

Beschreibung	Inhaltsverzeichnis
	0. Einleitung 2
	1. Start des Editor/Assemblers 3
	2. Editor EDIT 3
	2.1. Programmspezifikation 3
	2.2. Arbeitsweise des Editors 4
	2.3. Bedienung 5
	2.4. Kommandos 6
	2.4.1. Kommandoaufbau 6
	2.4.2. Beschreibung der Kommandos 7
	2.4.2.1. E/A- und Steuerkommandos 7
	2.4.2.2. Aufbereitungskommandos 12
	2.4.3. Vereinfachung der Kommandoeingabe 15
	2.5. Fehlermeldungen 16
	3. Absolutassembler ASM 17
	3.1. Programmspezifikation 17
	3.2. Arbeitsweise des Absolutassemblers 18
	3.3. Bedienung 18
	3.4. Speicheraufteilung 20
	3.5. Programmstruktur 20
	3.6. Beschreibung der Kommandos für die Übersetzung 20
	3.7. Realisierter Sprachumfang 22
	3.8. Fehlermeldungen 22
	3.8.1. Allgemeine Fehler 22
	3.8.2. Syntaxfehler 23
EDITOR/ASSEMBLER	
zum Heimcomputer	
robotron Z9001	

VEB ROBOTRON_MESSELEKTRONIK >OTTO SCHÖN< DRESDEN
DDR 8012 Dresden, Lingnerallee 3, Postschließfach 211

0. Einleitung

Mit dem Programmsystem Editor/Assembler hat der Anwender die Möglichkeit, Programme auf Assemblerniveau zu schreiben. Es wird die Assemblersprache SYPS K 1520 realisiert. Sie ist eine maschinenorientierte Programmiersprache, die auch für das Mikrorechnersystem K 1520 verwendet wird.

Der wesentliche Vorteil dieser Programmierungsform besteht darin, daß der Programmierer unter Beachtung einfacher Regeln und mit einprägsamen (mnemonischen) Operationscodes und symbolischen Adressen Programme effektiver zu schreiben vermag als in rechnerinterner Form.

Im Gegensatz zu den höheren Programmiersprachen (z.B. BASIC) gestattet die Assemblersprache andererseits durch die 1:1-Übersetzung in den Maschinencode die gleiche maximale Zugriffsmöglichkeit zu allen Funktionen des Rechners wie der direkte Maschinencode.

Damit ist die Assemblersprache das ideale Programmierwerkzeug für die Programmierung zeitkritischer und/oder speicheroptimaler Programme.

Durch den Editor (EDIT) ist eine einfache Korrektur der Assemblerprogramme möglich. Mit dem Editor werden die Programme für den Assembler (ASM) bereitgestellt, der daraus Maschinencode erzeugt.

Die vorliegende Schrift ist keine Sprachbeschreibung der Assemblersprache SYPS K 1520. Es wird davon ausgegangen, daß Anwender, die das Programmsystem EDIT/ASM benutzen, die Assemblersprache kennen bzw. schon eine Sprachbeschreibung besitzen. Als Anhang wird eine Übersicht der Sprache SYPS K 1520 in Tabellenform mitgeliefert.

1. Start des Editor/Assemblers

Der Editor/Assembler kann als ROM- und RAM-Variante benutzt werden. Befindet sich der Editor/Assembler auf einem ROM-Modul, werden die beiden Programmteile des Editor/Assemblers durch die Eingabe von

EDIT

bzw.

ASM

gestartet.

Befindet sich der Editor/Assembler auf einer Magnetbandkassette (RAM-Variante), muß er erst durch die Eingabe des Monitorkommandos

CLOAD ASM

geladen werden.

Anschließend kann der Start der beiden Programmteile durch die Eingabe von EDIT bzw. ASM erfolgen. Die weiteren Kommandos sind bei der RAM- und ROM-Variante gleich.

2. Editor EDIT

2.1. Programmspezifikation

EDIT ist ein Dienstprogramm zum Erstellen und Aufbereiten von Quelltexten. Die Aufbereitung der Quelltexte, in diesem Falle Quellen in der Assemblersprache SYPS K 1520, erfolgt in einem frei wählbaren Textpuffer.

Der Editor arbeitet zeilenorientiert, d.h., es werden stets vollständige Zeilen verarbeitet. Die Spezifizierung der Zeilen kann sowohl durch die Zeilennummer als auch durch eine zu Beginn der Zeile stehende Zeichenkette erfolgen.

