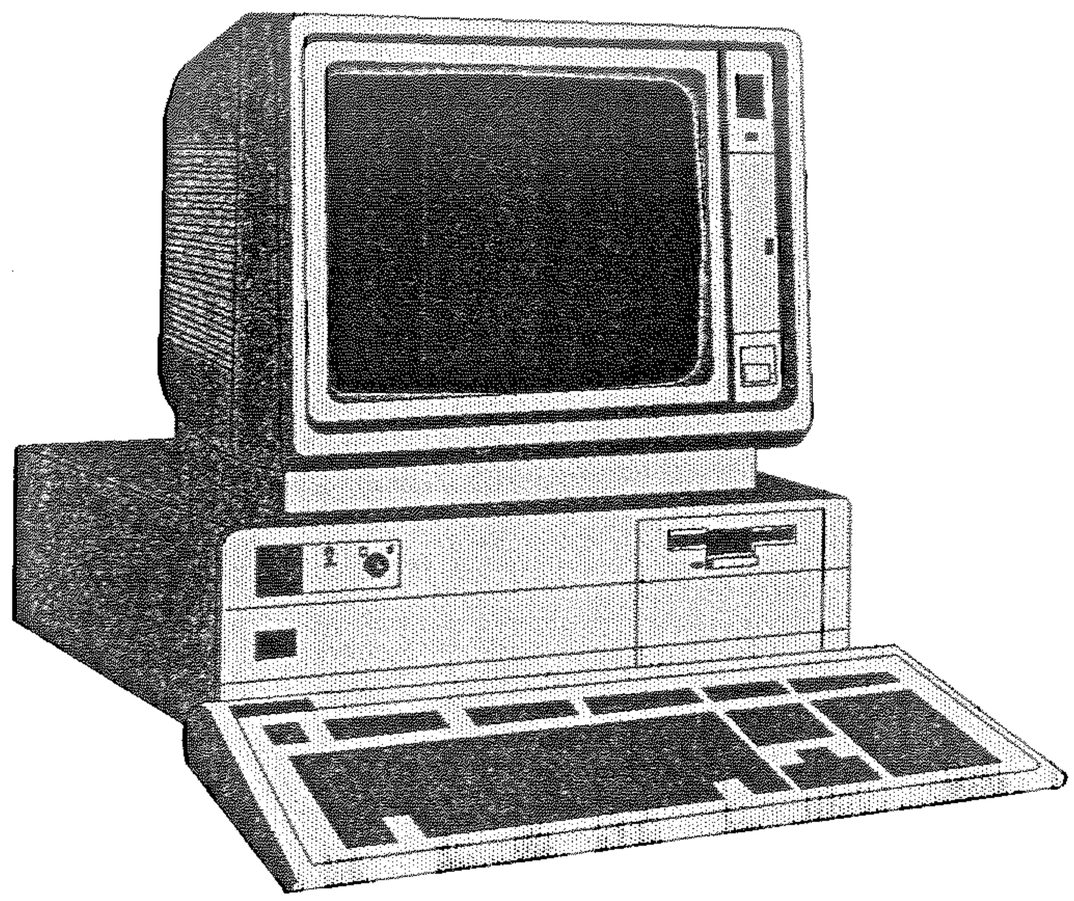
20.2.89

PCA - ASSEMBLER

Bedienerhandbuch

Version MS-DOS

Provisorische Ausgabe 2.89



SAIA AG
CH-3280 MURTEN

Tel 037 / 727 727
FAX 037 / 711 983

(c) SAIA AG 1987
Alle Rechte vorbehalten. Nachdruck, auch auszugsweise, verboten.

INHALTSVERZEICHNIS

INSTALLATION

1 Uebersicht
4 Installation auf System mit Festplatte
6 Installation auf System mit 2 Laufwerken
11 Arbeiten mit Hilfe der Menüs
12 Fehlermeldungen
15 Setup

ALLGEMEINE BESCHREIBUNG DER SOFTWARE

18 Ablauf einer Programmentwicklung
20 Ablauf des Assemblers
22 Ablauf des Linkers

ERSTELLEN DES QUELLPROGRAMMES

24 Editoraufruf
27 Format des Quellprogrammes
40 Macros

ASSEMBLIEREN VON QUELLPROGRAMMEN

44 Aufruf
46 Syntaxkontrolle
47 Fehlermeldungen

LINKEN VON PROGRAMMEN

50 Aufruf
52 Fehlermeldungen

LADEN VON PROGRAMMEN

54 Aufruf

ANWENDUNGSBEISPIEL

56 Beschreibung des Programmbeispieles
57 Ausdruck des Quellprogrammes (Datei.SRC)
61 Ausdruck des Assemblerlisting (Datei.LST)

ONLINE DEBUG

66 Aufruf

COMMUNICATION INTERFACE
68 Aufruf
70 Beschreibung der Befehle
79 Fehlermeldungen des comm Interface

TEXTASSEMBLER
80 Aufruf des Textassemblers
83 Format der Quelldatei
84 Befehlssatz
88 Anwendungsbeispiel

DISASSEMBLER
92 Aufruf
94 Fehlermeldungen
95 Format des Quellcodes
96 Beispiel

PCFLOW
98 Aufruf
100 Fehlermeldungen

XRF
102 Aufruf
105 Fehlermeldungen

FILE HANDLING
106 Aufruf

108 QUIT

A) Übersicht

Dieses provisorische Handbuch enthält Informationen über die Softwareinstallation und den Einsatz der neuen PCA-ASSEMBLER Softwarepakete.

Die neue Ausgabe des SAIA°PCA-ASSEMBLER verfügt über:
- Einfach handhabbare Menüs, wodurch die Verwendung der alten Stapelverarbeitungsdateien überflüssig geworden ist
- Hilfsseiten
- Neue Up/Downloader Programme mit Datenvergleich
- Neue EPROM Programmier utility
- Neues ONLINE Debug Programm
- Neue Utility Programme wie P10-Simulator, PCLIST Dokumentations-Programm,etc.
- Lauffähig auch auf IBM PS/2 Systemen

Die Softwarepakete sind so ausgelegt, dass sie auf folgenden Systemen lauffähig sind:

Minimalkonfiguration Hardware

- IBM-PC/XT/AT oder kompatibel sowie IBM PS/2
- min. 512K Bytes Arbeitsspeicher
- 2 Floppy-Laufwerke zu je 360K
- 1 oder besser 2 parallele Printerports (LPT1 zu Printer, LPT2 zu PCA)
- Monochrombildschirm
- SAIA PCA2.K43 Kabel für die Verbindung mit der PCA

Minimalkonfiguration Software

- Betriebssystem MS-DOS 3.0 oder höher
- Personal Editor (PE II) oder einen anderen Editor mit der Möglichkeit reine ASCII-Dateien anzulegen.

Ideal ist ein Gerät mit Festplatte, wobei der häufige Wechsel der Disketten wegfällt, sowie 640 KByte Arbeitsspeicher.
Der SAIA°PCA-ASSEMBLER ist in 3 sich ergänzenden Paketen erhältlich:

- PCASS 1 ONLINE	Das Basispaket enthält die Inbetriebnahme - Werkzeuge, mit welchen auch kleine Programme erstellt werden können.
- PCASS 2 PROGRAMMING	Das Hauptpaket, welches neben den Funktionen des PCASS 1 zusätzlich ASSEMBLER und LINKER enthält. Dieses Paket erlaubt die Entwicklung von grossen Programmen. Verwendung von mnemotechnischen Befehlen mit Symbolen und Labels erhöhen den Komfort und erlauben praxisgerechte Dokumentationen.Dazu ist ihr Textverarbeitungsprogramm erforderlich.
- PCASS 3 TRANSFER	Enthält Transfer- und Prüfprogramme

Paket PCASS 1 ONLINE

Lieferung auf 1 Diskette 360K: - Disk 1 PCASS 1 ONLINE

Inhalt der Diskette 'ONLINE':

LIESMICH	Diese Datei
PGUINI.COM	Initialisiert die Parallelschnittstelle zur PCA
PCA.EXE	Menü-Programm
PCAHLP.0xx	Hilfs-Dateien zu den Menüs
PCDNLD.EXE	Laden von Programmen zur PCA
PCUPLD.EXE	Programme von PCA auf Diskette laden (sichern)
ONLINE.EXE	Online-Zugriff zu PCA-Daten im RUN-Betrieb
CI-EXE	Zugriff zu PCA-Daten im Nicht-RUN-Betrieb
PCXRF.EXE	Crossreferenz-Liste von PCA-Programmen
PCFILE.EXE	Datei-Handhabung wie kopieren, löschen usw.
PCLIST.EXE	Erstellt eine Dokumentations-Datei vom PCA Speicher
P10.EXE	Simuliert das Programmiergerät P10

Paket PCASS 2 PROGRAMMING

Lieferung auf 3 Disketten 360K: - Disk 1 PCASS 1 ONLINE
- Disk 2 PCASS 2 ASSEMBLER
- Disk 3 PCASS 1-2-3 HELP

Inhalt von Diskette 1 'ONLINE':

LIESMICH	Diese Datei
PGUINI.COM	Initialisiert die Parallelschnittstelle zur PCA
PCA.EXE	Menü-Programm
PCDNLD.EXE	Laden von Programmen zur PCA
PCUPLD.EXE	Programme von PCA auf Diskette laden (sichern)
ONLINE.EXE	Online-Zugriff zu PCA-Daten im RUN-Betrieb
CI-EXE	Zugriff zu PCA-Daten im Nicht-RUN-Betrieb
PCXRF.EXE	Crossreferenz-Liste von PCA-Programmen
PCFILE.EXE	Datei-Handhabung wie kopieren, löschen usw.
PCLIST.EXE	Erstellt eine Dokumentations-Datei vom PCA Speicher
P10.EXE	Simuliert das Programmiergerät P10

Inhalt von Diskette 2 'ASSEMBLER':

PCKEY.PWD	Ihre registrierte Kunden-Identifikation
PCAS.EXE	PCA Macro Assembler
PCAS.00x	PCAS Überlagerungs-Dateien
PCLINK.EXE	Linker, Zusammenfügen von PCA Programmen
PCLINK.00x	PCLINK Überlagerungs-Dateien
PCTEXT.EXE	Text-Assembler
PCDIS.EXE	Disassembler
PCCOMP.EXE	Programmvergleich
PCFLOW.EXE	Flussdiagramm-Dokumentation
xxxx.SRC	Quellenprogramm Beispiele

Inhalt von Diskette 3 'HELP':

\D\PCAHLP.0xx	Hilfs-Dateien zu den Menüs in Deutsch
\F\PCAHLP.0xx	Hilfs-Dateien zu den Menüs in Französisch
\E\PCAHLP.0xx	Hilfs-Dateien zu den Menüs in Englisch

Paket PCASS 3 TRANSFER

Lieferung auf 1 Diskette 360K: - Disk 1 PCASS 3 TRANSFER

Inhalt:

LIESMICH	Diese Datei
PCPROM.EXE	EPROM-Ladeprogramm
ERASE.OBJ	RAM-Speicher Löschdatei (FFFF's)
RTA.COM	Laufzeit-Analyse (für PCA Kommunikation)
PCCONVST.EXE	Konvertiert OBJ-Dateien nach Standard INTEL Hex-Format
PCCONV.EXE	Konvertiert OBJ-Dateien nach INTEL MCS-86 HEX-Format

B) Wichtige MS-DOS Befehle

Um zu verstehen, wie der PCA-ASSEMBLER auf einem IBM-PC/XT/AT installiert wird, ist ein Minimum an MS-DOS oder PC-DOS Wissen erforderlich. Sie sollten sich (sofern Festplatte verwendet wird) über den Aufbau von Verzeichnissen (directories) auskennen oder wissen wie eine Diskette formatiert und Dateien kopiert werden. Auch einige Kentnisse über Stapeldateien sind erforderlich. Wenn Ihnen diese Begriffe noch nicht geläufig sind, sehen Sie sich vorgängig die Beschreibung der folgenden Kapitel in Ihrem DOS-Handbuch an:

Verwendung von Verzeichnissen (directories) in Baumstruktur
DOS-Befehle:

CHDIR bzw. CD	Wechselt Verzeichnis
COPY	Kopiert Dateien
FORMAT	Formatiert eine Diskette vor dem Gebrauch
MKDIR bzw. MD	Erstellt ein neues Verzeichnis
RMDIR bzw. RD	Löscht ein Verzeichnis

Stapelverarbeitungsdateien (Batch Files):
Diese enthalten mehrere DOS-Befehle und verkürzen langes Eintippen. (besonders die AUTOEXEC.BAT DATEI beachten)

C) Software Installation auf Festplatte

Dies ist eigentlich sehr einfach. In Kürze zusammengefasst: Kopieren Sie alle Dateien von allen Disketten, die Sie erhalten haben, in ein spezielles PCA-Verzeichnis und definieren Sie den Pfad so, dass DOS diese Dateien finden kann.

Am besten folgen Sie aber diesen 9 Schritten:

1) Erstellen Sie ein leeres Verzeichnis.
   Am besten benennen Sie es "\PCA". Sie können es natürlich auch anders nennen, wie z.B. "\PCANEU". Für die folgenden Beispiele benützen wir den Namen "\PCA". Erstellen Sie nun das Verzeichnis mit dem Befehl:

   MD C:\PCA

2) Kopieren Sie alle PCA-Assembler Dateien von allen Disketten in dieses Verzeichnis.
   Kopieren der Disketten PCASS 1 ~~PCASS 2~~ und PCASS 3 :
   Diskette ins Laufwerk A: einlegen und jedesmal den folgenden Befehl ausführen lassen:

   COPY A:*.* C:\PCA

   Kopieren der Hilfe-Dateien auf der Diskette PCASS 1-2-3 für:

   Hilfe in Deutsch: COPY A:\D C:\PCA oder
   Hilfe in Englisch: COPY A:\E C:\PCA

3) Erstellen Sie eine Sicherungskopie der AUTOEXEC.BAT Datei

   COPY C:\AUTOEXEC.BAT C:\AUTOEXEC.BAK

4) Definition des Suchpfades für MS-DOS und Initialisieren der parallelen Schnittstelle:
   Oeffenen Sie mit Hilfe Ihres Textverarbeitungsprogrammes die Datei AUTOEXEC.BAT und editieren folgende Zeilen:

   SET PCA=C:\PCA
   C:\PCA\PGUINI

   Dieser Befehl wird die parallele Schnittstelle für die Kommunikation mit der PCA initialisieren. Wenn er nicht ausgeführt wird und man eine PCA, die sich im RUN Mode befindet, mit dem IBM-PC verbindet, wird das PCA Programm gestoppt.
   Fügen Sie der Zeile mit der PATH Anweisung folgendes hinzu

   ;C:\PCA

   Fehlt die PATH Anweisung geben Sie folgende Zeile ein:

   PATH C:\PCA

   Der PATH-Befehl definiert die Suchpfade, in welchen das PCA-Menü (und das MS-DOS) Programme sucht, welche sich beim Aufruf nicht im atuellen Verzeichniss befinden.

Wenn DOS diese Stapeldatei ausführt, wird nun automatisch die Umgebung Ihres Systemes für die Arbeit mit dem PCA-Assemblerpaket konfiguriert. Programmierer haben oft verschiedene Stapelverarbeitungsdateien im Stammverzeichnis zur Konfiguration des Systemes für verschiedene Aufgaben. Beispielsweise konfiguriert TC.BAT das System für den Gebrauch von TURBO C, TP.BAT für das TURBO PASCAL und PCA.BAT für den Gebrauch des PCA-ASSEMBLER usw.

Achtung! Zu diesem Beispiel müssen Sie anstelle von C:\PCA den von
-------- Ihnen im Schritt 1 gewählten Verzeichnisnamen benützen.

Speichern Sie die AUTOEXEC.BAT Datei und verlassen Sie das Textverarbeitungsprogramm.

5) Editieren Sie die folgende Zeile in der CONFIG.SYS Datei:

DEVICE=ANSI.SYS

Dies erlaubt die Unterstützung der Standard ANSI Ein/Ausgabe-Sequenzen, die für die Bildschirmsteuerung gebraucht werden (siehe DOS-Handbuch).

Bemerkung: Einige IBM kompatible PC's haben ein ANSI.EXE (oder ANSI.COM) Programm anstelle von ANSI.SYS. Dieses Programm kann direkt von der AUTOEXEC.BAT Datei aufgerufen werden ohne die CONFIG.SYS Datei zu ändern.

6) Mit den Tasten Ctrl-Alt-Del rufen Sie die DOS Grundstellung auf, so dass die Dateien CONFIG.SYS und AUTOEXEC.BAT ausgeführt werden.

7) Lesen Sie nun das Kapitel E, "Verwendung von Verzeichnissen auf Festplatte". Erstellen Sie dann ein Arbeitsverzeichnis, wechseln Sie in das erstellte Verzeichnis (mit CD) und geben Sie folgenden Befehl ein:

PCA

Als Variante könnte eine Stapelverarbeitungsdatei sogar Ihr Arbeitsverzeichnis direkt aufrufen und den PCA-Assembler von dort aus aufstarten. Dies geschieht mit folgenden Befehlen:

```
CD C:\%1
C:\PCA\PCA
```

Das Zeichen %1 gilt für einen Parameter. Auf diese Weise kann man mit der gleichen Stapeldatei verschiedene Arbeitsverzeichnisse aufrufen. Der Befehl PCA PROJEKT1 zum Beispiel, würde die Stapeldatei PCA.BAT mit PROJEKT1 als Parameter ausführen. So würden wir uns im Verzeichnis PROJEKT1 befinden.
Bemerkung: Bei dieser Variante müssen Sie, bevor Sie die Stapeldatei aufrufen, sicher sein, dass das Projektverzeichnis existiert (wurde mit "MD Projekt1" erstellt)

8) Wählen Sie "Setup" vom Hauptmenü aus, indem Sie die Taste "S" drücken und geben Sie den Namen von Ihrem Editor-Programm (wird nur vom Paket PCASS 2 Programming unterstützt) und das Seitenformat für den Drucker ein.
Der Editor wird zusammen mit dem Namen der zu erstellenden Datei aufgerufen. Zum Beispiel wenn Ihr Editor PE genannt wird und Sie die Datei FRITZ.SRC erstellen, führt das PCA-Assembler Menü den Befehl "PE FRITZ.SRC" aus.

Wenn dieses Vorgehen nicht geeignet ist für Ihren Editor, können Sie ihn auch mit einer Stapelverarbeitungsdatei, die die zusätzlichen Informationen beinhaltet, aufrufen. Um dies zu machen, müssen Sie anstelle des Editor Namens im "Setup" den Namen der Stapeldatei eingeben.

9) Lesen Sie jeden Help Bildschirm, indem Sie im Hauptmenü das jeweilige Untermenü (z.B. ONLINE) und dann die Taste "F1" anwählen.

10) Nun sind Sie startbereit.

D) Software Installation auf System mit 2 Floppy Laufwerken

(ohne Festplatte)

Da die üblichen 2 Diskettensysteme mit 360K Byte Datenlaufwerken nicht ausreichen um alle Programme vom PCA-ASSEMBLER zu speichern, ist eine Aufteilung auf verschiedene Disketten notwendig.Diese Disketten müssen dann abwechslungsweise ins Laufwerk A eingelegt und wieder hinaus genommen werden, je nach Programm, das Sie ausführen möchten.

Das Laufwerk A sollte für das DOS Betriebssystem einige DOS und PCA-ASSEMBLER Programme benützt werden. Das Laufwerk B sollte für Ihre Arbeitsdiskette mit den Anwenderprogramm-Dateien reserviert werden Während der Arbeit bleiben Sie im Laufwerk A. Dies bedingt, dass Sie jeweils bei jeder Dateinamen-Eingabe die Laufwerkbezeichnung "B:" benützen, so dass Ihre Arbeitsdatei gefunden und/oder auf Ihre Arbeitsdiskette im laufwerk B gespeichert wird.

Wir empfehlen Ihnen für die Arbeit im Laufwerk A, 3 Disketten vorzubereiten. Jede Diskette muss für das Enthalten von Betriebssystemdateien formatiert sein. Benützen Sie das Untermenü "File handling, option FORMAT" oder den DOS-Befehl "FORMAT B:/S" für die Initialisierung dieser Disketten, bevor Sie Dateien kopieren.

Das Kopieren wird wie folgt ausgeführt:
Legen sie die SAIA°PCA-ASSEMBLER Diskette mit dem zu kopierende Programm ins Laufwerk A und eine von den 3 vorbereiteten Disketten ins Laufwerk B. Mit dem Befehl COPY A:DATEINAME B:, kopieren sie nun die gewünschten Programme.

Die Arbeitsdisketten für das Laufwerk B können ohne Platz für das Betriebssystem formatiert werden (mit dem Befehl FORMAT B:). Der Gebrauch von Verzeichnissen ist bei der Arbeit mit 360K Byte Disketten nicht empfehlenswert.
Die 3 Disketten können die in der Folge aufgelisteten Dateien beinhalten. Welches Programm auf welcher Diskette sein muss, ist abhängig von der Häufigkeit des Gebrauchs vom jeweiligen Programm.

Wenn die hier unten aufgelisteten Dateien nicht auf einer Diskette Platz haben, verschieben Sie eine oder mehrere Dateien auf eine andere Diskette. Beachten Sie aber, dass die PCAS.xxx, die PCLINK.xxx und die PE.xxx Dateien als Blöcke zusammengehören und dass es gewisse Vorteile bringt, wenn diese 3 Dateiblöcke auf der gleichen Diskette zu finden sind. Falls Help Dateien gewünscht werden, müssen die PCAHLP.xxx Dateien auf jeder Diskette sein. Für einen normalen Ablauf des PCA-ASSEMBLER Pakets sind sie jedoch nicht notwendig. Sehr Hilfreich ist es, die PCFILE.EXE Datei auf jeder Diskette zu haben.
Wenn Sie vom Hauptmenü oder von einem Untermenü versuchen, ein Programm zu starten und die Meldung "PROGRAMM NOT AVAILABLE" erscheint, heisst das, dass Sie die Diskette mit dem jeweilig gewünschten Programm ins Laufwerk A einlegen müssen.

Diskette 1: Startup Diskette, im Laufwerk A beim Aufstarten.

```
COMMAND.COM
AUTOEXEC.BAT    (siehe nachfolgend)
CONFIG.SYS      (siehe nachfolgend)
ANSI.SYS        (oder ANSI.EXE auf einigen Systemen)
KEYBxx.COM      (Ihr Tastatur-definitionsprogramm, z.B. KEYBGR.COM)
PCA.EXE
PCKEY.PWD       (wenn Sie mit dem Paket PCASS2 oder 3 arbeiten)
```

Diskette 2: Editer, Assembler, Linker und Downloader im Laufwerk A zum Editieren, Assemblieren, Linken und Downloaden von PCA Programmen.

```
COMMAND.COM
PRINT.COM
FORMAT.COM
PGUINI.COM
PCAHLP.001 - PCAHLP.018
PCAS.EXE
PCAS.001 - PCAS.004
PCLINK.EXE
PCLINK.001 - PCLINK.003
PCDNLD.EXE
PCFILE.EXE
PE.EXE          }        (Editor Dateien, in diesem Beispiel für
PE.PRO          }         den Personal Editor)
PE.HLP          }
```

Diskette 3: Debug, Downloader, Uploader und Utilities, im Laufwerk A zum Testen von Programmen.

```
COMMAND.COM
PRINT.COM
FORMAT.COM
PCAHLP.001 - PCAHLP.018
PCFILE.EXE
ONLINE.EXE
CI.EXE
PCUPLD.EXE
PCDNLD.EXE
PCXRF.EXE
PCDIS.EXE
PCCOMP.EXE
PCFLOW.EXE
RTA.COM
PCCONV.EXE
PCCONVST.EXE
PCPROM.EXE
```

Die AUTOEXEC.BAT Datei muss folgende Zeilen beinhalten:

```
PGUINI
PATH A:\;B:\
SET PCA=A:\
PCA
```

Die CONFIG.SYS Datei muss nachfolgende Zeile enthalten (siehe auch die Bemerkung im Kapitel C, "Software Installation auf Festplatte", Schritt 5)

```
DEVICE=ANSI.SYS
```

Starten Sie Ihr System mit der Diskette 1 im Laufwerk A und Ihre Arbeitsdiskette im Laufwerk B. Das Hauptmenü erscheint automatisch, da die AUTOEXEC.BAT Datei den PCA-ASSEMBLER aufruft.

Mit der Diskette 2 können die Untermenüs Edit, Assemble, Download, und File Handling aufgerufen werden. Für die anderen Untermenüs legen Sie die entsprechende Diskette ins Laufwerk A ein, bevor Sie es vom Hauptmenü aus aufrufen.

E) Verwendung von Verzeichnissen (directories) auf Festplatte

Wenn das SAIA PCA-ASSEMBLER Paket richtig installiert ist, die Befehle "PATH" und "SET PCA=Verzeichnis" von der Stapelverarbeitungsdatei (die Sie in Schritt 3 erstellt haben) ausgeführt worden sind, dann kann das Paket von jedem Verzeichnis aus gestartet werden.
Es ist zu empfehlen, dass Sie für jedes Projekt oder grösseres Programm ein anderes Verzeichnis verwenden. Arbeiten Sie also nie im SAIA-PCA Programm Verzeichnis, wenn Sie nicht ein Verzeichnis wollen, das aus einer Mischung von SAIA-Programmen und Ihren SRC, LNK, OBJ etc. Dateien besteht!!
Jedes Arbeitsverzeichnis muss die folgenden Dateien beinhalten, um Assembler, Linker und Editor aufstarten zu können:

PCAS.001 - PCAS.004	Übergeordnete Assembler Dateien
PCLINK.001 - PCLINK.003	Übergeordnete Linker Dateien
PCKEY.PWD	Ihre Identifikations-Datei
PE.PRO (oder PE2.PRO)	Editor's Schlüssel PRODATEI (wenn Sie
PE.HLP (oder PE2.HLP)	Editor's Help Datei PE benützen)

Die Kopien von den Originaldateien sollen im SAIA-PCA Programmverzeichnis bleiben, sodass sie auf einfache Art in jedes Arbeitsverzeichnis kopiert werden können. Sie können auch eine Stapelverarbeitungsdatei erstellen, die Sie von jedem neuen Arbeitsverzeichnis aus aufrufen und so automatisch die benötigten Dateien kopiert werden.

