu handhaben.

- Fehlerhafte Eingaben können vor Kommando-Abschluß durch die [CR]- oder [SP]-Taste mit der [DEL]-Taste (Delete = löschen) gelöscht und dann entsprechend korrigiert werden.

- Die Rückkehr aus den Kommandoebenen in das Betriebsprogramm erfolgt durch Betätigen der [ESC]-Taste (Escape = flüchten). Siehe hierzu auch Bild 5.

H-Kommando

Mit dem Help-Kommando lassen sich die Namen aller zulässigen Kommandos des Betriebssystems MAT 85 in alphabetischer Reihenfolge ausdrucken.

Aufruf und Handhabung:

KMD> HELP	H[CR] oder H[SP] eintippen "ELP" wird ergänzt
(Kommando-Ausführung)	
KMD>_	Nächstes Kommando

Zur Kommando-Ausführung:

- Nach dem Ausdruck aller Kommandonamen (die obere Zeile "KMD> HELP" wird überschrieben) erfolgt ein Rücksprung in die Kommando-Routine (KMD>_).

- Zum Aufruf eines der Kommandos muß nur der 1. Buchstabe, gefolgt von der Taste [CR] (Carriage returm = Wagen-Rücklauf) oder der Taste [SP] (Space = Leerzeichen) eingegeben werden.
  Andernfalls erfolgt eine Fehlermeldung ohne Annahme der Eingabe.

- Eingaben, die vor Betätigung der [CR]- oder [SP]-Taste erfolgen, können mit der Taste [DEL] (Delete = streichen) gelöscht werden.

M1

M-Kommando

Mit dem Memory-Kommando lassen sich die Speicherinhalte in verschiedenen Formaten byte-weise anzeigen und ändern.

Aufruf und Handhabung:

```
KMD > MEMORY

 START-ADR = XXXX YYYY

 FORMAT    = X Y
```

M [CR] oder M [SP] eintippen
"EMORY" wird ergänzt

XXXX = Vorgabe; Neu: YYYY [CR] oder [SP]
Vorgabe: [CR] oder [SP]

X = Vorgabe; Neu: Y [CR] oder [SP]
Vorgabe: [CR] oder [SP]

Y = A: ASCII (Druckbare Zeichen)
= B: Binär (0,1)
= D: Dezimal (0...9)
= H: Hexadezimal (0...F)

— Beispiel für Adresse = F800 und Format = H:

Schirmbild	Eingabe	Wirkung
F800 C3 ⟶ zufällige Inhalte	[SP]	Inh. unverändert, ADR + 1
F801 20 22	22 [SP]	Inh. verändert, ADR + 1
F802 44	[SP]	Inh. unverändert, ADR + 1
F803 55 -	[-]	Inh. unverändert, ADR - 1
F802 44 54-	54 [-]	Inh. verändert, ADR - 1
F801 22 } ⟶ eingegebene Werte	[+]	Inh. unverändert, ADR + 1
F802 54 }	[CR]	Inh. unverändert, Fertig
KMD >_	Nächstes Kommando kann eingegeben werden.	

— Beispiel für Adresse = F850 und Format = A:

Schirmbild	Eingabe	Wirkung
F850 A4		Wenn Speicherinhalt kein ASCII-Zeichen, wird Hex-Code angezeigt.
	[SP]	Inh. unverändert, ADR + 1
F851 B7 R	R [SP]	Inh. verändert, ADR + 1
F852 A6 A	A [SP]	Inh. verändert, ADR + 1
F853 BA M-	M [-]	Inh. verändert, ADR - 1
F852 .A U		ASCII-Zeichen sind durch "." gekennz.
	U [SP]	Inh. verändert, ADR + 1
F853 .M	[CR]	Inh. unverändert, Fertig
KMD >_	Nächstes Kommando	

— Beispiel für Adresse = F860 und Format = D:

Schirmbild	Eingabe	Wirkung
F860 164	SP	Inh. unverändert, ADR + 1
F861 183 1	1 SP	Inh. verändert, ADR + 1
F862 247 0	0 SP	Inh. verändert, ADR + 1
F863 191 300	300 SP	Summer ertönt, da 300 > 255; 30 wird angenommen. Inh. verändert, ADR + 1
F864 160 -	-	Inh. unverändert, ADR - 1
F863 30 -	-	Inh. unverändert, ADR - 1
F862 0 -	-	Inh. unverändert, ADR - 1
F861 1	CR	Inh. unverändert, Fertig
KMD >_	Nächstes Kommando	

— Beispiel für Adresse = F870 und Format = B:

Schirmbild	Eingabe	Wirkung
F870 10100100	SP	Inh. unverändert, ADR + 1
F871 10110111 00001111	00001111 SP	Inh. verändert, ADR + 1
F872 10100000 -	-	Inh. unverändert, ADR - 1
F871 00001111	CR	Inh. unverändert, Fertig
KMD >_	Nächstes Kommando	

Verlassen des Kommandos Memory:

1. Durch Betätigung von CR
   Das Betriebssystem trägt die letzte Änderung in den RAM-Speicher ein und fordert zur Eingabe eines neuen Kommandos auf

2. Durch Betätigung von ECS
   Die Bearbeitung des Memory-Kommandos wird abgebrochen.
   Achtung! Eine gewünschte Änderung des Speicherinhaltes an der zuletzt angezeigten Adresse wird nicht ausgeführt.

MAT 85 **Name:**

M-Kommando **Datum:**

M3

Geben Sie mit dem M-Kommando ab Adresse F800 folgende Daten im Hex-Format ein:

33	48	20	12	FF	3F	0D	4D	6D	7C

Sehen Sie sich diese Daten in den vier verschiedenen Formaten an und tragen Sie die Ergebnisse in die Tabelle ein.
Üben Sie dabei auch den Gebrauch der Tasten [DEL] , [+] , [-] und [ESC]

Zeile Nr.	Adr.	Inhalt der Speicherzeilen im Format ... H	D	A	B
1	F800				
2	F801				
3	F802				
4	F803				
5	F804				
6	F805				
7	F806				
8	F807				
9	F808				
10	F809				

Besonderheiten im Format A (ASCII-Zeichen):

Zeile	Bemerkungen
3	Die Hex-Zahl 20 entspricht dem ASCII-Zeichen SP (Space, Leerzeichen). Daher ist hier nur der . zu erkennen, der anzeigt, daß die Speicherzeile ein ASCII-Zeichen enthält.
4,5,7	Diese Speicherzeilen enthalten hexadezimale Zahlen, deren Bitkombinationen keine ASCII-Zeichen entsprechen. Im A-Format wird daher der Inhalt solcher Speicherzeilen als Hex-Zahl dargestellt.
8,9	Obwohl in diesen beiden Speicherzeilen Inhalte stehen, die sich durch ein Bit voneinander unterscheiden, wird für beide Inhalte das ASCII-Zeichen M angezeigt. Tatsächlich aber entspricht der Hex-Zahl 4D das M und 6D das m. Das Video-Interface des Computers zeigt jedoch nur Großbuchstaben an.

P-Kommando **P1**

Mit dem Print-Kommando werden die Inhalte eines Speicherbereichs im gewünschten Format ausgedruckt. Dazu muß der anzuzeigende Speicherbereich durch Angabe einer Start- und Stop-Adresse definiert werden, die das Betriebsprogramm nach dem Kommandoaufruf vom Bediener erfragt. Die möglichen Formate entsprechen denen des M-Kommandos.

Im Protokoll werden je nach dem gewählten Format bis zu acht Speicherinhalte in einer Zeile ausgedruckt. Jedes Zeilenprotokoll beginnt mit der Adresse des ersten in der Zeile ausgedruckten Speicherinhalts.

Anwendung: Dokumentation von Programmen im Hex-Code,
Text im Speicher suchen.

Aufruf und Handhabung:

```
KMD > PRINT

 START-ADR =X1X1 Y1Y1

 STOP -ADR =X2X2 Y2Y2

 FORMAT    =X Y
```

P [CR] oder P [SP] eintippen
"RINT" wird ergänzt

X1X1 = Vorgabe; Neu: Y1Y1 [CR] oder [SP]
Vorgabe: [CR] oder [SP]

X2X2 = Vorgabe; Neu: Y2Y2 [CR] oder [SP]
Vorgabe: [CR] oder [SP]

X = Vorgabe; Neu: Y [CR] oder [SP]
Vorgabe: [CR] oder [SP]
Y = A, B, D, H wie beim M-Kommando.

— Beispiel Startadresse = 0080, Stopadresse = 0094, Format = H:

```
KMD > PRINT
 START-ADR =0000 0080
 STOP -ADR =0000 0094
 FORMAT    =H
 0080 43 29 20 43 4F 50 59 52
 0088 49 47 48 54 20 31 39 38
 0090 32 20 42 46 5A
KMD >_
```

Format-Vorgabe wurde akzeptiert.
Speicherzeile mit Adresse 0080 hat den Inhalt 43, Speicherzeile mit Adresse 0081 den Inhalt 29 usw.

— Beispiel für gleiche Start- und Stopadresse, jedoch Format D:

```
KMD > PRINT
 START-ADR =0080
 STOP -ADR =0094
 FORMAT    =H D
 0080  67  41  32  67  79  80  89  82
 0088  73  71  72  84  32  49  57  56
 0090  50  32  66  70  90
KMD >_
```

Die in obigem Beispiel dargestellten Speicherinhalte sind hier in dezimaler Schreibweise ausgedruckt.

Beispiel für gleiche Start- und Stopadresse, jedoch Format = A:

```
KMD > PRINT
 START-ADR =0080
 STOP -ADR =0094
 FORMAT    =D A
 0080 .C .) .  .C .0 .P .Y .R
 0088 .I .G .H .T .  .1 .9 .8
 0090 .2 .  .B .F .Z

KMD > _
```

Die im ersten Beispiel dargestellten Speicherinhalte sind hier im ASCII-Code dargestellt.

- Beispiel für gleiche Start- und Stopadresse, jedoch Format = B:

```
KMD > PRINT
 START-ADR =0080
 STOP -ADR =0094
 FORMAT    =A B
 0080 01000011
 0081 00101001
 0082 00100000
 0083 01000011
 0084 01001111
 0085 01010000
 0086 01011001
 0087 01010010
 0088 01001001
 0089 01000111
 008A 01001000
 008B 01010100
 008C 00100000
 008D 00110001
 008E 00111001
 008F 00111000
 0090 00110010
 0091 00100000
 0092 01000010
 0093 01000110
 0094 01011010

KMD > _
```

Die im ersten Beispiel dargestellten Speicherinhalte sind hier im Binär-Code dargestellt.

Der Bildschirmausdruck wird an dieser Stelle unterbrochen durch Ausgabe der Aufforderung, die SPACE-Taste (_==>SPACE) zu betätigen. Die oberen drei Zeilen werden aus dem Bild "gerollt".
Diese Meldung wird immer dann ausgegeben, wenn mehr als 16 Bildschirmzeilen ausgegeben werden sollen.

Übung: Drucken Sie die Inhalte des Speicherbereichs von 0190 bis 0250 in allen Formaten aus.

G-Kommando G1

Mit dem Go-Kommando wird der Prozessor veranlaßt, Anwender-Programme von einer bestimmten Startadresse an abzuarbeiten.

Aufruf und Handhabung:

```
KMD > GO

 START-ADR =XXXX YYYY

(Kommando-Ausführung)
 *** USER ***
KMD > _
      (oder)
 _
```

G [CR] oder [SP] eintippen
"O" wird ergänzt

XXXX = Vorgabe; Neu: YYYY [CR] oder [SP]
Vorgabe: [CR] oder [SP]

Meldung nach Abarbeitung eines nicht zyklischen Programms, das mit einem Restart-Befehl abgeschlossen wurde.
Meldung bei Abarbeitung eines zyklischen Programms.

Zur Kommando-Ausführung:

- Bei der Ausführung zyklischer Programme (Schleifen ohne Ende) kann eine Rückkehr zum Betriebsprogramm nur durch Betätigen der RESET-Taste erfolgen. Danach meldet sich das Betriebsprogramm mit dem Ausdruck *** RESET *** (Rücksetzen) und erwartet das nächste Kommando.

- Nicht zyklische Programme müssen mit einem Rücksprungbefehl (RST1, Restart, CFH) abgeschlossen sein. Wenn dieser Befehl ausgeführt wurde, meldet sich das Betriebsprogramm mit dem Ausdruck *** USER *** (Benutzer) und erwartet das nächste Kommando.

- Durch Fehlbedienungen des Gerätes während der ersten Experimente mit dem Betriebssystem kann es vorkommen, daß sich ein Programm nicht starten läßt, obwohl es richtig eingegeben wurde. Meist erfolgt nach dem Startversuch die Meldung *** RESET ***. Um das Programm starten zu können, muß der Stack-Pointer (Stapelzeiger), ein spezielles Register in der CPU, mit Hilfe des Register-Kommandos mit dem Wert FC32H geladen werden.
  Hinweise hierzu finden Sie unter der Beschreibung des Register-Kommandos.

MAT 85

Name:

G-Kommando

Datum:

G2

Für die folgenden Experimente benötigen Sie zusätzlich eine Eingabe- und eine Ausgabe-Baugruppe im Baugruppenträger. Stellen Sie vor dem Einschieben die Port-Adressen wie folgt ein:
Eingabe-Baugruppe: Adresse 12H
Ausgabe-Baugruppe: Adresse 13H

1. Laden Sie mit dem Memory-Kommando ab Adresse F800H das folgende zyklische Programm in den Speicher.

Adressen	Daten	Bemerkungen zu den Befehlen
F800 F801	DB 12	Eingabe-Port mit Port-Adr. 12 lesen und gelesenen Wert in den Akkumulator transportieren
F802 F803	D3 13	Akku-Inhalt zum Ausgabe-Port mit Port-Adr. 13 transportieren
F804 F805 F806	C3 00 F8	Sprung zum Anfang dieses Programms (bei Adr. F800)

2. Überprüfen Sie die Programmeingabe mit dem Print-Kommando.

3. Starten Sie das Programm mit dem Go-Kommando.
   Wirkung: Die Signalkombinationen, die Sie am Eingabe-Port einstellen, müssen auch am Ausgabe-Port erscheinen.

4. Beenden Sie den Programmlauf durch Betätigen der RESET-Taste.

5. Ersetzen Sie mit dem Memory-Kommando das Befehlsbyte C3 (Adresse F804) durch das Befehlsbyte CF (Restart-Befehl RST1).

6. Stellen Sie mit den Eingabeschaltern des Eingabe-Ports die Bitkombination 55H ein.

7. Starten Sie das geänderte Programm mit dem Go-Kommando.

G3

Wirkung: Am Ausgabe-Port erscheint die Bitkombination 55 und auf dem Bildschirm die Meldung *** USER ***.
Der Sprungbefehl (C3) zum Anfang des Programms wurde durch den Restart-Befehl (CF) ersetzt. Dadurch ist aus dem zyklischen Programm ein nicht zyklisches Programm mit dem erforderlichen Befehl für die Rückkehr zum Betriebsprogramm geworden.

8. Ersetzen Sie nun mit dem Memory-Kommando das Befehlsbyte CF (Adresse F804) durch das Befehlsbyte FF (Restart-Befehl RST7) und die Einstellung am Eingabe-Port auf die Bitkombination AAH.

9. Starten Sie das geänderte Programm mit dem Go-Kommando.
Wirkung: Auch dieses nicht zyklische Programm wird einmal abgearbeitet (Eingabe-Bitkombination = Ausgabe-Bitkombination) ehe der Rücksprung zum Betriebsprogramm erfolgt. Diesesmal wird jedoch die Meldung *** PROGRAMM-ABORT *** ausgegeben. Eine solche Meldung erfolgt immer dann, wenn der Prozessor im Verlauf seiner Befehlsbearbeitung auf den Datenwert FF trifft. Dies ist z.B. immer der Fall, wenn die Startadresse im G-Kommando in einem Speicherbereich liegt, in dem gar kein RAM-Speicher vorhanden ist.

10. Drucken Sie sich die Inhalte des Betriebsprogramm-RAM's zwischen FC00 und FFFF mit dem Print-Kommando aus.

Sie erkennen das häufige Auftreten des Datums FF. Wenn der Prozessor dieses Byte findet - meist ist das der Fall, wenn ein Anwenderprogramm nicht mehr kontrolliert arbeitet (Fehler im Programm) - wird die weitere Programmbearbeitung abgebrochen.

Mit dem Disassembler-Kommando können Programme, die in Maschinensprache geschrieben sind und sich im Speicher des Systems befinden, in den "Mnemo-Code" oder "Assembler-Code" übersetzt werden.
(Symbolische Namen für Adressen werden immer dann eingesetzt, wenn sie vorher bei der Eingabe des Programms mit dem Assembler-Kommando definiert wurden. Siehe hierzu Beschreibung des Assembler-Kommandos).

Aufruf und Handhabung:

```
KMD > DISASSEMBLER

 START-ADR =X1X1 Y1Y1

 STOP -ADR =X2X2 Y2Y2

(Kommando-Ausführung)
KMD >_
```

D [CR] oder D [SP] eintippen
"ISASSEMBLER" wird ergänzt

X1X1 = Vorgabe; Neu: Y1Y1 [CR] oder [SP]
Vorgabe: [CR] oder [SP]

X2X2 = Vorgabe; Neu: Y2Y2 [CR] oder [SP]
Vorgabe: [CR] oder [SP]

Zur Kommando-Ausführung:

- Die zwischen den eingegebenen Start- und Stop-Adressen liegenden Maschinen-Bytes werden disassembliert, d.h., in den zugehörigen Mnemo-Code übersetzt.

- Zur richtigen Übersetzung eines Maschinenprogramms ist es notwendig, daß die Start-Adresse auf ein Befehlsbyte zeigt.

Beispiel für einen Disassembler-Ausdruck:

```
KMD > DISASSEMBLER
 START-ADR =0000 F800
 STOP -ADR =0000 F806
F800 3E 03          MVI  A,03
F802 3D             DCR  A
F803 C2 00F8        JNZ  F800
F806 CF             RST  1
                    END

KMD >_
```

Es werden nur die Speicheradressen angegeben, unter denen das jeweils 1. Byte eines Befehls gespeichert ist. Dieses 1. Byte eines jeden Befehls wird häufig Befehlsbyte genannt.

MAT 85

Name:

D-Kommando

Datum:

D2

1. Laden Sie mit dem M-Kommando ab Adresse F850 das folgende Programm in den Speicher:

Adressen	Daten	Bemerkungen
F850	DB	Befehlsbyte
F851	12	
F852	2F	Befehlsbyte
F853	D3	Befehlsbyte
F854	13	
F855	C3	Befehlsbyte
F856	50	
F857	F8	

2. Disassemblieren Sie das Programm erst ab F850 (richtige Adresse) und dann ab F851 (falsche Adresse). Tragen Sie die verschiedenen Ergebnisse in die vorbereiteten Tabellen ein.