EDIT realisiert folgende Funktionen:

- Streichen von Zeilen,
- Einfügen von Zeilen,
- Substituieren von Zeichenketten,
- Anfügen von Kommentar,
- Streichen von Kommentar,
- Anzeigen des Textpufferinhaltes auf dem Bildschirm,
- Ausgeben des Textpufferinhaltes auf Magnetbandkassette,
- Eingeben von Quelltexten von der Magnetbandkassette in den Textpuffer.

2.2. Arbeitsweise des Editors

Jede Zeile wird bei Eingabe über die Tastatur in einen Zeilenpuffer übernommen, dort analysiert, aufbereitet und in den Textpuffer übernommen.

Ein Zeilenzähler zählt intern die Anzahl der verarbeiteten Zeilen. Die Bezugnahme auf diesen Zeilenzähler erfolgt durch die Angabe der Zeilennummer in den Kommandos. Die Aufbereitung des Quelltextes ist grundsätzlich nur mit steigender Zeilennummer möglich. Über die Tastatur des Heimcomputers eingegebene Quellzeilen werden an der aktuellen Position in den neuen Quelltext übernommen und beeinflussen den Zeilenzähler nicht.

Der Textpuffer des EDIT ist ein Speicherbereich im Arbeitsspeicher des Heimcomputers, der durch seine Anfangs- und Endadresse festgelegt wird. Eingegebene Quelltexte von der Magnetbandkassette werden an eventuell schon vorhandenen Quelltext im Textpuffer angefügt.

Soll der im Textpuffer aufbereitete Quelltext durch andere Programme weiterverarbeitet werden, so ist zu beachten, daß die Abspeicherung des Textes ab der höherwertigen Adresse in Richtung auf die niederwertige Adresse erfolgt.

Anfangs- und Endadresse des Textpuffers sind standardmäßig festgelegt und können bei Bedarf durch das A-Kommando angezeigt und verändert

werden. Als Kennzeichen für Quelltextende verwendet der Editor die Codierung 03.

2.3. Bedienung

Nach dem Start (siehe Abschnitt1) meldet sich der Editor mit der Ausschrift

EDIT:

und fordert die Eingabe eines Pflichtkommandos an:

N (Neustart) - Diese Eingabe ist bei Erststart des Editors zur Initialisierung der Zuweisungen und des Textpuffers notwendig.

(Restart) - Alle anderen Zeichen, z.B. nur Betätigen der ENTER-Taste, bewirken einen Restart des Editors. Ein im Textpuffer stehender Text bleibt erhalten. Restart ohne einen vorangegangenen Neustart führt zu undefinierbaren Fehlern.

Durch die Ausgabe von "+" auf dem Bildschirm wird die nächste Bedienereingabe angefordert.

Jede Tastatureingabe wird erst nach Abschluß der Eingabezeile, d.h. mit dem Drücken der ENTER-Taste, analysiert und verarbeitet. Korrekturen während der Eingabe einer Zeile sind durch Betätigen der Kursortaste <-- und anschließende Eingabe der richtigen Zeichen möglich. Beginnt die Bedienereingabe mit dem Zeichen # (Kommandokennzeichen), so wird die nachfolgend eingegebene Zeile als Kommentar interpretiert. Andernfalls wird sie als neue Zeile in den Quelltext übernommen.

Soll eine Quelltextzeile mit "#" beginnen, so ist das Zeichen doppelt einzugeben.

Die maximale Länge einer Zeile beträgt 72 Zeichen.

2.4. Kommandos

2.4.1. Kommandoaufbau

Im Editor stehen Aufbereitungs-, E/A- und Steuerkommandos zur Verfügung.

Mit den Aufbereitungskommandos kann der Quelltext im Textpuffer modifiziert werden.

Mit Hilfe der E/A-Kommandos erfolgt die Übertragung von Quelltexten zwischen dem Textpuffer und der Magnetbandkassette.

Die Aufbereitungskommandos besitzen allgemein folgendes Format:

#Xn,m oder #X [parameter]

Dabei bedeuten:

#	Kommandokennzeichen
X	Kommandosymbol als Klein- oder Großbuchstabe
n,m	Zeilenspezifikation
,	Trennzeichen (Leerzeichen oder Komma)

Nach Abarbeitung des Kommandos erfolgt die Anforderung eines neuen Kommandos.

EDIT bietet die Möglichkeit, die Zeilenspezifikation in 2 Formen anzugeben. Beide Formen können in einem Kommando auftreten.

Zeilennummer: vorzeichenlose Dezimalzahl ($0 < n \leq 65535$)

Zeichenkette: Die Zeichenkette muß zu Beginn der anzugebenden Zeile stehen. Sie wird rechts oder links durch das gleiche Sonderzeichen begrenzt. Dieses Sonderzeichen ist frei wählbar, darf aber nicht selbst Bestandteil der Zeichenkette und kein Trennzeichen sein. Für die Länge "l" der Zeichenkette gilt $0 < l < 72$. Durch die Angabe einer Verschiebung ist die Bezugnahme auf eine Zeile möglich, die nach der durch die Zeichenkette spezifizierten steht.