Hier ein Beispiel; wir wollen die Stapeldatei MAKEWORK.BAT erstellen (weiter unten aufgelistet). Diese sollte sich im SAIA-PCA Programmverzeichnis befinden. Für die Benützung dieser Datei, eröffnen Sie mit dem Befehl MD PROJEKT das neue Arbeitsverzeichnis, gehen sie dort hinein mit dem Befehl CD PROJEKT und geben sie schliesslich

MAKEWORK <CR> ein.

Auflistung der Stapelverarbeitungsdatei MAKEWORK.BAT:

```
COPY \PCA\PCLINK.00?
COPY \PCA\PCAS.00?
COPY \PCA\PCKEY.PWD
COPY \PCA\PE.PRO
COPY \PCA\PE.HLP
```

Wenn Sie mit der DOS-Version 3.3 oder höher arbeiten, können Sie den Befehl APPEND benützen, um temporär ihr Arbeitsverzeichnis mit dem PCA Programmverzeichnis zusammenzufügen. In diesem Fall ist es nicht notwendig, diese Dateien zu kopieren (sehen sie im DOS-Handbuch nach).

```
APPEND C:\PCA
```

Von den SAIA°PCA-ASSEMBLER Menüs können Sie nicht ein anderes Verzeichnis adressieren, da einige SAIA-Programme keine Verzeichnisnamen, sondern nur Laufwerknamen ausfindig machen können. Um dies trotzdem machen zu können, benützen Sie den SUBST Befehl um ein Verzeichnis mit einem Laufwerknamen umzubenennen. Dieser neue Laufwerkname kann jetzt von den SAIA-Menüs aus benützt werden, anstelle von Verzeichnissen. Laufwerke D und E (oder höher) können verwendet werden.

Laufwerk B wird von Systemen, die nur ein Floppy A besitzen, nicht unterstützt (MS-DOS reassigniert automatisch Laufwerk B als Laufwerk A, wenn Laufwerk B aufgerufen worden ist, das physikalisch nicht existiert).

Wenn Sie von einem Menü aus versuchen, Laufwerk B zu adressieren das jedoch physikalisch nicht existiert, erscheint die Fehlermeldung "Invalid file name". Sie können auch Laufwerke F bis Z für die Umbenennung benützen, wenn Sie in der Datei CONFIG.SYS den DOS-Befehl LASTDRIVE=n entsprechend anpassen. Sehen Sie im DOS-Handbuch nach.

ACHTUNG: Wenn ein SUBST Befehl aktiv ist, dürfen Sie die folgenden Befehle für das umbenannte Laufwerk nicht benützen:

ASSIGN, BACKUP, DISKCOMP, DISKCOPY, FDISK, FORMAT, JOIN, LABEL or RESTOR.

F) Arbeiten mit Hilfe der Menüs (mit Fehlermeldungen)

Um das Menü-Programm aufzurufen, wird nach dem DOS-Prompt "PCA" eingegeben. Es erscheint das SAIA-Logo mit Angaben über Eigentumsvorbehalt und Ihre registrierte Kundenidentifikation. Durch Betätigen einer beliebigen Taste wird das Hauptmenü sichtbar. Werden schnell nacheinander zwei Tasten gedrückt (z.B. PCA<CR><CR>), so wird direkt zum Hauptmenü gesprungen.

Über das Hauptmenü können jetzt die verschiedenen Programme des SAIA°PCA-ASSEMBLERS aufgerufen werden. Bevor Sie aber zu arbeiten beginnen (sofern Sie das Paket PCASS 2 besitzen), sollten Sie im Untermenü "Setup" den Namen Ihres Editors (z.B. PE oder WS) und das Seitenformat für den Printer eingeben.

Ist eine Arbeit abgeschlossen, so wird über das Untermenü "Quit" wieder auf die DOS-Ebene zurückgekehrt. Das Programm "Quit" bewirkt aber, dass alle Informationen, welche Sie zu den einzelnen Untermenüs (im aktuellen Verzeichnis) eingetippt haben, in der Datei PCAMENU.DAT abgespeichert werden. Wenn Sie aus diesem Verzeichnis das nächste Mal den PCA-ASSEMBLER aufrufen, so können Sie dank der abgespeicherten Daten alle Untermenüs wieder in der genau gleichen Weise betreiben, wie Sie diese das letzte Mal verlassen haben.

Drei verschiedene Arten von Eingabefeldern werden in den Menüs verwendet. Wenn sich der Cursor auf einem Feld befindet, wird es immer in Negativschrift dargestellt. Um von einem zum nächsten Feld zu springen, werden die Pfeiltasten oder der Tabulator verwendet.

1) Wahlfeld

Dieses erscheint im Hauptmenü. Das Wahlfeld erhält ein Stichwort für das entsprechende Untermenü. Das angewählte Feld erscheint in heller Negativschrift. Durch Drücken von <CR>(Enter) kommt das zugehörige Untermenü auf den Bildschirm.
Um ein anderes Feld anzuwählen, können sowohl die Tasten "Leerschlag" als auch "Tab" oder die Cursorpfeile verwendet werden. Alternativ dazu kann auch über die Taste entsprechend den hellen Grossbuchstaben, die zugehörige Funktion direkt gewählt werden.

2) Eingabefeld

In dieses Feld können numerische oder Text-Daten eingegeben werden. Es werden nur gültige Eingaben angenommen, ungültige Angaben werden mit Piepston zurückgewiesen.

Alle Editiertasten wie "Einschieben" (INS), "Löschen" (DEL), "Rück" (BS), Links-/Rechtspfeile sowie "Home" und "End" können verwendet werden.

Felder zur Eingabe von Dateinamen nehmen nur gültige MS-DOS Dateinamen an. Hinweis: Nur 1 Punkt "." ist im Dateiname erlaubt. Ist schon ein Punkt im Eingabefeld, so muss dieser zuerst gelöscht werden, falls er an einer anderen Stelle erscheinen soll.

MS-DOS Dateinamen bestehen aus max. 8 Charaktern mit einer optionalen Erweiterung von nochmals max. 3 Charaktern. Werden mehr als 8 bzw. 3 Charakter ins Feld eingegeben, so werden die überzähligen ignoriert. Endet der Dateiname mit "." , so wird keine Erweiterung hinzugefügt. Mit der Taste "Leerschlag" wird der Rest des Eingabefeldes leer gelassen und zum folgenden Feld gewechselt.

3) Umschalt-Feld

In diesem Feld können zwei oder mehrere Alternativen durch drücken der Taste "Leerschlag" gewählt werden (z.B. YES/NO).

Hilfsseiten zu den Menüs

Aus der Position des Hauptmenüs und wenn ein Untermenü auf dem Bildschirm steht, kann mit der Taste "F1" eine Hilfsseite auf den Bildschirm gebracht werden. Diese Textseiten sind einzeln abgelegt in den Dateien PCAHLP.001 bis 018.

Fehlermeldungen zu den Menüs

Wird ein Fehler festgestellt, so wird der negative Cursor über das Eingabefeld gelegt, wo der Fehler ist.

ERROR 1: CAN'T OPEN FILE <Dateiname>

Die entsprechende Datei befindet sich nicht im aktuellen Verzeichnis oder im angegebenen Laufwerk.

ERROR 2: NO FILE NAME SPECIFIED

Bevor die Operation gestartet werden kann, muss ein Dateiname eingegeben werden.

ERROR 3: CANNOT PRINT PGD AND LST AT THE SAME TIME

Im Untermenü "Assembler" wurde sowohl für LST als auch für PGD ein "YES" eingegeben. Der Ausdruck beider Listen muss nacheinander als zwei getrennte Operationen aufgerufen werden.

ERROR 4: PROGRAM LOAD FAILURE ON: <Programm-Name>

Das Programm kann nicht geladen und ausgeführt werden. Dies tritt auf, wenn die System-Datei COMMAND.COM nicht gefunden wird oder wenn nicht genügend Speicherplatz zum laden muss ausführen des Programmes vorhanden ist.

ERROR 5: WRITE ERROR ON FILE: <Dateiname>

Folgende Gründe:
- Diskette ist voll
- Diskette oder Datei ist schreibgeschützt
- keine Diskette im Laufwerk

ERROR 6: OUTPUT FILE HAS SAME NAME AS INPUT FILE

Im "Linker"-Programm wird eine provisorische LNK-Datei oder eine PGD-Datei erzeugt, die den gleichen Namen hat wie eine Eingabedatei. Der Name der Ausgabedatei muss geändert werden.

ERROR 7: COMMAND.COM NOT FOUND, OR NOT ENOUGH MEMORY

Ähnlich wie ERROR 4, jedoch ausgelöst, da das Betriebssystem (nicht das PCA-Menü) seinen Befehl nicht ausführen kann.
Dies weil:
- COMMAND.COM nicht gefunden wird
- nicht genügend Speicher vorhanden ist, um das Programm zu laden und auszuführen.

ERROR 8: INVALID VALUE

Der Wert im erhellten Feld ist ausserhalb des zulässigen Bereiches.

ERROR 9: INVALID ADDRESS

Die Adresse kann zwischen 0 und 8191 bzw. 16383 liegen, abhängig vom PCA-Typ.

ERROR 10: INVALID FILE NAME: <Dateiname>

Der erhellte Dateiname ist kein gültiger MS-DOS Dateiname:
- Dateinamen PRN, LPT1, LPT2, LPT3, CON, AUX, COM1 und COM2 können hier nicht als Eingangs-Dateinamen benützt werden.
- CON, AUX, COM1, und COM2 können hier nicht als Ausgangs-Dateinamen benützt werden.
- Sie versuchen das Laufwerk B aufzurufen; physikalisch ist es jedoch nicht vorhanden.
- Sie dürfen den Befehl SUBST für das Laufwerk B nicht verwenden; mit SUBST muss Laufwerk D: oder höher eingesetzt werden.

ERROR 11: PROGRAM NOT AVAILABLE: <Programm-Name>

Das angezeigte Programm existiert nicht (ist nicht Bestandteil des gekauften PCA-Softwarepaket) oder kann nicht gefunden werden.
Der Pfad-Befehl befindet sich normalerweise in der Datei AUTOEXEC oder in einer Stapeldatei. Von hier aus muss das Verzeichnis aufgerufen werden, in welchem sich die SAIA-Programme befinden. Lesen Sie die Informationen in den Kapiteln Systeminstallation dieses provisoriches Handbuches nach oder konsultieren Sie Ihr DOS-Handbuch.
Tritt dieser Fehler beim Arbeiten mit dem Programm "Edit" auf, so muss geprüft werden, ob der im "Setup"-Menü eingegebene Editor-Name mit dem gewünschten Namen übereinstimmt und ob der Editor über den Pfad gefunden werden kann.

ERROR 12: CANNOT PRINT RTA AND MOD AT THE SAME TIME

Im Untermenü "Runtime analysis", können Sie entweder die RTA-Datei oder die MOD-Datei an den Drucker senden aber nicht beide zusammen.

G) Verbindung zwischen PCA und IBM-PC

Auf dem IBM-PC muss ein freies Parallel-Port vorhanden sein für den Anschluss des Kabels PCA2.K43. Dieses wird am anderen Ende am PGU-Stecker (Programmgeräte-Anschluss) der PCA eingesteckt.

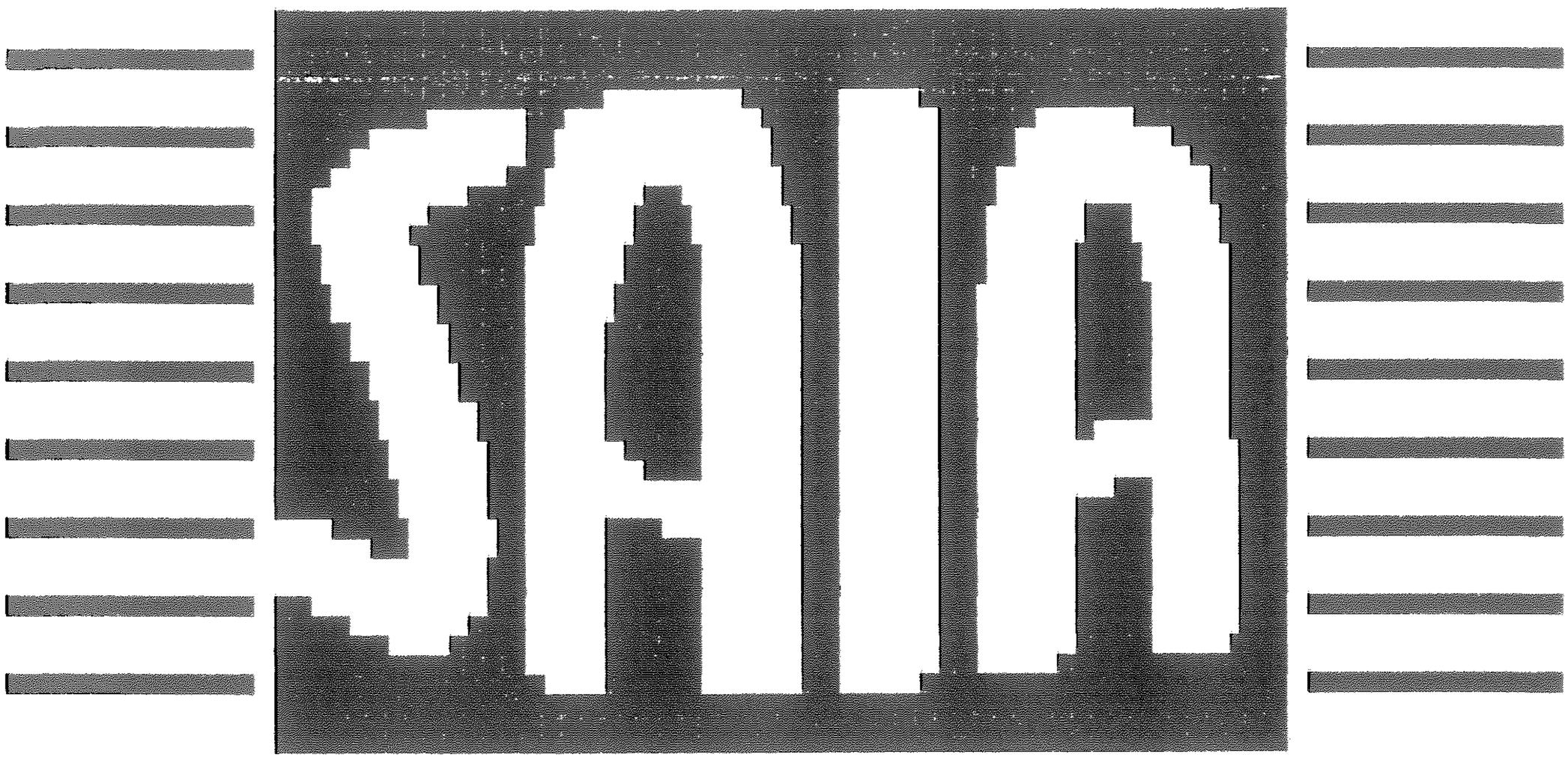
Das Port mit der höheren Nummer muss verwendet werden (LPT1, LPT2 oder LPT3). Z.B. wenn Sie zwei Parallel-Ports (LPT1 und LPT2) haben, verwenden Sie LPT1 für den Printer und LPT2 für die PCA. Verfügt Ihr IBM-PC nur über 1 Parallel-Port, so kann nicht gleichzeitig mit beiden Geräten kommuniziert werden.

*** ACHTUNG !! Wichtig ***

Befindet sich Ihre PCA im RUN-Mode, so ist es unerlässlich, dass die parallele Schnittstelle von Ihrem IBM-PC mit dem Befehl PGUINI (siehe Kapitel C-4) initialisiert wurde bevor sie die Verbindung mit dem PCA2.K43 Kabel herstellen.
Um Störungen auf der PCA, dem Kabel oder auf dem IBM-PC zu vermeiden, beachten Sie bitte folgendes:

- Der IBM-PC und die PCA müssen an der gleichen Erdung angeschlossen sein. Es darf kein Potentialunterschied sein zwischen den beiden Erdungen.
- Beim Ein- und Ausstecken des Kabels, dürfen Kabel-Pins nicht mit den geerdeten Teile des Steckers in Berührung kommen.
- Das Kabel PCA2.K43 darf nicht auf ein serielles Port eingesteckt werden (mit einem Adapter könnte es erzwungen werden).

Wenn Sie all diese Ratschläge beherzigen, werden Sie mit Freude rationell PCA-Programme erstellen können .



```
SAIA PCA ASSEMBLER - VERSION 19-07-88
Copyright (C) SAIA AG 1984-1988
Press any key to continue
```

```
SAIA PCA ASSEMBLER V1.1                                                MAIN MENU
          ***** SAIA AG   Marktbereich CH   3280 Murten  *****
Directory: C:\PCASS                                          16.12.88 13.39

   Edit                    Text assembler          Compare programs
   Assemble                Disassembler            Xref listing
   Link                    flow chart              File handling
   Up/download             Runtime analysis        Ms-dos command
   Online debug            Hex converter           Setup
   comms Interface         Program eproms          Quit

<ARROW>, <SPACE> or <Tab> selects operation, <CR> or <Command letter> executes
```

```
SAIA PCA SETUP MENU

   Name of EDITOR program . . . . . . . . .   PE_____

   PRINTER PAGE SIZE DEFINITION:
   Blank lines at top of page . . .  (0-20)    0_
   Lines per page . . . . . . . .  (20-255)    66_
   Number of characters on a line  (80-104)    104
   Blank lines at end of page . . .  (0-20)    6_
   Spaces for left margin . . . . .  (0-10)    0_
   Use form feed character  . . . . . . .      Yes

Enter editor program name, <ARROW> or <Tab> moves cursor, <CR> or <ESC> exits.
```

SETUP HILFSSEITE

Definiert den Namen Ihres Textverarbeitungsprogrammes (Editor) und das Seitenformat für die Ausgabe von LST, PGD, CRF, DIS, SYM oder TED Dateien. Durch den "PATH" Befehl muss festgelegt werden, wo die Editordatei vom Typ EXE, COM oder BAT gesucht werden muss.
Im Editoraufruf wird ausserdem die zu editierende Datei angegeben. Kann die Angabe in Ihrem Editor nicht gemacht werden, wird der Aufruf via Batchdatei realisiert. Der Batch-Parameter %1 ist der Name der zu editierenden Datei. Die Batchdatei muss resident im SAIA PCASS Verzeichnis sein. Für Details sehen Sie in Ihrem Handbuch nach.

Das Festlegen des Seitenformates ist selbsterklärend, ausser vielleicht dem Seitenvorschub. Normalerweise wird der Seitenvorschub mit einem ASCII-Charakter gesteuert; funktioniert das bei Ihrem Drucker nicht, setzen Sie diese Option auf "NO" (der Zeilenvorschub übernimmt diese Funktion).

Setzen Sie beim Drucken einer LST-Datei die "Number of characters on a line" auf 104. Kann Ihr Drucker die Anzahl Charakter nicht verarbeiten (werden Zeichen unterdrückt), kann der Drucker auf komprimierte Schrift umgeschaltet werden (siehe Druckerhandbuch).

ALLGEMEINE BESCHREIBUNG

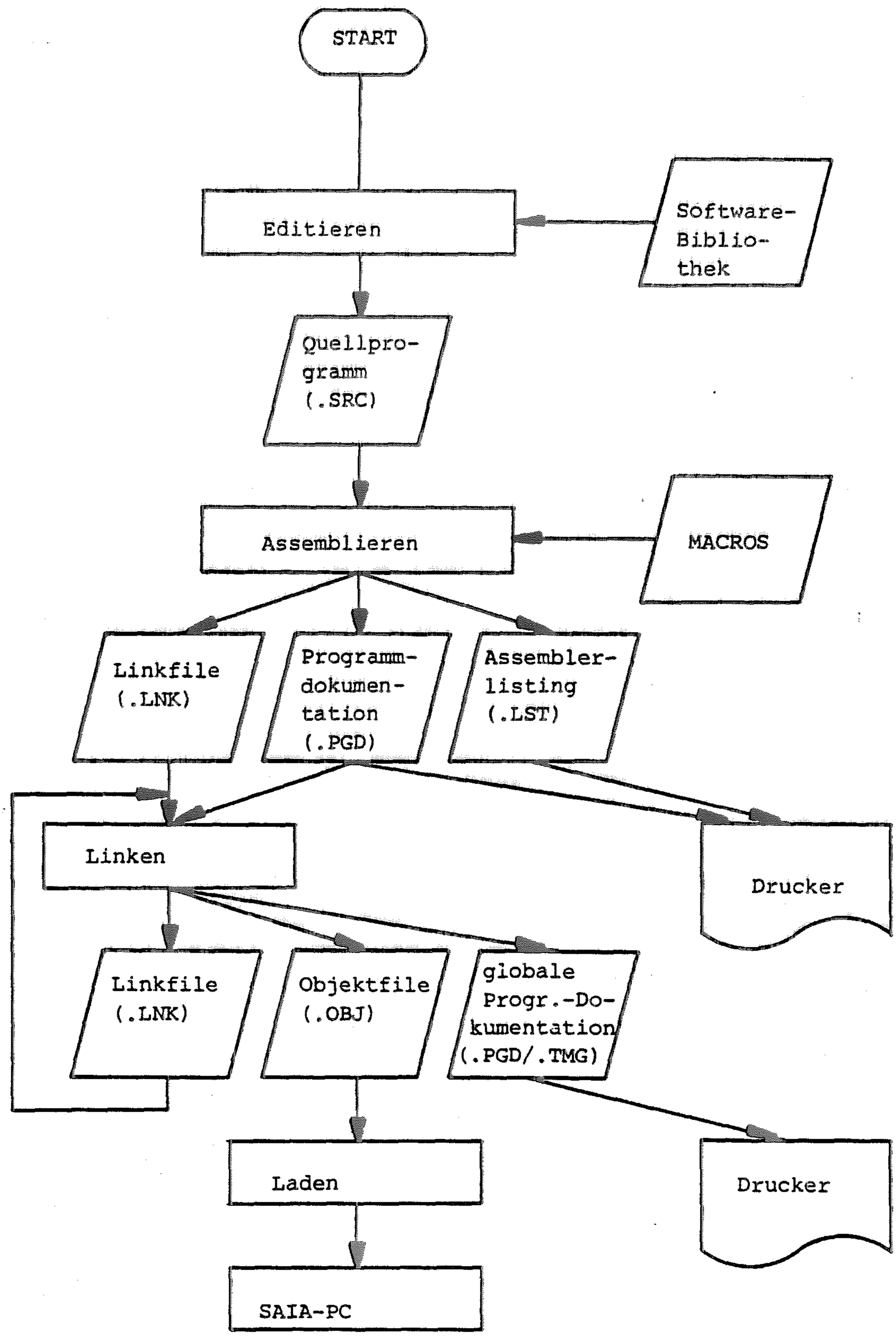
ABLAUF EINER PROGRAMMENTWICKLUNG

(Siehe Abbildung auf der folgenden Seite)
Mit dem Editor oder einem Textverarbeitungsprogramm wird ein Quellprogramm vom Typ SRC erstellt, das Macros und Programme aus einer Programmbibliothek enthalten kann.
Die Assemblierung des Quellprogrammes erzeugt wahlweise Dateien (Files), die zur Dokumentation des Programms oder zum Linken (Verbinden) mit andern assemblierten Programmen dienen. Die Ausgabedateien des Assemblers sind:

- eine Linkdatei (LNK)
- eine Programmdokumentation (PGD)
- ein Assemblerlisting (LST)

In einem nächsten Schritt - dem Linkvorgang - werden die Linkdateien zu einem einzigen Programm zusammengefügt. Dabei kann wahlweise eine wieder linkbare Datei (LNK) oder eine ladbare Datei (OBJ) erzeugt werden.
Die Programmdokumentationen können ebenfalls gelinkt werden.
Im letzten Schritt wird die Objektdatei über die parallele Schnittstelle in die SAIA-PC geladen.
Druckbar sind die Quelldatei, das Assemblerlisting und die Programmdokumentation. Linkdateien oder Objektdateien können nicht ausgedruckt werden.

PROGRAMMENTWICKLUNG MIT DEM SAIA-PC-ASSEMBLER



BESCHREIBUNG DER SOFTWARE

ABLAUF DES ASSEMBLERS

Der Assembler erstellt in drei Programmdurchläufen (PASS) ein linkbares PC-Programm.

PASS 1: Aus dem Quellprogramm und den Macros wird ein erweitertes Quellprogramm (Quellprogramm mit Macrocode), eine Tabelle der verwendeten Symbole und eine Tabelle der verwendeten Macros erzeugt.

PASS 2: Aus dem erweiterten Quellprogramm und der Symboltabelle werden folgende dem Benutzer zur Verfügung stehende Dateien erzeugt:

- Assemblerlisting Dateiname.LST
- linkfähige Datei Dateiname.LNK
- Programmdokumentation Dateiname.PGD

In diesen Dateien sind allen Symbolen und Labels bereits relative oder absolute Adressen zugeordnet.

PASS 3: Die Symboltabelle (mit absoluten Adressen) und die Cross Reference-Tabelle werden erzeugt und in die gewünschte Form gebracht.

Wieviele Labels, Symbole und Macros können maximal verarbeitet werden ?