START- ADR= F850 STOP - ADR= F857		
Adressen	Daten	Mnemo- Code

START- ADR= F851 STOP - ADR= F857		
Adressen	Daten	Mnemo-Code

3. Laden Sie mit dem Memory-Kommando den Wert C3 in die Speicherzeile, die vor Beginn des Programms liegt (F84F). Disassemblieren Sie dann ab F84F und vergleichen Sie dieses Ergebnis mit dem richtigen Programm (ab F850).

4. Fassen Sie die Versuchsergebnisse zusammen!

N1

N-Kommando

Mit dem Kommando "Next Instruction" (nächsten Befehl bearbeiten) läßt sich ein Anwenderprogramm schrittweise abarbeiten. Nach jedem ausgeführten Befehl werden die Inhalte der CPU-Register protokolliert. Die Anzahl der abzuarbeitenden Befehle (die Steps oder Schritte) kann vorgewählt werden. Die Programmausführung läuft nicht in Echtzeit ab.
Das N-Kommando kann besonders dazu dienen, die Wirkung einzelner Befehle zu studieren und den Lauf eines zu testenden Programms zu verfolgen.

Aufruf und Handhabung:

```
KMD > NEXT INSTRUCTION

 START-ADR =XXXX YYYY

 STEPS     =XX YY

(Kommando-Ausführung)
KMD >_
```

N [CR] oder N [SP] eintippen
"EXT INSTRUCTION" wird ergänzt

XXXX = Vorgabe; Neu: YYYY [CR] oder [SP]
Vorgabe: [CR] oder [SP]

XX = Vorgabe; Neu: YY [CR] oder [SP]
Vorgabe: [CR] oder [SP]

YY: zwischen 00 und 99 möglich

Zur Kommando-Ausführung:

- Die nach der Startadresse (XXXX oder YYYY) folgenden (XX bzw. YY) Befehle werden ausgeführt. Nach jedem abgearbeiteten Befehl werden die Inhalte der CPU-Register ausgedruckt.

- Wenn sich im Anwenderprogramm ein Halt-Befehl (HLT, 76H) oder ein unbekannter Befehls-Code befindet, wird die weitere Bearbeitung des Programms abgebrochen und folgende Meldung ausgedruckt:

```
*** HALT ODER ILLEGALER OPCODE ***
```

- Bei mehrmaligem Aufruf des N-Kommandos wird das Anwenderprogramm an der jeweils vorher unterbrochenen Stelle fortgesetzt.

- Die Start-Adresse muß auf ein Befehlsbyte des zu untersuchenden Programms zeigen.

Beispiel für die Ausführung des N-Kommandos:
(Das Programm wurde vorher mit dem Memory-Kommando in den Speicher geladen).

```
KMD > NEXT INSTRUCTION
 START-ADR =1E09 F850
 STEPS     =  0 4
 PC   LABEL:  OP   ADR.FELD  A  NZHPC B  C  D  E  H  L  I  SP
 F850         IN   12        F0 00000 00 00 00 00 00 00 80 FC32
 F852         CMA            0F 00000 00 00 00 00 00 00 80 FC32
 F853         OUT  13        0F 00000 00 00 00 00 00 00 80 FC32
 F855         JMP  F850      0F 00000 00 00 00 00 00 00 80 FC32
 F850         IN   12
KMD > _
```

Dieser Befehl wird nicht mehr ausgeführt (5. Schritt).

Die Abkürzungen der Kopfzeile und ihre Bedeutungen:

PC-Programm Counter (Programmzähler, 16 Bit); unter dieser Spalte werden die Adressen der Speicherzeilen angezeigt, die das Befehlsbyte des jeweiligen Befehls enthalten.

LABEL: Symbolische Adresse (siehe Assembler-Kommando).

OP Operations-Code; enthält den mnemonischen Code des Befehlsbytes.

ADR.FELD Adreß-Feld; enthält Adressen bzw. Daten zum Befehl.

A - REGISTER A (Akkumulator, oft kurz Akku, 8 Bit)

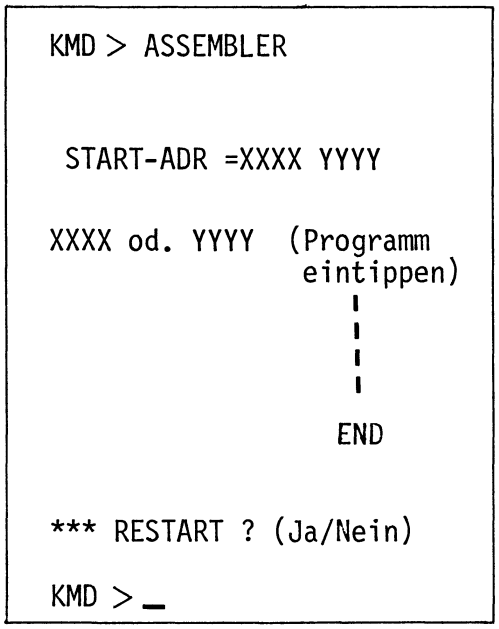
N	- NEGATIV -FLAG	(Negativ-Bit)	FLAG-Bits des Prozessor-Status-Registers (Zustandsregister)
Z	- ZERO -FLAG	(Null-Bit)	
H	- HALF CARRY-FLAG	(Zwischenübertrags-Bit)	
P	- PARITY -FLAG	(Paritäts-Bit)	
C	- CARRY -FLAG	(Übertrags-Bit)	

B - REGISTER B (8 Bit)
C - REGISTER C (8 Bit)
D - REGISTER D (8 Bit)
E - REGISTER E (8 Bit)
H - REGISTER H (8 Bit)
L - REGISTER L (8 Bit)
I - INTERRUPT CONTROL REGISTER (Interrupt-Masken-Register, 8 Bit)
SP - STACK POINTER (Stapel-Zeiger, 16 Bit)

Gliederung:

Mit dem Assembler-Kommando wird ein Hilfsprogrammm, das man Assembler nennt, aufgerufen. Aufgabe dieses Hilfsprogramms ist die Übersetzung jeder im mnemonischen Code eingegebenen Programmzeile in den zugehörigen Maschinencode. Der Maschinencode wird dann ab einer bestimmten Speicheradresse in den RAM-Speicher geschrieben. Diese Speicheradresse wird nach Aufruf des A-Kommandos durch die Eingabe einer "START-ADRESSE" festgelegt.
Um mit dem "Assemblerprogramm" (kurz Assembler) arbeiten zu können, bedarf es der Einhaltung einiger Regeln, die im folgenden schrittweise erklärt und geübt werden.

Aufruf des Assemblers:



A [CR] oder A [SP] eintippen
"SSEMBLER" wird ergänzt

XXXX = Vorgabe; Neu: YYYY [CR] oder [SP]
Vorgabe: [CR] oder [SP]

Siehe folgende Beschreibungen

Jede eingetippte Zeile muß mit [CR] abgeschlossen werden!

END [CR]; Jedes Programm muß mit END beendet werden

J [CR] oder N [CR] eintippen.
Sinn und Beschreibung folgen.
Nächstes Kommando wird erwartet.

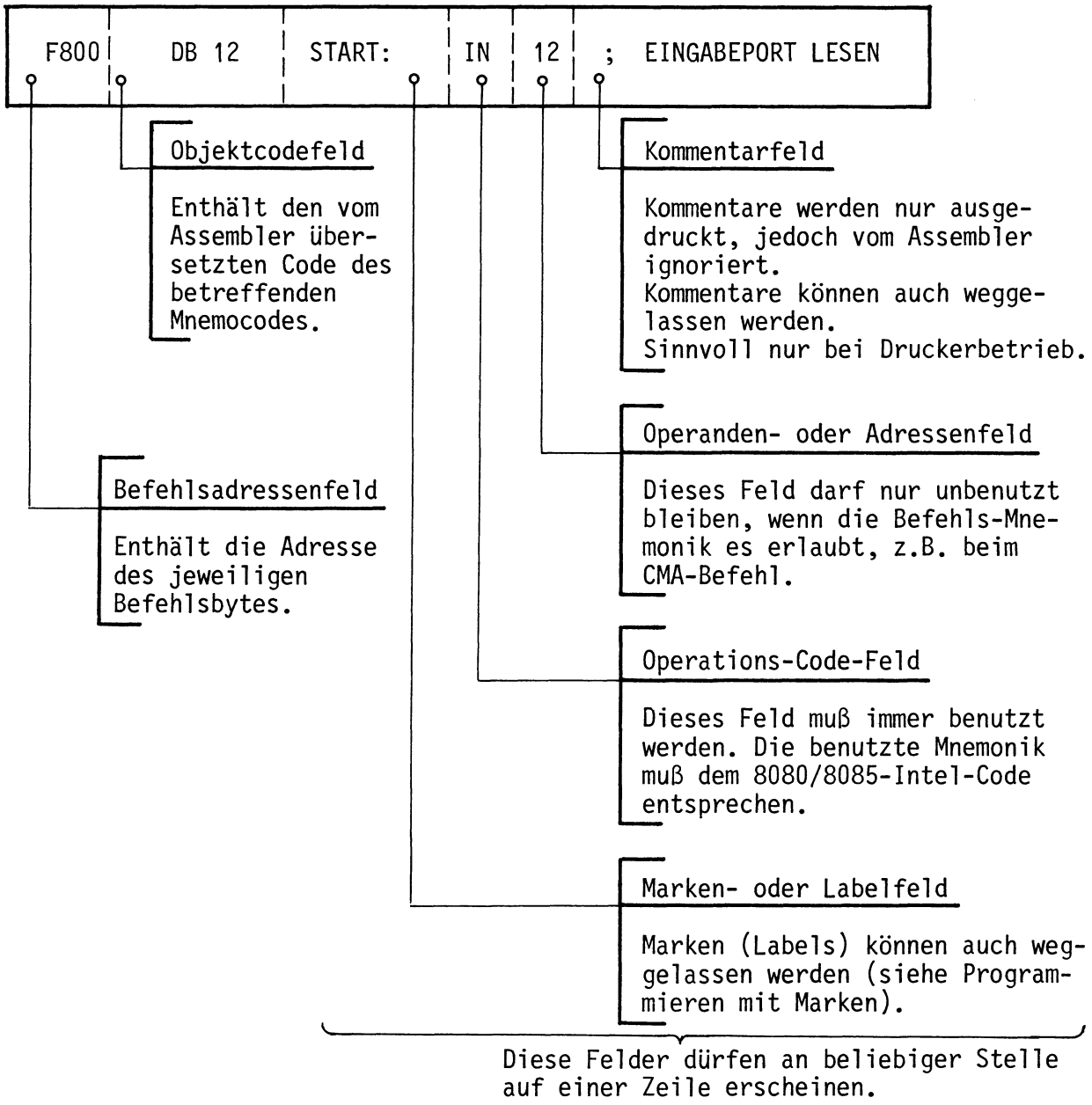
Übung:
Rufen Sie den Assembler auf, geben Sie die RAM-Start-Adresse F800 und den einzigen "Befehl" END ein und üben Sie den Austritt aus dem Assembler mit den verschiedenen "RESTART-Antworten".

Aufbau einer Assembler-Befehlszeile:

Rufen Sie den Assembler auf und geben Sie die Startadresse F800 ein.
Tippen Sie folgende Zeile ein:

START: IN 12; EINGABEPORT LESEN [CR]

([CR] wird im folgenden nicht mehr angegeben).
Die nach der Assemblierung ausgedruckte Zeile (Befehlszeile) kann in die dargestellten Felder zerlegt werden:

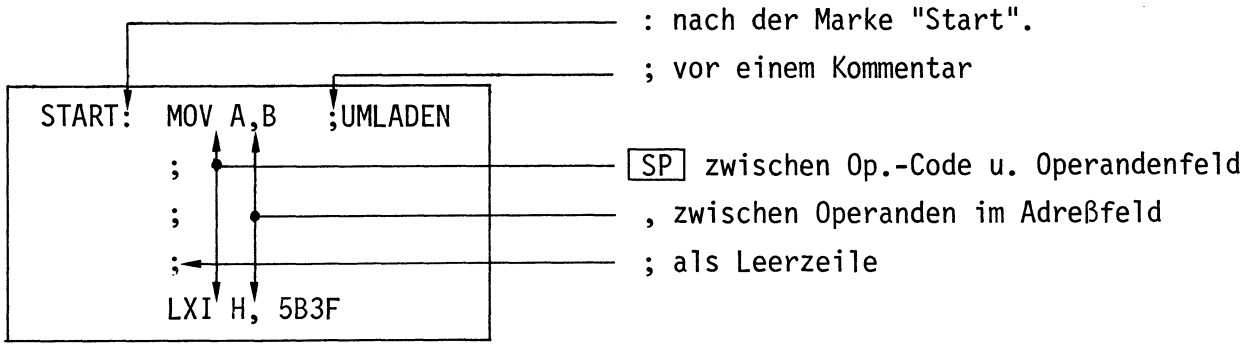


Damit der Assembler die Felder trennen kann, müssen ihm Begrenzungszeichen am Ende oder Beginn eines Feldes mitgeteilt werden.

Begrenzungszeichen für die Felder:

Zwischenraum [SP]	zwischen Operations-Code- und Operandenfeld
,	zwischen Operanden im Adressenfeld
;	1. vor einem Kommentar 2. wenn als einziges Zeichen in der Zeile, zur Erzeugung einer Leerzeile (wird vom Disassembler ignoriert)
:	Unmittelbar nach einer Marke

Beispiele:



Übung:

Tippen Sie das folgende Programm ab Startadresse F900 ein. Füllen Sie in der Tabelle das Befehlsadressenfeld und das Objektcodefeld aus. Kontrollieren Sie Ihre Eingabe mit dem Disassembler. Starten Sie das Programm.

Mit Schalter B0 des E-Ports müssen sich die vier unteren LED's (B4-B7) des A-Ports ein- oder ausschalten lassen.

F900	IN 12 ;EINGABEPORT LESEN
	ANI 01 ; B0 BETÄTIGT ?
	JNZ 0F90E ;NEIN,ALLE LEDS AUS
	MVI A,0F0;JA,LEDS B4-B7 EIN
	OUT 13
	JMP 0F900;ZURÜCK ZUM START
	MVI A,00;ALLE LEDS AUS
	OUT 13
	JMP 0F900
	END ;ENDE

Besonderheit: Wenn im Operanden- oder Adressenfeld eine Zahl mit A-F beginnt, muß dieser Zahl eine 0 vorgestellt werden (Erklärung später).

Fehlermeldungen:

Unbekannte und fehlerhaft eingegebene Befehle und Daten (Op.-Codes und Operanden) werden ignoriert (nicht angenommen) und mit "?" unterhalb der Zeile in der Umgebung des Fehlers kenntlich gemacht. Die Befehlsadresse wird nicht weitergezählt.

Beispiele für typische Fehler:

1.	F907 3E 00 MVI A, 0F0 ;..... ? F907	Nach, darf kein [SP] sein F907 wurde nicht geändert
2.	F909 OUT13 ? F909	[SP] zwischen Op.-Codefeld und Operand fehlt
3.	F912 C3 0000 JMP F900 F915	keine Fehlermeldung, jedoch wurde falsche Adresse assembliert (0000).0 vor F900 vergessen!

Fehlerbeseitigung:

- Fehler 1 und 2 können berichtigt werden, indem einfach die richtige Zeile eingegeben wird, denn die Befehle wurden ignoriert.

- Fehler der Art 3 können zunächst berichtigt werden, indem man nach Eingabe aller Befehle und Abschluß mit END etc. wieder den Assembler aufruft, und die START-ADR. auf die Adresse des zu ändernden Befehls setzt (hier z.B. "START-ADR = XXXX F912" [CR].
  Geben Sie dann den richtigen Befehl ein und verlassen Sie den Assembler mit der Taste [ESC]

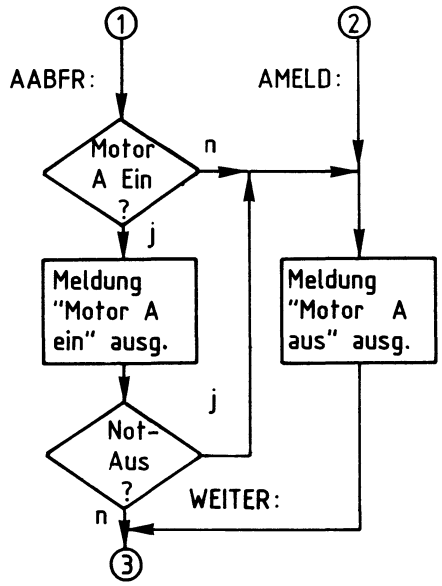
- Wenn Sie einen Fehler vor Abschluß der Zeileneingabe mit [CR] bemerken, können Sie die Fehleingabe mit Taste [DEL] löschen und den richtigen Text eingeben.

Programmieren mit Marken (Label): (Label = Markierung)

Bei der Entwicklung eines Programms für einen Mikrocomputer zur Lösung irgend einer Aufgabe geht man so vor:

- Definition der zu lösenden Aufgabe
- Erstellung eines Flußdiagramms
- Schreiben des Programms
- Testen des Programms, Fehlersuche und deren Beseitigung, dabei evtl. Änderung des Flußdiagramms und des Programms
- Dokumentation

Das folgende Bild zeigt ein Flußdiagramm für ein Programm, das noch zu erstellen ist.



Beim Entwurf des Flußdiagramms weiß man noch nicht, wo das spätere Programm im Speicher liegen wird und wieviele Programmschritte zur Lösung der Blöcke (z.B. Not-Aus?) nötig sein werden. Um trotzdem die Sprungziele kennzeichnen zu können - z.B. wohin, wenn Motor A ausgeschaltet oder wenn Not-Aus betätigt? - bedient man sich der Hilfe von Marken oder Labels.