Beispiel:

```
+MARKE2+  
/ZYKL:/+5  
:A:+100
```

Im 1. Fall ist die Zeile spezifiziert, die mit der Zeichenkette "MARKE2" beginnt. Bei den 2 folgenden Spezifikationen einer Zeile kommt die Angabe einer Verschiebungszahl zur Anwendung. Findet z.B. der Editor eine Zeile, die mit "ZYKL:" beginnt, zählt er anschließend noch 5 Zeilen weiter. Diese ist dann die gewünschte Zeile.

2.4.2. Beschreibung der Kommandos

2.4.2.1. E/A- und Steuerkommandos

A (assign) Anzeigen und Modifizieren des Textpufferbereiches

Syntax: #A

Nach Eingabe dieses Kommandos werden anschließend 3 Adressen in folgender Form angezeigt:

```
TB CXXXXX-XXXXX,XXXXX :
```

1. Adresse: aktuelle Anfangsadresse des Textpuffers
2. Adresse: Ende des Textes im Textpuffer
3. Adresse: aktuelle Endadresse des Textpuffers

Die Anzeigen sind durch ENTER (keine Änderung) oder durch die Eingabe neuer Textpufferanfangs- und Endadressen in der Form XXXXX,XXXXX zu quittieren. Soll nach einer Erweiterung oder Verkürzung des Textpuffers mit dem vorhandenen Textpufferinhalt weitergearbeitet werden, so ist die Adresseneingabe nur mit ENTER abzuschließen. Andernfalls (auch bei einer Zuweisung nach Programmneustart) ist vor Abschluß mit ENTER noch ein "N" einzugeben.

Jede Zuweisung des Textpuffers bewirkt einen Programm-
restart, d.h., der interne Textpufferzeiger des Editors wird auf
Null gestellt.

Beispiel:

```
+#A  
TB BFFF-BF00, BF10: B000, 8000N  
+■
```

Durch die Ausgabe von "+" fordert der Editor zur Eingabe auf. Nach der
Eingabe des Kommandos #A wird der aktuelle Textpufferbereich ange-
zeigt. Der Textpuffer wird von der höherwertigen zur niederwertigen
Adresse gefüllt. In diesem Fall ist der Textpuffer bis zur Adresse BF10H
belegt. Durch die Eingabe von

```
B000, 8000N
```

wird der Textpufferanfang etwas verschoben und der Bereich vergrößert.
Die Eingabe von N bewirkt eine Initialisierung der Arbeitszellen des
Editors, z.B. erhält der Textendezeiger die Anfangsadresse von Textpuffer.

B (begin) Neustart des Aufbereitungsvorganges

Syntax: #B bzw. #Bn

Das Kommando bewirkt das Rücksetzen aller Arbeitszellen
auf den Anfang des Quelltextes.

Bei Angabe einer vorzeichenlosen Dezimalzahl werden die
Zellen beginnend mit der angegebenen Zahl gezählt. Damit
kann der Aufbereitungsvorgang erneut begonnen werden.

Beispiel:

```
•  
•  
•  
+#B  
+TEXTPUFFERANFANG  
+■
```

Nach der Eingabe von #B erfolgt ein Restart des Editors. Die folgende
Eingabe "TEXTPUFFERANFANG" wird als Quellzeile in den Textpuffer als

1. Zeile eingefügt. Schon vorhandener Quelltext wird im Puffer nach hinten
verschoben.

E (end) Beenden der Arbeit mit dem Editor

Syntax: #E

Der Editor wird verlassen, und es wird der Rücksprung in das
Monitorprogramm veranlaßt. Der Inhalt des Textpuffers bleibt
erhalten und kann nach Restart des EDIT erneut aufbereitet
werden.

Beispiel:

```
+#E  
OS  
>■
```

Nach der Eingabe des Kommandos #E meldet sich das Monitorprogramm
mit ">" und fordert zur Eingabe auf. Der Cursor ■ steht am
Anfang der Zeile.

R (read) Eingabe von der Kassette in den Textpuffer

Syntax: #R"Name"

Das Kommando ermöglicht das Einlesen eines Quelltextes
(Programm) von einer Magnetbandkassette in den Textpuffer
des EDIT. Der eingelesene Text wird an einen eventuell
schon vorhandenen Text im Textpuffer angefügt. Auf diese
Weise kann man auch durch mehrere R-Kommandos den
Textpuffer füllen.