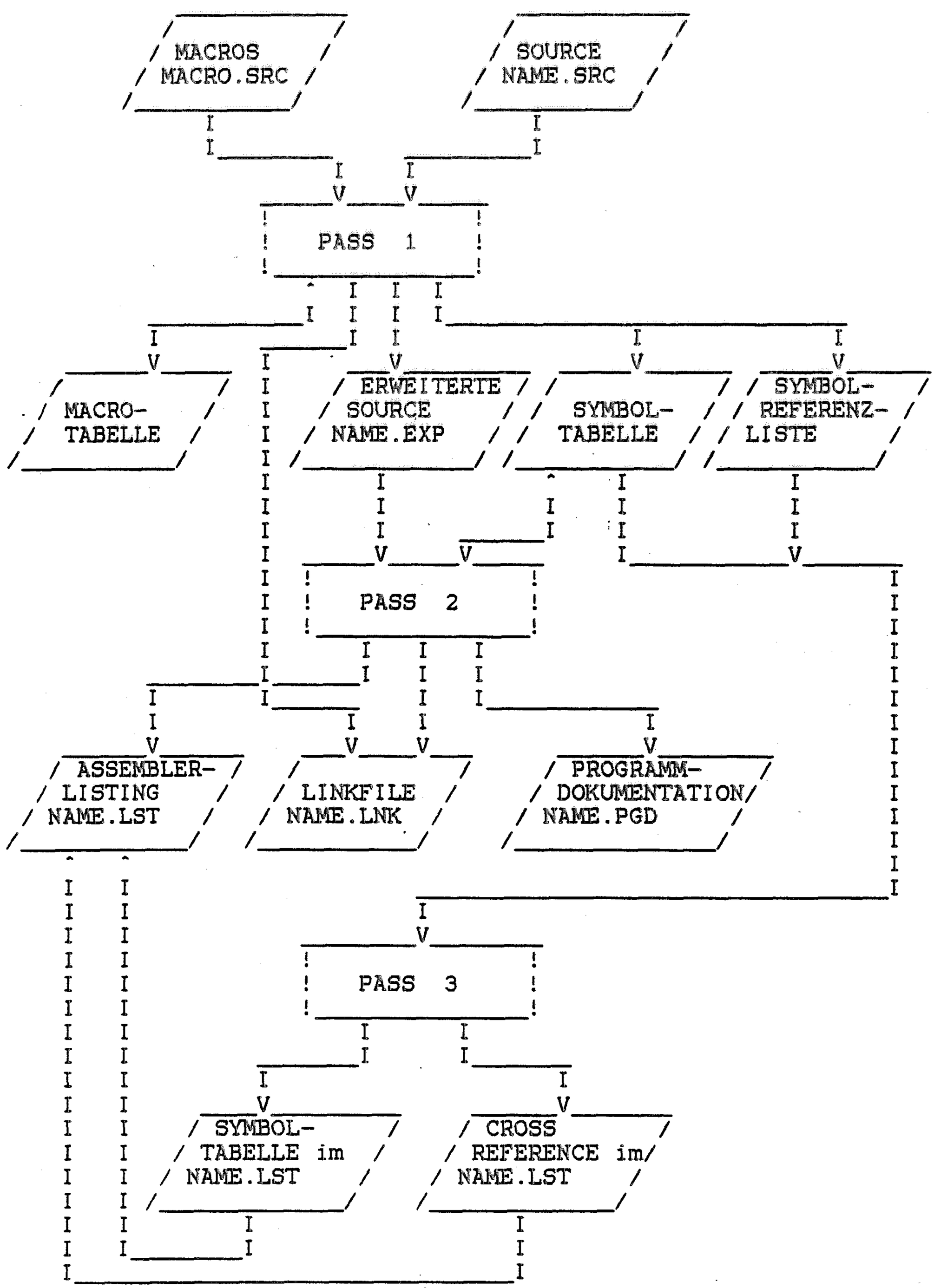
Der Bedarf an Speicherkapazität hängt von verschiedenen Faktoren wie Symbollänge, Anzahl der Referenzen bei Cross Reference usw. ab. Folgende Angaben mögen einen Hinweis geben, wo die Grenzen des Systems liegen:

Anzahl Labels plus Symbole : ca. 1500
Anzahl Macros : ca. 75
Diese Werte gelten unter der Annahme, dass eine Cross Reference mit 2 Referenzen pro Symbol oder Label erstellt wird. Ohne Cross Reference erhöht sich der zur Verfügung stehende Speicherplatz.

Die Abbildung auf Seite A-14 zeigt den Ablauf des Assembliervorganges.

BESCHREIBUNG DER SOFTWARE

PHASEN des ASSEMBLERS



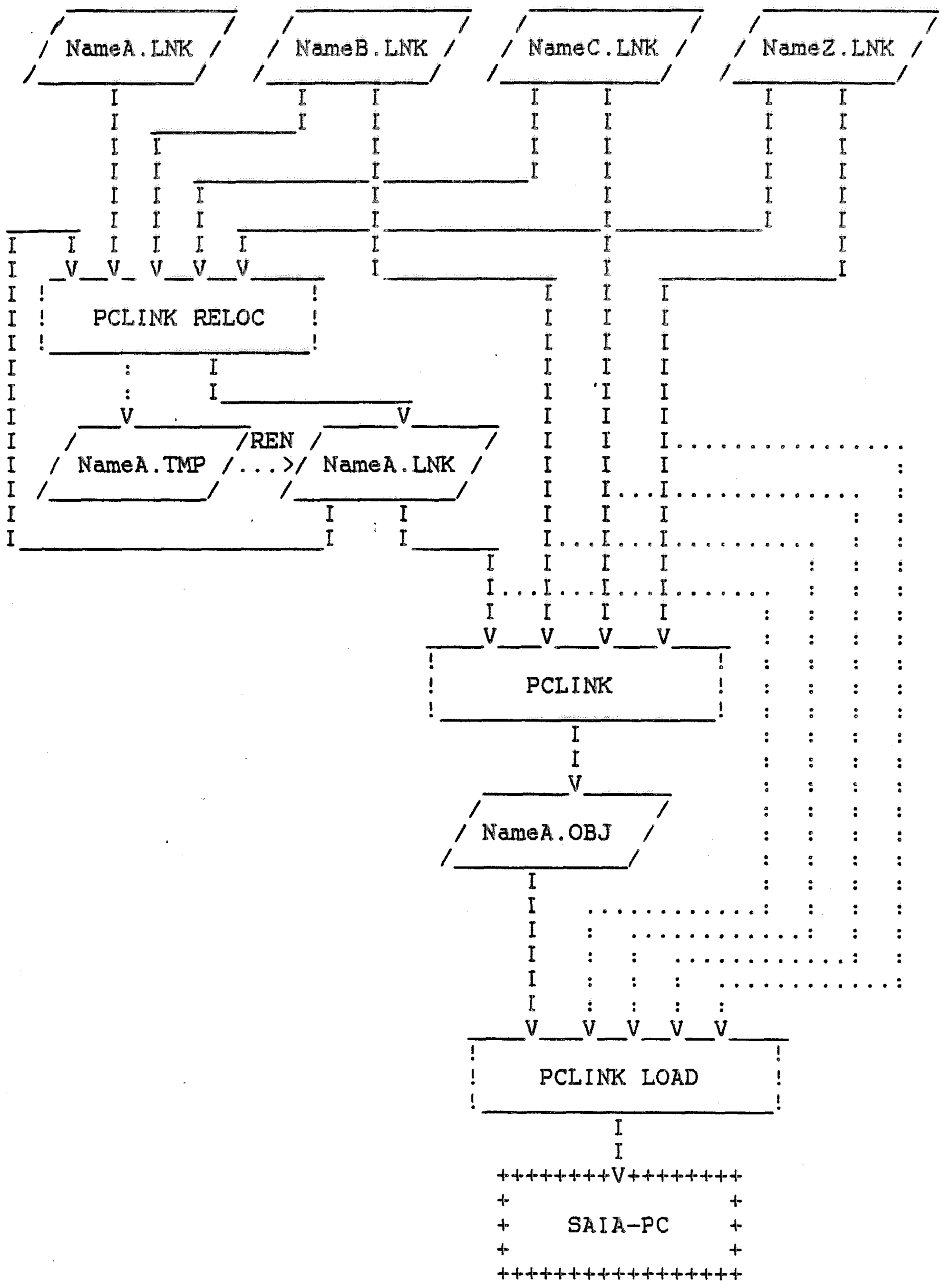
BESCHREIBUNG DER SOFTWARE

ABLAUF DES LINKER/LOADERS

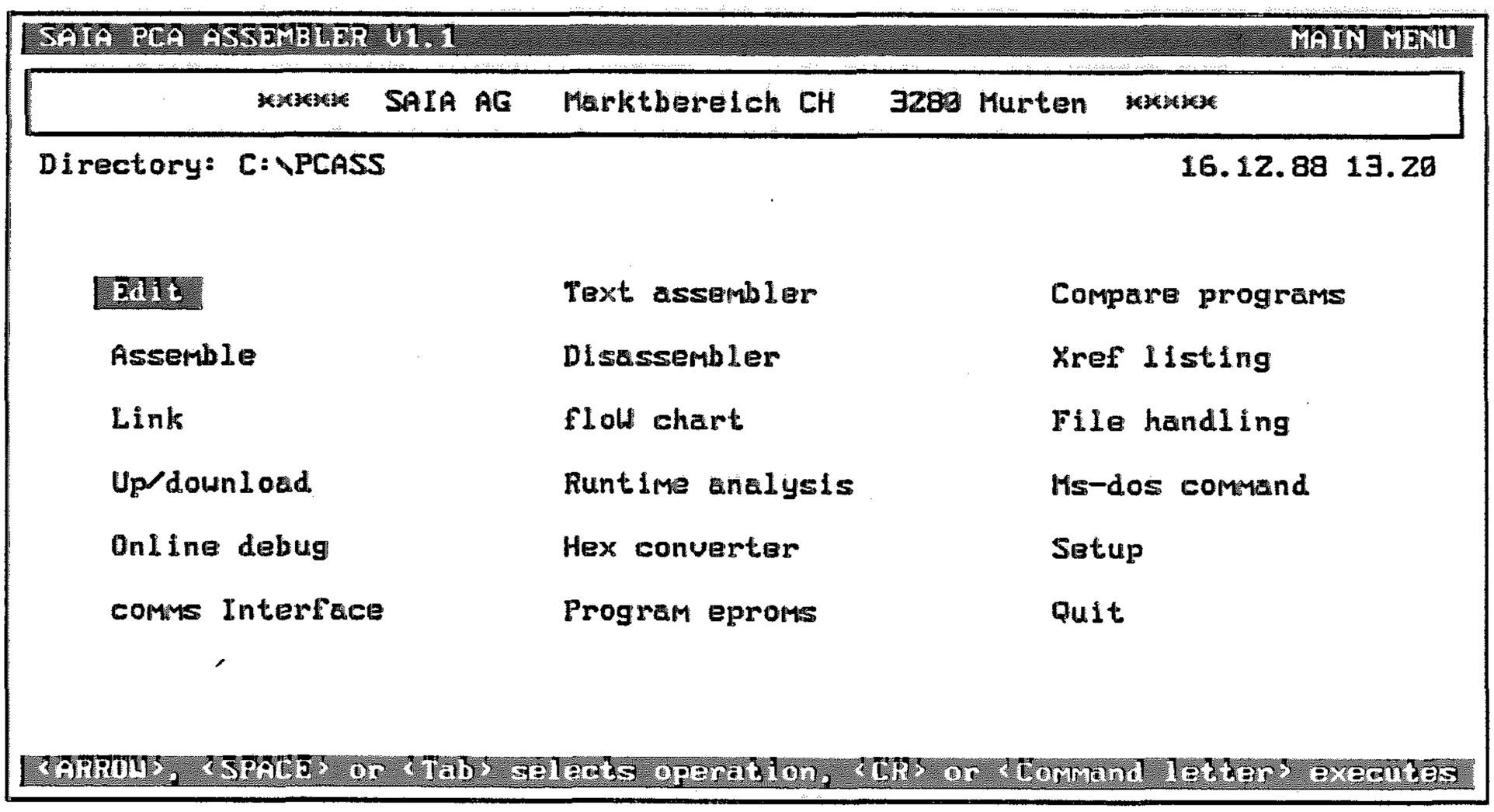
Der Linker fügt die assemblierten Teilprogramme zu einer ladbaren Datei zusammen. Es ist möglich, gelinkte Dateien mit weiteren linkbaren Dateien zu verbinden. Es können maximal 50 Dateien gelinkt werden. Der Linker kann etwa 1500 globale Symbole verarbeiten. Die ganze Linkprozedur gilt analog auch für die Programmdpkumentation, wobei der Ladevorgang entfällt.
Der Loader lädt die jetzt ladbare Datei über den PGU-Stecker in jede SAIA-PC. Hinweise zu eventuell notwendigen Modifikatitionen der PC finden sich im Anhang.
Die Abbildung auf Seite A-15 zeigt den Ablauf des Linkvorganges.

BESCHREIBUNG DER SOFTWARE

PHASEN des LINKERS



SAIA PCA ASSEMBLER - VERSION 1.1
Copyright (C) SAIA AG, 1984-1988
Press any key to continue


SAIA PCA ASSEMBLER V1.1
MAIN MENU
***** SAIA AG Marktbereich CH 3280 Murten *****
Directory: C:\PCASS
16.12.88 13.20
Edit
Text assembler
Compare programs
Assemble
Disassembler
Xref listing
Link
flow chart
File handling
Up/download
Runtime analysis
Ms-dos command
Online debug
Hex converter
Setup
comms Interface
Program eproms
Quit
<ARROW>, <SPACE> or <Tab> selects operation, <CR> or <Command letter> executes

EDITOR MENU

Name of file to edit (.SRC) _____

Enter file name, <CR> executes, <ESC> aborts.

SAIA PCA ASSEMBLER - VERSION 19-07-88 MAIN MENU

EDIT HILFSSEITE

Aufruf des Textverarbeitungs-Programmes. Dazu ist die Eingabe eines Dateinamens erforderlich. Die eingestellte Dateierweiterung (bzw. Dateityp) ist '.SRC'.
Sie wird automatisch zum eingegebenen Dateinamen hinzgefügt.
Wird eine andere Erweiterung gewünscht, so muss diese bezeichnet werden, z.B. 'FRITZ.TEX'.
Wird keine Erweiterung gewünscht, so ist der Dateiname mit '.' abzuschliessen, z.B. 'FRITZ.'

Es kann eine Laufwerkbezeichnung angegeben werden (A:, B: usw.). Beim Arbeiten mit einer Festplatte kann die Bezeichnung D: oder E: usw. verwendet werden, um den Bezug zu einem Verzeichnis herzustellen (siehe
DOS-Manual, Befehl SUBST).

```
Beispiele: NAME        -> Datei NAME.SRC im aktuellen Laufwerk editieren
           B:NAME      -> Datei NAME.SRC im Laufwerk B editieren
           NAME.TEX    -> Datei NAME.TEX im aktuellen Laufwerk editieren
           E:NAME.ABC  -> Datei NAME.ABC im "Verzeichnis E" editieren
```

Diese Dateibezeichnungen gelten für alle NAME-Eingabefelder der PCASS-Menüs.

Der Aufruf-Name des Texteditors wird im Untermenü 'Setup' eingegeben.
Press any key to return to the menu

PERSONAL EDITOR

KURZBESCHREIBUNG DES PERSONAL EDITOR (PE)

Aufruf: A>PE <CR>

Datei eröffnen : e lw:name.typ <CR>
Umschalten zwischen Textfeld und Kommandozeile: <ESC>
Hilfe : <F1>
Abspeichern ohne Löschen der Datei : <F2> <CR> (SAVE)
Abspeichern und Löschen der Datei : <F3> <CR> (FILE)
Verlassen der Datei ohne Abspeichern : <F4> <CR> (QUIT)
Verlassen des Editors : <F4> <CR> (QUIT)

Zeile entfernen : <CTRL><←> (Back Space)
Zeileninhalt löschen : <F5>
Bis Zeilenende löschen : <F6>
Datei ausdrucken : <F7> <CR>
Dateien umschalten : <F8>
Zeile einfügen : <F9>
Zeichen löschen : <DEL>
Einfügen/Ueberschreiben : <INS>

Cursorbewegungen

Zeilenanfang	: <HOME>	Zeilenende	: <END>
nach links	: <←>	nach rechts	: <→>
nach oben	: <↑>	nach unten	: <↓>
eine Seite vor	: PgDn	eine Seite zurück	: PgUp
Dateianfang	: <CTRL><HOME>	Dateiende	: <CTRL><END>
Bildschirmanfang	: <CTRL><PgUp>	Bildschirmende	: <CTRL><PgDn>

Blockoperationen

Blockmarkierung	: <ALT><B>	Zeilenmarkierung	: <ALT><L>
Zeichenmarkierung	: <ALT><C>	Block kopieren	: <ALT><Z>
Block verschieben	: <ALT><M>	Block löschen	: <ALT><D>
Markierung löschen	: <ALT><U>		

Alle folgenden Befehle sind auf der Kommandozeile einzugeben.
(Gross- oder Kleinschreibung)

C/alt/neu/ <CR>	: ersetzt "alt" mit "neu"
<SHIFT><F5>	: bestätigt Ersetzen
C/alt/neu/* <CR>	: ersetzt "alt" mit "neu" von Cursorposition bis Dateiende
DIR lw: <CR>	: Inhaltsverzeichnis der Diskette in lw:
DIR lw:name <CR>	: Inhaltsverzeichnis der Dateien name
ERASE lw:name <CR>	: löscht Datei name
L/text/ <CR>	: Sucht "text" vorwärts
L/text/- <CR>	: Sucht "text" rückwärts
? TABS <CR>	: zeigt die Tabulatoren
? MARGINS <CR>	: zeigt die Randeinstellung
? MEMORY <CR>	: zeigt den verbleibenden Speicherbereich
S MARGINS l r <CR>	: setzt Rand links (l) und rechts (r)
S TABS t1 t2...t20	: setzt Tabulatoren

ERSTELLEN DES QUELLPROGRAMMES

FORMAT DES QUELLPROGRAMMES

ZEILENEINTEILUNG

Beim Erstellen des Quellprogrammes müssen die folgenden Regeln beachtet werden:

- Eine Zeile des Quellprogrammes besteht aus max. 80 Zeichen.
- Jede Zeile ist mit <CR> abzuschliessen.
- Jede Zeile ist in 4 Felder eingeteilt.
- Die Länge eines Feldes ist beliebig, darf aber die maximale Anzahl Zeichen (80) nicht überschreiten. Benützen Sie den "Personal Editor", so setzen Sie die Zeilenlänge mit der Anweisung "SET MARGINS 1 80 <CR>" auf das richtige Format.
- Der Inhalt der Felder 1 bis 3 ist fest definiert.
- Feld 4 wird mit ";" eröffnet und steht für freien Kommentar zur Verfügung.
- Jedes Feld muss mit mindestens einem <BLANK> oder einem <TAB> abgeschlossen werden.
  Ausnahme: Das letzte Feld wird mit <CR> abgeschlossen (= Zeilenabschluss).

<CR> = Carriage Return + Line Feed
<BLANK> = Leerschlag
<TAB> = Tabulatorfunktion

Beispiele:

1)

START	STH	E1	; Abfrage Startknopf
FELD 1	FELD 2	FELD 3	FELD 4

2)

; Diese Zeile enthält nur Kommentar!

Feld 1 bis 3 haben die Länge 0, Feld 4 kann daher max. 80 Zeichen (inklusiv Semikolon) enthalten.

3)

	ENDM		; Ende Makro

Feld 2 enthält eine Assembler-Anweisung. Feld 1 und 3 sind leer.

DEFINITION DER FELDER

Feld 1: Das Feld 1 enthält Labels, Symbole, den Namen einer Macrodefinition oder das Steuerzeichen "$" als Kennzeichen einer Assemblersteueranweisung.

Feld 2: Das Feld 2 enthält den Operationscode (Mnemonic- oder Zahlencode der SAIA-PC), Zuordnungsfunktionen, und innerhalb des Quellprogrammes verwendbare Steueranweisungen.

ERSTELLEN DES QUELLPROGRAMMES

Feld 3: Das Feld 3 enthält die Operanden. Als Operand bezeichnet man Argumente von PC-Befehlen oder von Assemblersteueranweisungen, z.B. Eingänge, Timerwerte, Zieladressen von Sprungbefehlen (Labels) usw.
Die Operanden können absolut, symbolisch oder als arithmetischer Ausdruck angegeben werden.

Feld 4: Feld 4 enthält beliebigen Kommentar. Dieses Feld wird immer mit ";" eröffnet und mit <CR> abgeschlossen.

LABELS UND SYMBOLE

LABELS sind symbolische Adressen, die zur Bezeichnung von Startadressen von Unterprogrammen, Parallelprogrammen und Zieladressen von Sprüngen verwendet werden.
SYMBOLE sind Namen für Konstanten und Variablen.

Labels und Symbole dürfen maximal 7 alfanumerische Zeichen enthalten, wobei das erste Zeichen ein Buchstabe sein muss.

Beispiel:

```
START   EQU     1
VENT1   EQU     33
END2    EQU     12

BEGIN   STH     START       ; Startknopf ein
        ANH     END2        ; und Endschalter 2 ein
        OUT     VENT1       ; Ventil 1 ein
        ...     ...
        ...     ...
        JMP     BEGIN       ; Rücksprung an Programmanfang
```

Falsch sind folgende Beispiele:

```
ENDSCHAL2          ; mehr als 7 Zeichen
EIN/AUS            ; enthält unerlaubtes Zeichen
1A                 ; beginnt mit einer Zahl
   LABEL           ; wird wegen den Leerstellen am Anfang nicht
                   ; als Label interpretiert  --> Fehlermeldung
```

ERSTELLEN DES QUELLPROGRAMMES

ASSEMBLER-BEFEHLSSATZ

Der Assembler-Befehlssatz umfasst den Operationscode der SAIA-PC sowie die Anweisungen SET, EQU, PUBL, EXTN, ORG, ASEG und RSEG, deren Funktionen im folgenden beschrieben werden.
In die gleiche Kategorie gehören auch die Anweisungen MACRO, ENDM und LOCAL, die im Kapitel "MACROS" beschrieben werden.

Alle oben erwähnten Anweisungen stehen im Feld 2 und sind in Grossbuchstaben einzugeben.

OPERATIONSCODE

Als Operationscode ist der gesamte Befehlssatz der SAIA-PC (Bit- und Wortprozessor) zulässig.

BIT PROZESSOR:

NC	MNEMOCODE	
00	NOP	(WENN OP >< 1248)
00	SWP	(WENN OP = 1248)
01	STH	START HIGH
02	STL	START LOW
03	ANH	AND HIGH
04	ANL	AND LOW
05	ORH	OR HIGH
06	ORL	OR LOW
07	XOR	EXCLUSIVE OR
08	NEG	NEGATE ACCU
09	DYN	DYNAMIC CONTROL
10	OUT	SET OUTPUT W ACCU
11	SEO	SET OUTPUT
12	REO	RESET OUTPUT
13	COO	COMPLEMENT OUTPUT
14	STR	*SET TIMER
15	SCR	*SET COUNTER
16	SEI	SET INDEX
17	INC	INCREMENT COUNTER
18	DEC	DECREMENT COUNTER
19	SEA	SET ACCU
20	JMP	*UNCONDITIONAL JUMP
21	JIO	*JUMP IF ACCU = 1
22	JIZ	*JUMP IF ACCU = 0
23	JMS	*JUMP TO SUBROUTINE
24	RET	RETURN FROM SUBROUT
25	WIH	WAIT IF HIGH
26	WIL	WAIT IF LOW
27	INI	INCREMENT INDEX
28	DEI	DECREMENT INDEX
29	PAS	*PROGRAM ASSIGNEMENT
30	DOP	DISPLAY OPERAND
31	DTC	DISPLAY TIMER OR COUNTER

WORT PROZESSOR:

NC	MNEMOCODE	
00	NOP	(WENN OP >< 1248)
00	SWP	(WENN OP = 1248)
01	RRG	READ REGISTER
02	WRG	WRITE INTO REGISTER
03	RRE	READ REG AND WRITE IN ELEM
04	WRE	WRITE REG WITH ELEMENTS
05	LAR	LOAD A0 WITH REGISTERS
06	SAR	STORE A0 INTO REGISTERS
07	LAC	LOAD A0 WITH COUNTER
08	SAC	STORE A0 INTO COUNTER
09	WEL	WRITE ELEM WITH LOWER DIGIT
10	WEU	WRITE ELEM WITH UPPER DIGIT
11	INR	INCREMENT REGISTER
12	DER	DECREMENT REGISTER
13	SNC	SKIP IF NO CARRY OR IF 0
14	CLA	CLEAR A
15	LAI	LOAD A IMMEDIAT
16	DBN	DECIMAL TO BINARY
17	BND	BINARY TO DECIMAL
18	ROR	ROTATE REGISTERS
19	ROA	ROTATE A
20	EXG	EXCHANGE A0 WITH A1
21	CLK	CLOCK SOURCE
22	SHI	SHIFT REGISTERS
23	TXT	TEXT
24	SEW	SKIP TO EWP OR SWP
25	CMP	COMPARE A0 WITH A1
26	SQR	SQUARE ROOT FROM A1
27	ADD	ADD A0 TO A1
28	SUB	SUBTRACT A0 FROM A1
29	MUL	MULTIPLY A1 BY A0
30	DIV	DIVIDE A1 BY A0
31	EWP	END WORD PROCESSOR

ERSTELLEN DES QUELLPROGRAMMES

Die mit * versehenen Befehle sind zwei- oder mehrzeilig. Hier sind je nach Befehl verschiedene Besonderheiten zu beachten. Weiteres dazu in den folgenden Abschnitten.

Timer- und Zählerbefehle:

1. ZEILE :

NC	MNC	OPRD
14	STR	256 - 287
15	SCR	256 - 287
		(256 - 319 für PCA23 und PCA14)

2.ZEILE :

NC	MNC		BEDEUTUNG:
00 - 15	TCV	*	TIMER-/COUNTER-VALUE
16	IUM		IN : 1 * 8 BIT BCD
17	ITM		IN : 10 * 8 BIT BCD
18	IHM		IN :100 * 8 BIT BCD
19	IFD		IN : 20 BIT BCD
20	OFD		OUT : 20 BIT BCD
21	OEB		OUT : 8 BIT BINAER
22	OTB		OUT : 12 BIT BINAER
23	OSB		OUT : 16 BIT BINAER
24	IEB		IN : 8 BIT BINAER
25	ITB		IN : 12 BIT BINAER
26	ISB		IN : 16 BIT BINAER
27	ADD		Z = Z + X
28	SUB		Z = Z - X
29	MUL		Z = Z * X
30	DIV		Z = Z / X
31	ICV		Z = INDEX/COUNTER-VALUE

*: TCV-Werte können im Bereich 0...32767 definiert werden. Die Umcodierung erfolgt automatisch.

Für PCA 14/23 gilt:

Die Zeit- bzw. Zählregister wurden von 15 auf 16 Bit erweitert, was einer Erweiterung der Dezimalzahlen von 32767 auf 65535 entspricht. Dieser erweiterte Zeit- bzw. Zählbereich ist nur als Resultat einer arithmetischen PC-Operation oder über die Einlesebefehle IFD (19) oder ISB (26) in der zweiten Zeile zugänglich. Im Assemblercode dürfen also keine Werte grösser als 32767 auftreten. Diese Einschränkung ist systembedingt.

Der in der zweiten Zeile von STR und SCR verwendete Mnemocode ist <u>ausschliesslich</u> auf die Anwendung im SAIA-PC-Assembler beschränkt. Insbesondere ist es nicht möglich, diesen Mnemocode auch im Communication-Interface PCA2.T13 zu verwenden. Hingegen ist es natürlich möglich, im PC-Assembler in der zweiten Zeile den numerischen Code zu gebrauchen.