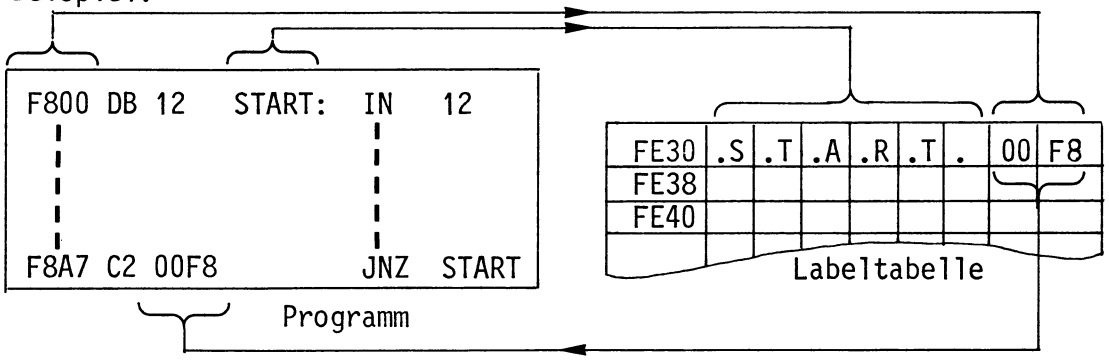
Marken oder Label sind Namen für Adressen, deren Werte während des Programmierens noch nicht bekannt sind.

- Marken werden am häufigsten in Sprung-Aufruf- und Verzweigungsbefehlen verwendet.
- Marken erleichtern das Auffinden von Programmstellen.
- Während des Programmierens braucht man sich nicht um die Berechnung von Adressen zu kümmern.
- Marken machen Programme verständlicher.

Wie behandelt der Assembler eine Marke?:

Wenn das Markenfeld einer Befehlszeile eine Marke enthält, trägt der Assembler diese Marke und die zugehörige Adresse des Befehlsbytes in eine "Labeltabelle" im Betriebsprogramm-RAM ein.
Man kann danach diese Marke als Adresse (oder als Datum) im Adressenfeld eines anderen Befehls verwenden. Der Assembler ersetzt dann die Marke durch den Adressenwert aus der Labeltabelle, wenn er das Maschinenprogramm erzeugt.

Beispiel:



Das Setzen einer Marke mit nachfolgendem Doppelpunkt in das Marken- oder Labelfeld nennt man auch "Definieren einer Marke oder eines Labels".

Regeln für den Gebrauch von Marken:

- Marken dürfen eine Länge von 1 bis 6 Zeichen haben, das erste Zeichen muß ein Buchstabe sein.

- Damit der Assembler zwischen den Hex-Zahlen A-F und dem 1. Buchstaben einer Marke unterscheiden kann, muß den Hex-Zahlen A-F eine 0 vorangestellt werden.

- Im Programmverlauf erst später definierte Marken dürfen vorher schon im Adreßfeld benutzt werden. Der Assembler trägt die zugewiesenen Adressen nach, sobald sie definiert werden.

- Ein Markenname darf nur einmal definiert werden.

- Mehr als 57 Marken sind nicht erlaubt.

Fehlermeldungen beim Umgang mit Marken:

— Wird eine Marke innerhalb eines Programms mehrfach definiert, so macht der Assembler nach Abschluß der Zeile durch [CR] mit einem "?" auf diesen Fehler aufmerksam. Die gleiche Fehlermeldung tritt auch dann auf, wenn in einem ganz anderen Programm dieser Markenname schon einmal benutzt wurde (häufig hat man mehrere Übungsprogramme in verschiedenen Speicherbereichen gespeichert).

● Abhilfe: Verwenden Sie an der Stelle eine neue Marke!

— Die oben genannte Fehlermeldung tritt auch dann auf, wenn Sie sich beim Eintippen einer Befehlszeile mit Marke im Operationscode- oder im Operandenfeld vertippt haben und im zweiten Anlauf versuchen, diesen Fehler zu beheben. Die Marke ist vom Assembler bereits angenommen und wird bei erneuter Eingabe als "schon definiert" behandelt.

● Abhilfe: Geben Sie nur den Mnemo-Code neu ein!

— Alle verwendeten Marken (auch die aus anderen Programmen) werden immer dann automatisch nach Abschluß der Programmeingabe mit "END" ausgedruckt, wenn das Programm noch undefinierte Marken enthält. Diese werden dann mit * gekennzeichnet. Die rechts daneben angegebene Adresse zeigt auf den Speicherplatz, der das niederwertige Byte der zur Marke gehörenden Adresse enthält.

Beispiele:

```
E000 DB 12          START:   IN 12
E002 C2 0000        JNZ E012
E005 C3 0000        JMP WEITER
E008                END
E012  *E003
START  E000
WEITER*E006
```

Start wurde auf E000 definiert

Weil vor E012 die 0 fehlt, faßt der Assembler die Adresse als Marke auf. Da sie noch nicht definiert ist, wird zunächst 0000 eingesetzt.
WEITER ist ebenfalls noch undefiniert.
Abschluß mit END.
Automatischer Ausdruck der Marken:
"E012" wird als nichtdefinierte Marke ausgedruckt (da mit Buchstaben beginnend), ebenso "WEITER". "START" ist definiert.

Anweisungen an den Assembler zum Löschen und Ausdrucken der Marken:

– Löschen der Marken

Wenn man ohne Rücksicht auf bereits vorher verwendete Marken ein neues Programm eingeben möchte, muß die im RAM liegende Labeltabelle (ab FE30) gelöscht werden. Nach dem Löschen sind alle, auch in früher eingegebenen Programmen, verwendeten Marken verschwunden. Die Programme bleiben trotzdem lauffähig, denn ihr Maschinen-Code befindet sich ja noch im Speicher. Lediglich beim Disassemblieren der Programme fehlen die ursprünglich verwendeten Marken.

Das Löschen der Labeltabelle erfolgt nach dem Eintritt in den Assembler durch Eingeben der Anweisung "LC" (Label Clear).

Beispiel:

```
KMD> ASSEMBLER
 START-ADR =F800 E000
E000             LC
E000       –
```

Aufruf des Assemblers;
Löschen der Labeltabelle;
Programmeingabe wird erwartet;

– Ausdrucken der Marken

Wollen Sie sich zur Orientierung nach dem Eintritt in den Assembler (z.B. um das Programm zu ändern) oder während des Programmierens oder am Ende der Programmeingabe die bisher verwendeten Marken ausdrucken lassen, so müssen Sie die Anweisung "LP" eingeben. Die Befehlsadresse wird dadurch nicht verändert. (LP = Label Print)

Beispiel:

```
   ¦                ¦
   ¦                ¦
   ¦                ¦
   ¦                ¦
E100 7D          MOV A,L
E101 C3 0000     JMP WARTE
E104             LP
START  E000
WARTE *E102
Z1     E01B
Z2     E01D

E104           –
```

Ausdruck aller verwendeten Marken, auch der evtl. noch nicht definierten (mit *) und der in anderen Programmen verwendeten.

Weiter mit Programmeingabe.

Assembler-Anweisungen:

Assembler-Anweisungen sind Anweisungen für das Assemblerprogramm, die nicht in Maschinensprache übersetzt werden. Mit ihrer Hilfe läßt sich z.B. ein Maschinenprogramm einem bestimmten Speicherbereich zuweisen, oder ein RAM-Bereich für die Ablage von Datenbytes oder Adressen festlegen.

Um diese Anweisungen - man nennt sie auch Pseudo-Operationen (vorgetäuschte Op.) - zu verwenden, müssen Sie die Mnemonik dieser Anweisungen in das Op.-Codefeld und Adressen oder Daten (falls erforderlich) in das Adressenfeld setzen.

Dieser Assembler gestattet die Verwendung folgender Pseudo-Operationen:

ORG	— ORIGIN (Ursprung)
EQU	— EQUATE (Gleichsetzen)
DB	— DEFINE BYTE, definiere Byte, 8-Bit
DW	— DEFINE WORD, definiere Wort, 16-Bit
LP, LC	— LABEL-PRINT u. CLEAR (bereits erklärt)
END	— Ende einer Programmeingabe

— Die ORG-Anweisung:

Mit dieser Anweisung wird die Befehlsadresse neugesetzt. Der Assembler erhält dadurch Bescheid, ab welcher Adresse er die Maschinensprache der folgenden Befehle in den Speicher schreiben soll. Es können mehrere ORG's an verschiedenen Stellen im Programm verwendet werden. Mit ORG kann man auch zu bereits verlassenen Adressen zurückkehren, um dort z.B. eingegebene Fehler zu berichtigen.

Beispiele:

= Überspringen eines Speicherbereiches

```
  ¦                ¦
  ¦                ¦
  ¦                ¦
  ¦                ¦
E010 3E 04         MVI A,04
E012 2F            CMA
E013               ORG 0FA00
FA00               —
```

— Die Befehlsadresse wird von E013 auf FA00 gesetzt. Die 0 vor der Adresse dient der Unterscheidung der Adresse von einer Marke.

= Nachträgliches Berichtigen einer Befehlszeile

```
  ¦                ¦
  ¦                ¦
  ¦                ¦
  ¦                ¦
F900 79            MOV A,C
F901 00            NOP
F902 00            NOP
F903 00            NOP
F904 A6            ANA M
F905               ORG 0F900
F900 78            MOV A,B
F901               ORG 0F905
F905               —
```

— Hier bemerkt der Programmierer, daß er sich bei Adresse F900 vertippt hatte. Nach Rückkehr zu F900 mit der ORG-Anweisung und Berichtigung des Fehlers springt er mit ORG 0F905 zur ursprünglichen Befehlsadresse vor.

= Label in der ORG-Anweisung

```
E00A 06 FF ZEIT:   MVI B, 0FF
  ¦                ¦
  ¦                ¦
  ¦                ¦
E010 3E FE         MVI A,0FE
E012               ORG ZEIT
E00A               —
```

— Die Adresse ist hier durch die Marke "Zeit" angegeben. Ihr wurde vorher im Programm E00A zugewiesen.

— Die EQU-Anweisung:

Mit dieser Anweisung können bestimmten Adressen oder Daten Namen (Marken) zugeordnet werden. Diese Namen und die ihnen gleichgesetzten (zugewiesenen) Adressen oder Daten werden ebenfalls in die Labeltabelle eingetragen.
Mit der EQU-Anweisung werden jedoch keine Daten in den Programmspeicher geladen.

Setzen Sie EQU-Anweisungen immer an den Anfang einer Programms, es wird dadurch besser lesbar.

Beispiel:

```
KMD > ASSEMBLER
 START-ADR =E010 F800
F800              LC                 — alte Labeltabelle löschen
F800              CR EQU 0D          — Namen CR wird 0D zugewiesen
F800              EPORT EQU 12       — Namen EPORT wird Adr. 12 zugewiesen
F800              APORT EQU 13       — Namen APORT wird Adr. 13 zugewiesen
F800              ;                  — Leerzeile zur Trennung
F800 DB 12        IN EPORT             Der Assembler ersetzt bei der
F802 2F           CMA                  Erzeugung des Maschinen-Codes
F803 D3 13        OUT APORT          —die Namen durch die ihnen zuge-
F805 3E 0D        MVI A,CR             wiesenen Daten oder Adressen.
F807               ¦
                   ¦
```

Ein Blick in die Labeltabelle:

```
KMD > PRINT
 START-ADR =FE30
 STOP -ADR =FE48
 FORMAT    =A
 FE30 .A .P .0 .R .T .  13 00
 FE38 .C .R .  .  .  .  0D 00
 FE40 .E .P .0 .R .T .  12 00
 FE48 FF
```

Die EQU-Anweisung kann auch dazu dienen, nicht definierte Marken nachzudefinieren:

— Wenn ein Programm bereits mit "END" abgeschlossen wurde, müssen Sie dazu den Assembler neu aufrufen und die mit "*" gekennzeichneten Marken definieren.

— Wenn Sie sich noch im Programm befinden, können Sie dies bei der gerade aktuellen Befehlsadresse tun, denn sie wird dadurch ja nicht verändert.

— Die DB-Anweisung:

Diese Anweisung setzt Datenbytes oder ASCII-Zeichen in den Programmspeicher und zwar ab der Adresse, bei der die DB-Anweisung erteilt wird.

Beispiele:

= Absetzen von Datenbytes

```
  ¦              ¦
  ¦              ¦
F800 0A3F5BFF    DB 0A,3F,5B,0FF,7C,3A,0E4,55,3E
F804 7C3AE455
F808 3E          —
```

Inhalt des Programmspeichers

- DB und erstes Byte durch Leerzeichen trennen;
- Bytes durch Komma trennen;
- max. 20 Bytes pro DB möglich;
- nach letztem Byte kein Komma;
- vor A-F 0 setzen;

Es empfiehlt sich, jeweils höchstens die Bildschirmzeile zu füllen und dann mit [CR] abzuschließen. Sollen mehr Bytes abgesetzt werden, erneut DB verwenden.

= Absetzen von ASCII-Zeichen

```
  ¦              ¦
  ¦              ¦
F900 5343484E    DB 'SCHNAPS STINKT'
F904 41505320
F908 5354494E
F90C 4B54        —
```

- Zeichen in ' ' einschließen;
- max. 40 Zeichen pro DB möglich;

= Absetzen von ASCII- und Hex-Zeichen

```
  ¦              ¦
  ¦              ¦
FA00 FA3A4D49    DB 0FA,3A,'MILCH ',0D,'KLARER',00
FA04 4C434820
FA0B 0D4B4C41
FA0C 52455200
FA10             —
```

Übung:

Probieren Sie die gezeigten Beispiele aus!

— Die DW-Anweisung:

Mit der DW-Anweisung kann man 16-Bit-Worte in den Speicher eingeben. Die Anzahl der Worte pro Zeile sollte 9 nicht übersteigen (Bildschirmzeile voll).

Beispiele:

= Absetzen von Adressen

```
FA00 3412FCAA    DW 1234,0AAFC,55AF,0F800
FA04 AF5500F8    —
```

im Speicher sind höher- und niederwertige Bytes vertauscht.

- DW und erste Adresse durch Leerzeichen trennen;
- Adressen durch Komma trennen;
- nach letzter Adresse kein Komma;
- vor A-F 0 setzen;

= Absetzen von Adressen, die vorher durch die EQU-Anweisung definiert wurden

```
E010             AADR EQU 1234
E010             EADR EQU 0F850
E0F0 341250F8    DW AADR,EADR
E0F4             —
```

DB- und DW-Anweisungen setzt man häufig ans Ende eines Programms.

— Die END-Anweisung:

An der END-Anweisung erkennt der Assembler das Ende der Programmeingabe. Nach Abschluß dieser Anweisung mit [CR] fragt der Assembler mit dem Ausdruck

```
*** RESTART ? (JA/NEIN)
```

danach, ob er einen RST-1-Befehl ans Ende des Programms setzen soll (J-[CR]) oder nicht (N-[CR] oder [CR]). Dieser Befehl bewirkt einen Sprung ins Betriebsprogramm.
Mit einem solchen Rücksprung ins Betriebsprogramm soll verhindert werden, daß durch unkontrollierten Lauf des Computers wichtige Speicherinhalte im Betriebsprogramm-RAM überschrieben werden. Abschluß eines "Endlos-Programms" mit RST1 bewirkt den Ausdruck:

```
*** USER ***
```

(Anwender, Benutzer)

und Rückkehr in die "KMD>-Routine".

Beispiel:

```
ASSEMBLER
 START-ADR =E000
E000 DB 12        START:IN 12
E002 E6 04        ANI 04
E004              END
 *** RESTART ? (JA/NEIN) J
```

— Eingabe eines Programms ohne ein Ende durch Rücksprung zum Anfang (Schleife).

— Restart-Frage mit "Ja" beantwortet.

```
KMD > DISASSEMBLER
 START-ADR =E000
 STOP -ADR =E007 E004
E000 DB 12    START:  IN   12
E002 E6 04            ANI  04
E004 CF               RST  1
```

— Der Assembler hat einen RST-1-Befehl ans Programmende gesetzt.

```
KMD > GO
 START-ADR =E0F5 E000

 *** USER ***
```

— Start des Programms bewirkt Rücksprung ins Betriebsprogramm.

Operations-Symbole + und —:

Dieser Assembler gestattet die Verwendung der Operations-Symbole "+" und "-" in Verbindung mit der Verarbeitung von Marken, Daten und Adressen. Addition und Subtraktion erfolgen dabei hexadezimal. Überläufe (etwa FE+3) dürfen nicht auftreten. Durch die Verwendung der Operations-Symbole lassen sich häufig Marken einsparen. Dies kann bei langen Programmen notwendig werden, da die Labeltabelle nur 57 Marken aufnehmen kann.

Beispiele:

— Einsparung einer Marke

```
KMD > ASSEMBLER
 START-ADR =E000 F800
F800              LC
F800 DB 12        START:IN 12
F802 06 07        MVI B,07
F804 4B           MOV C,B

F81B C3 04F8      JMP START+4
F81E
```

— In dieser Zeile wird eine Marke gespart

● Das "+" muß ohne Zwischenraum auf "START" folgen;
(Der Sprung muß auf eine Adresse zeigen, unter der ein Befehlsbyte steht).

— Operations-Symbole in Verbindung mit Assembler-Anweisungen:

```
KMD > ASSEMBLER
 START-ADR = F800
F800              LC
F800              EPORT EQU 12
F800              APORT EQU EPORT+1
F800              ;
F800 DB 12        START:IN EPORT
F802 2F           CMA
F803 D3 13        OUT APORT
F805              ORG START+F5
                              ?
F805              ORG START+0F5
F8F5 C3 00F8      JMP START
F8F8              END
 *** RESTART ?  (JA/NEIN)
```

— Die APORT-Adresse ergibt sich aus der Addition.

— Die APORT-Adresse ist berechnet.
— Hier fehlt die 0 vor F.

— Richtige Eingabe.
— Neue Befehlsadresse

Formatierte Programm-Eingabe:

Mit der Taste [→] kann der Cursor um je 8 Schreibstellen nach rechts versetzt werden.
Dadurch ist während der Eingabe des Operations-Codes eine übersichtlichere Darstellung aller Zeichen auf dem Bildschirm möglich.
Unabhängig vom Format der Programme im Assemblerbetrieb druckt der Disassembler die Programme jedoch formatiert aus.