Der angegebene Name muß mit dem übereinstimmen, mit
dem der Quelltext (Assemblerprogramm) auf die Kassette
geschrieben wurde (EDIT-Kommando #W"Name").

Stimmt der gelesene Name, nicht mit dem angegebenen
überein, wird der gefundene Name von EDIT auf dem
Bildschirm ausgeschrieben.

Der Editor beendet das Einlesen von Quelltext, wenn eine der folgenden Bedingungen erfüllt ist:

- Erkennen des Quelltextendes (Codierung 03),
- Erkennen eines Endblockes (Blocknummer FFH),
- Erreichen des Textpufferendes.

Bei Lesefehlern erscheinen Fehlerausschriften, die in der Bedienungsanleitung bzw. im Programmierhandbuch beschrieben sind.

Sonderfall:

Fehlt die Angabe des Namens, erfolgt kein Namensvergleich, d.h., beginnend mit dem 1. Block (Blocknummer 1) werden die Daten in den Textpuffer eingelesen.

Beispiel:

```
+ #R "PROGRAMM"  
start tape <ENTER>  
+ ■
```

Nach der Eingabe des R-Kommandos erscheint die Ausschrift

```
start tape .
```

Die Magnetbandkassette muß auf die gewünschte Stelle positioniert werden.

Durch Betätigen der ENTER-Taste beginnt der Lesevorgang. Nach dem erfolgreichen Einlesen des Blockes (128 Bytes) wird der Cursor jeweils um eine Position nach rechts bewegt. Ist das Programm mit dem Namen "PROGRAMM" vollständig eingelesen, meldet sich der Editor wieder und wartet auf die nächste Eingabe.

T (text) Ausschreiben des Textpufferinhaltes

Syntax: #Tn,m

Durch das Kommando werden die Zeilen n bis einschließlich m des Textpuffers mit Angabe der Zeilennummer auf dem Bildschirm ausgegeben.

Der interne Zeilenzähler wird dabei nicht verändert.

Durch Betätigen der PAUSE-Taste kann die Bildschirmausgabe angehalten und durch Drücken einer beliebigen anderen Taste fortgesetzt werden.

Beispiel:

```
+ #T20,/MARKE:✓  
+ ■
```

Beginnend mit der Zeile 20 wird der Quelltext bis einschließlich der Zeile, die mit "MARKE:" beginnt, auf dem Bildschirm ausgegeben.

U (uppercase) Umwandlung von Kleinbuchstaben in Großbuchstaben

Syntax: #UN oder #U[Y]

#U[Y] Die Umwandlung der Kleinbuchstaben in Großbuchstaben wird während der Ausführung eines Aufbereitungskommandos durchgeführt

#UN Es erfolgt keine Umwandlung der Kleinbuchstaben.

Das Kommando #UN ist Standard, d.h., nach Neustart des EDIT erfolgt während der Ausführung eines Aufbereitungskommandos keine Umwandlung der Kleinbuchstaben in Großbuchstaben.

Das U-Kommando wirkt auf die folgenden Aufbereitungskommandos und kann während eines Aufbereitungsvorganges beliebig oft eingegeben werden.

W (write) Ausgabe des Textpufferinhaltes auf Magnetband

Syntax: #W"Name"

Der Inhalt des Textpuffers wird auf die Kassette ausgelagert.

Bis zu 8 Zeichen sind für den Namen signifikant. Nach erfolgter Eingabe des Kommandos wird auf dem Bild die Aufforderung

```
start tape
```

ausgeschrieben. Nach Betätigung der entsprechenden Tasten am Kassettengerät und Drücken der ENTER-Taste erfolgt der Aufzeichnungsvorgang. Nach Beendigung der Ausgabe kann ein Vergleichslesen durchgeführt werden. Das Vergleichs-lesen ist in der Bedienungsanleitung bzw. im Programmier-handbuch beschrieben. Jeder ausgegebene Block (128 Bytes des Textpuffers) wird auf dem Bildschirm angezeigt.

Beispiel:

```
+#W"PROG1"  
start tape <ENTER>  
+■
```

Ein im Textpuffer befindlicher Quelltext wird unter dem Namen "PROG1" auf die Kassette geschrieben. Dabei verwendet der Editor den Typ "TXT" für seine erstellten Textdateien (Assemblerprogramme) auf der Kassette.

2.4.2.2. Aufbereitungskommandos

Durch die unten aufgeführten Aufbereitungskommandos kann ein im Textpuffer vorhandener Text verändert werden.