Beispiel:	STR	256			STR	256
	TCV	31562		entspricht	15	842

ERSTELLEN DES QUELLPROGRAMMES

Bedeutung des Mnemocodes in der 2.Zeile von STR/SCR Befehlen

NC	MNC	Bedeutung
00 .. 15	TCV	Timer Counter Value
16	IUM	In Units Multiply
17	ITM	In Tenth Multiply
18	IHM	In Hundreds Multiply
19	IFD	In Five Digits
20	OFD	Out Five Digits
21	OEB	Out Eight bits Binary
22	OTB	Out Twelve bits Binary
23	OSB	Out Sixteen bits Binary
24	IEB	In Eight bits Binary
25	ITB	In Twelve bits Binary
26	ISB	In Sixteen bits Binary
27	ADD	Add
28	SUB	Subtract
29	MUL	Multiply
30	DIV	Divide
31	ICV	Index/Counter-Value

Sprungbefehle:

Sprungbefehle müssen im Quellprogramm einzeilig eingegeben werden. Für Sprungadressen > 2047 und für die Adresse 0 müssen die Sprungbefehle mit einem "L" (Long) versehen werden. Die Sprungbefehle lauten dann JMPL, JIOL, JIZL und JMSL. Ab Assemblerversion C8C1/1.1.05 wird auch der erweiterte Adressbereich der PCA2.M32 unterstützt. Die höchste zulässige Adresse beträgt somit 15999.

Beispiele:	JMPL	4320	entspricht	JMP	0
				2	224
	JIOL	0	entspricht	JIO	0
				0	0

PAS-Befehle:

1. ZEILE			2. ZEILE :		
NC	MNC	OPRD	NC	MNC	
29	PAS	0 - 15	00 - 31	00 - 31	
29	PAS	16 - 99	00 - 31	00 - 31	
29	PAS	100 - 2047	00 - 31	00 - 31	(Nächste 9 Zeilen)

Bei den PAS-Befehlen sind ab der 2. Zeile für den Operationscode (Feld 2) keine Symbole zulässig. Es darf hier also nur numerischer Code verwendet werden.
Bei Verwendung von PAS-Befehlen in Macros (siehe Kapitel 2.6) ist es zulässig, für den Operationscode (!) Symbole zu verwenden, die in der Macrodefinition als Parameter definiert sind.
Beim Aufruf des Macros im Quellprogramm müssen diese Parameter jedoch als Absolutwerte (0...31) angegeben werden.

ERSTELLEN DES QUELLPROGRAMMES

DEFINITION EINER KONSTANTEN EQU

Mit der Definition EQU wird einem Symbol ein konstanter Wert zugeordnet. Dieser Wert kann nicht mehr verändert werden. Insbesondere ist es nicht mehr möglich, diesem Symbol im weiteren Quellprogramm mit der Anweisung SET einen andern Wert zuzuweisen.
Als Wertzuweisungen sind numerische Werte, Symbole und Konstanten, die bereits definiert sind, und arithmetische Ausdrücke erlaubt.
Extern definierte Werte (siehe EXTN) können nicht zugewiesen werden.
Diese Wertzuweisungen können auch nach der Verwendung des entsprechenden Symbols vorgenommen werden, d.h. sie dürfen an beliebiger Stelle des Programmes auftreten.

Beispiele:
```
NOTSTOP  EQU   8
LAMPE1   EQU   32
LAMPE2   EQU   LAMPE1 + 1
```

Folgende Wertzuweisung ist <u>nicht zulässig</u> (Vorwärtsreferenz):

```
VENTIL1  EQU   VENTIL2 + 1
VENTIL2  EQU   38
```

DEFINITION EINER VARIABLEN SET

Mit der Definition SET wird einem Symbol ein variabler Wert zugeordnet, der mit weiteren SET-Definitionen im Quellprogramm beliebig verändert werden kann.
Als Wertzuweisungen sind numerische Werte, Symbole und Konstanten, die bereits definiert sind, und arithmetische Ausdrücke erlaubt.
Extern definierte Werte (siehe EXTN) können nicht zugewiesen werden.
Diese Wertzuweisungen dürfen an beliebiger Stelle im Programm stehen, müssen jedoch vor dem entsprechenden Symbol verwendet werden.

Beispiele:
```
VERZOEG SET   25

        STR     256
        TCV     VERZOEG

VERZOEG SET   25 + 10

        STR     256
        TCV     VERZOEG
        usw.
```

Folgende Wertzuweisung ist <u>nicht zulässig</u>:

```
DELAY  SET     DELAY + 10
```

ERSTELLEN DES QUELLPROGRAMMES

EXTERN BENUTZBARE LABELS UND SYMBOLE PUBL

Der Assembler erlaubt, verschiedene Programmteile, die einzeln assembliert wurden, zu einem einzigen Programm zu verbinden (linken).
In einem Programmteil können deshalb Labels und Symbole definiert werden, die auch von einem andern, getrennt assemblierten Programmteil aus angesprungen bzw. benützt werden können.
Diese Labels und Symbole müssen in demjenigen Programmteil, in dem sie definiert werden, als PUBL (public) bezeichnet werden.
Ein Label muss dort als definiert betrachtet werden, wo es im ersten Feld steht. Ein Beispiel finden Sie im folgenden Abschnitt.
Im Assemblerlisting erscheint bei den als public definierten Symbolen und Labels in der Spalte UI jeweils ein "P". Siehe Programmbeispiel Kap. 6.

! Die Anweisung PUBL muss am Schluss des zugehörigen Programmteiles stehen.

LABELS UND SYMBOLE AUSSERHALB DES MODULS EXTN

Labels und Symbole, die in einem andern Programmteil (Modul) als PUBL definiert werden, müssen in allen übrigen Modulen, in denen sie gebraucht werden, als EXTN (extern) bezeichnet werden. Ein als extern deklariertes Symbol darf bei den arithmetischen Operationen nur in einer Addition als Operand auftreten. Subtraktion, Multiplikation und Division sind nicht zugelassen.
Im Assemblerlisting erscheint bei den als extern definierten Symbolen und Labels in der Spalte UI jeweils ein "E". Sie- Programmbeispiel Kap. 6.

! Die Anweisung EXTN muss am Schluss des zugehörigen Programmteiles stehen.

Beispiel:

PROGRAMM A

```
! Feld 1  ! Feld 2  ! Feld 3        ! Feld 4
SYM1     EQU      12
SYM2     EQU      36

         STH      SYM1
         ANH      SYM2
         usw.

         PUBL     SYM1,SYM2         ; Diese Symbole werden auch
                                    ; im Programm B verwendet
```

ERSTELLEN DES QUELLPROGRAMMES

PROGRAMM B

```
STH     SYM2
XOR     SYM1
usw.

EXTN    SYM1,SYM2       ; Diese Symbole sind im
                        ; Programm A definiert
```

DEFINITION EINER STARTADRESSE ORG

Die Startadresse eines Programmes kann mit ORG (origin) definiert werden. Eine Startadresse darf nur in einem Absolutsegment (Siehe Kap. 2.4.7) vorgegeben werden.
ORG steht im Feld 2, die Startadresse im Feld 3.

Beispiele:
```
ORG     100
ORG     PSTART
ORG     PSTART + 10
```

In ORG-Anweisungen dürfen keine extern definierten Symbole oder Labels verwendet werden.

ABSOLUTES SEGMENT ASEG

Ein Programmteil, der die nach dem Assemblieren bestimmten Adressen auch nach dem Linken beibehalten soll, wird mit ASEG (absolutes Segment) bezeichnet.
ASEG bezeichnet den Beginn eines absoluten Segmentes. Die Anweisung ORG legt die Startadresse des absoluten Segmentes fest.

Beispiel:
```
ASEG            ; Dieses Programm beginnt
ORG     100     ; bei Adresse 100
```

RELATIVES SEGMENT RSEG

Mit RSEG wird der Beginn eines Programmteils bezeichnet, dessen Position nicht von vornherein festgelegt ist. Die absoluten Adressen werden erst vom Linker zugeordnet.
ASEG bezeichnet das Ende eines relativen Segmentes.
Die Anweisung ORG ist in einem relativen Segment nicht zulässig.
Wird ein Programmteil weder mit ASEG noch mit RSEG bezeichnet, so ist er automatisch als relatives Segment definiert.

ERSTELLEN DES QUELLPROGRAMMES

DARSTELLUNG VON OPERANDEN

Bei Symbolen von Operanden werden maximal 7 alphanumerische Zeichen erkannt. Das erste Zeichen muss ein Buchstabe sein.
Der Assembler akzeptiert Operanden als absolute, symbolische oder arithmetische Ausdrücke.
Es sind absolute Werte von 0...2047 für alle Befehle zulässig.

Ausnahmen:

- Sprungbefehle und 2.Zeile von PAS 0...15: 0...15999
- TCV (Timer/Counter Value) : 0...32767

Als Operanden dürfen auch arithmetische Ausdrücke mit Symbolen verwendet werden. Folgende Operationen sind möglich:

- Addition +
- Subtraktion -
- Multiplikation *
- Division / (Resultat ganzzahlig)
- Modulo \
- Klammerausdruck ()
- Spezialzeichen $
  Für das Zeichen $ im Operand wird die Adresse eingesetzt, die gerade für diese Programmzeile gilt.

  Bsp: JIO $ - 4
  d.h Rücksprung um 4 Programmzeilen.

  Bei den Sprunganweisungen JMPL, JIOZL, JIOL, JMSL (erzeugen zweizeilige Befehle) muss die zweite Zeile mitberücksichtigt werden.

```
Bsp: JIOL   END  <--:
     STH    1       :
     ANL    12      :
     JIZL   $-5  ---:
```

Ausnahme bei den arithmetischen Operationen:

Symbole, die extern definiert werden, können <u>nur mit der Addition</u> verwendet werden.

```
Beispiele:   STL   FLAG1 + 1000    ; Indexierung

             ORG   ($ +9)/10 * 10  ; nächster Programmschritt
                                   ; kommt auf eine "Zehner"-
                                   ; Adresse

             SYMBOL1 EQU   SYMBOL3 - SYMBOL2
```

ERSTELLEN DES QUELLPROGRAMMES

ASSEMBLERSTEUERANWEISUNGEN IM QUELLPROGRAMM

Steueranweisungen im Quellprogramm:
Diese Steueranweisungen unterstützen die Gestaltung der Listings, ermöglichen das Einbinden externer Dateien u.a.

Die Steueranweisungen im Quellprogramm sind durch das Steuerzeichen "$", das im Feld 1 steht, gekennzeichnet. Diese Steueranweisungen können mehrmals im Quellprogramm gebraucht werden. Sie werden auch im Assemblerlisting wieder aufgeführt. Alle diese Steueranweisungen stehen im Feld 2.

STEUERUNG DES ASSEMBLERLISTINGS UND DER PROGRAMMDOKUMENTATION — NOLIST / LIST

Die Anweisung NOLIST im Quellprogramm bewirkt, dass im Assemblerlisting und in der Programmdokumentation die auf diese Anweisung folgenden Zeilen unterdrückt werden.

Die Anweisung LIST hebt NOLIST wieder auf.

Die Möglichkeit, die Abspeicherung von Zeilen im Assemblerlisting zu steuern, dient vor allem dazu, Macrodefinitionen, die im Quellprogramm aufgerufen werden, nicht jedesmal wieder vollständig aufzulisten.
Die Grundeinstellung ist LIST. Es ist daher nicht notwendig, am Anfang des Quellprogrammes LIST zu programmieren.

TITEL — TITLE

Mit der Anweisung TITLE kann ein Seitenüberschriftstitel, der im Kopf jeder Seite wieder erscheint, bestimmt werden.
Dieser Titel sollte immer eine Versionsnummer des Anwenderprogrammes enthalten (zur Identifikation der eigenen Programme).
In der darauf folgenden Zeile erscheint jeweils automatisch die Firmenanschrift. Siehe das Beispiel unter 2.5.3.

UNTERTITEL — STITLE

Mit der Anweisung STITLE ist es möglich, im Kopf der Seite auch einen Untertitel zu definieren.

ERSTELLEN DES QUELLPROGRAMMES

Beispiel:

```
$          TITLE    DEMOPROGRAMM  V11.4
$          STITLE   VERZOEGERUNG
```

ergibt im Assemblerlisting und in der Programmdokumentation folgenden Seitenkopf:

```
*** SAIA - PC ***     DEMOPROGRAMM V11.4                    PAGE 1
MACRO - ASSEMBLER     VERZOEGERUNG

Firma PROCONTROL  3000 Bern Schweiz
```

Die Versionsnummer des Anwenderprogrammes (V11.4) muss im Titel eingegeben werden.
Die Firmenanschrift wird gemäss Kundenangaben vor der Auslieferung der Software erstellt. Diese erscheint jeweils automatisch und kann nicht verändert werden.

VORSCHUB AUF SEITENANFANG — EJECT / EJECTL

Die Anweisung EJECT bewirkt, dass im Assemblerlisting und in der Programmdokumentation eine neue Seite begonnen wird.
Die Anweisung EJECTL bewirkt, dass nur im Assemblerlisting eine neue Seite begonnen wird.

EINFÜGEN VON n LEERZEILEN — SPACE(n)

Mit der Anweisung SPACE(n) können n Leerzeilen ins Assemblerlisting und in die Programmdokumentation eingefügt werden.

Beispiel:

```
$          SPACE(12)    ; erzeugt 12 Leerzeilen im Assemblerlisting
```

EINFÜGEN EINER QUELLPROGRAMMDATEI — INCLUDE

Mit der Anweisung INCLUDE können andere Quellprogramme oder Macros in das Assemblerlisting eingefügt werden. Die Programme müssen im Quellprogramm an der betreffenden Stelle mit

```
$          INCLUDE  lw:name.typ
```

aufgerufen werden. Die einzufügenden Macros oder Quellproprogramme müssen im Feld 3 vollständig mit Laufwerkbezeichnung, Name und Datentyp angegeben werden. Sie müssen Quellformat haben. Der Datentyp kann beliebig sein.

Beispiele:

```
$        INCLUDE  A:text.txt      ; Einbinden einer Textdatei

$        NOLIST                   ; Einbinden eines Quellpro-
$        INCLUDE  B:delay.src     ; grammes, wobei dessen In-
$        LIST                     ; halt nicht im Assembler-
                                  ; listing erscheint.
```

Die Verwendung der Anweisung INCLUDE im Zusammenhang mit Macros ist im Kapitel 2.6 beschrieben.

UNTERTITEL IN DER PROGRAMMDOKUMENTATION

MARK
MARKL

Mit der Anweisung MARK können im Quellprogramm Untertitel definiert werden, die an der entsprechenden Stelle nur in der Programmdokumentation PGDOK eingesetzt werden. Das erste Zeichen im Feld 3 wird in der Programmdokumentation vor dem eigentlichen Untertitel automatisch 19 mal wiederholt.

Beispiel:

```
         ASEG
         ORG    : 1150
$        MARK     - EINSCHALTVERZOEGERUNG; Untertitel
         STH      E1                     ; Eingang
         DYN      PULS                   ; 1 Puls erzeugen
         STR      TR1                    ; Timer
         VAL      T1                     ; Verzoegerungszeit
         ANH      PULS
         OUT      A1                     ; verz. Ausgang
```

ergibt in der Programmdokumentation:

```
--------------------- EINSCHALTVERZOEGERUNG
1150 01  STH      0  Eingang
1151 09  DYN    500  1 Puls erzeugen
1152 14  STR    256  Timer
1153 16   16     10  Verzoegerungszeit
1154 03  ANH    500
1155 10  OUT     32  verz. Ausgang
```

Mit MARKL wird zusätzlich vor dem Untertitel eine Leerzeile eingeschoben.

ERSTELLEN DES QUELLPROGRAMMES

DATUM und ZEIT

DATE
TIME

Die Anweisungen DATE und TIME lesen das Systemdatum und die Systemzeit beim Assemblieren ein und fügen sie ins Assemblerlisting und in die Programmdokumentation ein.

Aufruf im Quellprogramm:

```
$       DATE
$       TIME
```

Assemblerlisting:

```
$       DATE    tt-MM-yyyy
$       TIME    hh:mm
```

Programmdokumentation:

```
DATE    tt-MM-yyyy
TIME    hh:mm
```

(tt = Tag, MM = Monat, yyyy = Jahr, hh = Stunden, mm = Minuten)

ERSTELLEN DES QUELLPROGRAMMES

MACROS

Die Leistungsfähigkeit des SAIA-MACRO-ASSEMBLER wird durch die Verwendung von Macros ganz wesentlich gesteigert. Damit ist es möglich, eine umfangreiche SW-Bibliothek mit Programmbausteinen aufzubauen. Sogar eine eigene, amwendungsbezogene Programmiersprache kann auf diese Weise definiert werden.
Immer wiederkehrende Programmsequenzen werden mit einer einzigen Zeile programmiert. Neben einer sehr schnellen Programmierung garantiert die Verwendung von Macros auch eine Vereinheitlichung innerhalb der Programmbausteine. Ein bestehender Programmbaustein kann jedoch jederzeit individuell dem Problem angepasst werden.

MACRODEFINITION MACRO — ENDM

Als Macro bezeichnet man einen vom Anwender geschriebenen Programmteil, der im Quellprogramm an beliebigen Stellen beliebig oft aufgerufen werden kann. Ein Macroprogramm kann bis zu 10 Parameter enthalten, d.h. vom Quellprogramm aus können zur Assemblierzeit bis zu 10 Namen von Konstanten und Variablen (Symbole) übergeben werden.
Ein Macroprogramm muss im aufrufenden Programm definiert sein, d.h. es muss beim Assemblieren als Text vollständig vorhanden sein. Es bestehen nun zwei Möglichkeiten, ein Macroprogramm zu definieren:

1) Das Macroprogramm wird irgendwo im Quellprogramm gemäss den untenstehenden Regeln definiert.
2) Das Macroprogramm wird ausserhalb des aufrufenden Quellprogrammes als eigene Datei definiert und auf Diskette abgelegt (z.B. als Bestandteil einer Macrobibliothek). In diesem Fall muss diese Datei durch die Anweisung INCLUDE zur Assemblierzeit eingebunden werden.

Eine Macrodefinition wird mit der Anweisung MACRO eröffnet und mit der Anweisung ENDM abgeschlossen.
Die Anweisungen MACRO und ENDM stehen im 2. Feld.

Aufbau eines Macros:

```
mname    MACRO  p1,p2,...p10
         ...
         ...
         code
         ...
         ...
         ENDM
```

mname: frei definierbarer Macroname mit bis zu 7 alfanumerischen Zeichen

ERSTELLEN DES QUELLPROGRAMMES

MACRO: reserviertes Wort, bezeichnet den Beginn eines Macros
ENDM : reserviertes Wort, bezeichnet das Ende eines Macros
p1,p2,...p10: Parameterliste (oft Formalparameter genannt), sind Platzhalter für Werte, die im Hauptprogramm zur Assemblierzeit eingesetzt werden (sogenannte Aktualparameter). Die Syntax der Parameter unterliegt den gleichen Regeln wie diejenige der Symbole und Labels (siehe dazu Kap. 2.3.3).

PARAMETER IN EINEM MACRO %

Alle als Parameter verwendeten Symbole müssen im Macroquellcode mit dem Steuerzeichen "%" versehen werden. In der Parameterliste werden sie ohne dieses Steuerzeichen aufgeführt.

LOKALE LABELS UND SYMBOLE IN EINEM MACRO LOCAL

Werden Labels und Symbole nur innerhalb eines Macros gebraucht, müssen sie am Anfang des Macros mit der Anweisung LOCAL (Feld 2) als solche definiert werden. Die zugehörigen Labels und Symbole werden im Feld 3 aufgeführt. Die Anweisung LOCAL bewirkt also, dass sich der Geltungsbereich eines Symbols oder eines Labels auf den Macro beschränkt. Ein Konflikt mit gleichen Symbolen und Labels ausserhalb des Macros wird so vermieden. Im Macroprogramm selbst sind die lokalen Labels und Symbole mit dem Steuerzeichen "?" zu bezeichnen. Im Assemblerlisting werden sie durch die Ersatzsymbole(-labels) "??0", "??1" usw. ersetzt (siehe Beispielprogramm Kap. 6).
Es können maximal 10 lokale Symbole oder Labels definiert werden. Arithmetische Operationen sind zusammen mit lokalen Symbolen und Labels nicht zulässig.

Beispiel: LOCAL SYM1,LAB1,LAB2

EINBINDEN EINES MACROS INCLUDE

Mit der Anweisung INCLUDE werden im Quellprogramm alle benötigten Macros als Text eingebunden, sofern sie nicht im Quellprogramm selbst definiert werden.
INCLUDE ist eine Assembler-Steueranweisung und muss daher im 1. Feld mit "$" (Siehe Kap. 2.3.5) versehen werden.
Die INCLUDE-Anweisung bewirkt, dass im PASS 1 die Macros in das Quellprogramm eingebunden werden, wobei noch keine Parameter übergeben werden. PASS 1 erzeugt eine Temporärdatei vom Typ EXP, die am Schluss des Assemblierens automatisch wieder gelöscht wird.

ERSTELLEN DES QUELLPROGRAMMES

Beispiel:

```
  Feld 1    Feld 2    Feld 3            Feld 4
$          INCLUDE   lw:MTR1A1.SRC ; fügt die Macrodatei in das
                                     ; Quellprogramm ein
```

Man beachte, dass diese Anweisung <u>nicht</u> den eigentlichen Aufruf des Macros im Programmcode darstellt.

AUFRUF EINES MACROS UND PARAMETERUEBERGABE

An der Stelle im Programmcode, wo das Macroprogramm aktiv werden soll, wird dieses unter dem Namen (ohne Angabe des Datentypes) im Feld 2 aufgerufen.
Im Feld 3 werden die aktuellen Parameter vollständig und in richtiger Reihenfolge gemäss Definition aufgeführt.

Werden PAS-Befehle in Macros verwendet, ist eine Besonderheit zu beachten: Es ist zulässig, für den Operationscode (Code im 2. Feld) Symbole zu verwenden, die in der Macrodefinition als Parameter definiert sind.
Beim Aufruf des Macros im Quellprogramm müssen diese Parameter jedoch als Absolutwerte (0...31) angegeben werden.

Beispiel: Macro zum Setzen oder Lesen der Zeit

```
Macrodefinition: SLZEIT  MACRO   STD,MIN,SEC
                         PAS     50
                         %STD    256
                         PAS     50
                         %MIN    257
                         PAS     50
                         %SEC    258
                         ENDM
```

Aufruf im Quellprogramm: (Lesen der Zeit)

```
                         SLZEIT  15,16,17
```

Macroaufrufe können bis zu einer Tiefe von 10 verschachtelt sein und können auch wieder Macrodefinitionen enthalten.
Parameter können aber nicht durch mehrere Macroebenen hindurch ersetzt werden. Ein verschachtelt aufgerufener Macro kann also in seinen Parametern keinen direkten Bezug auf die Parameter des übergeordneten Macros nehmen, sondern nur über seine Parameterliste. Das Steuerzeichen "%" darf also auch in einer Parameterliste eines Macros stehen.

ERSTELLEN DES QUELLPROGRAMMES

BEISPIEL FÜR MACRODEFINITION UND MACROAUFRUF

Macrodefinition

Mit dem Editor wird eine Macrodefinition geschrieben, die eine Einschaltverzögerung beinhaltet. Der Dateiname sei "VERZ.SRC"und befinde sich auf der Diskette im Laufwerk B. Die INCLUDE-Anweisung bezieht sich auf den Dateinamen, der Aufruf auf den Macronamen. Diese beiden Namen müssen entgegen dem Beispiel nicht notwendigerweise übereinstimmen. Eine Macrodatei kann mehrere Macros mit verschiedenen Macronamen enthalten.

```
FELD 1    FELD 2    FELD 3                    FELD 4

VERZ      MACRO     E1,M1,TIMER,TWERT,A1

          STH       %E1                       ; Flanken-
          DYN       %M1                       ; triggerung
          STR       %TIMER                    ; Timer Start
          TCV       %TWERT                    ; Verzoegerungszeit
          STL       %TIMER                    ;
          ANH       %M1
          OUT       %A1                       ; Einschaltverz. Ausg.
          ENDM
```

Macroaufruf im Quellprogramm

```
FELD 1    FELD 2    FELD 3

; Equates:

HAUPT     EQU       1
TASTE1    EQU       5
MERKER1   EQU       600
MOTOR     EQU       34
TMR       EQU       260
VERZEIT   SET       15

$         INCLUDE   B:VERZEIN.SRC

BEGIN     STH       HAUPT
          ANL       MOTOR
          JIZ       BEGIN
          VERZ      TASTE1,MERKER1,TMR,VERZEIT,MOTOR
          JMP       BEGIN

; Ende des Programmes
```

Die Uebergabe der aktuellen Werte ("Equates") erfolgt über die Symbole der Parameterliste im Aufruf. Die Zuordnung der Werte zu den entsprechenden Parameter in der Macrodefinition ist abhängig von der Position und nicht von der Namengebung.

```
SAIA PCA ASSEMBLER V1.1                                          MAIN MENU
          ***** SAIA AG   Marktbereich CH   3280 Murten  *****
Directory: C:\PCASS                                    16.12.88 13.22

   Edit                    Text assembler          Compare programs

   Assemble                Disassembler            Xref listing

   Link                    flow chart              File handling

   Up/download             Runtime analysis        Ms-dos command

   Online debug            Hex converter           Setup

   comms Interface         Program eproms          Quit

<ARROW>, <SPACE> or <Tab> selects operation, <CR> or <Command letter> executes
```

```
SAIA PCA ASSEMBLER MENU
File(s) to assemble . . . . . . (.SRC)   HANDLING__   ________   ________

                                         ________   ________   ________

                                         ________   ________   ________

Create link file . . . . . . . (.LNK)    Yes
Create listing . . . . . . . . (.LST)    No
      Listing with:     Symbol table     No
                  Cross reference list   No
Create documentation . . . . . (.PGD)    Yes
Print documentation OR listing . . .     No

Enter file name, <ARROW> or <Tab> moves cursor, <CR> executes, <ESC> aborts,
```

ASSEMBLER HILFSSEITE

Uebersetzt Dateien vom Typ SRC in Dateien vom Typ LNK und bereitet sie für die Weiterverarbeitung mit dem Linker vor. Es können bis zu 9 SRC-Dateien gleichzeitig für das Assemblieren eingegeben werden. Als Option kann zusatzlich eine LST- oder PGD-Datei erzeugt werden.

Die Menüführung erlaubt die Wahl von LNK, LST oder PGD Dateien und die Angabe des Zielortes (Datei oder Drucker). Das Seitenformat für die LST- und PGD-Datei oder für den Ausdruck, wird im Untermenü "Setup" definiert.