Beispiele:

- Unformatiertes Assemblerprogramm

```
KMD > ASSEMBLER
 START-ADR =F800
F800              LC
F800              EPORT EQU 12
F800              APORT EQU 13
F800 DB 12        IN EPORT
F802 D3 13        OUT APORT
F804 C3 00F8      JMP START
F807              END
 *** RESTART ? (JA/NEIN)
```

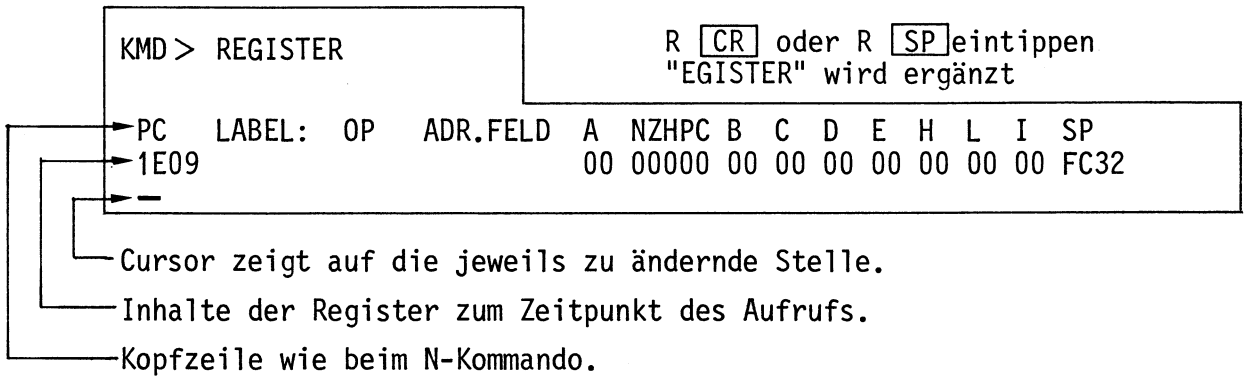
- Formatiertes Assemblerprogramm

```
KMD > ASSEMBLER
 START-ADR =F800
F800              LC
F800              EPORT EQU 12
F800              APORT EQU 13
F800 DB 12        START:  IN EPORT
F802 D3 13                OUT APORT
F804 C3 00F8              JMP START
F807              END
 *** RESTART ? (JA/NEIN)
```

R-Kommando R1

Mit dem Register-Kommando können die Inhalte der CPU-Register angezeigt und vor dem Start eines Anwender-Programms vorbelegt werden.

Aufruf und Handhabung:



Zur Kommando-Ausführung:

- Es können nur die Inhalte der Register, auf die der Cursor zeigt, geändert werden.
- Korrekturen eingegebener Werte sind mit der [DEL] -Taste möglich, sofern noch nicht mit [SP] abgeschlossen wurde.
- Mit der [SP]-Taste kann der Cursor von Register zu Register bewegt werden. Mit ihr beendet man auch die Änderung eines Registerinhaltes.
- Es lassen sich nur Hex-Werte in die Register eingeben; nur die Stelle, unter der sich der Cursor befindet, kann geändert werden.
- In die einzelnen Bits des Status-Registers lassen sich nur Binärwerte (0,1) eingeben.
- Mit der [CR]-Taste beendet man alle Eingaben, das nächste Kommando wird erwartet.

Die Ausgedruckten bzw. eingegebenen Registerinhalte verbleiben zunächst im Schreib-Lese-Speicher. Diese Speicherstellen nennt man auch "Schattenregister". Bevor ein Anwender-Programm gestartet wird, werden die Inhalte der Schattenregister durch das Betriebsprogramm in die CPU-Register geladen.

Beim Experimentieren mit dem R-Kommando sollte man den Inhalt des "Stack-Pointers" (SP) nicht verändern!

● Wenn sich ein Anwender-Programm nicht starten läßt, obwohl es richtig eingegeben wurde (Prüfen z.B. mit dem D-Kommando), hat das Betriebsprogramm durch vorhergegangene Bedienungsfehler den Inhalt von SP geändert (meist um FC80H). Sehen Sie sich mit dem R-Kommando diesen Inhalt an und stellen Sie ihn gegebenenfalls wieder auf FC32.
Danach läßt sich das Anwenderprogramm starten.

MAT 85

Name:

R-Kommando

Datum:

R2

Rufen Sie das R-Kommando auf.
Tragen Sie die angezeigten Registerinhalte in die erste Zeile der folgenden Tabelle ein. Ändern Sie dann die Inhalte der Register in die Werte, die in der zweiten Zeile der Tabelle angegeben sind (keine Angabe ≙ keine Änderung).

	KMD > REGISTER											
	PC	LABEL: OP ADR.FELD	A	NZHPC	B	C	D	E	H	L	I	SP
1.												
2.	F800		55	10101		AA		FF	2A		60	FC30

Nun soll überprüft werden, ob die Register die neuen Werte enthalten.

Rufen Sie hierzu das R-Kommando erneut auf.
Tragen Sie die angezeigten Registerinhalte in die Tabelle ein und vergleichen Sie diese Anzeige mit der 1. Zeile der obigen Tabelle.

KMD > REGISTER											
PC	LABEL: OP ADR.FELD	A	NZHPC	B	C	D	E	H	L	I	SP

B-Kommando B1

Mit dem Breakpoint-Kommando (Breakpoint = Haltepunkt) wird das Einsetzen von Haltepunkten in Anwender-Programme freigegeben bzw. gesperrt.
Dieses Kommando ermöglicht es, bestimmte Programmteile (z.B. Zeitschleifen) in Echtzeit durchlaufen zu lassen und ab dem Haltepunkt die Programmausführung mit dem N-Kommando schrittweise zu beochachten. Die Eingabe der Haltepunkt-Adressen erfolgt innerhalb der Ausführung des Go-Kommandos nach der Eingabe der Programm-Startadresse.

Aufruf und Handhabung:

```
KMD > BREAKPOINT

 BREAK-ADR1=X1X1
 BREAK-ADR2=X2X2
 BREAK-ADR3=X3X3
 BREAK-ADR4=X4X4

 EIN/AUS  =X Y
```

B [CR] oder B [SP] eintippen
"REAKPOINT" wird ergänzt

X1X1 = Breakadresse 1 ⎫ Änderungen nur
⋮ ⎬ unter Go-
X4X4 = Breakadresse 4 ⎭ Kommando möglich

X = Vorgabe; Ein: Y = E [CR] oder E [SP]
Aus: Y = A [CR] oder A [SP]
Unverändert: Y = [CR] oder [SP]

Zur Kommando-Ausführung:

Das Breakpoint-Kommando wird unter dem Go-Kommando ausgeführt, wenn...

- Breakpoints eingeschaltet sind und
- Breakadressen nicht alle 0 sind und
- das Anwenderprogramm eine Adresse erreicht, auf die ein Breakpoint gesetzt ist. Diese Adresse muß auf ein Befehlsbyte zeigen.

Übung:

Rufen Sie das B-Kommando auf und schalten Sie die Breakpoint-Routine ein bzw. aus.

Einsetzen der Break-Adressen:

Die Break-Adressen werden bei eingeschaltetem "Breakpoint" unter dem Go-Kommando wie folgt eingesetzt:

```
KMD > GO

 START-ADR =XXXX YYYY

 BREAK-ADR1=X1X1 Y1Y1
 BREAK-ADR2=X2X2 Y2Y2
 BREAK-ADR3=X3X3 Y3Y3
 BREAK-ADR4=X4X4 Y4Y4
```

G [CR] oder G [SP] eintippen
"0" wird ergänzt

XXXX = Vorgabe; Neu: YYYY [CR] oder [SP]
Vorgabe: [CR] oder [SP]

X1X1 = Vorgabe; Neu: Y1Y1 [SP] *)
Vorgabe: [SP]

X4X4 = Vorgabe; Neu: Y4Y4 [CR] oder [SP]
Vorgabe: [CR] oder [SP]

*) Wenn alle vier Break-Adressen gesetzt werden sollen, bei ADR1 - ADR3 mit [SP] abschließen!
Wenn weniger als vier Break-Adressen gesetzt werden sollen, jeweils mit [CR] abschließen!

Breakpoint-Ausführung:

Wenn die CPU bei der Ausführung des Anwender-Programms eine Break-Adresse erreicht hat, wird die weitere Programmausführung gestoppt und in das Betriebsprogramm zurückgesprungen. Das Betriebsprogramm meldet sich mit folgendem Ausdruck (Beispiel):

```
*** BREAKPOINT ***
PC  LABEL:   OP   ADR.FELD  A  NZHPC B  C  D  E  H  L  I  SP
                            87 00000 00 00 00 00 F8 2A 80 FC34
F802         OUT  02
KMD >_
```

└ Befehl, auf den die Break-Adresse zeigt.
Dieser Befehl ist noch nicht ausgeführt worden!

└ Break-Adresse

Der Rücksprung in das Betriebsprogramm nach dem Erreichen einer Haltepunkt-Adresse erfolgt dadurch, daß das Betriebsprogramm vor der Ausführung des Go-Kommandos einen Rücksprungbefehl (RST 4) in das Anwenderprogramm einbaut. Dazu wird das ursprünglich vorhandene Befehlsbyte aus dem Anwenderprogramm im RAM zwischengespeichert und nach dem Erreichen der Haltepunkt-Adresse wieder eingesetzt. Außerdem werden alle Register-Inhalte der CPU, die beim Erreichen des Haltepunktes vorlagen, in die Schattenregister gerettet.

Bei Neuaufruf des Go-Kommandos wird die jeweils letzte Break-Adresse als Go-Start-Adresse vorgegeben. Jeweils nach Abschluß mit [SP] *) werden dann der Reihe nach wieder alle vorgewählten Break-Adressen angezeigt, ehe das Programm bis zur nächsten Break-Adresse abgearbeitet wird.

*) Um die Übersicht zu behalten, sollte man nach dem Aufruf des Go-Kommandos immer alle Break-Adressen mit [SP] aufrufen. So hat man stets die Anfangsadressen der bereits untersuchten Programmteile und die der noch zu untersuchenden vor Augen.

Fehlermeldungen:

Wenn im Anwenderprogramm ein RST4-Befehl (E7H) gefunden wird, erfolgt der Registerausdruck mit der Überschrift *** BREAKPOINT ERROR ***.

MAT 85

Name:

Fehlersuche mit B und N.

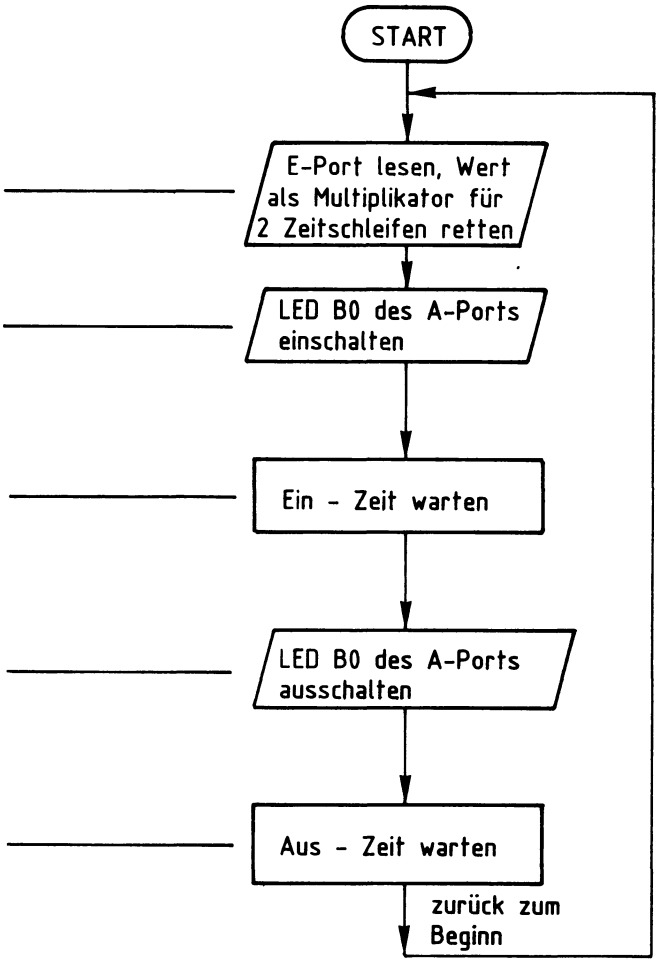
Datum:

B4

Anhand des folgenden Programms wird der Einsatz des B-Kommandos in Verbindung mit dem N-Kommando demonstriert.
Das Programm hat die Aufgabe, die obere LED B0 des Ausgabe-Ports (Adresse 13H) periodisch blinken zu lassen. Die Periodendauer soll mit der Schalterstellung der Schalter des Eingabe-Ports (Adresse 12H) veränderbar sein (Rechteckgenerator).

1. Geben Sie folgendes (zunächst fehlerhafte) Programm ab Adresse F800 in den Speicher ein.

Adresse	Op- Code	Operand
F800	IN	12
F802	MOV	B,A
F803	MOV	D,A
F804	MVI	A,01
F806	OUT	13
F808	MVI	C,0FF
F80A	DCR	C
F80B	JNZ	0F80A
F80E	DCR	B
F80F	JNZ	0F808
F812	MVI	A,00
F814	OUT	13
F816	MVI	C,0FF
F818	DCR	C
F819	JNZ	0F816
F81C	DCR	D
F81D	JNZ	0F818
F820	JMP	0F800



2. Stellen Sie alle Schalter des E-Ports auf L-Pegel (LED's aus) und starten Sie das Programm bei ausgeschalteten "Breakpoints".

   Wirkung: LED B0 des Ausgabeports leuchtet kurz auf und erlischt wieder.
   Ein Rücksprung ins Betriebsprogramm erfolgt nicht.

MAT 85 **Name:**

Fehlersuche mit B und N. **Datum:**

B5

Das Programm arbeitet in einer Schleife, die es wegen eines logischen Fehlers nicht mehr verlassen kann.

Anweisung	Wirkung/Kommentar
RESET betätigen	Rückkehr ins Betriebsprogramm
BREAKPOINT einschalten	In der Folge wird untersucht, ob die einzelnen Programmteile, die auch durch die Blöcke im Flußdiagramm dargestellt sind, funktionieren.
Mit R alle Register auf 0; Am E-Port 08 einstellen; Go aufrufen; START-ADR ⟶ F800 BREAK-ADR1 ⟶ F808 BREAK-ADR2...4 ⟶ 0000	} Startbedingungen Programmbeginn Beginn der Zeitschleife für die Ein-Zeit. Es wird nur mit einem Haltepunkt gearbeitet (übersichtlich). *** BREAKPOINT *** PC OP ADR.Feld A B D 01 08 08 F808 MVI C,FF LED B0 leuchtet Das am E-Port mit den Schaltern eingestellte Datum 08 wurde in die Register B und D geladen. Der Wert 01 wurde in den Akku geladen und zum A-Port ausgegeben.
Go aufrufen; START-ADR ⟶ F800 BREAK-ADR1 ⟶ F812 BREAD-ADR2...4 ⟶ 0000	Programmbeginn Beginn des Programmteils "LED B0 ausschalten". *** BREAKPOINT *** PC OP ADR.FELD A B C D 01 00 00 08 F812 MVI A,00 LED B0 leuchtet Die Ein-Zeit-Schleife wurde in Echtzeit durchlaufen (N-Kommando wäre hierzu nicht geeignet). Durch die DCR-Befehle in der Schleife enthalten die Register B und C den Wert Null.

MAT 85

Name:

Fehlersuche mit B und N.

Datum:

B6

Anweisung	Wirkung/Kommentar
Go aufrufen START-ADR ⟶ F812 BREAK-ADR1 ⟶ F816 BREAK-ADR2...4 ⟶ 0000	ab hier soll weitergeprüft werden. Beginn der Zeitschleife für die Aus-Zeit. *** BREAKPOINT *** PC OP ADR.FELD A B D 00 00 08 F816 MVI C,FF LED B0 dunkel Der Wert 00 wurde in den Akku geladen und zum A-Port ausgegeben.
Go aufrufen START-ADR ⟶ F814 BREAK-ADR1 ⟶ F820 BREAK-ADR2...4 ⟶ 0000	Die Startadresse muß außerhalb der zu prüfenden Zeitschleife liegen. Läge sie innerhalb, würde die Programmabarbeitung nach jedem Dekrementieren angehalten. Ende außerhalb der Schleife. Haltepunkt F820 wird nicht erreicht. Das Programm "hängt fest". Der Fehler liegt in der Aus-Zeit-Schleife.
N aufrufen; START-ADR ⟶ F816 STEPS ⟶ 10	PC OP ADR.FELD C D F816 MVI C,FF FF 08 F818 DCR C FE 08 F819 JNZ F816 FE 08 F816 MVI C,FF FF 08 F818 DCR C FE 08 F819 JNZ F816 FE 08 F816 MVI C,FF FF 08 Hier liegt der Fehler! Der Sprung nach dem Dekrementieren von Register C (wenn C≠0) führt wieder nach F816. Dort aber wird Register C wieder mit FF geladen. Diese Schleife wird nicht mehr verlassen.

MAT 85

Name:

Fehlersuche mit B und N.

Datum:

B7

Anweisung	Wirkung/Kommentar
JNZ F816 ändern in JNZ F818 Go aufrufen; START-ADR ⟶ F814 BREAK-ADR1 ⟶ F820 BREAK-ADR2...4 ⟶ 0000	mit dem M- oder A-Kommando Beginn vor die Aus-Zeit-Schleife gelegt. Ende der Zeitschleife *** BREAKPOINT *** PC OP ADR.FELD A B C D 00 00 00 00 F820 JMP F800 Die Schleife ist in Echtzeit durchlaufen worden. Die Inhalte der Register C und D sind auf Null herabgezählt worden.
BREAKPOINT ausschalten Programm mit G starten Verändern Sie die Stellung der Schalter des E-Ports.	Das Programm arbeitet richtig. Die Blinkfrequenz verändert sich.
Bauen Sie eigene Fehler in das Programm ein und versuchen Sie, diese nach der gezeigten Methode zu finden. Setzen Sie später auch mehrere Breakpoints. Setzen Sie auch Breakpoints innerhalb einer Zeitschleife.	

T-Kommando **T1**

Mit dem Trace-Interval-Kommando wird ...

- die Protokollierung der Registerinhalte für einen gewünschten Programmabschnitt eines Anwenderprogramms ein- oder ausgeschaltet und
- Start- und Stop-Adresse dieses Programmsabschnittes eingegeben.

Aufruf und Handhabung:

KMD> TRACE INTERVAL	T [CR] oder T [SP] eintippen "RACE INTERVAL" wird ergänzt
EIN/AUS =X Y	X = Vorgabe; Ein: Y = E [CR] oder E [SP] Aus: Y = A [CR] oder A [SP] Unverändert: Y = [CR] oder [SP]
START-ADR =X1X1 Y1Y1	X1X1 = Vorgabe; Neu: Y1Y1 [CR] oder [SP] Alt: [CR] oder [SP]
STOP -ADR =X2X2 Y2Y2	X2X2 = Vorgabe; Neu: Y2Y2 [CR] oder [SP] Alt: [CR] oder [SP]
	Y1Y1 und Y2Y2 müssen auf ein Befehlsbyte zeigen

Zur Kommando-Ausführung:

Nach dem Einschalten des "Trace-Interval" und Aufruf des G-Kommandos wird das Anwenderprogramm unter der Kontrolle des Tracers Befehl für Befehl durchlaufen (verlängerte Bearbeitungszeit!).