C (comment) Anfügen von Kommentar

Syntax: #Cn,m

Die Zeilen bis einschließlich n-1 bleiben unverändert erhalten. Für die Zeilen n bis einschließlich m wird durch Ausgabe des Kommentarkennzeichens (;) die Eingabe eines Kommentartextes angefordert. Das Kommentarkennzeichen und der Kommentartext werden an die Zeile angefügt bzw. überschreiben den alten Kommentar. Die Abarbeitung des

Kommandos ist erst beendet, wenn kein Kommentar mehr angefordert wird.

I (insert) Einfügen von Zeilen

Syntax: # In

Die Zeilen bis einschließlich der n-ten Zeile bleiben im Textpuffer erhalten.

Der interne Textpufferzeiger des Editors erhält den Wert n. Anschließend können über die Tastatur Quellzeilen eingegeben werden.

Die eingegebenen Zeilen werden nach der Zeile n in den Textpuffer eingefügt.

Beispiel:

```
+# I100  
+AUSG:LD DE,TEXT; STRINGAUSGABE  
+LD C,9  
+■
```

Nach der Zeile 100 des Textpuffers werden diese 3 Zeilen eingefügt.

K (kill) Streichen von Zeilen

Syntax: #Kn,m

Die Zeilen bis einschließlich Zeile n-1 bleiben erhalten. Die Zeilen n bis einschließlich m des alten Quelltextes werden gelöscht.

Beispiel:

```
+#K10,12  
+LD A,60H  
+■
```

Die Zeilen 10, 11 und 12 des Textpuffers werden gelöscht. Es wird dafür eine Zeile eingefügt.

N (no comment) Streichen von Kommentar

Syntax: #Nn,m

Die Zeilen bis einschließlich n-1 bleiben unverändert erhalten. In den Zeilen n bis einschließlich m wird der Kommentar gelöscht. Leerzeilen und Zeilen, die nur das Kommentarzeichen enthalten, werden gestrichen. Auf diese Weise kann der Speicherplatzbedarf für den Textpuffer verringert werden.

S (substitute) Substituieren von Teilzeichenketten

Syntax: #Sn,m
/ZKa1/ZKn1/,/ZKa2/ZKn2/,.../ZKan/ZKnn

Dabei bedeuten: ZKa alte Teilzeichenkette
ZKn neue Teilzeichenkette
/ Begrenzungszeichen analog der Zeilenspezifikation

Die Zeilen bis einschließlich Zeile n-1 bleiben erhalten. In den Zeilen n bis einschließlich m wird nacheinander versucht, die alten Teilzeichenketten durch die neuen zu ersetzen. Im Kommando muß mindestens eine Substitution ausgeführt werden, da sonst eine Fehlermeldung erfolgt.

Zwei Sonderfälle sind möglich:

/ZK/ Die angegebene Zeichenkette wird gestrichen
//ZK/ Die angegebene Zeichenkette wird an den Anfang der Zeile gesetzt.

Beispiel:

```
+#S10,100
+ /HL/DE/
+#S105
+ /WIRD GELOESCHT//
+ ■
```

In den Zeilen 10 bis 100 des Textpuffers wird die Zeichenkette "HL" durch "DE" ersetzt. In der Zeile 105 wird die angegebene Zeichenkette gelöscht. Die Spezifizierung nur einer Zeile ist möglich (siehe Abschnitt 2.4.3.).

2.4.3. Vereinfachung der Kommandoeingabe

Zur Vereinfachung der Kommandoeingabe können in bestimmten Fällen Teile der Zeilenspezifikationen weggelassen werden. Es sind Vereinfachungen bei den Kommandos C, D, K, N, S und T möglich.

#Xn,	von Zeile n bis Textende
#X,m	vom aktuellen Stand des internen Zeilenzählers bis einschließlich Zeile m (Ausnahme T-Kommando)
#X,	vom aktuellen Stand des internen Zeilenzählers bis Textende (Ausnahme T-Kommando)
#Xn	nur Zeile n
#X	nur die nächste Zeile (Ausnahme T-Kommando)

Wird beim T-Kommando keine Anfangszeile angegeben, erfolgt die Ausgabe des Textes ab Textpufferanfang.

Beispiele:

```
+#T,10
+ ■
```

Die Zeilen 1 bis 10 werden auf dem Bildschirm angezeigt.

```
+#T,
+ ■
```

Es wird der gesamte Textpufferinhalt auf dem Bildschirm ausgegeben.

```
+#K56
+ ■
```

Die Zeile 56 wird aus dem Textpuffer gestrichen

```
+#K100,
+ ■
```


Ab der Zeile 100 werden alle Zeilen aus dem Textpuffer gelöscht.

```
+#S98  
//MARKE:/  
+■
```

Die Zeichenkette "MARKE:" wird am Anfang der Zeile 98 eingefügt.

2.5. Fehlermeldungen

+++CE (command error)

Das eingegebene Kommando ist syntaktisch fehlerhaft oder nicht sinnfäll.