Die Angabe "Symbol table" und/oder "Cross reference" wird ignoriert, wenn bei "Create listing" "NO" definiert wird.

Um eine LST- oder PGD-Datei (sofern erzeugt) zum Drucker zu senden, muss "Print documentation OR listing" mit "Yes" beantwortet werden. Wenn die LST- und PGD-Dateien ausgedruckt werden, bleiben die eventuell vorhandenen vorherigen Dateien auf der Disk unverändert; sind also nicht mehr auf dem neusten Stand. Es ist nicht möglich beide Dateien gleichzeitig auszudrucken.

Hinweis: Für DOS-Versionen ohne Befehl "APPEND", müssen sich die Ueberlagerungsdateien PCAS.001 bis PCAS.004 und PCKEY.PWD im aktuellen Verzeichnis befinden.

ASSEMBLIEREN

SYNTAXKONTROLLEN DES ASSEMBLERS

Der Assembler prüft im Pass 1 und Pass 2 die Operanden und den Operationscode auf Zulässigkeit und gibt bei unzulässigem Operationscode und/oder bei unzulässigen Operanden eine Fehlermeldung aus.

SYNTAXKONTROLLE DER OPERANDEN

Einzeilige Befehle: Bereich 0...2047

Mehrzeilige Befehle:

Sprungbefehle JMP, JMS, JIO, JIZ:
Der Operand liegt im Bereich 1...2047

Sprungbefehle JMPL, JMSL, JIOL, JIZL:
Der Operand liegt im Bereich 0...15999

PAS 00...15:
Operand in der 2. Zeile: 0...15999

PAS 15...99:
Operand in der 2. Zeile: 0...2047

PAS >99:
Operand in der 2. bis 10. Zeile: 0...2047

STR/SCR-Befehle:
Operand in der 2. Zeile: 0...32767

SYNTAXKONTROLLE DES OPERATIONSCODES

Einzeilige Befehle:
Numerischer Operationscode: 00...31
Gesamter Mnemocode des Bit- und Wortprozessors

PAS-Befehle:
2. bis 10. Zeile: nur numerischer Code 00...31.
Werden PAS-Befehle in Macros mit Parametern aufgerufen, die sich auf Operationscode in der zweiten bis zehnten Zeile beziehen, so sind diese Parameter im Aufruf des Macros mit numerischem Code im Bereich 0...31 anzugeben. Sie dürfen kein Symbol sein, dem vorher mit einer SET- oder EQU-Anweisung ein Wert zugewiesen wurden.
Die Anzahl der Zeilen, die einem PAS-Befehl folgen müssen, wird geprüft.

ASSEMBLIEREN

FEHLERMELDUNGEN

Bei Fehlermeldungen ist zwischen Fehlermeldungen des MS-DOS Betriebssystems und Fehlermeldungen des Assemblers zu unterscheiden.
Fehlermeldungen des DOS-Betriebssystems sind im DOS-Handbuch unter dem Kapitel "Nachrichten" nachzulesen.
Der Assembler gibt folgende Arten von Fehlermeldungen:

- FATAL ERROR
- ERROR
- WARNING

FATAL ERROR führt immer zum Abbruch des laufenden Programmes. ERROR und WARNING werden angezeigt, das Programm läuft jedoch weiter.

Liste der FATAL ERRORs

Alle diese FATAL ERRORS führen bei ihrem Auftreten zum sofortigen Abbruch des Assemblerprogramms.
Alle bereits eröffneten Dateien werden nicht geschlossen.

ARGUMENTSTACK OVERFLOW: Dabei dürfen max. 10 Argumente pro Macro angegeben sein.

ARGUMENTTABLE OVERFLOW: Dabei ist die Summe der Charakter der Argumente beschränkt.

CALLSTACK OVERFLOW: In einem Macro dürfen max. 10 weitere Macros aufgerufen werden.

CANNOT CREATE FILE: Eine Zieldatei kann nicht eröffnet werden.

CANNOT OPEN FILE: Eine zu assemblierende Datei oder eine Datei, die mit INCLUDE in das Quellprogramm eingebunden werden soll, existiert nicht.

CANNOT WRITE TO DISK: Fehler beim Schreiben auf Diskette, z.B. Diskette voll.

HEAP OVERFLOW: Die Anzahl der verwendeten Symbole und Symbolreferenzen (auch von Macrodefinitionen) ist für den freien noch verfügbaren Speicher zu gross.

ILLEGAL COMMAND LINE SYNTAX: Die Syntax des Assembleraufrufs ist nicht korrekt.

INCLUDE CANNOT BE NESTED: Der Befehl INCLUDE kann nicht ausgeführt werden.

MACROTABLE OVERFLOW: Der Speicherbedarf für die Macrodefinitionen ist zu gross.

ASSEMBLIEREN

NO MACRO-CMD IN ARGUM.LIST: Ein Macroname wurde in der Parameterliste benutzt.

SOURCE FILE NOT SPECIFIED: Die zu assemblierende Datei ist nicht spezifiziert worden.

TOO MANY MACROS: Die Anzahl bzw. der Speicherbedarf der Macrodefinitionen ist zu gross.

TOO MANY SYMBOLS: Die Anzahl der verwendeten Symbole ist für den freien noch verfügbaren Stringspeicher zu gross.

Liste der ERROR Meldungen

Die Ausführung des Assemblerprogramms wird nicht unterbrochen. Dem fehlerhaften Symbol oder Label wird jedoch kein Wert zugeordnet.

ILLEGAL EXPR WITH RELATIVE OR EXTN SYM: Das entsprechende Symbol wurde in einem nicht zulässigen Ausdruck verwendet z. B. Multiplikation mit einem extern definierten Symbol.

ILLEGAL OR MISSING OPCODE OR ASSEMBL-DIR: Der genannte Operationscode oder die Assembleranweisung ist nicht erlaubt oder fehlt.

ILL. SYNTAX OR TOO MANY ARGS IN ARG-LIST: Die Syntax der Parameterliste ist nicht eingehalten worden, oder es sind zu viele Parameter aufgeführt.

MISSING FIRST OPERAND: Der erste Operand fehlt.

MISSING PARENTHESIS IN EXPRESSION: Im genannten Ausdruck fehlt eine Klammer.

MULTIPLE DEFINED SYMBOL: Das genannte Symbol ist an mehr als einer Stelle definiert worden.

OPCODE NOT OF APPROPRIATE CLASS: Der genannte Operationscode ist an dieser Stelle nicht erlaubt.

OPERAND OUT OF RANGE: Der Wert des genannten Operanden ist nicht im erlaubten Bereich.

PARAM NOT SPEC. IN MACRO- OR LOCAL LIST: Der genannte Parameter wurde in der Liste der Macrodefinition oder in der Liste der lokal verwendeten Symbole nicht spezifiziert.

SYMBOL OR ASSEMBL-CNTL EXPECTED: An der genannten Stelle wird ein Symbol oder ein Assembler-Steuerzeichen erwartet.

ASSEMBLIEREN

Liste der WARNINGs

Die Warnungen führen ebenso wie die Fehler nicht zu einem Abbruch des Assemblerprogramms.
Der Wert des entsprechenden Symbols wird jedoch beibehalten, auch wenn er falsch ist.

ILLEGAL ASSEMBL-DIR WITHIN REL SEGM: Die genannte Assembler-Anweisung darf in einem relativen Segment nicht verwendet werden.

NO LABEL ALLOWED HERE: An der genannten Stelle ist kein Label erlaubt.

UNDEFINED SYMBOL: Das genannte Symbol ist nicht definiert.

```
SAIA PCA ASSEMBLER V1.1                                                MAIN MENU
          ***** SAIA AG   Marktbereich CH   3280 Murten   *****
Directory: C:\PCASS                                          16.12.88 13.24

   Edit                    Text assembler          Compare programs
   Assemble                Disassembler            Xref listing
   link                    flow chart              File handling
   Up/download             Runtime analysis        Ms-dos command
   Online debug            Hex converter           Setup
   comms Interface         Program eproms          Quit

<ARROW>, <SPACE> or <Tab> selects operation, <CR> or <Command letter> executes
```

```
SAIA PCA LINKER MENU
File(s) to link . . . (.LNK)   HANDLING__  COUNTER___  _________  _________
                               _________  _________  _________  _________
                               _________  _________  _________  _________
                               _________  _________  _________  _________
                               _________  _________  _________  _________
                               _________  _________  _________  _________
Output file name . . . . . .   _________  (.OBJ, .LNK, .PGD, .SYM)
Relocatable file . . .(.LNK)   No
Global documentation  (.PGD)   No        Organize at base address   No
Global symbol file . .(.SYM)   No        Base address . . . . . .   0____

Enter file name, <ARROW> or <Tab> moves cursor, <CR> executes, <ESC> aborts.
```

LINK HILFSSEITE

Bearbeitet vom Assembler erzeugte LNK-Dateien und verbindet mehrere Dateien zu einer in der SAIA-PLC lauffähigen OBJ-Datei oder einer relokativen, wieder linkbaren Datei. Ausserdem kann eine PGD-Datei und eine Symboltabelle (SYM-Datei) erzeugt aber nicht ausgedruckt werden.

Bis zu 24 LNK-Dateien können im Menü definiert und gelinkt werden. Müssen mehr als 24 Dateien gelinkt werden, muss eine relokative Datei erzeugt werden, welche dann mit den weiteren Dateien verbunden werden kann.

Die Angabe des Zieldateinamens ist optional, wenn nur OBJ- und/oder SYM- Dateien erzeugt werden. In diesem Fall wird als "Output File Name" der Name der ersten Link-Datei verwendet. Wird eine wieder linkbare Datei LNK oder eine globale Programm-dokumentation PGD verlangt, so muss ein neuer Name angegeben werden; die Ausgabedatei darf nicht den gleichen Namen haben wie eine existierende LNK- oder PGD-Datei. Soll eine PGD-Datei erzeugt werden, so muss bei allen zu linkenden Dateien bereits beim Assemblieren "PGD" definiert worden sein. Für relocative Segmente "RSEG" kann als Option eine Basisadresse angegeben werden. Dateien die den Befehl ASEG beinhalten MUSSEN in der korrekten Reihenfolge gelinkt werden.
Hinweis:Für DOS-Versionen ohne Befehl APPEND, müssen sich die Ueberlagerungsdateien PCLNK.001 bis PCLNK.003 und PCKEY.PWD im aktuellen Verzeichnis befinden.

LINKEN VON PROGRAMMEN

FEHLERMELDUNGEN

FATAL ERRORS führen immer zum sofortigen Abbruch der weiteren Programmausführung.
Der Grund eines Fatal Error wird gemeldet.

- READ PAST END OF FILE
- CHECKSUM MISMATCH
- LINK FILE FORMAT ERROR
- TRANSIENT SYMBOLS STRING OVERFLOW

Diese Fehler dürfen bei ordnungsgemäss von PCAS und PCLINK generierten Dateien nicht vorkommen, es sei denn, es handelt sich um einen systembedingten Lesefehler.

- TOO MANY ABSOLUTE SEGMENTS
- TOO MANY PUBLIC SYMBOLS
- FILENAME TABLE OVERFLOW
- HEAP OVERFLOW

Diese Fehler können entstehen beim Ueberschreiten der internen Speicherlimiten. Abhilfe bringt normalerweise ein Zusammenfassen einer Gruppe von Dateien in einem weiteren Linkerdurchlauf (ausgenommen bei zu vielen PUBLIC Symbolen, deren Anzahl beschränkt ist).

- SOURCE FILE NOT SPECIFIED

Es muss in jedem Fall mindestens eine Linkdatei spezifiziert werden.

Folgende Fehler sind nur dann FATAL ERRORS, wenn eine absolute, ladbare Datei erzeugt wird. Sie enthalten dann den Zusatz: "IN LOADMODULE".

- UNDEFINED SYMBOL *)
- ILLEGAL RELOCATION VALUE AT: *)
- EXCEEDING OPERAND AT: *)
- DUPLICATE SYMBOL FOUND
- UNSATISFIED REFERENCES IN LOADMODULE

Die drei Fehlermeldungen mit *) haben bei Erzeugung einer wieder mit anderen Dateien linkbaren Datei (RELOC) keine Auswirkung.

Folgende Fehler bewirken bei Erzeugung einer ladbaren oder wieder mit anderen Dateien linkbaren Datei, dass am Schluss der Inhalt der resultierenden Datei gelöscht wird, da der Linkprozess nicht korrekt ausgeführt wurde. Beim Laden werden sie nur angezeigt.

- ADDRESS SPACE COLLISION BETWEEN
- EXCEEDING ADDRESS SPACE AT:

Das Gleiche gilt bei nur ladbaren Dateien für:

- EXCEEDING OPERAND AT:

und bei wieder linkbaren Files für:

- DUPLICATE SYMBOL FOUND

```
SAIA PCA ASSEMBLER V1.1                                                    MAIN MENU
            ***** SAIA AG   Marktbereich CH   3280 Murten  *****
Directory: C:\PCASS                                          16.12.88 13.29

  Edit                    Text assembler          Compare programs
  Assemble                Disassembler            Xref listing
  Link                    flow chart              File handling
  Up/download             Runtime analysis        Ms-dos command
  Online debug            Hex converter           Setup
  comms Interface         Program eproms          Quit

<ARROW>, <SPACE> or <Tab> selects operation, <CR> or <Command letter> executes
```

```
SAIA PCA UPLOADER AND DOWNLOADER MENU
Operation, SPACE to select . . . . . .   Download from OBJ file to PCA

Download Operation Information:
      Object file name  . . . .  (.OBJ)  HANDLING_____
      Start and End address . . . . . .  0____    8191_
      Fill unused instructions  . . . .  No fill

Upload Operation Information:
      Object file name  . . . .  (.OBJ)  _____________
      Start and End address . . . . . .  0____    8191_
      Remove instructions . . . . . . .  Remove nothing

Enter file name, <ARROW> or <Tab> moves cursor, <CR> executes, <ESC> aborts,
```

UP/DOWNLOAD HILFSSEITE

"Download" -> Laden einer durch den Linker erstellten OBJ-Datei in die PCA. "Upload" -> Programme von der PCA zurückladen in eine OBJ-Datei. Mit der Taste <SPACE> (Leerschlag) kann eine der Funktionen angewählt werden.

Der Betriebswahlschalter der PCA muss auf 'PROG' stehen. Die PCA wird via PGU-Stecker und dem Kabel PCA2.K43 mit dem PC verbunden (parallel Interface). Anschliessend kann das Laden der Daten gestartet werden.

Beim "Download" muss der Name der entsprechenden OBJ-Datei angegeben werden. Beim "Upload" muss der Name der neuen Zieldatei (OBJ) angegeben werden.

Achtung: Bereits existierende Dateinamen (OBJ) werden beim "Upload" überschrieben.

Um bei "Upload" zu vermeiden, dass der ganze Programm- und Textspeicher von der PCA übertragen wird, können die Start- und Endadressen eingegeben werden.

Beim "Download" können die leeren RAM-Speicherplätze mit NOP 0000 oder FFFF gefüllt werden (wichtig für Checksum PAS 31..46).
Beim "Upload" hingegen, ist es möglich, die NOP 0000 und/oder FFFF zu unterdrücken.

ANWENDUNGSBEISPIEL

BESCHREIBUNG DES PROGRAMMBEISPIELES

Das Programmbeispiel besteht aus den zwei Parallelprogrammen HANDLING und ZAEHLER. Das Parallelprogramm 0 ist eine Handlingaufgabe mit folgenden Funktionen:

> Ein Greifarm kann sich horizontal und vertikal bewegen und zugeführte Werkstücke greifen. Das erfolgreiche Greifen eines Werkstückes wird durch einen Kontakt signalisiert. Ist die erste Greifoperation nicht erfolgreich, so soll diese Bewegung wiederholt werden. Falls auch der zweite Greifversuch erfolglos ist, soll die Bewegung gestoppt und eine Störung gemeldet werden. Nach Behebung der Störung kann durch Betätigung eines Quittierschalters die Arbeit wieder fortgesetzt werden.

Die Vertikalbewegung wird als Macroprogramm geschrieben.
Als Parallelprogramm 1 wird ein Auf/Abwärtszähler mit Anzeige des Zählerstandes realisiert.
Die beiden Programme werden unabhängig erstellt und später gelinkt.

ANWENDUNGSBEISPIEL

LISTING DES QUELLPROGRAMMES HANDLING.SRC

```
;*****************************************
;*   DEMOPRO:   DEMONSTRATIONSPROGRAMM   *
;*****************************************

$         LIST
$         TITLE DEMONSTRATIONSPROGRAMM
$         STITLE HANDLING

; PARAMETER-DEFINITIONEN
; ----------------------

; EINGAENGE:

READY     EQU       0          ; STARTKONTAKT
LEFT      EQU       1          ; ENDSCHALTER LINKS
RIGHT     EQU       2          ; ENDSCHALTER RECHTS
DOWN      EQU       3          ; ENDSCHALTER UNTEN
UP        EQU       4          ; ENDSCHALTER OBEN
WORK      EQU       5          ; WERKSTUECK DA
QUIT      EQU       17         ; QUITTIERUNGSSCHALTER

; AUSGAENGE:

MOVRI     EQU       39         ; BEWEGUNG NACH RECHTS
MOVLE     EQU       43         ; BEWEGUNG NACH LINKS
MOVUP     EQU       42         ; BEWEGUNG NACH OBEN
MOVDO     EQU       40         ; BEWEGUNG NACH UNTEN
ERROR     EQU       44         ; STOERUNG
GRIP      EQU       41         ; GREIFER

; TIMER:

TIM1      EQU       256
VAL1      SET       40         ; 4 SEK WARTEZEIT

$         EJECTL               ;

; HAUPTPROGRAMM
; -------------

$         NOLIST
$         INCLUDE B:MVERT.SRC        ; MACRO MVERT BEREITSTELLEN
$         LIST

          ASEG                 ; ABSOLUTSEGMENT
          ORG       200        ; BASISADRESSE 200

$         MARKL     * HAUPTPROGRAMM

          PAS       1          ; ZUWEISUNG PARALLELPROGRAMM 1
          00        START2     ; ZAEHLER
          JMP       START1     ; ZUWEISUNG PARALLELPROGRAMM 0
                               ; WERKSTUECKHANDLING
```

ANWENDUNGSBEISPIEL

(Fortsetzung des Quellprogrammes HANDLING.SRC)

```
$        MARKL    * HANDLING (PP1)
START1   STH      READY    ; ABFRAGE, OB STARTKONTAKT GESCHLOSSEN
         ANL      LEFT     ; UND ENDSCHALTER LINKS OFFEN
         JIZ      $ - 2    ; -> START1, FALLS NICHT ERFUELLT
         SEO      MOVRI    ; SONST BEWEGUNG NACH RECHTS
         WIH      RIGHT    ; SOLANGE ENDSCHALTER RECHTS ZU
         REO      MOVRI    ; STOP BEWEGUNG NACH RECHTS

         JMS      DOWNUP   ; => AB/AUF-BEWEGUNG

         STH      WORK     ; WERKSTUECK DA
         JIO      GOLEFT   ; -> BEWEGUNG NACH LINKS

         JMS      DOWNUP   ; => VERTIKAL-BEWEGUNG

         STH      WORK     ; WERKSTUECK DA
         JIO      GOLEFT   ; -> BEWEGUNG NACH LINKS
         SEO      ERROR    ; SONST STOERUNGSLAMPE EIN
         WIL      QUIT     ; QUITTIERUNGSSCHALTER ABWARTEN
         REO      ERROR    ; STOERUNGSLAMPE AUS
GOLEFT   SEO      MOVLE    ; BEWEGUNG NACH LINKS
         WIH      LEFT     ; ENDSCHALTER LINKS ZU
         REO      MOVLE    ; STOP BEWEGUNG NACH LINKS
         REO      GRIP     ; GREIFER OEFFNEN

         JMS      WAIT4    ; => 4 SEK WARTEN

         JMP      START1   ; -> ANFANG PP1

$        MARKL    - SUBROUTINE TIMER

WAIT4    STR      TIM1     ; TIMER SETZEN
         TCV      VAL1     ; WARTEZEIT
         WIH      TIM1     ; WARTEN BIS ZEIT ABGELAUFEN
         RET      0

$        EJECT             ; VORSCHUB AUF SEITENANFANG

DOWNUP   MVERT    MOVDO,DOWN,GRIP,MOVUP,UP   ; MACRO MVERT
         RET      0

         EXTN     START2   ; BEGINN DES PARALLELPROGRAMMES 1

; ENDE DES PARALLELPROGRAMMES 0 (WERKSTUECKHANDLING)
```

ANWENDUNGSBEISPIEL

In gleicher Weise werden das Macroprogramm VERT.SRC und das Parallelprogramm ZAEHLER.SRC erstellt.

LISTING DES MACROPROGRAMMES VERT.SRC

```
;*************************
;* MACROPROGRAMM MVERT *
;*************************

; DAS MACROPROGRAMM MVERT STEUERT EINE VERTIKAL-BEWEGUNG.
; ENDSCHALTER LIEFERN DIE SIGNALE DER ENDSTELLUNGEN.
; AN EINEM ENDE DER BEWEGUNG WIRD EINE AKTION AUSGELOEST.
; DIE EINZELNEN BEWEGUNGEN UND DIE AKTION WERDEN DURCH
; ZEITINTERVALLE GETRENNT.

MVERT   MACRO   GODOWN,SWIDO,GR,GOUP,SWIUP
        LOCAL   WAIT2,END
$       MARKL   = VERTIKAL-BEWEGUNG

; PARAMETER:
; ----------

VAL2    EQU     20      ; WARTEZEIT
TIM2    EQU     257     ; TIMER

; PROGRAMM
; --------

        JMS     ?WAIT2  ; => 2 SEK WARTEZEIT
        SEO     %GODOWN ; BEWEGUNG NACH UNTEN BIS
        WIH     %SWIDO  ; ENDSCHALTER OFFEN
        REO     %GODOWN ; STOP BEWEGUNG
        JMS     ?WAIT2  ; => 2 SEK WARTEZEIT
        SEO     %GR     ; GREIFER SCHLIESSEN
        JMS     ?WAIT2  ; => 2 SEK WARTEZEIT
        SEO     %GOUP   ; BEWEGUNG NACH OBEN BIS
        WIH     %SWIUP  ; ENDSCHALTER OFFEN
        REO     %GOUP   ; STOP BEWEGUNG
        JMS     ?WAIT2  ; => 2 SEK WARTEZEIT
        JMP     ?END    ; SUBROUTINE INNERHALB DES MACROS
                        ; UEBERSPRINGEN

; SUBROUTINE
; ----------
$       MARKL   - SUBROUTINE TIMER (VERT. BEWEGUNG)

?WAIT2  STR     TIM2    ; TIMER
        TCV     VAL2    ; AUF 2 SEK SETZEN
        WIH     TIM2    ; TIMER ABWARTEN
        RET     0
?END    NOP     0

$       MARKL   = ENDE VERTIKAL-BEWEGUNG
        ENDM
```

ANWENDUNGSBEISPIEL

LISTING DES PARALLELPROGRAMMES ZAEHLER.SRC

```
;*****************************************
;*  DEMOPRO :  DEMONSTRATIONSPROGRAMM    *
;*****************************************

; AUF/ABWAERTSZAEHLER MIT ANZEIGE DES ZAEHLERSTANDES: ZAEHLER.SRC

$        LIST
$        TITLE DEMONSTRATIONSPROGRAMM
$        STITLE ZAEHLER

; PARAMETER-DEFINITIONEN:
; ----------------------

COUNTER EQU     268      ; ZAEHLER
RESET   EQU     10       ; RESET-EINGANG
INCREM  EQU     11       ; INCREMENT-EINGANG
DECREM  EQU     12       ; DECREMENT-EINGANG
STATUS  EQU     32       ; ZAEHLERSTATUS
CVAL    SET     0        ; ZAEHLERANFANGSWERT

; HAUPTPROGRAMM:
; -------------

        ASEG
        ORG     250      ; BASISADRESSE DES PROGRAMMES

$       MARKL   * ZAEHLER (PP2)

START2  STH     RESET    ; ZAEHLER RUECKSETZEN AUF
        SCR     COUNTER
        TCV     CVAL     ; ANFANGSWERT

        STH     INCREM   ; ZAEHLER ERHOEHEN
        DYN     500      ; FLANKENTRIGGERUNG
        INC     COUNTER  ; CVAL + 1

        STH     DECREM   ; ZAEHLER ERNIEDRIGEN
        DYN     501      ; FLANKENTRIGGERUNG
        DEC     COUNTER  ; CVAL - 1

        STH     COUNTER  ; ZAEHLERZUSTAND ABFRAGEN
        OUT     STATUS   ; STATUSANZEIGE

        SEA     0        ; ACCU = 1 FUER DTC
        DTC     COUNTER  ; ANZEIGE DES ZAEHLERINHALTES
        JMP     START2   ; -> ANFANG PP2

        PUBL    START2

; ENDE DES PARALLELPROGRAMMES 1 (ZAEHLER)
```

ANWENDUNGSBEISPIEL

ASSEMBLERLISTING HANDLING.LST

Das Assemblerlisting kann mit dem DOS-Befehl TYPE auf dem Bildschirm dargestellt werden:

```
A>TYPE B:handling.LST     <CR>
```

Falls diese Datei auf einen Drucker ausgegeben werden soll, muss ein Drucker mit 95 Zeichen/Zeile verwendet werden. Damit kann der gesamte Kommentar dargestellt werden.

```
*** SAIA - PC *** DEMONSTRATIONSPROGRAMM                    PAGE 1
MACRO - ASSEMBLER HANDLING

SODECO-SAIA CH-3280 MURTEN

LINE   ADDR MC OPRD UI  LABEL   MNC     OPRD            COMMENT
   1                    ;*****************************************
   2                    ;*  DEMOPRO :  DEMONSTRATIONSPROGRAMM    *
   3                    ;*****************************************
   4
   5                    $       LIST
   6                    $       TITLE DEMONSTRATIONSPROGRAMM
   7                    $       STITLE HANDLING
   8
   9                    ; PARAMETER-DEFINITIONEN
  10                    ; ----------------------
  11
  12                    ; EINGAENGE:
  13
  14             0      READY   EQU     0       ; STARTKONTAKT
  15             1      LEFT    EQU     1       ; ENDSCHALTER LINKS
  16             2      RIGHT   EQU     2       ; ENDSCHALTER RECHTS
  17             3      DOWN    EQU     3       ; ENDSCHALTER UNTEN
  18             4      UP      EQU     4       ; ENDSCHALTER OBEN
  19             5      WORK    EQU     5       ; WERKSTUECK DA
  20            17      QUIT    EQU     17      ; QUITTIERUNGSSCHALTER
  21
  22                    ; AUSGAENGE:
  23
  24            39      MOVRI   EQU     39      ; BEWEGUNG NACH RECHTS
  25            43      MOVLE   EQU     43      ; BEWEGUNG NACH LINKS
  26            42      MOVUP   EQU     42      ; BEWEGUNG NACH OBEN
  27            40      MOVDO   EQU     40      ; BEWEGUNG NACH UNTEN
  28            44      ERROR   EQU     44      ; STOERUNG
  29            41      GRIP    EQU     41      ; GREIFER
  30
  31                    ; TIMER:
  32
  33           256      TIM1    EQU     256
  34            40      VAL1    SET     40      ; 4 SEK WARTEZEIT
  35
  36                    $       EJECTL          ;
```