- Solange sich der Tracer außerhalb des angegebenen Trace-Intervalls aufhält, wird kein Bildschirmausdruck sichtbar.
- Gerät der Tracer in das angegebene Intervall (einschließlich Start- und Stop-Adresse), so werden nach Ausführung jedes Befehls in diesem Bereich alle Registerinhalte ausgedruckt.
- Breakpoints können ohne Einschränkung im gesamten Adreßbereich weiter verwendet werden.
- Im Trace-Betrieb werden der Halt-Befehl (76H) und illegale OP-Codes erkannt und führen zum Programmabbruch (ohne Trace-Betriebsart wird ein Halt-Befehl ohne eine Meldung ausgeführt).

MAT 85

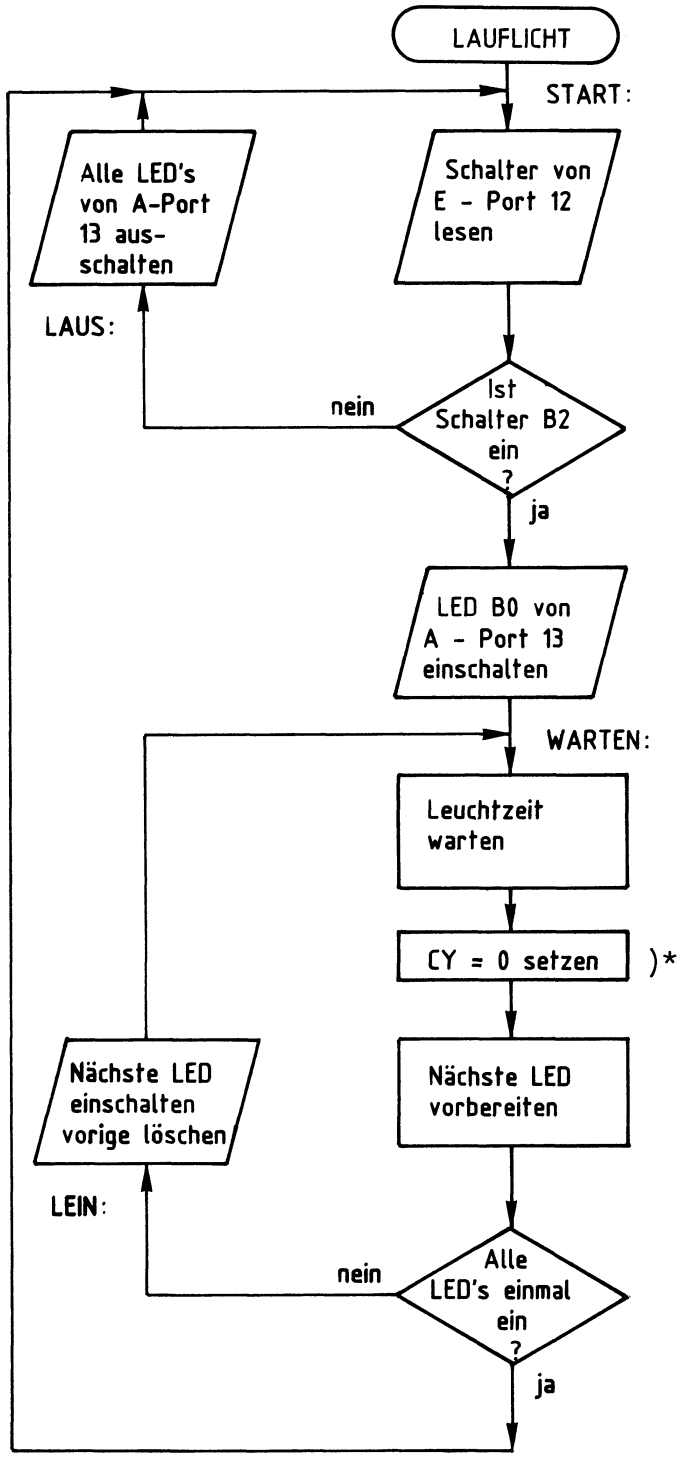
Name:

Übung T-Kommando

Datum:

T2

Laden Sie das folgende "Lauflicht-Programm"
ab Adresse F800 in den Speicher.
Überprüfen Sie Ihre Eingabe mit dem D-Kommando.



Adresse	Label	Op-Code	
F800	START:	IN	12
F802		ANI	04
F804		JZ	LAUS
F807		NOP	
F808		MVI	A,01
F80A		OUT	13
F80C	WARTEN:	MVI	B,33
F80E	ZA:	MVI	C,99
F810	ZI:	DCR	C
F811		JNZ	ZI
F814		DCR	B
F815		JNZ	ZA
F818		ANA	A
F819		RAL	
F81A		CPI	00
F81C		JNZ	LEIN
F81F		JMP	START
F822	LAUS:	MVI	A,00
F824		OUT	13
F826		JMP	START
F829	LEIN:	OUT	13
F82B		JMP	WARTEN
F82E	END		

MAT 85 **Name:**

Übung T-Kommando **Datum:**

T3

)* Der RAL-Befehl schiebt den Akku-Inhalt (hier 01) über das Carry-Bit um eine Bitstelle nach links. Sollte das Carry-Bit zufällig 1 sein, würde diese 1 in Bit 0 des Akkus geschoben, was nicht erwünscht ist. Daher wird hier vor RAL mit ANA A das Carry-Bit auf 0 gesetzt.

Anweisung	Wirkung/Kommentar
Schalter B2 vom E-Port einschalten; Programm mit G bei F800 starten	Die CPU arbeitet das Anwender-Programm in "Echtzeit" ab . Die LED's am A-Port werden in schneller Folge von oben nach unten ein- und ausgeschaltet. Es entsteht der Eindruck eines "Laufenden Lichtes".
RESET betätigen	Die CPU kehrt vom Anwender-Programm zum Betriebsprogramm zurück. Das Betriebsprogramm erwartet ein neues Kommando.
Mit T "Trace Interval" einschalten; START-ADR ⟶ F800 STOP -ADR ⟶ F80A Schalter B2 ⟶ Aus Mit R die Register A,B, C und alle Flags auf 0 setzen; Mit G bei F800 starten;	Trace Interval liegt damit zwischen F800 und F80A. PC LABEL OP ADR.FELD A ..Z.. B C F800 START: IN 12 00 0 00 00 F802 ANI 04 00 1 00 00 F804 JZ LAUS 00 1 00 00 F822 LAUS: MVI A,00 ├──── Kopfzeile ────┤ Text wie oben, außer Zeile F800 ├──── Kopfzeile ────┤ Text wie vorher ==> SPACE Alle Programmteile, die im vorgegebenen Trace Intervall (Fenster) liegen, werden ausgeführt und ausgedruckt. Weil SB2 auf "AUS" steht, erfolgt ein Sprung zum Programmteil "LAUS". Da dieser Teil außerhalb des vorgegebenen "Fensters" liegt, wird die Zeile F822 nicht vollständig ausgedruckt.

T4

Anweisung	Wirkung/Kommentar
	Am Ende dieses Programmteils (bei Adr. F826) erfolgt der Rücksprung nach "START". Da der Tracer nun wieder innerhalb des "Fensters" arbeitet, werden die darin liegenden Programmteile erneut ausgedruckt. "==>SPACE" signalisiert, daß die weitere Programmbearbeitung durch den Tracer erst nach Betätigung der SPACE-Taste erfolgt. Bei dieser Meldung kann die Programmbearbeitung aber auch durch Betätigen von [ESC] abgebrochen werden.
Schalter B2 ⟶ Ein [SP] betätigen	PC LABEL OP ADR.Feld A ..Z.. B C F800 START: IN 12 04 1 00 00 F802 ANI 04 04 0 00 00 F804 JZ LAUS 04 0 00 00 F807 NOP 04 0 00 00 F808 MVI A,01 01 0 00 00 F80A OUT 13 01 0 00 00 F80C WARTEN: MVI B,33 – (Die LED's werden der Reihe nach ein- und nach ca. 11s ausgeschaltet. Nach ca. 90s erscheint:) Kopfzeile Protokollanfang wie oben = = > SPACE Weil SB2 eingeschaltet wurde, wird der Sprung zum Programmteil "LAUS" ignoriert (siehe Z-Flag) und bei F807 weitergemacht. Nach Bearbeitung der Zeile F80A verläßt der Tracer das "Fenster" und tritt in die Schleife ein, die mit "WARTEN:" beginnt. Innerhalb dieser Schleife müssen erst alle LED's einmal ein- bzw. ausgeschaltet worden sein, ehe der Tracer wieder in den vorgegebenen "Fensterbereich" eintritt (Sprung bei F81F). Da er die Befehle nicht in Echtzeit bearbeitet, erfolgt eine erneute Protokollierung erst nach ca. 90s. (Diese Zeit hängt vom Inhalt der Register B und C ab und vom Traceprogramm).

MAT 85 **Name:**

Übung T-Kommando **Datum:**

T5

Anweisung	Wirkung/Kommentar
Legen Sie folgende Intervall-Adressen fest: START-ADR⟶F819 STOP -ADR⟶F81F	Das Protokoll zeigt Ihnen, unter welchen Bedingungen die Sprünge nach "LEIN" und "START" erfolgen.
Legen Sie folgende Intervall-Adressen fest: START-ADR⟶F80C STOP -ADR⟶F818	Das Protokoll zeigt Ihnen, wie die Warteschleife abgearbeitet wird. Wenn die Bearbeitungszeit verkürzt werden soll, müssen die Register B und C mit kleineren Zahlenwerten geladen werden. Hierzu müssen Sie das Programm entsprechend ändern.
Fügen Sie anstelle des NOP-Befehls einen HLT-Befehl ein. Starten Sie das Programm einmal bei ausgeschaltetem und bei eingeschaltetem Tracer.	Tracer aus: Der HLT-Befehl wird ausgeführt, ohne daß eine Meldung erfolgt. (Dabei sendet die CPU die nächst höhere Adresse F808 auf den Adreßbus und schaltet die Daten- und Steuersignale in den hochohmigen Zustand. Rückkehr ins Betriebsprogramm ist nur durch RESET möglich. Tracer ein: Es erfolgt Programmabbruch und die Meldung *** HALT ODER ILLEGALER OPCODE ***
Schalten Sie nach den Experimenten den Tracer aus.	

I-Kommando

Mit dem In-Kommando (nicht zu verwechseln mit dem 8085-Befehl IN) können Daten von Eingabe-Baugruppen gelesen werden.

Aufruf und Handhabung:

```
KMD > IN

 PORT-NR=X1 Y1



 DATEN  =X2 Y2
```

I [CR] oder I [SP] eintippen
"N" wird ergänzt

X1 = Vorgabe; Vorgabe - 1: [-] eintippen
Vorgabe + 1: [+] eintippen
Neu: Y1 [CR] oder [SP]
Vorgabe: [CR] oder [SP]

X2 = Gelesener Wert (Hex)
Y2 = [CR] : Fertig, nächstes Kommando
Y2 = [SP] : Nochmal lesen
Y2 = [+/-] : Neue Port-Adresse

Übung:

Lesen Sie die Daten Ihres Eingabe-Ports.
Ändern Sie die Bitkombination mit Hilfe der Schalter und lesen Sie erneut.
Versuchen Sie auch Daten von Ports zu lesen, die gar nicht existieren.

O-Kommando

O

Mit dem Out-Kommando (nicht zu verwechseln mit dem 8085-Befehl OUT) lassen sich Daten zu Ausgabe-Baugruppen übertragen.

Aufruf und Handhabung:

```
KMD > OUT

 PORT-NR=X1 Y1



 DATEN  =X2 Y2
```

O [CR] oder O [SP] eintippen
"UT" wird ergänzt

X1 = Vorgabe; Vorgabe - 1: [-] eintippen
Vorgabe + 1: [+] eintippen
Neu: Y1 [CR] oder [SP]
Vorgabe: [CR] oder [SP]

X2 = Vorgabe; Neu: Y2 [CR] (Fertig)
Y2 [SP] (Nochmal)
Vorgabe: [SP]

Y2 = [CR] : Fertig (auch [ESC])
Y2 = [+/-] : Neue Port-Adresse
Y2 = [SP] : Nochmal schreiben

Übung:
Schreiben Sie der Reihe nach die Datenbytes
01 - 10 - 00 - FF - 55 - AA in Ihr Ausgabe-Port.
Versuchen Sie auch Daten in ein Port zu schreiben, das gar nicht existiert.

S-Kommando S

Mit dem Save-Kommando können Programme und Daten über einen Kassetten-Recorder auf Magnetband gespeichert werden. Die Verwendung des S-Kommandos erfordert zusätzlich die Baugruppe "Kassetten-Interface BFZ/MFA 4.4".

Aufruf und Handhabung:

```
KMD> SAVE

 START-ADR =X1X1 Y1Y1

 STOP -ADR =X2X2 Y2Y2

 BAND EINSCHALTEN, DANN SPACE
 (Kommando-Ausführung)
KMD>_
```

S [CR] oder S [SP] eintippen
"AVE" wird ergänzt

X1X1 = Vorgabe; Neu: Y1Y1 [CR] oder [SP]
Vorgabe: [CR] oder [SP]

X2X2 = Vorgabe; Neu: Y2Y2 [CR] oder [SP]
Vorgabe: [CR] oder [SP]

Reihenfolge beachten!
Vorspann am Anfang
der Bänder kann nicht
beschrieben werden!

Zur Kommando-Ausführung:

- Die Startadresse (X1X1 oder Y1Y1) wird für den späteren Ladevorgang mit auf das Band übertragen.
- Zur Erkennung von Lesefehlern werden mit den Daten, die als ASCII-Zeichen übertragen werden, auch Prüfsummenbytes übertragen.
- Soll die Kommando-Ausführung abgebrochen werden, ist die RESET-Taste auf der CPU-Baugruppe zu betätigen.

L-Kommando

L

Mit dem Load-Kommando werden Daten vom Kassetten-Recorder in den Speicher zurückgelesen. Sollen die Daten nicht in den beim Save-Kommando angegebenen Speicherbereich übertragen werden, so kann eine neue Startadresse angegeben werden.

Aufruf und Handhabung:

```
KMD> LOAD TAPE

 START-ADR =YYYY




 SPACE, DANN BAND EINSCHALTEN
 (Kommando-Ausführung)
KMD>_
```

L [CR] oder L [SP] eintippen
"OAD TAPE wird ergänzt

YYYY = 0: Startadresse, die auf der Kassette gespeichert ist, wird genommen.
Neu: YYYY [CR] oder YYYY [SP] eintippen.

Reihenfolge beachten!

Zur Kommando-Ausführung:

- Die Daten vom Band werden eingelesen, eine Kontrollsummenbildung findet dabei statt.
- Ausdruck bei fehlerfreiem Empfang: [READY]
- Ausdruck bei fehlerbehaftetem Empfang: [CHECKSUM ERROR]
- Ausdruck, wenn ein empfangenes Zeichen nicht den Hex-Zeichen 0 bis F im ASCII-Code entspricht (z.B. 0≙30, F≙46): [*** NICHT HEX = XX],wobei die empfangene Bitkombination XX (z.B. 0A≙LF) ausgegeben wird.
- Abbruch des L-Kommandos ist nur mit RESET möglich.

1. Anschluß einer Datensichtstation

1.1. Bedingungen für eine fehlerfreie Datenübertragung

Um eine fehlerfreie Datenübertragung zwischen dem MFA-Mikrocomputer und der Datensichtstation zu gewährleisten, sind folgende Punkte zu beachten:

1. Arbeiten beide Geräte mit den gleichen Strom- oder Spannungspegeln?
2. Übertragen beide Geräte die gleiche Anzahl Daten-Bits?
3. Ist in beiden Geräten die Paritätsprüfung ein- oder ausgeschaltet?
4. Wird auf gerade oder ungerade Parität geprüft?
5. Stimmt die Anzahl der Stop-Bits in beiden Geräten überein?
6. Mit welcher Baudrate sendet die Datensichtstation?

Informationen hierzu für den MFA-Mikrocomputer:

Zu 1.: Es sind möglich eine 20-mA-Stromschnittstelle und eine V-24-Spannungsschnittstelle. Beide müssen zusätzlich verdrahtet werden (siehe Anschlußpläne auf den folgenden Seiten).
Pegel der 20-mA-Stromschnittstelle:
log. 1 = unterbrochener Stromkreis
log. 0 = Strom von 20 mA
Pegel der V-24-Spannungsschnittstelle:
log. 1 = -12 V
log. 0 = +12 V

Zu 2.-5.: Der MFA-Mikrocomputer sendet 1 Start-, 7 Daten-, 1 Paritäts- und 2 Stop-Bits aus. Die gleiche Bitfolge kann er auch empfangen. Die Paritätsbits werden nicht überprüft.

Zu 6.: Nach Einschalten des MFA-Mikrocomputers muß die Space-Taste der Datensichtstation betätigt werden. Aus dem empfangenen Datenwort bestimmt das Betriebsprogramm dann die Übertragungsgeschwindigkeit der Datensichtstation und paßt an diese die eigene Baudrate an.

1.2. Anschlußplan

Die meisten Datensichtgeräte verwenden einen genormten Buchsenstecker mit 25 Anschlüssen (ITT-Cannon DB-25S oder Harting 09 67 025 2704). Die zugehörigen Stiftstecker haben die Bezeichnung DB-25P (Cannon) und 09 67 025 2604 (Harting). Die Steckerbelegung ist genormt, die wichtigsten Anschlüsse zeigt Bild 7.

Pin-Nr.	Funktion	Bemerkungen
1	Gehäusemasse	–
2	Transmit Data (TxD)	Sendeleitung
3	Receive Data (RxD)	Empfangsleitung
4	Request to Send (RTS)	Sendeaufforderung
5	Clear to Send (CTS)	Sender betriebsbereit log. 1 = Sender freigeben
6	Data Set Ready (DSR)	Empfänger bereit?
7	Signalmasse	–
8	Data Carrier detect (DCD)	Signal vorhanden
20	Data Terminal Ready (DTR)	Empfänger bereit

Bild 7: Anschlußbelegung V-24-Norm

In Bild 8 ist gezeigt, wie nun die Verbindungen zwischen einer Datensichtstation und dem MFA-Mikrocomputer herzustellen sind.