Ursachen: - fehlerhafte Syntax,
- unzulässiges Kommandosymbol,
- Zeilennummer kleiner als Nummer der nächsten zu bearbeitenden Zeile,
- erste Zeilennummer größer als zweite.

Die Kommandoeingabe ist zu wiederholen.

+++LE (length error)

Ursachen: - Bei der Tastatureingabe wurde die maximale Zeilenlänge (72) überschritten. Die Eingabe ist mit reduzierter Zeilenlänge zu wiederholen.
- Beim S-Kommando wurde die maximale Zeilenlänge überschritten. Die Zeile wird mit Zeilennummer in ihrem aktuellen Zustand angezeigt. Sie ist mit zulässiger Länge über die Tastatur einzugeben.

+++NF (not found)

Eine spezifizierte Zeile wurde im Textpuffer nicht gefunden.

+++TB (textbuffer)

Das Ende des Textpuffers ist erreicht. Eine weitere Speicherung von Zeilen führt zum Überlauf des Textpuffers. Der Textpuffer ist zu erweitern (A-Kommando).

+++SE (substitution error)

Bei einem S-Kommando konnte keine der angegebenen Substitutionen ausgeführt werden, da keine der angegebenen alten Teilzeichenketten in den spezifizierten Zeilen gefunden wurde.

Weiterhin können Fehler beim Einlesen von der Magnetbandkassette auftreten. Diese Fehler sind in der Bedienungsanleitung bzw. im Programmierhandbuch beschrieben.

3. Absolutassembler ASM

3.1. Programmspezifikation

Der Assembler ASM dient zum Übersetzen von Quellprogrammen in der Assemblersprache SYPS K 1520 in die Maschinensprache. ASM erzeugt einen absolut adressierten Maschinencode.

Die Übersetzung erfolgt aus dem Textpuffer des Editors.

ASM realisiert folgende Funktionen:

- Quellprogrammeingabe aus dem Textpuffer des Editors EDIT,
 - Maschinencodeausgabe in den Speicher,
 - Maschinencodeausgabe auf das Magnetband,
 - Ausgabe der Assemblerprotokolle auf dem Listgerät (Bildschirm oder Drucker)
- Es kann zwischen Übersetzungsliste, Fehlerliste, Crossreferenztablelle und Symbolreferenztablelle gewählt werden.

Die Quellprogrammzeilen, die ASM verarbeitet, müssen im Format der Sprache SYPS K 1520 vorliegen. Jede Zeile ist mit dem

Zeilenendezeichen (CR) abzuschließen und kann aus maximal 80 Zeichen bestehen.

Farbsteuerzeichen sind in einer Quellprogrammzeile nur innerhalb einer Zeichenkette der DB-Anweisung zulässig. ASM versteht nur Befehle (Mnemoniks) mit Großbuchstaben. Mnemoniks mit Kleinbuchstaben weist ASM als syntaktisch falsch aus.

3.2. Arbeitsweise des Absolutassemblers

ASM übersetzt den Inhalt des Textpuffers des Editors EDIT in den Maschinencode. Die Ausgabe erfolgt auf das Magnetband oder in den Arbeitsspeicher (RAM) des Heimcomputers.

Das Assemblerprotokoll kann über den Bildschirm oder einen Drucker, falls ein Drucker und das entsprechende Druckerbedienungsprogramm vorhanden sind, ausgegeben werden.

ASM benötigt 2 Durchläufe (Pässe) zur Übersetzung. Im 1. Durchlauf erzeugt er die Symboltabelle. Im 2. Durchlauf kann gewählt werden zwischen:

- Ausgabe der Übersetzungsliste,
- Ausgabe des Maschinencodes in den Arbeitsspeicher oder auf Kassette,
- Ausgabe der Fehlerliste der Assemblerquelle,
- Ausgabe der Crossreferenztabelle,
- Ausgabe der Symbolreferenztabelle.

3.3. Bedienung

Der Assembler meldet sich nach dem Start mit der Ausschrift

ASM--HC:

und erwartet die Eingabe eines Pflichtkommandos.

Bei Programmneustart muß ein "N" eingegeben werden. Damit übernimmt der Assembler die Textpufferadressen des Editors und initialisiert seine Arbeitszellen. Jede andere Eingabe führt zum Restart des ASM. Erfolgt nach dem 1. Aufruf kein Neustart, arbeitet der Assembler nicht ordnungsgemäß. Bei der RAM-Variante kann das auch zur Zerstörung des EDIT/ASM führen.

Das zu übersetzende Assemblerprogramm muß im Textpuffer des EDIT bereitstehen, d.h. mit Hilfe des EDIT erstellt oder von der Magnetbandkassette eingelesen worden sein.