ANWENDUNGSBEISPIEL

```
*** SAIA - PC *** DEMONSTRATIONSPROGRAMM                                    PAGE   2
MACRO - ASSEMBLER HANDLING

SODECO-SAIA CH-3280 MURTEN

LINE   ADDR MC OPRD UI  LABEL   MNC     OPRD            COMMENT
  37
  38                    ; HAUPTPROGRAMM
  39                    ; -------------
  40
  41                $       NOLIST
  92                $       LIST
  93
  94                        ASEG            ; ABSOLUTSEGMENT
  95                        ORG     200     ; BASISADRESSE 200
  96
  97                $       MARKL   * HAUPTPROGRAMM
  98
  99   200 29    1          PAS     1       ; ZUWEISUNG PARALLELPROGRAMM 1
 100   201 00    0 E        00      START2  ; ZAEHLER
 101   202 20  203          JMP     START1  ; ZUWEISUNG PARALLELPROGRAMM 0
 102                                        ; WERKSTUECKHANDLING
 103
 104                $       MARKL   * HANDLING (PP1)
 105   203 01    0    START1 STH     READY   ; ABFRAGE, OB STARTKONTAKT GESCHLOSSEN
 106   204 04    1          ANL     LEFT    ; UND ENDSCHALTER LINKS OFFEN
 107   205 22  203          JIZ     $ - 2   ; -> START1, FALLS NICHT ERFUELLT
 108   206 11   39          SEO     MOVR1   ; SONST BEWEGUNG NACH RECHTS
 109   207 25    2          WIH     RIGHT   ; SOLANGE ENDSCHALTER RECHTS ZU
 110   208 12   39          REO     MOVR1   ; STOP BEWEGUNG NACH RECHTS
 111
 112   209 23  228          JMS     DOWNUP  ; => AB/AUF-BEWEGUNG
 113
 114   210 01    5          STH     WORK    ; WERKSTUECK DA
 115   211 21  218          JIO     GOLEFT  ; -> BEWEGUNG NACH LINKS
 116
 117   212 23  228          JMS     DOWNUP  ; => VERTIKAL-BEWEGUNG
 118
 119   213 01    5          STH     WORK    ; WERKSTUECK DA
 120   214 21  218          JIO     GOLEFT  ; -> BEWEGUNG NACH LINKS
 121   215 11   44          SEO     ERROR   ; SONST STOERUNGSLAMPE EIN
 122   216 26   17          WIL     QUIT    ; QUITTIERUNGSSCHALTER ABWARTEN
 123   217 12   44          REO     ERROR   ; STOERUNGSLAMPE AUS
 124   218 11   43    GOLEFT SEO     MOVLE   ; BEWEGUNG NACH LINKS
 125   219 25    1          WIH     LEFT    ; ENDSCHALTER LINKS ZU
 126   220 12   43          REO     MOVLE   ; STOP BEWEGUNG NACH LINKS
 127   221 12   41          REO     GRIP    ; GREIFER OEFFNEN
 128
 129   222 23  224          JMS     WAIT4   ; => 4 SEK WARTEN
 130
 131   223 20  203          JMP     START1  ; -> ANFANG PP1
 132
 133                $       MARKL   - SUBROUTINE TIMER
 134
 135   224 14  256    WAIT4 STR      TIM1    ; TIMER SETZEN
 136   225 00   40          TCV     VAL1    ; WARTEZEIT
 137   226 25  256          WIH     TIM1    ; WARTEN BIS ZEIT ABGELAUFEN
 138   227 24    0          RET     0
 139
 140                $       EJECT           ; VORSCHUB AUF SEITENANFANG
```

ANWENDUNGSBEISPIEL

```
*** SAIA - PC *** DEMONSTRATIONSPROGRAMM                                    PAGE   3
MACRO - ASSEMBLER HANDLING

SODECO-SAIA CH-3280 MURTEN

LINE   ADDR MC OPRD UI  LABEL   MNC     OPRD            COMMENT
  141
  142              M   DOWNUP  MVERT   MOVDO,DOWN,GRIP,MOVUP,UP  ; MACRO MVERT
  143              I   $       MARKL   = VERTIKAL-BEWEGUNG
  144              I
  145              I                                   ; PARAMETER:
  146              I                                   ; ---------
  147              I
  148           20 I   VAL2    EQU     20              ; WARTEZEIT
  149          257 I   TIM2    EQU     257             ; TIMER
  150              I
  151              I                                   ; PROGRAMM
  152              I                                   ; --------
  153              I
  154   228 23 240 I           JMS     ??0             ; => 2 SEK WARTEZEIT
  155   229 11  40 I           SEQ     MOVDO           ; BEWEGUNG NACH UNTEN BIS
  156   230 25   3 I           WIH     DOWN            ; ENDSCHALTER OFFEN
  157   231 12  40 I           REQ     MOVDO           ; STOP BEWEGUNG
  158   232 23 240 I           JMS     ??0             ; => 2 SEK WARTEZEIT
  159   233 11  41 I           SEQ     GRIP            ; GREIFER SCHLIESSEN
  160   234 23 240 I           JMS     ??0             ; => 2 SEK WARTEZEIT
  161   235 11  42 I           SEQ     MOVUP           ; BEWEGUNG NACH OBEN BIS
  162   236 25   4 I           WIH     UP              ; ENDSCHALTER OFFEN
  163   237 12  42 I           REQ     MOVUP           ; STOP BEWEGUNG
  164   238 23 240 I           JMS     ??0             ; => 2 SEK WARTEZEIT
  165   239 20 244 I           JMP     ??1             ; -> SUBROUTINE INNERHALB DES MACROS
  166              I                                   ; UEBERSPRINGEN
  167              I
  168              I                                   ; SUBROUTINE
  169              I                                   ; ----------
  170              I   $       MARKL   - SUBROUTINE TIMER (VERT. BEWEGUNG)
  171              I
  172   240 14 257 I   ??0     STR     TIM2            ; TIMER
  173   241 00  20 I           TCV     VAL2            ; AUF 2 SEK SETZEN
  174   242 25 257 I           WIH     TIM2            ; TIMER ABWARTEN
  175   243 24   0 I           RET     0
  176   244 00   0 I   ??1     NOP     0
  177              I
  178              I   $       MARKL   = ENDE VERTIKAL-BEWEGUNG
  179              M           ENDM
  180   245 24   0             RET     0
  181
  182
  183                          EXTN    START2  ; BEGINN DES PARALLELPROGRAMMES 1
  184
  185                  ; ENDE DES PARALLELPROGRAMMES 0 (WERKSTUECKHANDLING)
```

ANWENDUNGSBEISPIEL

```
*** SAIA - PC *** DEMONSTRATIONSPROGRAMM                                    PAGE   4
MACRO - ASSEMBLER HANDLING

************************** SYMBOL - TABLE **************************
(xx = E (EXTERNAL), xx = P (PUBLIC), xx = C (RELOCATIVE), xx = PC (P & C))
LABELS:
??0------239     ??1------244     DOWNUP---228     GOLEFT---218
START1---203     START2-----0 E   WAIT4----224

SYMBOLS:
DOWN-------3     ERROR-----44     GRIP------41     LEFT-------1
MOVDO-----40     MOVLE-----43     MOVRI-----39     MOVUP-----42
QUIT------17     READY------0     RIGHT------2     TIM1-----256
TIM2-----257     UP----------4    VAL1------40     VAL2------20
WORK-------5
```

```
***** SAIA - PC ***** DEMONSTRATIONSPROGRAMM                                 PAGE   5
PC - ASSEMBLER  C8  - HANDLING

************************ CROSS REFERENCE LIST ************************
(THE LINE WHICH DEFINES THE SYMBOL IS MARKED WITH A "*")
??0      154   158   160   164   172*
??1      165   176*
DOWN      17*  156
DOWNUP   112   117   142*
ERROR     28*  121   123
GOLEFT   115   120   124*
GRIP      29*  127   159
LEFT      15*  106   125
MOVDO     27*  155   157
MOVLE     25*  124   126
MOVRI     24*  108   110
MOVUP     26*  161   163
QUIT      20*  122
READY     14*  105
RIGHT     16*  109
START1   101   105*  131
START2   100
TIM1      33*  135   137
TIM2     149*  172   174
UP        18*  162
VAL1      34*  136
VAL2     148*  173
WAIT4    129   135*
WORK      19*  114   119

   0   Error(s)
```

```
SAIA PCA ASSEMBLER V1.1                                                    MAIN MENU

            ***** SAIA AG   Marktbereich CH   3280 Murten  *****

Directory: C:\PCASS                                          16.12.88 13.30

    Edit                    Text assembler          Compare programs

    Assemble                Disassembler            Xref listing

    Link                    flow chart              File handling

    Up/download             Runtime analysis        Ms-dos command

    Online debug            Hex converter           Setup

    comms Interface         Program eproms          Quit

<ARROW>, <SPACE> or <Tab> selects operation, <CR> or <Command letter> executes
```

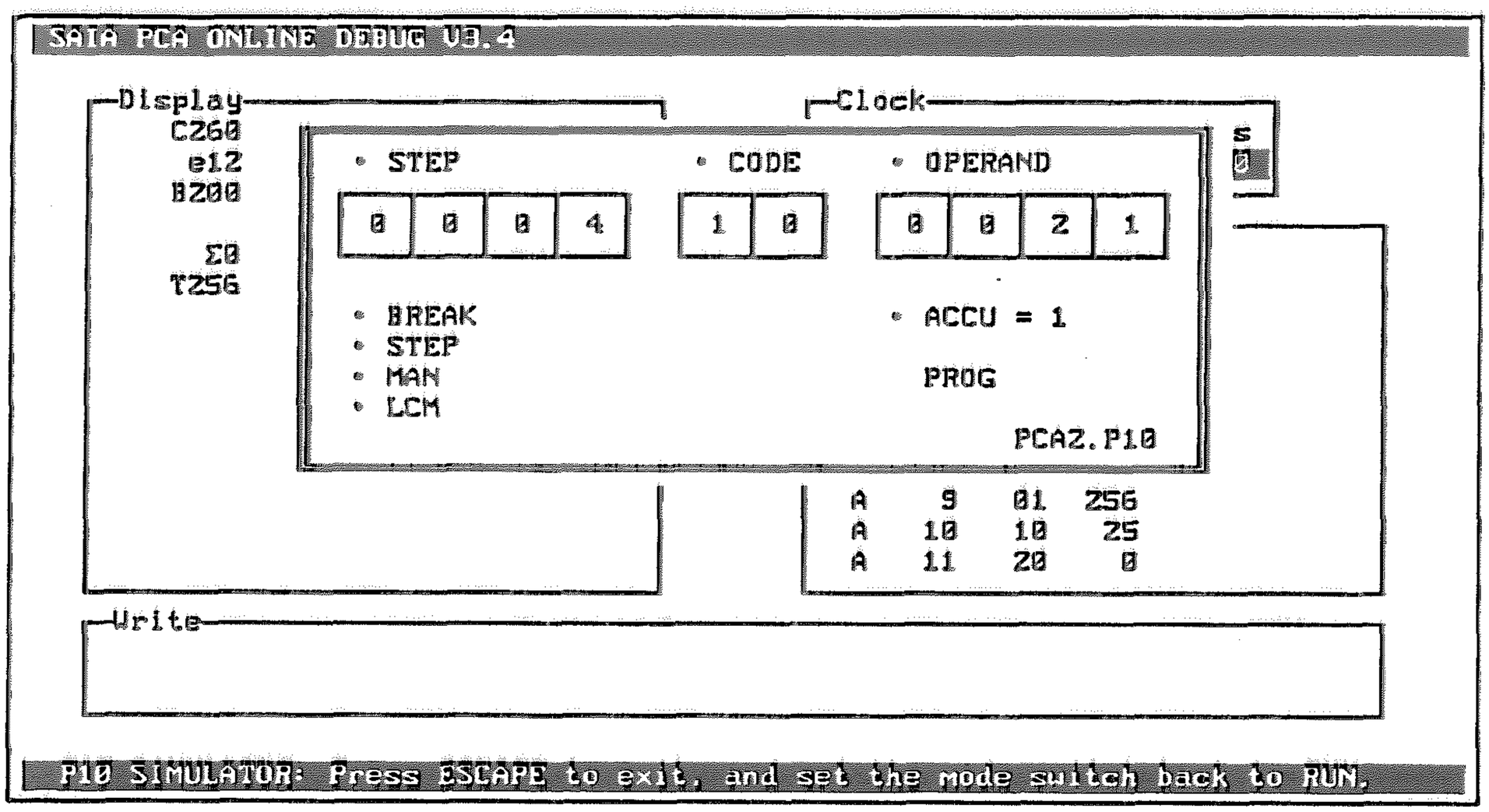
```
SAIA PCA ONLINE DEBUG V3.4

Display                                   Clock
    C260    2090                          yy dw yy mo dm hh mm ss
     e12    0011 0010                     88 03 88 12 15 17 05 58
    B200    EE EF FF 3B 67

                                          Program
     Σ0     1101 0000 0100 0011           A    1   03    2
    TZ56    2163                          A    2   03    7
                                          A    3   10   12
                                          A    4   10   21
                                          A    5   10   13
                                          A    6   10   24
                                          A    7   14  256
                                          A    8   00  500
                                          A    9   01  256
                                          A   10   10   25
                                          A   11   20    0

Write

Display  Write  Program  clock  Save  Load  comms-Interface  Reset  Quit
```



ONLINE DEBUG HILFSSEITE

Mit dem "Online Debug"-Programm können im RUN-Betrieb folgende Daten angezeigt und verändert werden: Elemente, Timer, Zähler, Register, Uhr und Programmcode. Der Betriebswahlschalter der PCA muss auf "RUN" stehen. Die PCA ist vor der "Online Debug"-Anwahl via PGU-Stecker (Kabel PCA2.K43) mit dem IBM-PC (parallel Interface) zu verbinden.

Die Befehle sind aus dem Menü am unteren Bildrand zu entnehmen. Anwahl durch Eingabe der Grossbuchstaben.

Werte im "Display"- und "clocK"-Fenster werden dauernd aufgefrischt. Die Auffrischzeit ist von der Anzahl der Werte abhängig. Die Anzeigen im Displayfenster können mit "Save" gespeichert und mit "Load" wieder aufgerufen werden (Dateiname angeben).

Mit dem "comms Interface", das mit der Taste "I" oder vom Hauptmenu aus ge- startet werden kann, ist ein Arbeiten in den anderen Betriebsmodi möglich. Dazu kann mit der Taste "F10" der P10 Simulator (P10.EXE) in einem zusätzlichen Fenster aufgerufen werden (siehe Bild oben).

Hinweis: Ältere PCA-Systemprogramme und die Baureihen PCA13 und PCA210 ermöglichen die "Online Debug" Funktionen nicht (Liste im Handbuch).

```
SAIA PCA ASSEMBLER V1.1                                          MAIN MENU
          ***** SAIA AG   Marktbereich CH   3280 Murten  *****
Directory: C:\PCASS                                     16.12.88 13.31

    Edit                    Text assembler          Compare programs
    Assemble                Disassembler            Xref listing
    Link                    flow chart              File handling
    Up/download             Runtime analysis        Ms-dos command
    Online debug            Hex converter           Setup
    comms interface         Program eproms          Quit

<ARROW>, <SPACE> or <Tab> selects operation, <CR> or <Command letter> executes
```

(C) SAIA AG

SAIA - PLC

COMMUNICATION INTERFACE

DOS VERSION: 1988.02

```
SAIA PCA ASSEMBLER - VERSION 19-07-88                                  MAIN MENU
                    COMMUNICATIONS INTERFACE KOMMANDOÜBERSICHT
              BETRIEBSART MUSS AM WAHLSCHALTER EINGESTELLT WERDEN !!
RUN  Run.          Setze die Startadresse mit SST
BRK  Break.        AAA:yyyy Ausführen bis yyyy, <CR>/<+> steps, <ANY CHAR> stop
SST  Step.         AAA:xxxx Sprung bis xxxx, dann <CR>/<+> steps, ACCU anzeigen
MAB  ManBit.       AAA:eee  Wähle Element, EEE:0/1 Setzen oder Rücksetzen
MAW  ManBcd.       AAA:rrr  Wähle Register, EEE:nn Eingabe neuer Wert.
TEX  Text.         AAA:tctc Charakter-Adresse, EEE:nn Eingabe char nn dezimal
LCM  LoadCopyMem.  LCM:xxxx,:yyyy Kopiere von xxxx bis yyyy, nur PCA2.M1../M3..
COM  Compare Mem.  COM:xxxx,:yyyy Vergleiche xxxx bis yyyy, Anzeige letzte Adr.
PRG  Prog.         AAA:aaaa Adresse,EEE:mnc oprd Schreibe neue Programmzeile
                   CCC lösche Eingabezeile   , <CR>/<+> nächste, <-> zurück,
                   IAD Adresse einschieben, KAD Adresse entfernen
             KOMMANDO, BETIEBSARTENSCHALTER MUSS ENTSPRECHEND STEHEN
SCH:xxxx,:yyyy,:oprd Suche Operand      | CLM:xxxx,:yyyy  Programm löschen (RAM)
LST:xxxx,:yyyy       Liste Flussdiagr.  | LWC:xxxx,:yyyy  Liste mit Kommentar
DME:xxxx,:yyyy       Anzeige Programm   | DWC:xxxx,:yyyy  Anzeige mit Kommentar
DIO:eee,:eee         Anzeige Elemente   | DTX:ttt,:ttt    Anzeige von Text
DRG:rrr,:rrr         Anzeige Register   | PAG             Seitenformat
MET:yyyy    Arbeitsbereich für IAD/KAD  |
Ctrl-P  Printer ein/aus.  Ctrl-S  Pause.  Ctrl-Q  weiter.   X  Stop.  G weiter

Press any key to return to the menu
```

COMMUNICATION-INTERFACE

Wichtig: Die Fernwahl der PC-Betriebsarten ist nicht vom IBM-PC aus möglich. Die richtige Betriebsart wird am Bildschirm angezeigt und muss von Hand an der SAIA-PC eingestellt werden.

Anschluss der SAIA-PC: Die Verbindung zwischen IBM-PC und PCA ist die gleiche wie beim Assembler. Siehe Kap. 1.3.3 oder 5.4.

BESCHREIBUNG DER BEFEHLE

Für die Beschreibung der Befehle wird in den Kap. 14.3 und 14.4 folgende Symbolik verwendet:

Eingabe von Ziffern: Nur die letzten 4 eingegebenen Ziffern sind signifikant (1220125 = 0125).

xxxx	Anfangsadresse
yyyy	Endadresse
zzzz	Vergleichsadresse
oooo	Operand allgemein
eee	Elementadresse
rrr	Registeradresse (Wortregister)
ttt	Textnummer
tctc	Adresse eines Textcharakters
MMM	Mnemocode, bestehend aus 3 Charakter (keine Ziffern) mit folgenden Bedeutungen: - Programmierbefehle der SAIA-PC-Anwendersprache (nur in der Betriebsart PRG). - Steuerbefehle für das Programm, das das Communication-Interface simuliert.
+ - ; ,	Charakter, welche verschiedenen Bedeutungen haben können.

<ANY CHAR> Betätigung einer beliebigen Taste.

AAA	ADDRESS	(entspricht Taste A auf P10)
EEE	ENTER	(entspricht Taste E auf P10)
CCC	CLEAR	(entspricht Taste C auf P10)

AKTIVIERUNG DES DRUCKERS CTRL/P

Im Zusammenhang mit allen folgenden Betriebsarten und Befehlen kann ein Drucker jederzeit aktiviert oder deaktiviert werden. Damit ist es möglich, alle Ausgaben wahlweise nur auf den Bildschirm oder auf den Bildschirm und auf den Drucker auszugeben.

Der Drucker wird mit dem Befehl CTRL/P aktiviert und mit dem gleichen Befehl wieder deaktiviert.

COMMUNICATION-INTERFACE

PC-BETRIEBSARTEN

Im Gegensatz zum Hardware-Communication-Interface ist eine Fernwahl der Betriebsart vom Computer aus nicht möglich. Am Bildschirm erscheint ein Hinweis, die CPU in die entsprechende Betriebsart zu schalten. Dies muss von Hand mit dem Betriebsartenschalter an der CPU vor genommen werden.

RUN — Betriebsart RUN (Programmablauf)
Ausgabe: PROGRAMM RUNNING
Soll das Programm ab einer bestimmten Startadresse laufen, so ist diese auf SST vorzuwählen.
Anschliessend RUN. Dieser RUN-Betrieb soll nicht für Dauerbetrieb verwendet werden.

TEST — Diese Betriebsart ist nicht ausführbar.

BRK — Betriebsart BREAK

<CR> bewirkt Abarbeiten einer Programmzeile.
AAA(:)yyyy <CR> bewirkt Abarbeiten aller Programmzeilen im automatischen Schrittbetrieb bis yyyy.
<ANY CHAR> stoppt die automatische Schrittverarbeitung.
Anschliessend ist mit <CR> oder + die Abarbeitung des Programmes in Einzelschritten möglich.

SST — Betriebsart STEP

<CR> bewirkt Abarbeiten einer Programmzeile.
AAA(:)xxxx <CR> bewirkt Springen auf Adresse xxxx.
Weiteres <CR> oder + bewirkt Abarbeitung des Programmes in Einzelschritten.
Zu jeder Programmzeile wird auch der Status des ACCU ausgegeben.

Hinweis:
Falls Parallelprogramme bestehen und die Assignierung vorgängig abgearbeitet wurde, so folgt das Programm im Einzelschrittbetrieb auch dem Programmwechsel entsprechend dem RUN-Betrieb (siehe Softwarebibliothek PP1).

MAB — Betriebsart MAN-BIT

AAA(:)eee <CR> bewirkt Abfragen und Ausgeben des logischen Zustandes eines Elementes.
Zusätzliche Eingabe von EEE erlaubt über die Tasten 0 oder 1 Setzen bzw. Rücksetzen des adressierten Elementes (ausser Eingängen).
<CR> oder + erhöht die Adresse des Elementes um 1,
- erniedrigt die Adresse um 1.

COMMUNICATION-INTERFACE

MAW

Betriebsart MAN-BCD (nur mit PCA23)

Durch manuellen Zugriff zum Wortregister können Werte ausgelesen oder eingegeben werden. Der Betriebsartenschalter muss auf MAN-BCD stehen.

AAA(:)rrr <CR> bewirkt Ausgabe des Registerinhaltes (2 Stellen).
Eingabe eines Wertes (z.B. 65):
AAA(:)rrr EEE(:) 65 +
Für die Adressweiterschaltung und Korrektur gelten die gleichen Regeln wie bei PRG.

TEX

Betriebsart TEXT (nur mit PCA 23 oder PCA 14)

Die Texte werden auf das entsprechende RAM-Modul geschrieben. Vorteilhafterweise verwendet man dazu die serielle Schnittstelle der PC, welche über einen komfortablen Editor verfügt.
Der Betriebsartenschalter muss auf der Stellung TEXT stehen.
AAA(:)tctc <CR> bewirkt Ausgabe des Charakters mit der Adresse tctc als ASCII-Nummer (dezimal) und der Charakter selbst. Beispiel: 83.S
Soll auf der Charakter-Adresse tctc der Charakter S eingegeben werden, so wird wie folgt vorgegangen:
AAA(:)tctc EEE(:) 83 +
D.h. zur Eingabe muss die Dezimalnummer der ASCII-Tabelle verwendet werden.
Für die Adressweiterschaltung und Korrektur gelten die gleichen Regeln wie bei PRG.

LCM

Betriebsart LCM (Load Copy Memory) nur PCA2

Es werden die Anfangs- und Endadressen für den Kopiervorgang verlangt. Trennung der Adressen durch Komma.
LCM(:)xxxx, (:)yyyy <CR>
In der folgenden Zeile wird mit der Zahl zzzz angegeben, bis zu welcher Schrittadresse der Vergleich erfolgreich war. Ist zzzz = yyyy, so sind die verglichenen Programmbereiche identisch.

COM

COMPARE MEMORY (nur mit PCA23)

Inhaltsvergleich des Hauptspeichers (oben) mit dem Kopierspeicher (unten). Es können Anwenderprogramme oder Texte verglichen werden.
Es werden die Anfangs- und Endadressen des zu vergleichenden Programmteiles verlangt.
COM(:)xxxx, (:)yyyy <CR>
In der folgenden Zeile wird mit der Zahl zzzz angegeben, bis zu welcher Schrittadresse der Vergleich erfolgreich war. Ist zzzz = yyyy, so sind die verglichenen Programmbereiche identisch.

COMMUNICATION-INTERFACE

PRG Betriebsart PROG (Programmieren)

Zusatzbefehle in dieser Betriebsart:

AAA ADDRESS: Eingabe der Schrittadresse. Mit <CR> abschliessen. Nur die letzten 4 Ziffern sind signifikant.