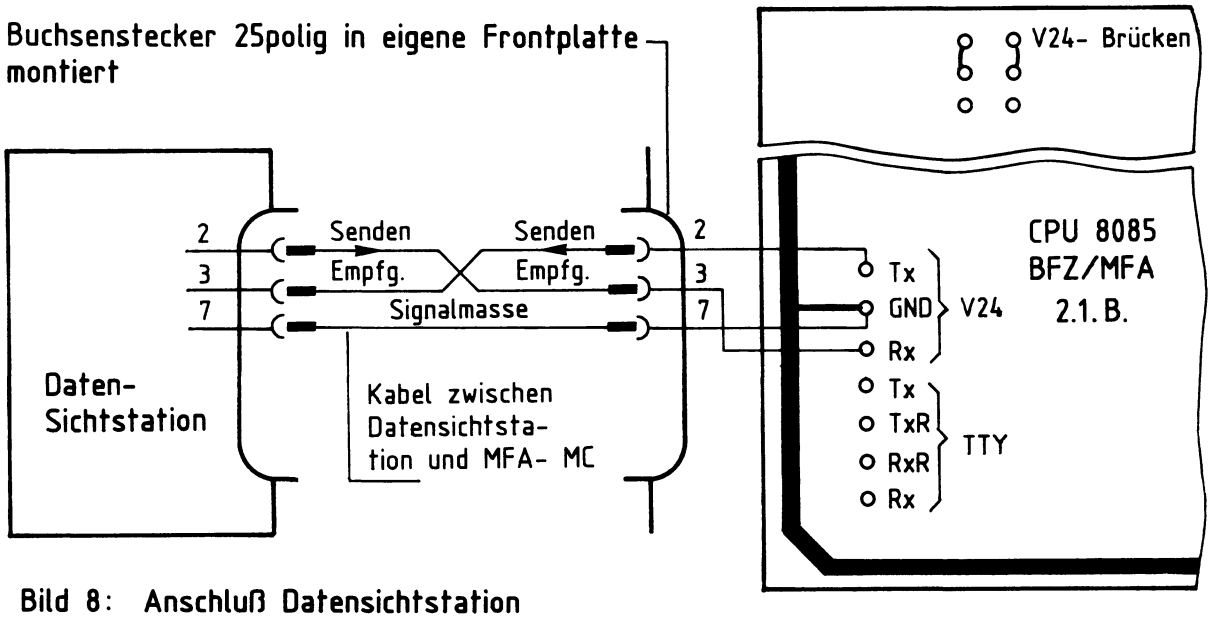


Bild 8: Anschluß Datensichtstation

2. Druckermodus, TTY-Betrieb

Das Betriebsprogramm unterscheidet je nach gemessener Übertragungsgeschwindigkeit zwischen einem Bildschirm- und einem Drucker-Modus. Der Drucker-Modus stellt sich immer dann ein, wenn die gemessene Übertragungsgeschwindigkeit gleich oder kleiner 300 Baud ist.

Im Gegensatz zum Bildschirm-Modus können bei Betrieb mit einer TTY falsch gedruckte Zeichen nicht gelöscht werden. Bei Betätigung der Taste [DEL] nach einem falsch eingegebenen Zeichen wird ein "/" (Slash-Schrägstrich) ausgedruckt und das falsch an den Mikrocomputer gesendete Zeichen gelöscht. Außerdem erfolgt bei längeren Protokollen kein Zwischenstop.

Bild 9 zeigt, wie ein Fernschreiber an den MFA-Mikrocomputer anzuschließen ist. Auch hier wird die Baudrate wieder vom Betriebsprogramm gemessen, wenn nach dem Einschalten der Geräte die [SP] -Taste betätigt wird.

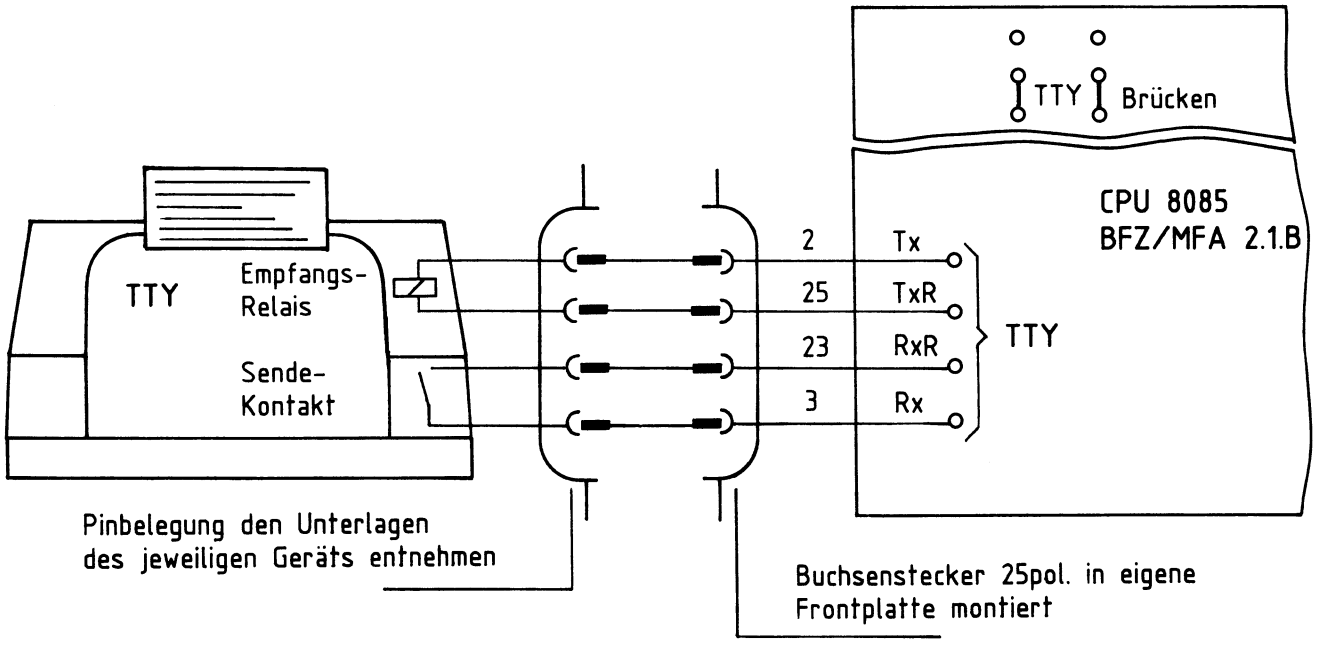


Bild 9: Anschluß TTY

3. Anschluß eines Matrix-Druckers

Matrix-Drucker arbeiten häufig mit paralleler Datenübertragung. Zum Anschluß solcher Drucker gibt es keine Norm. Der Druckerhersteller Centronics verwendete in seinen Druckern erstmals eine Steckerbelegung, die heute allgemeiner Standard ist und als "Centronics-Schnittstelle" bezeichnet wird. Für die parallele Daten-

übertragung zwischen MFA-Mikrocomputer und Drucker wird eine zusätzliche Baugruppe, die "Programmierbare parallele Schnittstelle BFZ/MFA 4.3." (Baugruppe mit Anschlußleitung) erforderlich. An diese Baugruppe wird der Drucker mit einem 25poligen Cannon-Stecker angeschlossen.
Das Betriebsprogramm enthält alle notwendigen Programmschritte, die zum Betrieb des Druckers erforderlich sind.

3.1. Betrieb des Matrix-Druckers

— Einschalten des Druckers
Drucker-Betriebsspannung einschalten!

```
KMD> GO

 START-ADR =XXXX 1E00
*** PRINTER ON ***
```

G [CR] oder G [SP] eintippen
"0" wird ergänzt
XXXX = Vorgabe; Neu: 1E00 [CR] oder [SP]
Meldung auf dem Bildschirm, daß der Drucker eingeschaltet ist. Der Drucker erzeugt zwei Zeilenvorschübe und bringt den Druckkopf in die Ausgangsposition.

— Eingabe eines kleinen Programms mit dem Assembler und Ausdruck.

```
KMD> ASSEMBLER
 START-ADR =0000 F800
F800 DB 12       START:  IN 12
F802 2F                  CMA
F803 D3 13               OUT 13
F805 C3 00F8             JMP START
F808             END
 *** RESTART ? (JA/NEIN) N
```

— Ausschalten des Druckers

```
KMD > GO

 START-ADR =1E09 1E03

*** PRINTER OFF ***
```

G [CR] oder G [SP] eintippen
"0" wird ergänzt
1E09 = Vorgabe; Neu: 1E03 [CR] oder [SP]

Meldung auf dem Bildschirm, daß der Drucker abgeschaltet ist. Auf dem Druckerpapier erscheint diese Meldung nicht!

— Fehlermeldung

```
*** PRINTER NOT READY ***
```

Diese Meldung erscheint auf dem Bildschirm, wenn bei Aufruf des Druckers mit 1E00 die Betriebsspannung nicht eingeschaltet wurde, oder wenn bei eingeschalteter Betriebsspannung kein Papier eingespannt ist.

4. ASCII-Code-Tabelle

Der ASCII-Code (American Standard Code for Information Interchange) ist ein in Amerika entwickelter Code für Fernschreiber, der heute allgemein für Datenübertragungseinrichtungen verwendet wird. Er setzt sich aus den Zeichengruppen Buchstaben, Ziffern, Sonderzeichen und Steuerzeichen zusammen.

Bild 10 zeigt eine Zuordnung der einzelnen Zeichen zu ihren hexadezimalen und binären Signaldarstellungen.

B6 B4 \ B3 B0	0 0000	1 0001	2 0010	3 0011	4 0100	5 0101	6 0110	7 0111	8 1000	9 1001	A 1010	B 1011	C 1100	D 1101	E 1110	F 1111
000	NUL	SOH	STX	ETX	EOT	ENQ	ACK	BEL	BS	HT	LF	VT	FF	CR	SO	SI
1 001	DLE	DC1	DC2	DC3	DC4	NAK	SYN	ETB	CAN	EM	SUB	ESC	FS	GS	RS	VS
2 010	SPC	!	"	#	$	%	&	'	(	)	*	+	,	-	.	/
3 011	0	1	2	3	4	5	6	7	8	9	:	;	<	=	>	?
4 100	@	A	B	C	D	E	F	G	H	I	J	K	L	M	N	O
5 101	P	Q	R	S	T	U	V	W	X	Y	Z	[	\	]	^	_
6 110	`	a	b	c	d	e	f	g	h	i	j	k	l	m	n	o
7 111	p	q	r	s	t	u	v	w	x	y	z	{	|	}	~	DEL

Bild 10: ASCII - Code - Tabelle

Das Code-Wort in hexadezimaler und binärer Darstellung ergibt sich aus Zeilen- und Spaltenwerten wie in folgenden Beispielen gezeigt:

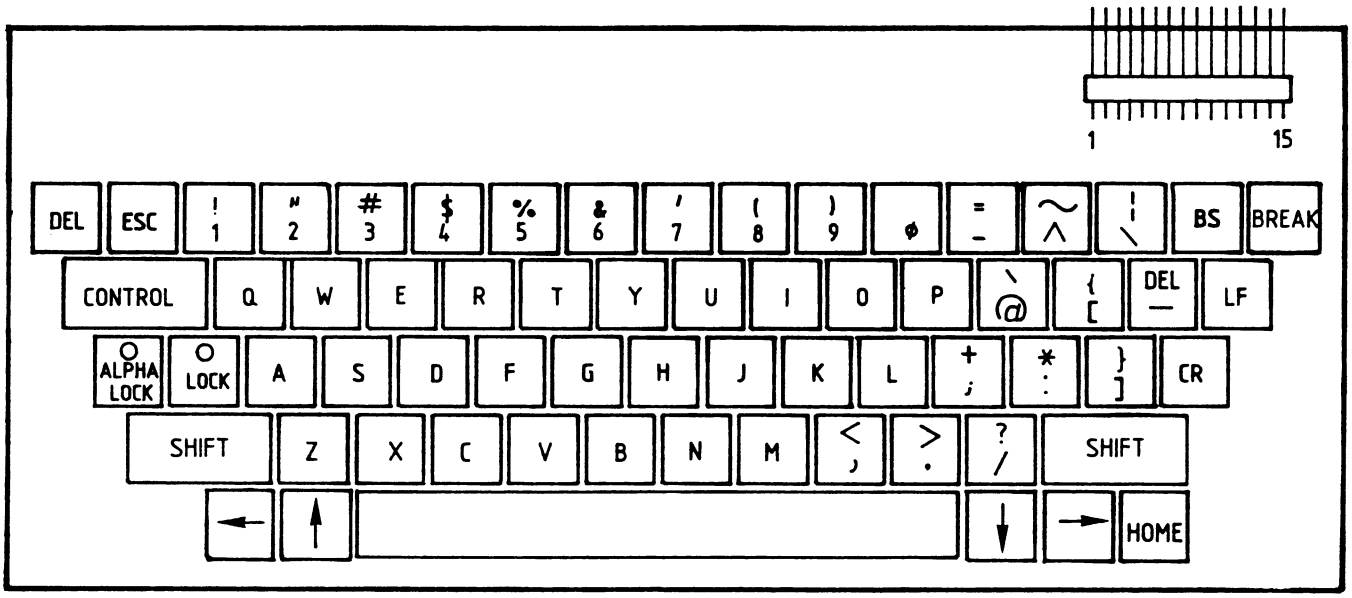
Zeichen	Code in Hex.-Darstellung	Code in Binärdarst.
A	41 (Zeile 4, Spalte 1)	100 0001
e	65 (Zeile 6, Spalte 5)	110 0101
}	7D (Zeile 7, Spalte D)	111 1101
?	3F (Zeile 3, Spalte F)	011 1111
5	35 (Zeile 3, Spalte 5)	011 0101

Die Bedeutung der Steuerzeichen (00H-20H und 7FH) zeigt die folgende Tabelle in Bild 11.

Code Hex	Code ASCII	Bedeutung	Steuerung des Cursors im Video-Interface BFZ/MFA 8.2.
00	NUL	Null, Nichts	
01	SOH	Kopfzellenbeginn	
02	STX	Textanfangszeichen	
03	ETX	Textendezeichen	
04	EOT	Ende der Übertragung	
05	ENQ	Aufforderung zur Datenübertragung	
06	ACK	Positive Rückmeldung	
07	BEL	Klingelzeichen	
08	BS	Rückwärtsschritt, (←)	um 1 Stelle nach links
09	HT	Horizontal Tabulator, (→)	um 1 Stelle nach rechts
0A	LF	Zeilenvorschub, (↓)	um 1 Stelle nach unten
0B	VT	Vertikal Tabulator, (↑)	um 1 Stelle nach oben
0C	FF	Seitenvorschub	Bildschirm löschen und zurück nach links oben (Zeitdauer dazu ca. 150ms)
0D	CR	Wagenrücklauf	zurück zum Zeilenanfang und Löschen des Zeilenendes.
0E	SO	Dauerumschaltzeichen	
0F	SI	Rückschaltungszeichen	
10	DLE	Datenübertragungsumschaltung	
11	DC1	Gerätesteuerzeichen 1	
12	DC2	Gerätesteuerzeichen 2	
13	DC3	Gerätesteuerzeichen 3	
14	DC4	Gerätesteuerzeichen 4	
15	NAK	Negative Rückmeldung	
16	SYN	Synchronisierung	
17	ETB	Ende des Datenübertragungsblocks	
18	CAN	Ungültig	
19	EM	Ende der Aufzeichnung	
1A	SUB	Substitution	
1B	ESC	Umschaltung	um 1 Zeile nach unten ohne Löschen der letzten Zeile
1C	FS	Hauptgruppentrennzeichen	zurück nach links oben
1D	GS	Gruppentrennzeichen	zurück zum Zeilenanfang
1E	RS	Untergruppentrennzeichen	
1F	US	Teilgruppentrennzeichen	
20	SP	Leerzeichen	
7F	DEL	Löschen des vorhergehenden Zeichens	

Bild 11: Bedeutung der Steuerzeichen

5. Die Tastatur Cherry G80-0177



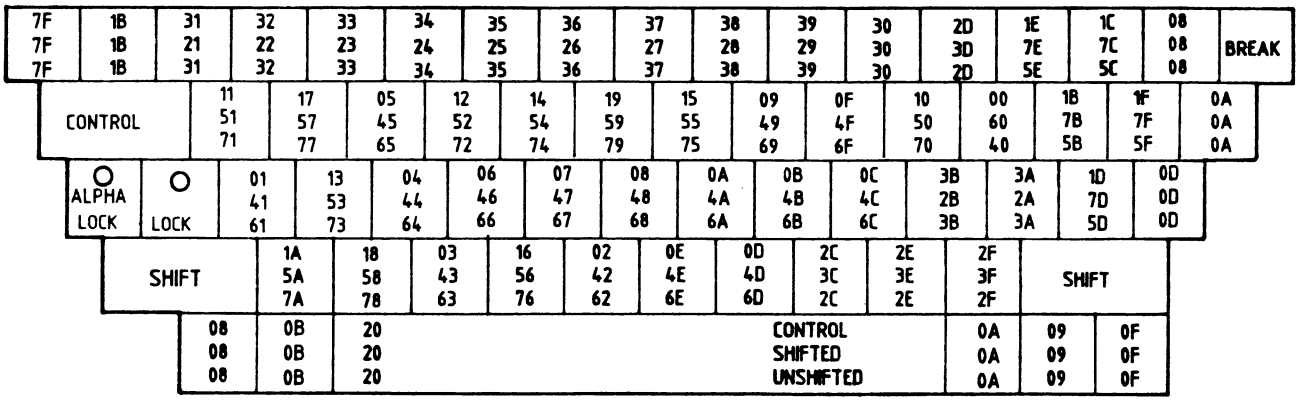


Bild 12: Beschriftung der Tasten und hexadezimale Verschlüsselung der Tastenfunktion

6. Häufig verwendete Symbole für Flußdiagramme (DIN 66001)

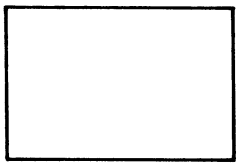
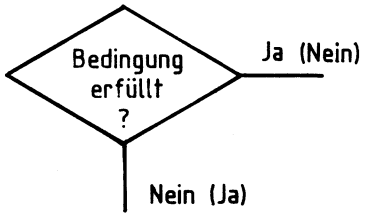
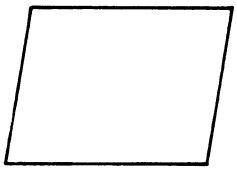
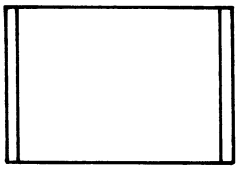
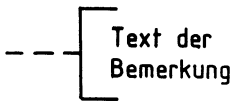
Symbole	Bedeutung
	Grenzstelle (Beginn, Ende ...)
	allgemeine Operation
	Verzweigung
	Ein-/Ausgabe- Operation
	Unterprogramm
	Flußlinie (Richtung der Abarbeitung)
	Zusammenführung
	Bemerkung
	Nahtstelle

Bild 13: Symbole für Flußdiagramme

7. Unterprogramme des Betriebsprogramms

Die in folgender Tabelle aufgeführten Unterprogramme aus dem Betriebsprogramm können Sie in eigenen Programmen verwenden. Wenn Sie die in der Tabelle angegebenen Namen der Unterprogramme in Ihren Programmen mitbenutzen wollen, müssen Sie diese Namen mit Hilfe der EQU-Anweisung vorher den zugehörigen Adressen zuweisen (siehe Beispiele).