Nach Neustart des Assemblers und dem Kommando "V" erfolgen folgende Ausschriften:

HL: Der Anwender hat die Möglichkeit, eine maximal 18 Zeichen lange Kopfzeile (z.B. Datum) für das Assemblerprotokoll einzugeben.

Nach der Ausschrift "MO:" wird durch die Eingabe von "T" oder "R" festgelegt, ob die MC-Ausgabe bei der Ausführung des Kommandos M (siehe Abschnitt 2.6.) in den RAM oder auf die Magnetbandkassette erfolgen soll.

MO: T MC-Ausgabe auf Magnetbandkassette.
R MC-Ausgabe in den Arbeitsspeicher (RAM) entsprechend den ORG-Anweisungen im Assemblerprogramm. Bei der Ausgabe in den RAM ist zu beachten, daß damit während der Übersetzung der Quelltext, die Symboltabelle, die Arbeitszellen des EDIT/ASM oder EDIT/ASM selbst (RAM-Variante) entsprechend den ORG-Anweisungen der Assemblerquelle überschrieben werden können.

Nach der Ausgabe von "LO:" wird das Listgerät festgelegt.

LO: C Übersetzungsliste, Fehlerliste, Crossreferenztabelle und Symboltabelle werden bei Bedarf auf dem Bildschirm (Console) angezeigt.
F Die Ausgabe erfolgt auf den angeschlossenen Drucker (Printer).

3.4. Speicheraufteilung

Der Editor/Assembler kann als ROM- und RAM-Variante benutzt werden.

EDIT/ASM als RAM-Variante:

300H	1000H	3800H	Ende RAM			
Arbeits- zellen EDIT/ASM	EDIT	ASM	Symbol- tabelle	Cross- referen- z	--> <- -	Text- puffer

EDIT/ASM als ROM-Variante (Zusatzmodul):

300H	1000H				C000H	E7FFH
Arbeits- zellen EDIT/ASM	Symbol- tabelle	Cross- referenz		Text- puffer	EDIT	ASM
V RAM					V ROM	

3.5. Programmstruktur

Ein Assemblerprogramm kann aus mehreren Moduln bestehen. Ein Modul beginnt mit der Pseudoanweisung PN und endet mit END.

ASM übersetzt Quelltext, bis er nach einem END das Zeichen für Quelltextende (03) erkennt.

3.6. Beschreibung der Kommandos für die Übersetzung

Mit der Ausschrift

PASS:

fordert ASM die Spezifikation des aktuellen Passes an. Es sind folgende Eingaben möglich:

- 1 Ausführung des 1. Passes. Von dem Quelltext, der im Textpuffer des EDIT steht, wird eine Symboltabelle im RAM erzeugt. Dieser Paß ist für die Übersetzung eines Programmes unbedingt notwendig.

Mit den weiteren Eingaben wird ein 2. Durchlauf des Assemblers angefordert.

- P [Zeilennummer 1, Zeilennummer 2 [, Programmende]]
Ausgabe der Übersetzungsliste auf dem Listgerät. Durch Angabe von Zeilennummern erfolgt die Ausgabe einer beschnittenen Liste. Sind mehrere Module, eingeschlossen in PN und END, im Quelltext vorhanden, wird nur der erste Modul bzw. der angegebene aufgelistet. Für den Programmnamen müssen die 2 Zeichen angegeben werden, die in der Pseudoanweisung PN stehen.

- M (MC-Output)
Ausgabe des Maschinencodes auf Magnetbandkassette oder in den Arbeitsspeicher entsprechend der Angabe "T" oder "R". Eventuell auftretende Fehler, die ASM erkennt, erscheinen dabei auf dem Listgerät.
Die MC-Ausgabe auf Kassette erfolgt fortlaufend Byte für Byte. Die Anfangsadresse, ab der das Programm beim Einlesen im RAM abgelegt wird, wird durch eine ORG-Anweisung bestimmt. Nichtbelegte Speicherbereiche, z.B. durch ORG-Anweisungen hervorgerufen, werden als Null-Bytes auf die Kassette ausgegeben.
Soll die MC-Ausgabe auf die Kassette erfolgen, sind weitere Angaben nötig.
Nach der Eingabe von "M" meldet sich der Assembler mit der Ausschrift

NAME:

Es muß ein in Anführungsstrichen eingeschlossener Name, 1 bis 8 Zeichen lang, eingegeben werden.

Mit dem angegebenen Namen wird das Maschinencodeprogramm nach Ausführung des M-Kommandos auf die Kassette ausgelagert. Im Monitorstatus kann das entsprechende Pro-

gramm mit

```
CLOAD name bzw.  
name
```

eingelezen werden.