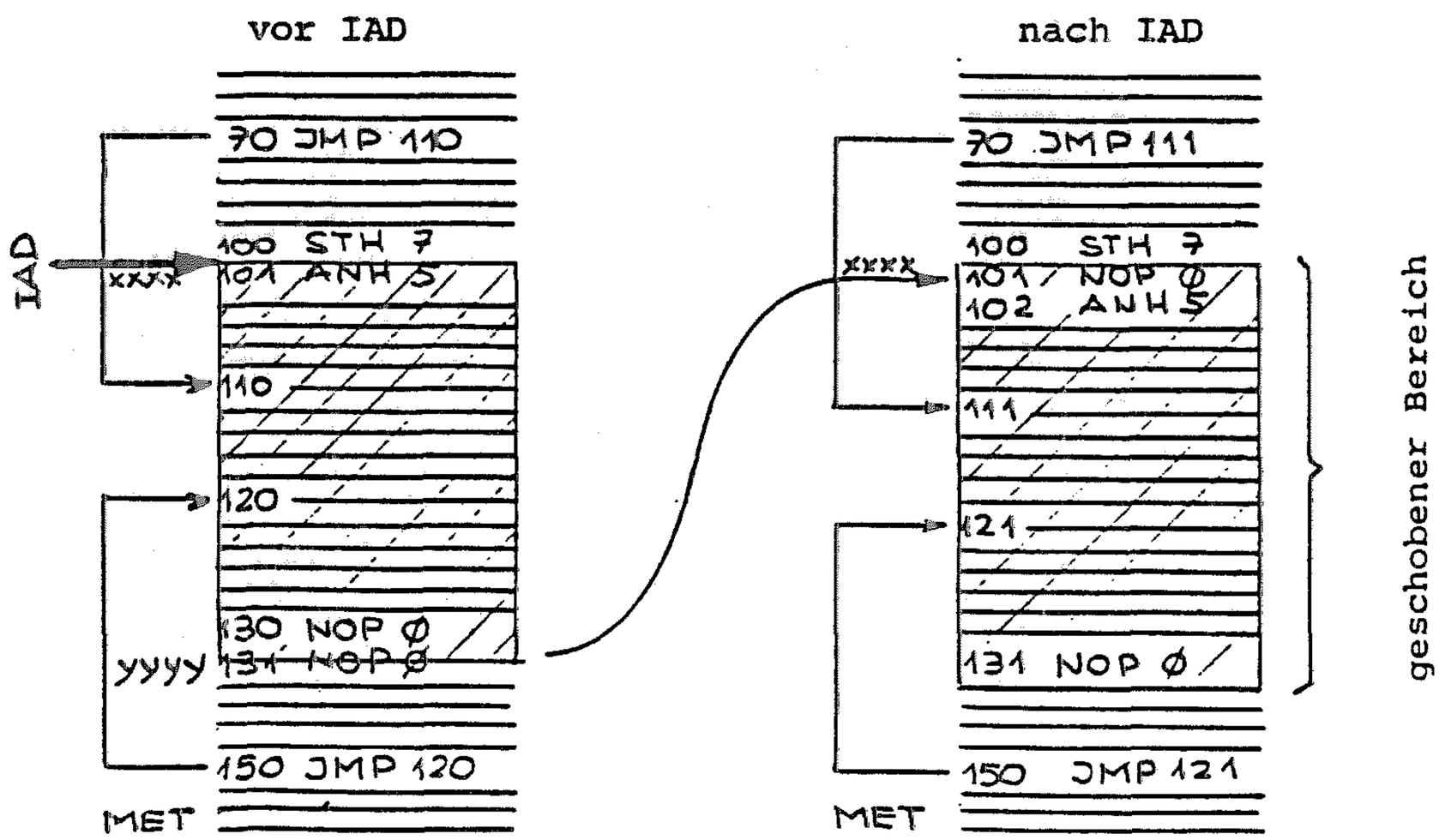
EEE ENTER: Bewirkt die Registerfreigabe für die Eingabe eines mnemonischen Befehlscodes (oder eines 2-stelligen Zahlencodes) und eines max. 4-stelligen Operanden.
Beispiel: EEE (375:) STH 12 <CR>
Die Programmzeile kann auch mit AAA, +, - oder EEE abgeschlossen werden.

\+ Bewirkt Erhöhen der Schrittadresse um 1 (gleiche Funktion wie Taste + auf P10). Für die Wiederholung dieser Funktion kann anschliessend <CR> gedrückt werden.

\- Bewirkt Erniedrigen der Schrittadresse um 1 (gleiche Funktion wie die Taste - auf dem P10). Die Syntax bei mehrzeiligen Befehlen und bei Wechsel von Bit- nach Wortprozessor und umgekehrt wird mit der Minustaste nicht korrekt ausgeführt.

CCC CLEAR: Löschen der eingegebenen Programmzeile (die noch nicht mit <CR> abgeschlossen wurde) und Bereitschaft (ohne EEE) zur Aufnahme der korrigierten Zeile.

IAD INSERT ADDRESS: Einfügen einer zusätzlichen Programmzeile.
Der Vorgang ist anhand des nachstehenden Schemas gut zu verfolgen:



COMMUNICATION-INTERFACE

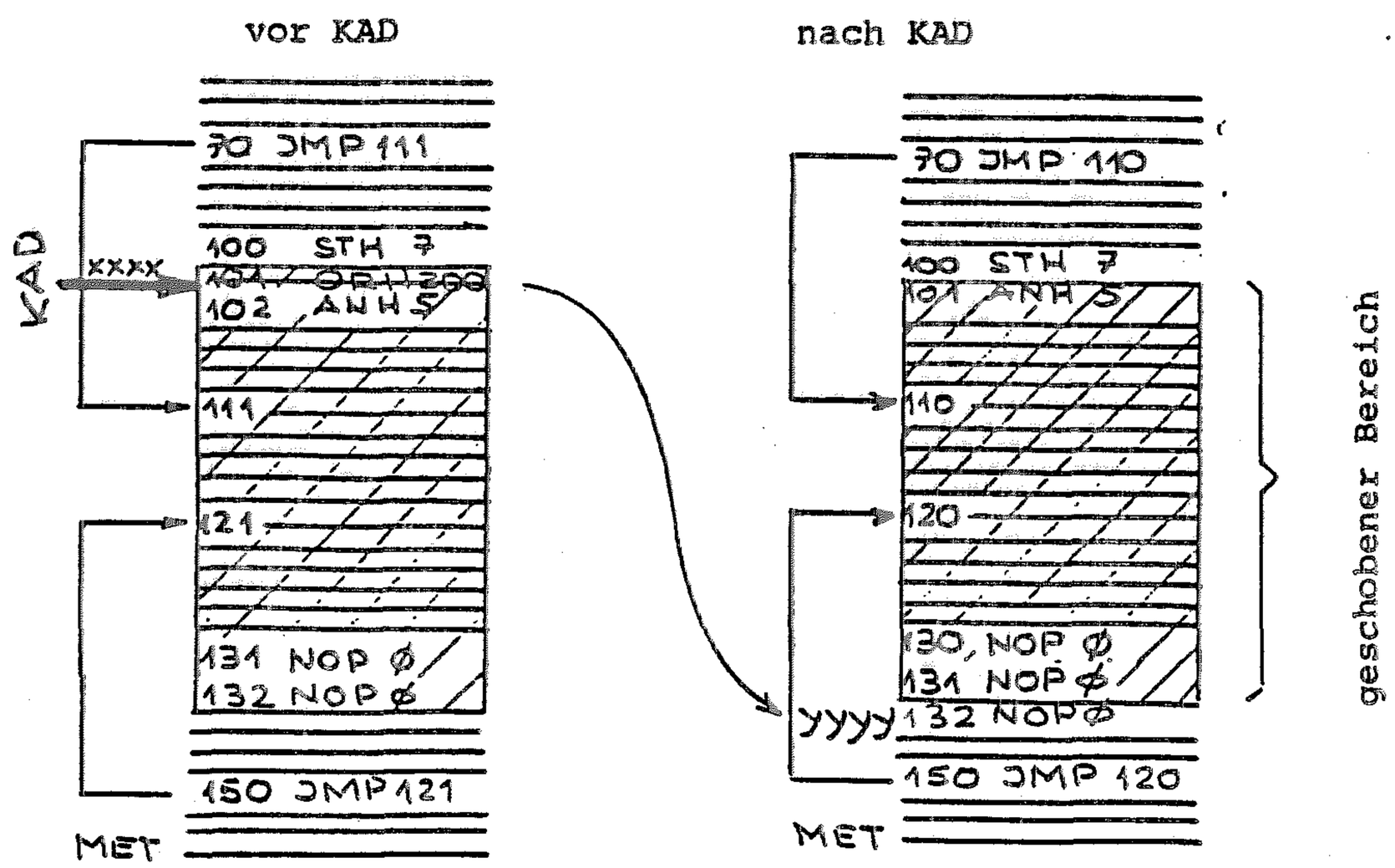
Mit AAA(:) xxxx <CR> die Schrittadresse anwählen, wo die Zeile eingeschoben werden soll.

Durch Eingabe des Befehls IAD werden anschliessend in den höheren Schrittadressen bis "MET" (Memory Top) mind. 2 sich folgende NOP 0 gesucht. Sobald diese NOP 0 gefunden sind, erscheint die Antwort: UPPER = yyyy (Adresse des 2. NOP 0) und verschiebt alle dazwischenliegenden Befehlszeilen um +1.

Dieses bei yyyy gefundenen NOP 0 wird auf die Adresse xxxx transportiert, und anschliessend werden im Bereich 0 bis MET alle Sprungadressen entsprechend korrigiert. Anschliessend kann die gewünschte Programmzeile an dieser Stelle eingegeben werden.
Wird IAD auf die <u>Zieladressen</u> von Sprungbefehlen anwendet, so werden deren Operanden um <u>+1 erhöht</u>.
In 4 Fällen wird IAD nicht akzeptiert:
- Wenn bis 2k Programmschritte bzw. bis "MET" keine Doppel-NOP gefunden wird.
- Wenn das INSERT auf der 2. Zeile eines zweizeiligen Befehles gemacht werden sollte.
- Innerhalb eines 10-zeiligen PAS-Befehles
- Wenn IAD über "MET" bzw. über 2k liegt

Die Antwort lautet in diesen Fällen: IAD/KAD REFUSED

KAD KILL ADDRESS: Eliminieren einer unerwünschten Programmzeile (sofern man diese nicht einfach mit NOP 0 überschreiben will).
Der Vorgang wird anhand des nachfolgenden Schemas erläutert:



COMMUNICATION-INTERFACE

Mit AAA(:) xxxx <CR> wird die zu eliminierende Pro-
Programmzeile angewählt. Durch Eingabe von KAD wird
diese Zeile gelöscht und als NOP 0 soweit Richtung
höherer Adresse verschoben (max. bis "MET") bis min.
zwei sich folgende NOP 0 gefunden worden sind.
Diese Adresse wird auf dem Bildschirm angegeben:
UPPER = yyyy
Die Befehlszeilen zwischen xxxx und yyyy werden an-
schliessend um -1 verschoben und die auf diesen Be-
reich adressierten Sprungadressen (bis "MET") korri-
giert.

Wird KAD auf die Zieladresse von Sprungbefehlen
angewendet, so werden deren Operanden nicht verän-
dert.

In 4 Fällen wird KAD verweigert:
- Wenn bis "MET" kein Doppel-NOP gefunden wird
- Wenn KAD auf NOP 1248 angewendet werden soll
- Wenn KAD auf EWP 0 angewendet werden soll
- für alle mehrzeiligen Befehle

Die Antwort lautet in diesen Fällen: IAD/KAD REFUSED

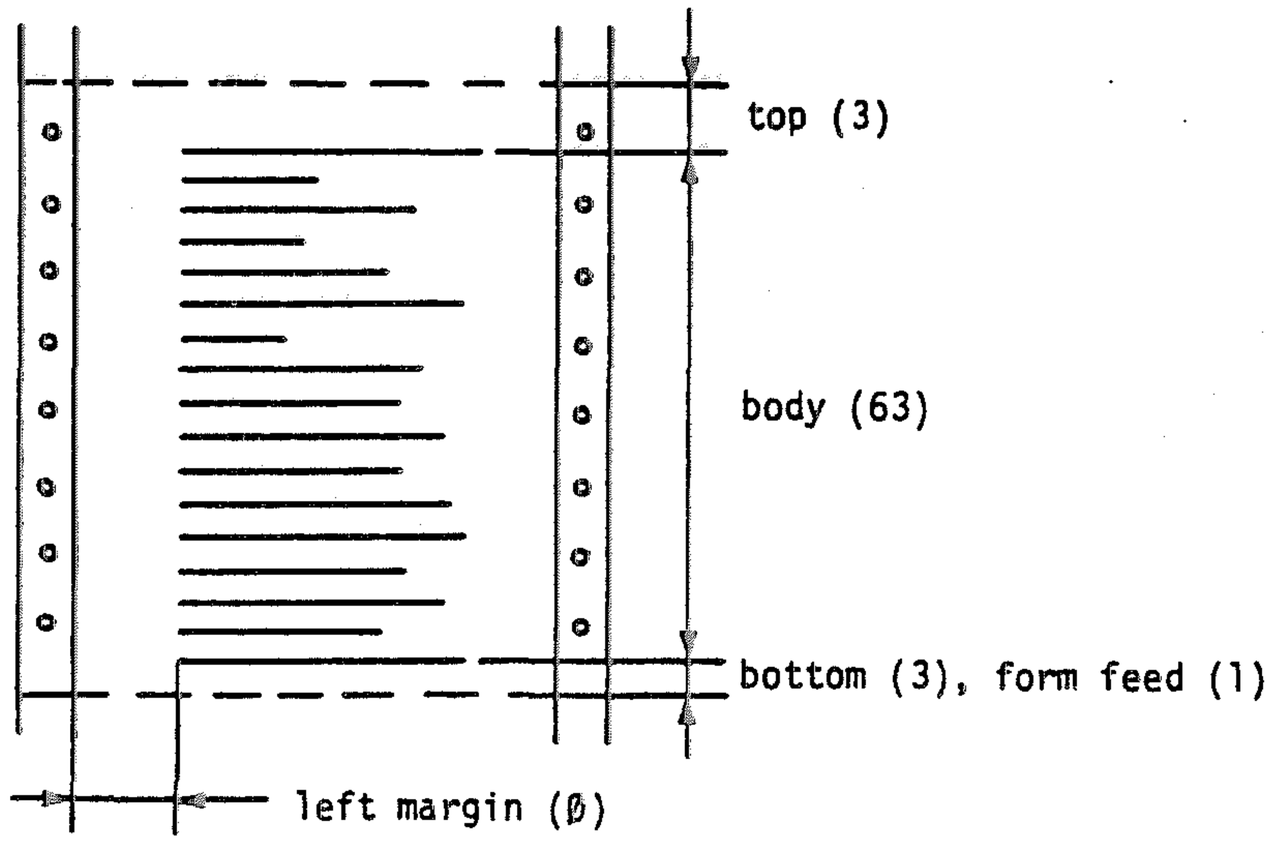
BEFEHLE

Allgemeiner Hinweis: JMP 0, JIO 0, JIZ 0 und JMS 0 werden als
zweizeiliger Sprungbefehl interpretiert.
Alle übrigen Sprungbefehle < 2k können
sowohl einzeilig als auch zweizeilig sein.

Die Stellung des Betriebswahlschalters für die folgenden Befehle
wird jeweils bei der Eingabe am Bildschirm angegeben.

PAG PAGE: Festlegen der Seiteneinteilung

Auf dem Bildschirm erscheint:
PAG top, body, bottom, form feed (1/0), left margin
Im Seitenformat bedeutet dies die entsprechende An-
zahl Charakter bzw. Zeilen (max. 255).



COMMUNICATION-INTERFACE

Wird PAG nicht eingegeben, so gelten automatisch die in Klammern angegebenen Werte.
Wird eine abweichende Seiteneinteilung gewünscht, so werden nach PAG die entsprechenden Zahlenwerte, durch Komma getrennt, eingegeben.
Beispiel: 6, 60, 6, 0, 10 <CR>
Soll kein Seitensprung und kein linker Randabstand erfolgen, so muss lediglich eingetippt werden:
0, 1 <CR>

SCH — SEARCH: Suche nach einem bestimmten Operanden.
Es wird die Anfangs- und Endadresse des zu untersuchenden Programmabschnittes, sowie der zu suchende Operand als Eingabe verlangt. Trennung durch Komma.
SCH(:) xxxx,(:) yyyy,(:) oooo <CR>

CLM — CLEAR MEMORY: Löschen eines bestimmten Abschnittes im im RAM-Anwenderspeicher.
Als Eingabe wird die Anfangs- und Endadresse des zu löschenden Abschnittes verlangt. Trennung durch Komma.
CLM(:) xxxx, (:) yyyy <CR>

LST — LISTING: Ausgabe des Anwenderprogrammes in Form eines Flussdiagrammes. Als Eingabe wird die Anfangs- und die Endadresse des auszugebenden Programmes verlangt. Trennung durch Komma.
LST(:) xxxx, (:) yyyy <CR>

LWC — LISTING with COMMENT: Nach dem Ausdruck jeder Programmzeile kann ein beliebig langer Kommentar eingegeben werden. Abschluss mit <CR>.
LWC(:) xxxx, (:) yyyy <CR> (;) freier Kommentar <CR>
Beispiel:
0726 03 ANH 0021 (;) Abfrage des Endschalters S21 <CR>
! ! ! !
ausgedruckte Programmzeile — freier Kommentar

DME — DISPLAY MEMORY: Erstellen eines Programmlistings wie LST, jedoch unter Verzicht einer Flussdiagrammdarstellung.
Als Eingabe wird die Anfangs- und Endadresse des auszudruckenden Programmes verlangt. Trennung durch Komma.
DME(:) xxxx, (:) yyyy <CR>

DWC — DISPLAY WITH COMMENT: Ausdruck der reinen Befehlsliste (DME) mit beliebiger Kommentarmöglichkeit (siehe LWC).

DIO — DISPLAY INPUTS and OUTPUTS: Ausdruck der adressierbaren Elemente (E, A, Merker, Timer, Zähler) unter Angabe ihres aktuellen logischen Zustandes.

COMMUNICATION-INTERFACE

Als Eingabe werden Anfangs- und Endadresse des auszugebenden Elementbereiches verlangt (0...999).
DIO(:) eee, (:) eee <CR>

DTX DISPLAY TEXT: (nur PCA23 oder PCA14)
Ausdruck des Textspeicherinhaltes
Nach Eintippen von DTX erscheint: PLEASE SET CPU MODE.
D.h. der Betriebsartenschalter muss in die Stellung TEXT gebracht werden.
Als Eingabe wird die Anfangs- und Endtextnummer des auszugebenden Bereiches verlangt.
DTX(:) ttt, (:) ttt <CR>
Der Ausdruck erfolgt in 10 Charakter pro Zeile.

DRG DISPLAY REGISTERS: (nur PCA23 oder PCA14)
Ausdruck des Wortregisterinhaltes
Nach Eintippen von DRG erscheint:
DRG PLEASE SET CPU MODE, d.h. der Betriebsartenschalter muss in die Stellung MAN-BCD gebracht werden.
Als Eingabe wird die Anfangs- und Endadresse des gewünschtenten Registerbereiches verlangt.
DRG(:) rrr, (:) rrr <CR>
Der Ausdruck erfolgt mit 10 Registerinhalten zu je 2 Digit pro Zeile.

MET MEMORY TOP:
Festlegen der oberen Grenze des Anwenderprogrammes
- Für die Befehle IAD, KAD und XRF kann durch MET die Bearbeitungsgrenze beliebig zwischen 0 und 8k festgelegt werden.
- Für XRF wird bewirkt, dass der Querverweis von 0 bis MET gemacht.
- Man beachte unbedingt, dass MET kleiner oder gleich dem eingesetzten Speichermodul gewählt wird. Bei einem 1 k-Speicher muss also die Bearbeitungsgrenze auf 1023 gesetzt werden.

Eingabe:
MET(:) yyyy <CR>
Ohne Eingabe von MET liegt die Bearbeitungsgrenze automatisch bei 2k.

XRF CROSS REFERENCE:
Ausgabe einer gegliederten Querverweisliste aller Operanden bis zur Schrittadresse 2k bzw. bis zur MET-Begrenzung (0 bis 8k).
Eingabe: XRF

Wichtig: Alle Zwischenräume zwischen den einzelnen Programmteilen sollten gelöscht sein.

Die Organisation der Ausgabe erfolgt in hierarchisch gegliederten Abschnitten:

PAS (PAS)

COMMUNICATION-INTERFACE

```
JMPs        (JIO, JIZ, JMP)
JMS         (JMS, RET)
LOGICAL     (STH, STL, ANH, ANL, ORH, ORL, XOR, DYN, OUT,
             SEO, REO, COO, STR, SCR, DEC, INC, WIH, WIL,
             DTC, WEU, WEL, SNC, CLA, CMP, LAC, SAC)
REGISTER    (RRG, WRG, RRE, WRE, LAR, SAR, INR, DER, SHI)
ARITHMETIC  (CLA, LAI, DBN, BND, ROR, ROA, EXG, SQR, ADD,
             SUB, MUL, DIV)
TEXT        (TXT)
OTHER       (SEI, INI, DEI, DOP, NEG, SEA, TXT, EWP,
             NOP 1248, RSV)
```

Diese Querverweisliste wird man vorwiegend am Ende der Dokumentation durchführen, da ein 1k-Programm für die Durcharbeitung ca. 45 min benötigt.
Unterbruch dieser Arbeit durch *, CTRL/S oder <ANY CHAR>.

Elementadressen von mehrzeiligen Befehlen werden in der Cross Reference nicht aufgeführt. Diese können aber mit SCH gezielt festgestellt werden.

HILFSBEFEHLE

<CR> – bei PRG: entspricht + oder wiederholt die zuletzt gewählte Funktion +, –, EEE.
– bei SST, BRK, MAN: entspricht +
– Beendigung einer numerischen Eingabe:
AAA(:) xxxx <CR>
AAA(:) xxxx, (:) yyyy <CR>
AAA(:) xxxx, (:) yyyy, (:) oooo <CR>

– Beendigung des freien Kommentars.

, Trennung von numerischen Adressangaben.

1/0 Bei MAB und nach EEE: Eingabe des logischen Zustandes eines Elementes.

* Verschiedene Stop-Funktionen

– Die laufende Arbeit wird verlassen
– Ausgabe des Arbeitsstatus
– Die Speicherung eines Uebertragungsfehlers wird gelöscht
Der Betriebsmodus kann neu eingegeben werden.

<ANY CHAR> Betätigung einer beliebigen Taste (mit Ausnahme von * und G) bewirkt Unterbruch der Datenausgabe.

G GO FORWARD: Fortsetzen der unterbrochenen Funktion.

COMMUNICATION-INTERFACE

CTRL/S Sofortige Unterbrechung der Datenausgabe bis CTRL/Q eingetippt wird.

CTRL/Q Freigabe der Datenausgabe nach einem CTRL/S

MELDUNGEN

: Das Programm verlangt Handeingabe von Daten. Erscheint nach der Eingabe abermals ":", so werden weitere Eingaben erwartet.

; Uebergang zu freiem Kommentar, der mit Ausnahme von * keine Programmsteuerfunktion hat.

#SAD#
#RDP# Ein Uebertragungsfehler wurde festgestellt. Der Fehler muss vor weiteren Arbeiten durch Eintippen von * korrigiert werden.

oder
Es wurde ein unübersetzbarer Befehl eingegeben bzw. der eingegebene Befehl ergibt in der laufenden Operation keinen Sinn (z.B. OUT in der Betriebsart MAB).

...# Falsche, sinnwidrige Eingabe(n). Es wird eine korrekte Eingabe erwartet.

```
SAIA PCA ASSEMBLER V1.1                                          MAIN MENU
      ***** SAIA AG   Marktbereich CH   3280 Murten  *****
Directory: C:\PCASS                                   16.12.88 13.31

   Edit                    Text assembler            Compare programs
   Assemble                Disassembler              Xref listing
   Link                    flow chart                File handling
   Up/download             Runtime analysis          Ms-dos command
   Online debug            Hex converter             Setup
   comms Interface         Program eproms            Quit

<ARROW>, <SPACE> or <Tab> selects operation, <CR> or <Command letter> executes
```

```
SAIA PCA TEXT ASSEMBLER MENU

   Text file to assemble . . . . . . (.TEX)    HANDLTEXT_
   Text memory address, <SPACE> to change      4096

Enter file name, <ARROW> or <Tab> moves cursor, <CR> executes, <ESC> aborts,
```

TEXT ASSEMBLER HILFSSEITE

Assembliert eine Text Quell-Datei vom Typ TEX und erzeugt eine OBJ-Datei, welche in die PCA geladen werden kann (mit dem "Up/Download" Untermenü). Es wird auch eine Text-Dokumentations-datei Typ TED erzeugt, welche im PCA-Textformat ausgedruckt werden kann. Vor dem Assemblieren des Textes muss die Textbasisadresse gewählt werden (mit der Leertaste):

PCA0: 2048 PCA14: 4096 PCA2.M22 und M31: 4096 PCA2.M32: 8192

Die Text Quell-Datei wird mit Ihrem Editor geschrieben. Folgende Textinstruktionen können dabei verwendet werden:

C	Kommentar
.	Textzeile
L	Text-Unterprogramm
U	Ende des Text-Unterprogrammes
T	Text
J	Textlinie ohne Zeilenvorschub
M	Textlinie ohne Wagenrücklauf und ohne Zeilenvorschub
@	Ende des Textes
Ax	Absolutes Segment, Text-Adresse "x"

Für Details sehen Sie in Ihrem Handbuch nach .

TEXTASSEMBLER PCTEXT

ALLGEMEINE BESCHREIBUNG

PCTEXT erlaubt Ihnen, eine Textdatei mit Hilfe eines Standard-editors (Personal Editor,Wordstar etc.) unter Beachtung einiger einfacher Regeln zu erstellen.
Dadurch ist es möglich, schon beim Editieren den Text im Ausgabeformat darzustellen.

PCTEXT konvertiert die erstellte Textdatei in eine SAIA-MACRO-ASSEMBLER-kompatible Objektdatei(.OBJ), welche danach mit PCLINK in die SAIA-PC geladen werden kann.
Zusätzlich zur Objektdatei wird eine Textdokumentations-datei(.TED) ähnlich dem SAIA-PC-Texteditor (PCA23 und PCA14) erstellt.

PCTEXT hilft Ihnen beim Erstellen und Modifizieren des PC-Textes auf einfachem Wege.
Sie können :
- Zeichen einfügen
- Zeichen löschen
- Texte anhängen
- Texte absolut adressieren

ERZEUGTE DATEIEN

PCTEXT erzeugt folgende Dateien vom Typ :

- .TED Textdokumentationsdatei, welche ähnlich dem Format des PCA-Editor ist.
- .OBJ Datei welche mit "PCLINK" in die SAIA-PC geladen werden kann.

ERSTELLEN DER QUELLDATEI

Die Quelldatei (Source) wird mit Hilfe eines Editors oder eines Textverarbeitungsprogrammes als Textdatei erzeugt.
Als Editor kann der zum MS-DOS gehörende Zeileneditor oder ein handelsübliches Textverarbeitungsprogramm verwendet werden, das für die Editierung geeignet ist.
Die dabei geltenden Regeln werden in den nächsten Kapiteln aus-führlich behandelt.

Beim Erstellen der Quelldatei dürfen keine Tabulatoren verwendet werden.