Unterpr. Name	Eingangs-Adresse	Veränd. Register	Funktion des Unterprogramms
KMD	0040		Rücksprung in die Kommandoroutine des Monitorprogramms und Ausdruck von KMD>.
RCHAR	0043	A	Liest ein Zeichen von der Tastatur ein. Der ASCII-Code des Zeichens steht im Akku. Bei [ESC] Rückkehr in die Kommandoroutine und Klingeln.
WCHARI	0055		Gibt 1 Zeichen, das nach dem CALL-Befehl im Speicher steht, auf Bildschirm und Drucker (wenn Ein) aus. Das auszugebende Zeichen muß mit der DB-Anweisung in den Speicher geschrieben werden. Beispiel: CALL 0055 DB 'A' ; A wird ausgegeben
WAHEX	0058		Gibt den Akku-Inhalt (8Bit) als zwei Hexadezimalziffern auf dem Bildschirm/Drucker aus.
WHLHEX	005B	H,L	Gibt den HL-Registerinhalt (16Bit) als vier Hexadezimalziffern auf dem Bildschirm/Drucker aus.
WABIN	005E	A	Gibt den Akku-Inhalt (8Bit) als Binärzahl am Bildschirm/Drucker aus. Beispiel: MVI A,23 ; 23 Hexadezimal CALL 005E ; wird als 00100011 ausgegeben
WADEZ	0061	A	Gibt den Akku-Inhalt (8Bit) als Dezimalzahl auf dem Bildschirm/Drucker aus. MVI A,23 ;23 Hexadezimal wird CALL 0061 ;als 35 ausgegeben

Fortsetzung Unterprogramme des Betriebsprogramms

Unterpr. Name	Eingangs-Adresse	Veränd. Register	Funktion des Unterprogramms
WAFOR	0064	A,C	Gibt den Akku-Inhalt (8Bit) in einem der zu wählenden Formate ASCII-Binär-Dezimal-Hex auf dem Bildschirm/Drucker aus. Das Format wird durch den Inhalt des Registers C wie folgt gewählt: 0 → ASCII-Zeichen 1 → Binärzahl 2 → Dezimalzahl 3 → Hexadezimalzahl Beispiel: MVI A,23 ; 23 Hexadezimal wird MVI C,1 ; als 00100011 ausge- CALL 0064 ; geben
WBLANK	0067		Gibt ein Leerzeichen (Blank) auf dem Bildschirm/ Drucker aus.
WBUFI	006D		Gibt den hinter dem CALL-Befehl stehenden Text auf dem Bildschirm aus. Der Text muß mit der DB-Anweisung in den Speicher (Textpuffer) geladen werden. Am Ende des Textes muß als Enderkennung eine 0 stehen. Beispiel: CALL 006D DB ' DIES IST EINE UEBUNG', 0 Gibt den Text DIES IST EINE UEBUNG aus.
WCRLFI	0073		Gibt einen Wagenrücklauf (CR), eine Neue Zeile (LF, Line Feed) und Text in diese neue Zeile aus. Der Text muß wie bei "WBUFI" vorher eingegeben werden.
HADR	08DF		Liest eine 16-Bit-Adresse (4 Hex-Stellen) von der Tastatur ein und speichert sie im Doppelregister H ab. Dabei gelangt der höherwertige Teil der Adresse ins H-Register und der niederwertige Teil ins L-Register. Die Eingabe der Adresse muß mit [CR] oder [SP] abgeschlossen werden.
BSTIME	0895	A,D,E	Zeitverzögerung von 0,24 s. Damit die Inhalte der Register A,D und E vor Aufruf des Unterprogramms gerettet und nachher

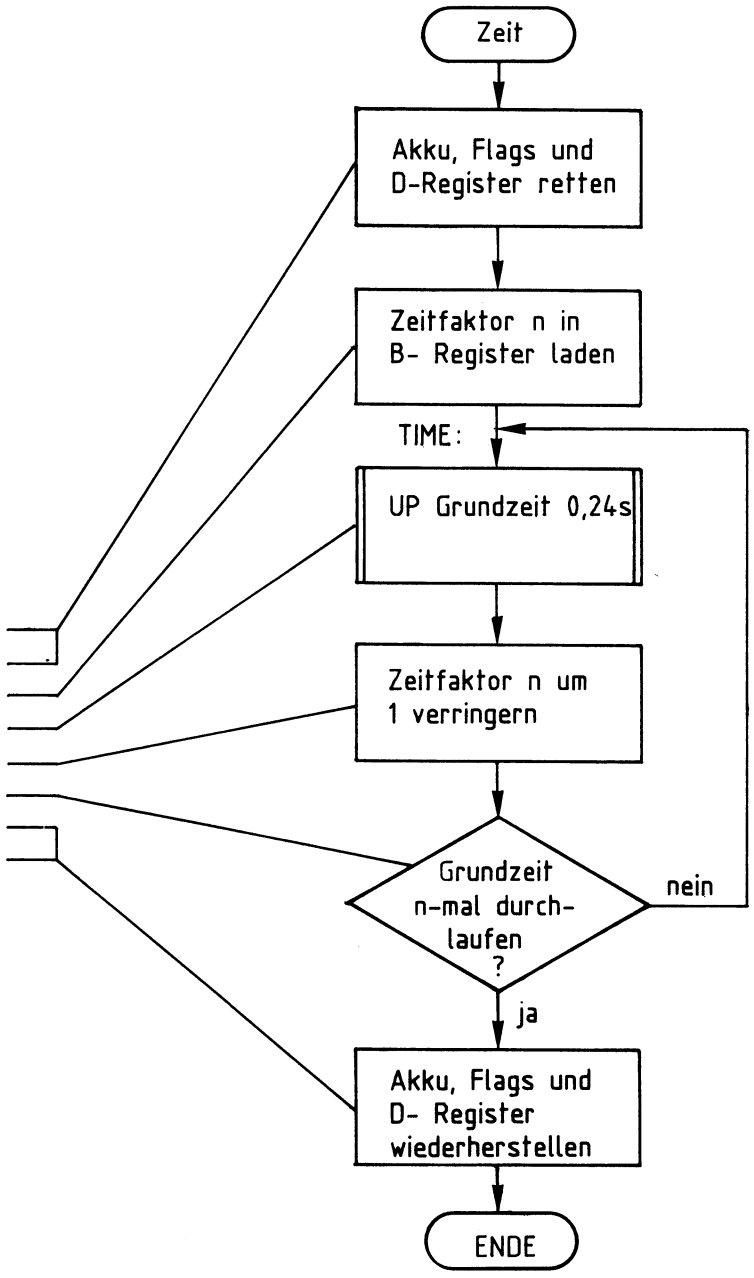
Fortsetzung Unterprogramme des Betriebsprogramms

Unterpr. Name	Eingangs-Adresse	Veränd. Register	Funktion des Unterprogramms
BSTIME			wiederhergestellt werden, muß die folgende Befehlsfolge eingehalten werden: PUSH PSW ;Registerinhalte A,D,E PUSH D ;retten CALL 0895 ;Zeitverzögerung POP D ;Registerinhalte wiederher- POP PSW ;stellen
CMP2	OEA8	A	Vergleicht die Inhalte der Register DE mit denen der Register HL. Wenn (HL)>(DE) ist, wird das Carry-Flag auf 1 gesetzt, sonst auf 0. Die zu vergleichenden Inhalte müssen vor Aufruf des Unterprogramms in die Doppelregister D und H geladen werden. Beispiel: LXI D, Zahl 1 LXI H, Zahl 2 CALL CMP2
SUB2	1039	A,HL, DE	Subtrahiert die 16-Bit-Zahl im Doppelregister D von der 16-Bit-Zahl im Doppelregister H. Das Ergebnis steht dann im Doppelregister H. [(HL) = (HL) - (DE)]
WBUF	OBA1		Gibt Text aus einem Textpuffer aus, dessen Anfangsadresse durch den Inhalt des HL-Registers adressiert ist. Der Text wird mit der DB-Anweisung ab dieser Adresse geladen, das Textende muß mit 0 gekennzeichnet sein. Nach der Ausgabe des Textes zeigt das HL-Register auf die Adresse nach dem Endezeichen.

8. Beispiele für den Gebrauch von Unterprogrammen aus dem Betriebsprogramm

8.1. Verzögerungszeit n x 0,24 s
(Hier wurde n = 10 (0AH)
gewählt)

```
KMD > ASSEMBLER
 START-ADR =F800
F800              LC
F800              BSTIME EQU 0895
F800 F5               PUSH PSW
F801 D5               PUSH D
F802 06 0A            MVI  B,0A
F804 CD 9508 TIME:    CALL BSTIME
F807 05               DCR  B
F808 C2 04F8          JNZ  TIME
F80B D1               POP  D
F80C F1               POP  PSW
```



Das Retten der Register ist nur nötig, wenn die Inhalte dieser Register vor Aufruf des Unterprogramms "BSTIME" Werte enthalten, die nach Abarbeitung des Unterprogramms erst im weiteren Programmverlauf benötigt werden.

8.2. Bildschirm löschen und Textausgabe

```
KMD > ASSEMBLER
 START-ADR =F807 F800
F800              ; BILDSCHIRM LOESCHEN UND
F800              ; CURSOR NACH LINKS OBEN
F800              ; AUSGABE EINES TEXTES
F800              LC
F800              FF EQU 0C  ; FF = SEITENVORSCHUB
F800              WCHARI EQU 55
F800              WCRLFI EQU 73
F800              BSTIME EQU 0895
F800              ;
F800 CD 5500              CALL WCHARI
F803 0C                   DB FF
F804 CD 9508              CALL BSTIME
F807 CD 7300              CALL WCRLFI
F80A 20444945             DB ' DIE WARTEZEIT "BSTIME" IST',0A,0D
F80E 20574152
F812 54455A45
F816 49542022
F81A 42535449
F81E 4D452220
F822 4953540A
F826 0D
F827 20455246             DB ' ERFORDERLICH ,WEIL DAS VIDEO-',0A,0D
F82B 4F524445
F82F 524C4943
F833 48202C57
F837 45494C20
F83B 44415320
F83F 56494445
F843 4F2D0A0D
F847 20494E54             DB ' INTERFACE CA. 150 MS ZUR LOE-',0A,0D
F84B 45524641
F84F 43452043
F853 412E2031
F857 3530204D
F85B 53205A55
F85F 52204C4F
F863 452D0A0D
F867 20534348             DB ' SCHUNG DES BILDSCHIRMS BRAUCHT',00
F86B 554E4720
F86F 44455320
F873 42494C44
F877 53434849
F87B 524D5320
F87F 42524155
F883 43485400
F887 CF                   RST 1
F888              END
```

Wagenrücklauf

Zeilenvorschub

Textende

Der ausgegebene Text:

```
DIE WARTEZEIT "BSTIME" IST
ERFORDERLICH ,WEIL DAS VIDEO-
INTERFACE CA. 150 MS ZUR LOE-
SCHUNG DES BILDSCHIRMS BRAUCHT
```

Wenn man Wagenrücklauf und Zeilenvorschub am Ende der jeweiligen DB-Anweisungen wegläßt, wird jeweils die ganze Bildschirmzeile vollgeschrieben.

8.3. Steuerung des Cursors per Programm

Zur Steuerung des Cursors auf dem Bildschirm muß man dem Video-Interface bestimmte Steuerzeichen senden, die dort als solche erkannt werden und unmittelbar zur Bewegung des Cursors in horizontaler oder vertikaler Richtung führen (siehe Bilder 11 u. 12).
Im folgenden Programmbeispiel soll dies demonstriert werden.
Aufgabe: In die Mitte des leeren Bildschirms soll ein Rechteck mit einer Seitenlänge von 30 Zeichen (-) und einer Höhe von 6 x Zeilenabstand (I) dargestellt werden.
In die Mitte des Rechtecks soll ein Text geschrieben werden; danach soll in die oberste Zeile des Bildschirms die Meldung zur Rückkehr ins Betriebsprogramm ausgegeben werden. Eine solche Rückkehr soll nur mit [ESC] möglich sein.

Bild 14 zeigt das zu programmierende Rechteck und seine Lage innerhalb des Bild schirms.

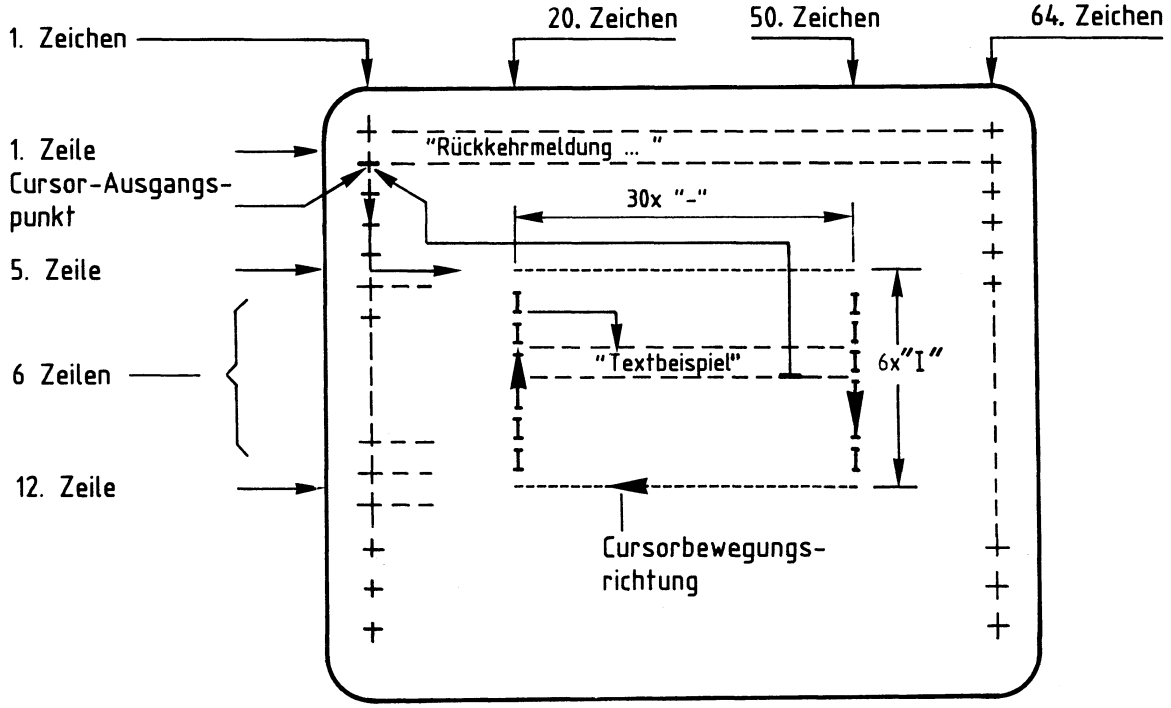


Bild 14: Einteilung des Bildschirms für das Programmierbeispiel

Bild 15 zeigt das Flußdiagramm zu dieser Aufgabe. Beachten Sie die Kommentare zu einzelnen Schritten und die Lösungen dazu im Programm-Listing.

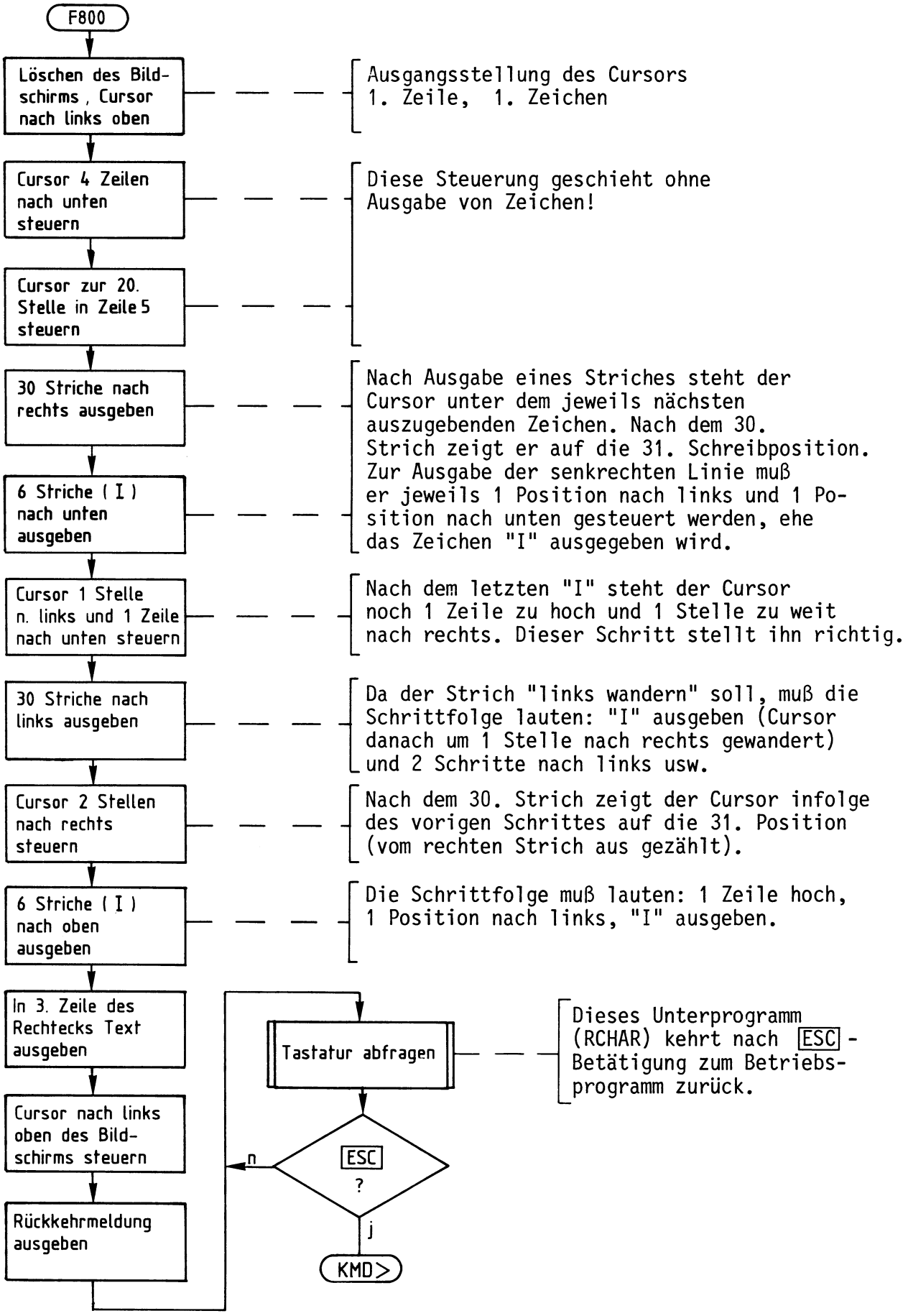


Bild 15: Flußdiagramm "Cursorsteuerung per Programm"