Nach der Meldung

```
ADDRESS:
```

kann eine Startadresse eingegeben werden. Wird nur die ENTER-Taste betätigt, erhält die Startadresse den Wert der Anfangsadresse des Maschinencodes.

Die Startadresse ist als hexadezimale Zahl (ohne Suffix H) einzugeben.

Mit dem M-Kommando ist die Möglichkeit gegeben, ein Maschinencodeprogramm für den Speicherbereich zu erzeugen, den der EDIT/ASM zur Zeit der Übersetzung selbst benutzt. Während der Ausführung des M-Kommandos sind keine ORG-Anweisungen möglich, die auf Speicheradressen verweisen, die durch das Maschinencodeprogramm schon belegt worden sind. In diesen Fällen erfolgt die Mitteilung

```
MC-OUTPUT WARNING
```

auf dem Bildschirm. Die MC-Ausgabe wird danach fortgesetzt. Beim Einlesen eines solchen Maschinencodeprogrammes von der Kassette kann es bei Nutzung der ORG-Anweisung zu Fehlern kommen.

F

(fault)

Syntaktische Prüfung des Quellcodes und Ausgabe der Fehlermeldungen bei syntaktisch falschem Programm.

Fehlerhafte Zeilen der Assemblerquelle werden auf dem Listgerät (Bildschirm oder Drucker) angezeigt. Es ist zweckmäßig, vor Ausgabe einer Übersetzungsliste oder des Maschinencodes zu überprüfen, ob das Programm syntaktisch richtig ist.

X

(cross-referenze)

Ausgabe der Crossreferenztabelle

A

(adress-dictionary)

Ausgabe der Symbolreferenztabelle

V

(variation)

Änderung des Datenflusses für die Ausgabe des Maschinencodes und der Assemblerprotokolle. Die Zuweisungen gelten so lange, auch nach Verlassen des Assemblers mit dem E-Kommando und erneutem Restart des Assemblers, bis mit dem nächsten V-Kommando oder nach Neustart von ASM die Datenflüsse geändert werden.

Beispiel:

```
PASS: V  
MO: R  
LO: P  
PASS: P  
PASS: E  
OS  
>■
```

Nach der Eingabe von "V" müssen der Datenfluß für die MC-Ausgabe (T oder R) und das Ausgabegerät für die Assemblerprotokolle (C oder P) festgelegt werden.

anschließend meldet sich der ASM wieder mit "PASS:". Hier wird eine Übersetzungsliste auf dem Drucker ausgegeben. Wird ASM mit dem E-Kommando verlassen, bleiben nach einem Wiederstart die vorher eingestellten Datenflüsse erhalten.

E

(end)

Rücksprung in das Steuerprogramm (Monitor).

Der 1. Paß ist Voraussetzung für die Übersetzungsläufe P, F, M, X und A. Diese können in beliebiger Reihenfolge durchgeführt werden.

Beim Erkennen des Quelltextendes (03) erfolgt wieder die Ausschrift "PASS:" und fordert damit eine Eingabe an.

Die Übersetzungsläufe P, F, X und A lassen sich durch Betätigen der PAUSE-Taste unterbrechen. Durch Drücken einer beliebigen anderen Taste erfolgt die Fortführung.

3.7. Realisierter Sprachumfang

ASM realisiert den gesamten Befehlsumfang des Prozessors U 880 in der Assemblersprache SYPS K 1520.

Macroverarbeitung und bedingte sind nicht möglich. Die Pseudobefehle

MACR, ENDM, ENIF, TITL, EJEG und DEF

führen zu Mnemonikfehlern.

Bei der Pseudoanweisung DB ist die zu definierende Zeichenkette pro Anweisung auf 35 Zeichen begrenzt.

3.8. Fehlermeldungen

3.8.1. Allgemeine Fehler

Folgende Fehlerausschriften können auftreten:

OVERFLOW - Überlauf der Symbol- oder Crossreferenztabelle,
- Textpuffer enthält kein vollständiges Programm
(PN ... END).

Beseitigung:

- Arbeitsspeicher erweitern oder Markenanzahl verringern und neu übersetzen
- PN bzw. END in die Quelle einfügen und neu übersetzen.

NO PN: - Programm beginnt nicht mit der PN-Anweisung.
Durch Eingabe eines "C" wird PN weiter im Quellcode gesucht.

3.8.2. Syntaxfehler

- +A+ Adressierungsfehler
- +C+ Falsche Mnemonik
- +F+ Formatfehler
- +M+ mehrfach definiertes Symbol
- +N+ nicht definiertes Symbol
- +O+ Operandenfehler