Programm "Cursorsteuerung per Programm".

```
KMD > ASSEMBLER
 START-ADR =F800
F800                ; CURSOR - STEUERUNG
F800                ;
F800                LC
F800                WBUFI EQU 6D
F800                WCHARI EQU 55
F800                BSTIME EQU 0895
F800                RCHAR EQU 43
F800                ;
F800                ;
F800 CD 5500          CALL WCHARI
F803 0C               DB 0C
F804 CD 9508          CALL BSTIME
F807 0E 04            MVI  C,04
F809 CD 5500 LF:      CALL WCHARI
F80C 0A               DB 0A
F80D 0D               DCR  C
F80E C2 09F8          JNZ  LF
F811 0E 13            MVI  C,13
F813 CD 5500 HT:      CALL WCHARI
F816 20               DB 20
F817 0D               DCR  C
F818 C2 13F8          JNZ  HT
F81B 0E 1E            MVI  C,1E
F81D CD 5500 HZ:      CALL WCHARI
F820 2D               DB 2D
F821 0D               DCR  C
F822 C2 1DF8          JNZ  HZ
F825 0E 06            MVI  C,06
F827 CD 6D00 VU:      CALL WBUFI
F82A 080A4900         DB 08,0A,'I',00
F82E 0D               DCR  C
F82F C2 27F8          JNZ  VU
F832 CD 6D00          CALL WBUFI
F835 080A00           DB 08,0A,00
F838 00               NOP
F839 0E 1E            MVI  C,1E
F83B CD 6D00 HL:      CALL WBUFI
F83E 2D080800         DB '-',08,08,00
F842 0D               DCR  C
F843 C2 3BF8          JNZ  HL
F846 CD 6D00          CALL WBUFI
F849 090900           DB 09,09,00
```

Definition der verwendeten Unterprogramme

Zeilenabstand

Bildschirm löschen und Cursor nach links oben

4 Zeilen nach unten

zur 20. Schreibposition nach rechts

30 Striche (Minuszeichen) nach rechts ausgeben

6 Zeichen "I" nach unten ausgeben

1 Pos. nach links und 1 Zeile nach unten steuern.

NOP kann entfallen;
30 Striche nach links ausgeben

Cursor 2 Stellen nach rechts

Fortsetzung des Programms "Cursorsteuerung per Programm".

```
F84C 0E 06          MVI  C,06
F84E CD 6D00 VO:    CALL WBUFI
F851 0B084900       DB 0B,08,'I',00
F855 0D             DCR  C
F856 C2 4EF8        JNZ  VO
```

6 Striche nach oben ausgeben

```
F859 CD 6D00        CALL WBUFI
F85C 0A0A2020   DB 0A,0A,20,20,20,2A,20,41,4C
F860 202A2041
F864 4C
F865 4C455320   DB 4C,45,53,20,50,41,4C,45,54
F869 50414C45
F86D 54
F86E 5449203F   DB 54,49,20,3F,20,2A,00
F872 202A00
```

Text innerhalb des Rechtecks

```
F875 CD 5500    CALL WCHARI
F878 1C         DB 1C
```

Cursor nach oben links auf Bildschirm

```
F879 CD 6D00    CALL WBUFI
F87C 4D495420   DB 'MIT ECS KOENNEN SIE NACH KMD > ZURUECK',00
F880 45534320
F884 4B4F454E
F888 4E454E20
F88C 53494520
F890 4E414348
F894 204B4D44
F898 203E205A
F89C 55525545
F8A0 434B00
```

Text

```
F8A3 CD 4300    ENDE:CALL RCHAR
F8A6 C3 A3F8    JMP ENDE
F8A9            END
 *** RESTART ? (Ja/NEIN)
```

Ende mit [ESC]

Anhang, 8085-Befehlsliste

Befehlsgruppe (Bedeutung der Spalten)	Mnemonik	Maschinen Code	Taktzyklen	Bytes	Masch. Zyklus	Funktion des Befehls	Veränderte Flags N	Z	P	C	H
Daten-Transport	MOV r1, r2		4	1	1	(r1) ← (r2)	X	X	X	X	X
	MOV M, r		7	1	2	(M) ← (r)	X	X	X	X	X
	MOV r, M		7	1	2	(r) ← (M)	X	X	X	X	X
	MVI r, n		7	2	2	(r) ← n	X	X	X	X	X
	MVI M, n	3 6	10	2	3	(M) ← n	X	X	X	X	X
	LXI B, m	0 1	10	3	3	(C) ← <B2>; (B) ← <B3>; m= <B3><B2>	X	X	X	X	X
	LXI D, m	1 1	10	3	3	(E) ← <B2>; (D) ← <B3>; m= <B3><B2>	X	X	X	X	X
	LXI H, m	2 1	10	3	3	(L) ← <B2>; (H) ← <B3>; m= <B3><B2>	X	X	X	X	X
	LXI SP, m	3 1	10	3	3	(SP) ← m	X	X	X	X	X
	SPHL	F 9	6	1	1	(SP) ← (H) (L)	X	X	X	X	X
	STAX B	0 2	7	1	2	((B) (C)) ← (A)	X	X	X	X	X
	STAX D	1 2	7	1	2	((D) (E)) ← (A)	X	X	X	X	X
	LDAX B	0 A	7	1	2	(A) ← ((B) (C))	X	X	X	X	X
	LDAX D	1 A	7	1	2	(A) ← ((D) (E))	X	X	X	X	X
	STA m	3 2	13	3	4	(m) ← (A)	X	X	X	X	X
	LDA m	3 A	13	3	4	(A) ← (m)	X	X	X	X	X
	SHLD m	2 2	16	3	5	(m) ← (L); (m+1) ← (H)	X	X	X	X	X
	LHLD m	2 A	16	3	5	(L) ← (m); (H) ← (m+1)	X	X	X	X	X
	XCHG	E B	4	1	1	(H) (L) ↔ (D) (E)	X	X	X	X	X
	XTHL	E 3	16	1	5	(H) (L) ↔ ((SP)+1) ((SP))	X	X	X	X	X
Arithmetik, Logik Vergleich	ADD r		4	1	1	(A) ← (A)+(r)	∎	∎	∎	∎	∎
	ADD M	8 6	7	1	2	(A) ← (A)+(M)	∎	∎	∎	∎	∎
	ADI n	C 6	7	2	2	(A) ← (A)+n	∎	∎	∎	∎	∎
	ADC r		4	1	1	(A) ← (A)+(r)+(C)	∎	∎	∎	∎	∎
	ADC M	8 E	7	1	2	(A) ← (A)+(M)+(C) C= Carry	∎	∎	∎	∎	∎
	ACI n	C E	7	2	2	(A) ← (A)+n+(C)	∎	∎	∎	∎	∎
	DAD B	0 9	10	1	3	(H) (L) ← (H) (L)+(B) (C)	X	X	X	∎	X
	DAD D	1 9	10	1	3	(H) (L) ← (H) (L)+(D) (E)	X	X	X	∎	X
	DAD H	2 9	10	1	3	(H) (L) ← (H) (L)+(H) (L)	X	X	X	∎	X
	DAD SP	3 9	10	1	3	(H) (L) ← (H) (L)+(SP)	X	X	X	∎	X
	SUB r		4	1	1	(A) ← (A)-(r)	∎	∎	∎	∎	∎
	SUB M	9 6	7	1	2	(A) ← (A)-(M)	∎	∎	∎	∎	∎
	SUI n	D 6	7	2	2	(A) ← (A)-n	∎	∎	∎	∎	∎
	SBB r		4	1	1	(A) ← (A)-(r)-(C)	∎	∎	∎	∎	∎
	SBB M	9 E	7	1	2	(A) ← (A)-(M)-(C) C= Carry	∎	∎	∎	∎	∎
	SBI n	D E	7	2	2	(A) ← (A)-n-(C)	∎	∎	∎	∎	∎
	ANA r		4	1	1	(A) ← (A)∧(r)	∎	∎	∎	0	1
	ANA M	A 6	7	1	2	(A) ← (A)∧(M)	∎	∎	∎	0	1
	ANI n	E 6	7	2	2	(A) ← (A)∧n	∎	∎	∎	0	1
	XRA r		4	1	1	(A) ← (A)∀(r)	∎	∎	∎	0	0
	XRA M	A E	7	1	2	(A) ← (A)∀(M)	∎	∎	∎	0	0
	XRI n	E E	7	2	2	(A) ← (A)∀n	∎	∎	∎	0	0
	ORA r		4	1	1	(A) ← (A)∨(r)	∎	∎	∎	0	0
	ORA M	B 6	7	1	2	(A) ← (A)∨(M)	∎	∎	∎	0	0
	ORI n	F 6	7	2	2	(A) ← (A)∨n	∎	∎	∎	0	0
	CMP r		4	1	1	(A) − (r)	∎	∎	∎	∎	∎
	CMP M	B E	7	1	2	(A) − (M)	∎	∎	∎	∎	∎
	CPI n	F E	7	2	2	(A) − n	∎	∎	∎	∎	∎
Register Inkrement Dekrement	INR r		4	1	1	(r) ← (r)+1	∎	∎	∎	X	∎
	INR M	3 4	10	1	3	(M) ← (M)+1	∎	∎	∎	X	∎
	DCR r		4	1	1	(r) ← (r)-1	∎	∎	∎	X	∎
	DCR M	3 5	10	1	3	(M) ← (M)-1	∎	∎	∎	X	∎
	INX B	0 3	6	1	1	(B) (C) ← (B) (C)+1	X	X	X	X	X
	INX D	1 3	6	1	1	(D) (E) ← (D) (E)+1	X	X	X	X	X
	INX H	2 3	6	1	1	(H) (L) ← (H) (L)+1	X	X	X	X	X
	INX SP	3 3	6	1	1	(SP) ← (SP)+1	X	X	X	X	X
	DCX B	0 B	6	1	1	(B) (C) ← (B) (C)-1	X	X	X	X	X
	DCX D	1 B	6	1	1	(D) (E) ← (D) (E)-1	X	X	X	X	X
	DCX H	2 B	6	1	1	(H) (L) ← (H) (L)-1	X	X	X	X	X
	DCX SP	3 B	6	1	1	(SP) ← (SP)-1	X	X	X	X	X
Akku rotieren	RLC	0 7	4	1	1	Links schieben C ← A7 A6 ... A1 A0 ←	X	X	X	∎	X
	RRC	0 F	4	1	1	Rechts schieben C → A7 A6 ... A1 A0 →	X	X	X	∎	X
	RAL	1 7	4	1	1	Links rotieren C ← A7 A6 ... A1 A0 ←	X	X	X	∎	X
	RAR	1 F	4	1	1	Rechts rotieren C → A7 A6 ... A1 A0 →	X	X	X	∎	X
Akku beeinfl.	CMA	2 F	4	1	1	(A) ← (Ā)	X	X	X	X	X
	DAA	2 7	4	1	1	(A) in BCD wandeln	∎	∎	∎	∎	∎
Carry setzen	STC	3 7	4	1	1	(C) ← 1	X	X	X	1	X
	CMC	3 F	4	1	1	(C) ← (C̄)	X	X	X	∎	X

Anhang, 8085-Befehlsliste

Bedeutung der Spalten / Befehls-gruppe	Mnemonik	Maschinen Code	Taktzyklen	Bytes	Masch. Zyklus	Funktion des Befehls	Veränderte Flags N	Z	P	C	H
Sprünge	JMP m	C 3	10	3	3	(PC) ← m	X	X	X	X	X
	PCHL	E 9	6	1	1	(PC) ← (H) (L)	X	X	X	X	X
	JC m	D A	10/7	3	3/2	(C)= 1	X	X	X	X	X
	JNC m	D 2	10/7	3	3/2	(C)= 0 Wenn Bedingung erfüllt	X	X	X	X	X
	JZ m	C A	10/7	3	3/2	(Z)= 1 (PC) ← m	X	X	X	X	X
	JNZ m	C 2	10/7	3	3/2	(Z)= 0	X	X	X	X	X
	JP m	F 2	10/7	3	3/2	(N)= 0	X	X	X	X	X
	JM m	F A	10/7	3	3/2	(N)= 1 Wenn Bedingung nicht erfüllt	X	X	X	X	X
	JPE m	E A	10/7	3	3/2	(P)= 1 (PC) ← (PC)+ 3	X	X	X	X	X
	JPO m	E 2	10/7	3	3/2	(P)= 0	X	X	X	X	X
Unter-programm-aufrufe	CALL m	C D	18	3	5	((SP)- 1) ((SP)- 2) ← (PC)+ 3, (PC) ← m (SP) ← (SP)- 2	X	X	X	X	X
	RST n		12	1	3	((SP)- 1) ((SP)- 2) ← (PC)+ 1, (PC) ← n x 8 (SP) ← (SP)- 2 wobei 0 ≦ n ≦ 7	X	X	X	X	X
	CC m	D C	18/9	3	5/2	(C)= 1	X	X	X	X	X
	CNC m	D 4	18/9	3	5/2	(C)= 0 Wenn Bedingung erfüllt	X	X	X	X	X
	CZ m	C C	18/9	3	5/2	(Z)= 1 ((SP)- 1) ((SP)- 2) ← (PC)+ 3	X	X	X	X	X
	CNZ m	C 4	18/9	3	5/2	(Z)= 0 (PC) ← m, (SP) ← (SP)- 2	X	X	X	X	X
	CP m	F 4	18/9	3	5/2	(N)= 0	X	X	X	X	X
	CM m	F C	18/9	3	5/2	(N)= 1 Wenn Bedingung nicht erfüllt	X	X	X	X	X
	CPE m	E C	18/9	3	5/2	(P)= 1 (PC) ← (PC)+ 3	X	X	X	X	X
	CPO m	E 4	18/9	3	5/2	(P)= 0	X	X	X	X	X
Unter-programm-Rück-sprünge	RET	C 9	10	1	3	(PC) ← ((SP)+ 1) ((SP)), (SP) ← (SP) + 2	X	X	X	X	X
	RC	D 8	12/6	1	3/1	(C)= 1	X	X	X	X	X
	RNC	D 0	12/6	1	3/1	(C)= 0 Wenn Bedingung erfüllt	X	X	X	X	X
	RZ	C 8	12/6	1	3/1	(Z)= 1 (PC) ← ((SP)+ 1) ((SP)), (SP) ← (SP)+ 2	X	X	X	X	X
	RNZ	C 0	12/6	1	3/1	(Z)= 0	X	X	X	X	X
	RP	F 0	12/6	1	3/1	(N)= 0	X	X	X	X	X
	RM	F 8	12/6	1	3/1	(N)= 1 Wenn Bedingung nicht erfüllt	X	X	X	X	X
	RPE	E 8	12/6	1	3/1	(P)= 1 (PC) ← (PC)+ 1	X	X	X	X	X
	RPO	E 0	12/6	1	3/1	(P)= 0	X	X	X	X	X
Ein-	IN n	D B	10	2	3	(A) ← (Eing. Puffer) ← (Eingang Daten von Gerät n)	X	X	X	X	X
Ausgabe	OUT n	D 3	10	2	3	(Ausgabe Gerät n) ← (A)	X	X	X	X	X
Unterbrech. Steuerung	EI	F B	4	1	1	(INTE) ← 1	X	X	X	X	X
	DI	F 3	4	1	1	(INTE) ← 0	X	X	X	X	X
Stapel-steuerung	PUSH PSW	F 5	12	1	3	((SP)- 1) ← (A), ((SP)- 2) ← (F), (SP) ← (SP)- 2	X	X	X	X	X
	PUSH B	C 5	12	1	3	((SP)- 1) ← (B), ((SP)- 2) ← (C), (SP) ← (SP)- 2	X	X	X	X	X
	PUSH D	D 5	12	1	3	((SP)- 1) ← (D), ((SP)- 2) ← (E), (SP) ← (SP)- 2	X	X	X	X	X
	PUSH H	E 5	12	1	3	((SP)- 1) ← (H), ((SP)- 2) ← (L), (SP) ← (SP)- 2	X	X	X	X	X
	POP PSW	F 1	10	1	3	(F) ← ((SP)), (A) ← ((SP)+ 1), (SP) ← (SP)+2	●	●	●	●	●
	POP B	C 1	10	1	3	(C) ← ((SP)), (B) ← ((SP) + 1), (SP) ← (SP)+2	X	X	X	X	X
	POP D	D 1	10	1	3	(E) ← ((SP)), (D) ← ((SP)+ 1), (SP) ← (SP)+2	X	X	X	X	X
	POP H	E 1	10	1	3	(L) ← ((SP)), (H) ← ((SP)+ 1), (SP) ← (SP)+2	X	X	X	X	X
Sonstige	HLT	7 6	5	1	1	(PC) ← (PC)+ 1	X	X	X	X	X
	NOP	0 0	4	1	1	(PC) ← (PC)+ 1	X	X	X	X	X
8085 Befehle	RIM	2 0	4	1	1	Liest Unterbrechungsmaske und seriellen Eingang in Akku	X	X	X	X	X
	SIM	3 0	4	1	1	Setzt Unterbrechungsmaske und seriellen Ausgang	X	X	X	X	X

Symbol	Erklärung
r	Register (A, B, C, D, E, H, L, M)
m	Zwei- Byte Daten
n	Ein- Byte Daten
⟨B2⟩, ⟨B3⟩	2. bzw. 3. Byte des Befehls
F	8- Bit- Daten-Wort aus N, Z, X, H, X, P, X, C
PC	Programm- Zähler
SP	Stapel- Zeiger
←	Datum in gezeichneter Richtung transportiert

Symbol	Erklärung
( )	Inhalt eines Registers oder Speichers
∨, ⊻	Inklusiv ODER, Exklusiv ODER
∧	Log. UND
—	1er Komplement
X	Flaginhalt bleibt unverändert
●	Flaginhalt wird gesetzt oder rückgesetzt
M	Inhalt der durch H und L adressierten Speicherzeile