Achtung: Mit dem Personal Editor muss beim Abspeichern der Quell-datei der Parameter <notabs> angegeben werden.

TEXTASSEMBLER

FORMAT DER QUELLDATEI

Beim Erstellen der Quelldatei müssen die folgenden Regeln beachtet werden:

- Jede Zeile ist mit <CR> abzuschliessen.
- Spalte 1 darf nur folgende Steuerzeichen enthalten :

  . C M J L U T @ A

  Alle Steuerzeichen können klein oder gross geschrieben werden.

- Spalte 2 bis Zeilenende enthält Text, Kommentar oder eine Adresse.
  Die Länge der Zeile wird durch den benützten Editor oder das Ausgabegerät bestimmt.

<CR> = Carriage Return + Line Feed

Beispiele :

```
.-->Spalte für Steuerzeichen
                <--Text-->

c  Kommentar in der Quelldatei

T            Anfang eines Textes
.               Textzeile
```

DARSTELLUNG DES ASCII-ZEICHENSATZES

Aus Kompatibilitätsgründen mit dem SAIA-PC-Editor muss der ASCII-Zeichensatz gemäss nachstehender Tabelle eingegeben werden.

	ASCII-Nr.	Editor-Eingabe	PCA-Editor-Ausgabe
*	0 bis 31	^^@ bis ^^-	^@ bis ^-
	32 " 126	sp bis .	.sp bis .
	128 " 159	^:@ bis ^:-	:@ bis :-
	160 " 254	^!sp bis ^!	!sp bis !

sp = Leertaste (space)

* ^sp/^sp L

TEXTASSEMBLER

TEXTASSEMBLER-BEFEHLSSATZ

C - Kommentar C , c

Kennzeichnet eine nicht zu verarbeitende Textzeile in der Quelldatei.

```
Source :   C
           c   Kommentarzeile in der Quelldatei
           C
```

. - Textzeile

Kennzeichnet eine zu verarbeitende Textzeile in der Quelldatei.

```
Source :   .          **********
           .          * PCA 14 *
           .          **********
```

L - Textsubroutine L , 1

Kennzeichnet die Adresse einer Textsubroutine in der Textdokumentation (.TED) mit '=>'.

```
Source :   L          **********
           .          * PCA 14 *
           .          **********

(.TED) :          =>  0  .$. .0.1.0.$.*.0.1.0
                      1  ^M^J.$. .0.1.0.*. .P
                      2  .C.A. .1.4. .*^M^J.$
                      3  . .0.1.0.$.*.0.1.0^M
                      4  ^J
```

TEXTASSEMBLER

U - Ende der Textsubroutine U , u

Schliesst die Textsubroutine mit '$U' ab und füllt den Rest der Zeile mit '^@' auf.Befindet sich das Steuerzeichen 'U' auf der letzten Zeile der Textsubroutine, werden Carriage Return/Line Feed dieser Zeile unterdrückt.

```
Source :    L          **********
            .          * PCA 14 *
            .          **********
            U
            L          **********
            .          * PCA 14 *
            U          **********

(.TED) :            =>  0   .$. .0.1.0.$.*.0.1.0
                        1   ^M^J.$. .0.1.0.*. .P
                        2   .C.A. .1.4. .*^M^J.$
                        3   . .0.1.0.$.*.0.1.0^M
                        4   ^J.$.U^@^@^@^@^@^@^@

                    => 10   .$. .0.1.0.$.*.0.1.0
                       11   ^M^J.$. .0.1.0.*. .P
                       12   .C.A. .1.4. .*^M^J.$
                       13   . .0.1.0.$.*.0.1.0.$
                       14   .U^@^@^@^@^@^@^@^@^@
```

T - Text T , t

Kennzeichnet die Startadresse eines Textes in der Textdokumentation (.TED) mit '->'.

```
Source :    TSODECO-SAIA        LANDIS & GYR
            .

(.TED) :            -> 20   .S.O.D.E.C.O.-.S.A.I
                       21   .A.$. .0.0.8.L.A.N.D
                       22   .I.S. .&. .G.Y.R^M^J
                       23   ^M^J
```

J - Textzeile ohne Line Feed J , j

Unterdrückt Line Feed (^J) am Ende einer Textzeile in der Quelldatei.

```
Source :    .   Die kompakte PCA14 fuer
            J   komplexe Aufgaben!

(.TED)              30   . . . .D.i.e. .k.o.m
                    31   .p.a.k.t.e. .P.C.A.1
                    32   .4. .f.u.e.r^M^J. .
                    33   . .k.o.m.p.l.e.x.e.
                    34   .A.u.f.g.a.b.e.n.!^M
                    35   ^@^@^@^@^@^@^@^@^@^@
```

TEXTASSEMBLER

M - Textzeile ohne Carriage Return/Line Feed M , m

Unterdrückt Carriage Return (^M)/Line Feed (^J) am Ende einer Textzeile in der Quelldatei.

```
Source :     .   Mit der PCA14 kann Ihre
           M     Steuerung mehr.

(.TED)          40  . . . .M.i.t. .d.e.r
                41  . .P.C.A.1.4. .k.a.n
                42  .n. .I.h.r.e^M^J. .
                43  . .S.t.e.u.e.r.u.n.g
                44  . .m.e.h.r..^@^@^@^@
```

@ - Ende des Textes @

Füllt den Rest der Zeile mit '^@' auf.
Befindet sich das Steuerzeichen '@' auf der letzten Textzeile, werden Carriage Return/Line Feed dieser Zeile unterdrückt.

```
Source :     T   Die kompakte PCA14 für
             .   komplexe Aufgaben!
             @

           1)T
             .   Mit der PCA14 kann Ihre
             @   Steuerung mehr.

(.TED)          -> 30  . . . .D.i.e. .k.o.m
                   31  .p.a.k.t.e. .P.C.A.1
                   32  .4. .f.u.e.r^M^J. .
                   33  . .k.o.m.p.l.e.x.e.
                   34  .A.u.f.g.a.b.e.n.!^M
                   35  ^J^@^@^@^@^@^@^@^@^@

                -> 40  . . . .M.i.t. .d.e.r
                   41  . .P.C.A.1.4. .k.a.n
                   42  .n. .I.h.r.e^M^J. .
                   43  . .S.t.e.u.e.r.u.n.g
                   44  . .m.e.h.r..^@^@^@^@
```

1) T<CR> ohne ^M/^J
   T <CR> mit ^M/^J

TEXTASSEMBLER

A - Absolutes Segment A , a

Format : Ax od. ax

'A' bezeichnet den Beginn eines absoluten Segmentes.
'x' legt die Startadresse des absoluten Segmentes fest.
Die Startadressen dürfen nur in aufsteigender Reihenfolge definiert werden.

```
Source :    A100
            T        $H
            .        Summenkontrolle des
            .        Systemprogrammes.
            @

            A110
            T         $H
            .         Summenkontrolle des
            .         Anwenderprogrammes.
            @

(.TED)            ->100  .$. .0.0.7.$.H^M^J.$
                    101  . .0.0.7.S.u.m.m.e.n
                    102  .k.o.n.t.r.o.l.l.e.
                    103  .d.e.s. ^M^J.$. .0.0
                    104  .7.S.y.s.t.e.m.p.r.o
                    105  .g.r.a.m.m.e.s..^M^J
                    106  ^@^@^@^@^@^@^@^@^@^@

                ->110  .$. .0.0.8.$.H^M^J.$
                    111  . .0.0.8.S.u.m.m.e.n
                    112  .k.o.n.t.r.o.l.l.e.
                    113  .d.e.s^M^J.$. .0.0.8
                    114  .A.n.w.e.n.d.e.r.p.r
                    115  .o.g.r.a.m.m.e.s..^M
                    116  ^J^@^@^@^@^@^@^@^@^@
```

TEXTASSEMBLER

ANWENDUNGSBEISPIEL

Source

```
C
C       Text für Beispiel 3b im PCA14 Handbuch
C       ------------------------------------
C
C       Textsubroutine :
A1
1)L
USAIA-
C
C
C       Text :
C
A10
T
.
.
.                    $H
.
.                    Die $L001 PC Reihe PCA14 ist das neueste
.
.                    $L001 Produkt innerhalb der $L001 PC Familie.
.
.                    $L001 PC gibt es in $c269 Laendern!
.
.
.                                                  SAIA
.
.                                                LANDIS & GYR
@
```

Die Sprungadresse der Textsubroutine ($L001) wird durch die absolute Adresse (A1) festgelegt.

1) L<CR>

TEXTASSEMBLER

Textdokumentation (.TED)

B:fname.TED

```
=> 1   .S.A.I.A.-.$.U^@^@^@
-> 10  . ^M^J. ^M^J.$. .0.1
   11  .6.$.H^M^J. ^M^J.$.
   12  .0.1.6.D.i.e. .$.L.0
   13  .0.1. .P.C. .R.e.i.h
   14  .e. .P.C.A.1.4. .i.s
   15  .t. .d.a.s. .n.e.u.e
   16  .s.t.e^M^J. ^M^J.$.
   17  .0.1.6.$.L.0.0.1. .P
   18  .r.o.d.u.k.t. .i.n.n
   19  .e.r.h.a.l.b. .d.e.r
   20  . .$.L.0.0.1. .P.C.
   21  .F.a.m.i.l.i.e..^M^J
   22  . ^M^J.$. .0.1.6.$.L
   23  .0.0.1. .P.C. .g.i.b
   24  .t. .e.s. .i.n. .$.c
   25  .10.6.9. .L.a.e.n.d.e
   26  .r.n.!^M^J. ^M^J. ^M
   27  ^J. ^M^J.$. .0.5.10.S
   28  .A.I.A^M^J. ^M^J.$.
   29  .0.4.8.L.A.N.D.I.S.
   30  .&. .G.Y.R^M^J^@^@^@
```

TEXTASSEMBLER

ZUSAMMENFASSUNG DES PCTEXT-BEFEHLSSATZES

C , c	Kommentar in der Quelldatei
.	Textzeile in der Quelldatei
L , l	Textsubroutine
U , u	Ende der Textsubroutine
T , t	Text
J , j	Textzeile ohne Line Feed
M , m	Textzeile ohne Carriage Return / Line Feed
@	Ende des Textes
Ax , ax	Absolutes Segment mit Adresse 'x'

```
SAIA PCA ASSEMBLER V1.1                                                    MAIN MENU

          ***** SAIA AG   Marktbereich CH   3280 Murten  *****

Directory: C:\PCASS                                          16.12.88 13.32

    Edit                    Text assembler          Compare programs

    Assemble                Disassembler            Xref listing

    Link                    flow chart              File handling

    Up/download             Runtime analysis        Ms-dos command

    Online debug            Hex converter           Setup

    comms Interface         Program eproms          Quit

<ARROW>, <SPACE> or <Tab> selects operation, <CR> or <Command letter> executes
```

```
SAIA PCA DISASSEMBLER MENU

    Object file to disassemble . . . (.OBJ)   HANDLING__

    Output file name . . . . . . . . (.DIS)   ________

Enter file name, <ARROW> or <Tab> moves cursor, <CR> executes, <ESC> aborts.
```

DISASSEMBLER HILFSSEITE

Übersetzt eine OBJ-Datei in eine Quelldatei vom Typ DIS. Es werden Symbole, Lables und Mnemonics erzeugt aber keinen Kommentar. Die DIS-Datei kann editiert, erneut kommentiert, assembliert, gelinkt und wieder zurück in die PCA geladen werden. Soll ein Programm aus der PCA disassembliert werden, muss es zuerst mit dem "Up/Download" geladen werden. Der "Disassembler" (aufgerufen vom Hauptmenü), kann nur die ganze OBJ-Datei bearbeiten. Die Bearbeitung von Datei-Teilen kann mit dem Untermenü "Ms-dos command" erfolgen, indem man die PCDIS Batch-Datei direkt anwählt (Details siehe Handbuch). Wird eine grosse OBJ-Datei disassembliert, so muss der IBM-PC genügend freien Speicher haben (ca. 5kB für 100 Programmzeilen).

Um eine disassemblierte Datei neu zu assemblieren, muss die Dateierweiterung vom Typ DIS in den Typ SRC umbenannt werden (im Untermenü "File Handling" oder "Ms-dos command").

Um den disassemblierten Code ausdrucken zu können, benützen Sie als "Output file name" PRN oder LPT1/2/3.

DISASSEMBLER

FEHLERMELDUNGEN

FATAL ERRORS führen zum Abbruch des Programmes und Rückkehr ins Betriebssystem.
ERRORS führen nicht zum Programmabbruch. Eine neue Eingabe ist möglich.
WARNINGS sind Mitteilungen an den Benutzer während des Programmablaufes ohne Programmabbruch.

FATAL ERRORS

Cannot Open File

Fehler beim Oeffnen der Datei, z.B. die Datei existiert nicht.

Cannot Write to Disk

Fehler beim Schreiben auf Diskette, z.B. wenn die Diskette voll ist.

Checksum Error

Fehlerhafte Datei. Dieser Fehler tritt bei korrumpierten Dateien auf.

Error Closing File

Eine Datei kann nicht ordnungsgemäss geschlossen werden.

Illegal File Name

Unzulässiger Dateiname, z.B. Angabe des Datentypes.

Inconsistant Objectfile

Die Eingabedatei hat ein falsches Format.

Too Many Counter Values
Too Many Labels

Die programminterne Speicherkapazität ist erschöpft.

ERRORS

Illegal Start- or Endaddress

Der Eingabewert liegt ausserhalb des zulässigen Bereiches.

DISASSEMBLER

WARNINGS

Undefined Destination Address

Eine Jump- oder PAS-Anweisung enthält einen Operanden, der sich ausserhalb des disassemblierten Bereiches befindet. Die betreffenden Operanden werden im Quellcode mit dem Buchstaben "U" (Undefined) versehen.

FORMAT DES QUELLCODES

Der Disassembler erstellt einen zum Assembler voll kompatiblen Quellcode mit folgenden Besonderheiten:

- Zieladressen von Sprung-Anweisungen und von PAS-Anweisungen (PAS 1 bis 15, PAS 190) werden durch symbolische Adressen (Labels) ersetzt. Diese symbolischen Adressen haben die Form L0, L1, L2 usw. Es können maximal 1000 Labels verarbeitet werden. Die Labels werden im Feld 1 (siehe Assembler) in aufsteigender Reihenfolge numeriert.

- Element- und Registeradressen werden durch Symbole der Form EAxxx bzw. Rxxx ersetzt, wobei xxx eine Zahl im Bereich 0..999 darstellt.

- Indexierte Element- oder Registeradressen werden in der Form EAxxx+1000 (bzw. Rxxx+1000) dargestellt.

- Timer/Counter-Werte werden in der Form Vxxxxx dargestellt, wobei xxxxx eine Zahl im Bereich 0..32767 ist. Sind diese Werte grösser als 32767 (PCA14 oder PCA23), erscheint eine negative Zahl. Solche Werte können vom Assembler nicht verarbeitet werden und müssen daher nach dem Laden direkt in die SAIA-PC eingegeben werden.

- Es werden Absolutsegmente erzeugt. Ist die Adressierung infolge Verwendung der Option C beim Programm PCSAVE nicht fortlaufend, so wird bei jedem Adresssprung mittels der Anweisung ORG eine neue Startadresse definiert.

- Die erzeugte Datei ist vom Typ DIS. Zum Assemblieren muss sie in eine Datei vom Typ SRC umbenannt werden.

Zuweisungslisten (Equate lists)

Der Disassembler erzeugt Zuweisungslisten mit der Assembler-Anweisung EQU (siehe SAIA-MACRO-ASSEMBLER). Diese Listen stehen am Anfang des disassemblierten Programmes.
Sie werden durch Ueberschriften unterteilt in:

- Inputs/Outputs
- Timers/Counters
- Timer/Counter Values
- Flags (Merker)
- Registers

DISASSEMBLER

BEISPIEL

Untenstehendes Beispiel zeigt ein Programm, dass mit Hilfe von PCDIS aus dem Objektcode erzeugt wurde.

```
;
; Inputs/Outputs
; --------------
EA0      EQU      0
EA1      EQU      1
EA2      EQU      2
EA32     EQU      32
EA48     EQU      48
;
;
; Timers/Counters
; ---------------
EA260    EQU      260
;
;
; Timer/Counter Values
; --------------------
V25      EQU      25
;
;
; Flags
; -----
EA500    EQU      500
EA501    EQU      501
;
;
; Registers
; ---------
;
;
;
         ASEG
;
         ORG      100
;
L0       STH      EA2
         SCR      EA260
         TCV      V25
         STH      EA0
         DYN      EA500
         DEC      EA260
         STH      EA1
         DYN      EA501
         INC      EA260
         STH      EA260
         OUT      EA48
         STH      EA260
         OUT      EA32
         SEA      0
         DTC      EA260
         JMP      L0
```

```
SAIA PCA ASSEMBLER V1.1                                                MAIN MENU
        ***** SAIA AG   Marktbereich CH   3280 Murten  *****
Directory: C:\PCASS                                          16.12.88 13.33

    Edit                    Text assembler          Compare programs
    Assemble                Disassembler            Xref listing
    Link                    Flow chart              File handling
    Up/download             Runtime analysis        Ms-dos command
    Online debug            Hex converter           Setup
    comms Interface         Program eproms          Quit

<ARROW>, <SPACE> or <Tab> selects operation, <CR> or <Command letter> executes
```

```
SAIA PCA FLOW CHART GENERATOR MENU

    Documentation file to process  . . (.PGD)  HANDLING__
    Flow chart file name . . . . . . . (.FLW)  ________

Enter file name, <ARROW> or <Tab> moves cursor, <CR> executes, <ESC> aborts.
```

FLOW CHART HILFSSEITE

Setzt eine vom Assembler oder Linker erzeugte PGD-Datei in ein Flussdiagramm um (Dateityp FLW).
Das Flussdiagramm ist eine strukturierte Darstellung des Programmes mit den spezifischen grafischen Symbolen. Die Seitentitel, Untertitel und der Programmkommentar der PGD-Datei sind auch in der Flussdiagrammdarstellung enthalten.

Die Flussdiagrammdarstellung kann sowohl von einer durch das Untermenü "Assemble" erzeugten PGD-Datei als auch aus einer globalen, durch "Link" erzeugten PGD-Datei erstellt werden.

Das Seitenformat wird im Menü "Setup" definiert.

Um das Flussdiagramm auf den Drucker auszugeben, benützen Sie als "Flow chart file name" die Befehle PRN oder LPT1/2/3.

PCFLOW

FEHLERMELDUNGEN

Alle Fehler sind "FATAL ERRORS" und führen zum Abbruch des Programmes PCFLOW mit Rückkehr ins Betriebssystem MS-DOS.

Cannot Open File filename

Datei filename kann nicht eröffnet werden, da nicht vorhanden.

Device I/O Error

Systembedingter Ein- oder Ausgabefehler.

Device unavailable

Die entsprechende Ein- oder Ausgabeeinheit steht nicht zur Verfügung.

Disk Error

Allgemeine Fehlermeldung der Massenspeicher.

Disk is full

Die Diskette oder die Festplatte ist voll. Neue Diskette benutzen und Programm noch einmal laufen lassen.

Disk Media Error

Fehlerhafte Diskette oder Hardwarefehler.

Disk not ready

Diskettenverriegelung offen oder keine Diskette im Laufwerk.

Disk write protected

Die Diskette ist mit einem Kleber gegen Schreiben und Löschen geschützt.

Illegal File Name

Ein ungültiger Dateinamen wurde benutzt.

Left margin too high

Der Wert für die linke Randeinstellung liegt ausserhalb des zulässigen Bereiches.

Too many Files

Im Inhaltsverzeichnis der Diskette sind zu viele Dateinamen eingetragen. Neue Diskette verwenden.

```
SAIA PCA ASSEMBLER V1.1                                                    MAIN MENU

                  ***** SAIA AG   Marktbereich CH   3280 Murten  *****

Directory: C:\PCASS                                          16.12.88 13.38


    Edit                      Text assembler            Compare programs

    Assemble                  Disassembler              Xref listing

    Link                      flow chart                File handling

    Up/download               Runtime analysis          Ms-dos command

    Online debug              Hex converter             Setup

    comms Interface           Program eproms            Quit


<ARROW>, <SPACE> or <Tab> selects operation, <CR> or <Command letter> executes
```

```
SAIA PCA CROSS REFERENCE LIST MENU

    Object file to process . . . . . . (.OBJ)  HANDLING__

    Cross reference list file name . . (.CRF)  _________

    Start address . . . . . . . . . . . . .    0____

    End address . . . . . . . . . . . . . .    8191_

                      Instruction selectors:

    PAS instructions      Yes     Text instructions                Yes

    Jump instructions     Yes     Word processor instructions      Yes

    JMS instructions      Yes     Other instructions               Yes

    Logic instructions    Yes

Enter file name, <ARROW> or <Tab> moves cursor, <CR> executes, <ESC> aborts,
```

XREF LISTING HILFSSEITE

Liest eine OBJ-Datei und erzeugt eine Cross Referenzliste vom Typ CRF. Ist der Dateiname für die Cross Referenz Liste nicht definiert, so wird der Name der OBJ-Datei übernommen, jedoch mit Erweiterung CRF.

Die CRF-Datei enthält Listen über alle Operanden-Typen, mit der Angabe wo und im Zusammenhang mit welchen Befehlen sie gebraucht sind. Es werden die im Menü mit YES/NO definierten Anweisungen ausgeführt.

Mit der Definition von Start- und Endadresse kann über einen bestimmten Programmbereich eine Cross Referenzliste erzeugt werden. Wenn die Endadresse nicht bekannt ist, setzen Sie die Adresse auf das Maximum (z.B. 8191). Die effektive Endadresse wird dann vom Programm automatisch erkannt.

Das Seitenformat wird über das Menü 'Setup' definiert.

Um die Cross Referenzliste auf den Drucker auszugeben, benützen Sie als "Cross reference list filename" die Befehle PRN oder LPT1/2/3.

PCXRF

Beispiel:

Inputfile(d:filename[.OBJ] : b:handling/w/a(200,850)/p l(64)

Outputfile (d:filename[.CRF]) : <CR>

PCXRF erzeugt aus dem Objektcode des Programmes handling von Adresse 200 bis 850 eine Querverweisliste ohne Wortprozessorbefehle. Die Ausgabedatei handling.xrf hat eine Seitenlänge von 64 Zeilen/Seite.

Gliederung der Ausgabedatei

Die Organisation der Ausgabe erfolgt in hierarchisch gegliederten Abschnitten:

PAS-Befehle	PAS
JMP-Befehle	JIO, JIZ, JMP
Subroutinen	JMS, RET
Elemente	STH, STL, ANH, ANL, ORH, ORL, XOR, DYN, OUT, SEO, REO, COO, STR, SCR, DEC, INC, WIH, WIL DTC, LAC, SAC, WEL, WEU, SNC, SEW, CMP, CLK
TXT und PAS 23	
Wortregister	INR, DER, RRG, WRG, RRE, WRE, LAR, SAR, EXG, SHI
Arithmetik im Wortprozessor	SQR, ADD, SUB, MUL, DIV
Andere	NOP, NEG, SEI, SEA, INI, DEI, DOP, SWP, EWP SNC, SEW, CLA, LAI, DBN, BND, ROR, ROA, EXG

Folgende Besonderheiten sind zu beachten:

Ist der Operationscode in der zweiten Zeile eines SCR- oder STR-Befehles grösser als 15, so wird er unter "ELEMENTS" aufgeführt. Der Operationscode (numerisch) wird dabei mit ":" gekennzeichnet.

Folgen sich mehr als eine Anweisung NOP 0, so wird nur die erste in der Querverweisliste aufgeführt.

PCXRF

FEHLERMELDUNGEN

Alle Fehler sind "FATAL ERRORS" und führen zum Abbruch des Programmes PCXRF mit Rückkehr ins Betriebssystem MS-DOS.

Cannot Create File filename

Eine Zieldatei kann nicht eröffnet werden. Ursache systembedingt.

Cannot Open File filename

Datei filename kann nicht eröffnet werden, da nicht vorhanden.

Cannot Read from File filename

Fehler beim Lesen der Datei. Ursache systembedingt.

Cannot Write to Disk

Fehler beim Schreiben auf Diskette. Ursache systembedingt.

Checksum Error filename

Fehlerhafte Datei. Ursache systembedingt.

End of File filename not found

Ende der Datei filename nicht gefunden, weil fehlerhafter Abschluss der Datei (kein EOF).

Error Closing File filename

Datei kann nicht ordnungsgemäss geschlossen werden. Ursache systembedingt.

Heap overflow

Kein verfügbarer Speicher mehr. Die Eingabedatei muss verkleinert werden.

Illegal Start- or Endaddress

Die Anfangs- oder Endadresse liegen nicht innerhalb des geladenen Programmes.

Inconsistent File filename

Falsches Format der Objektdatei. Keine SAIA-PC-Datei.

Incorrect Input Parameter

Unzulässige Eingabe bei der Option /x.

```
SAIA PCA ASSEMBLER V1.1                                                    MAIN MENU

            ***** SAIA AG   Marktbereich CH   3280 Murten  *****

Directory: C:\PCASS                                          16.12.88 13.38


    Edit                    Text assembler          Compare programs

    Assemble                Disassembler            Xref listing

    Link                    flow chart              File handling

    Up/download             Runtime analysis        Ms-dos command

    Online debug            Hex converter           Setup

    comms Interface         Program eproms          Quit


<ARROW>, <SPACE> or <Tab> selects operation, <CR> or <Command letter> executes
```

```
SAIA PCA FILE HANDLING UTILITIES V1.3

Directory: C:\PCASS                                  Files: 105   Bytes free: 8542208

.            <DIR>    2.06.88 15.06     HANDLING SRC   3155   3.11.88  1.00
..           <DIR>    2.06.88 15.06     HAUPTPCA SCR   8128  16.12.88 13.20
CI       EXE  10363  13.10.88  1.00     KOPF         <DIR>    2.09.88 10.31
COUNTER  LNK    128  16.12.88 13.28     LIESMICH      32271  16.12.88 11.42
COUNTER  SRC   1403   3.11.88  1.00     MENU     SCR   8128  16.12.88 13.20
ERASE    OBJ  71009   9.10.88  1.00     METPCA   TXT  21001  16.12.88 11.48
HANDLING FLW   6240  13.12.88 14.03     MVERT    SRC   1487   3.11.88  1.00
HANDLING LNK    384  16.12.88 13.25     M_AS     SCR   8128  16.12.88 13.22
HANDLING OBJ    384  16.12.88 13.29     M_AS1    SCR   8128  16.12.88 13.23
HANDLING PGD   3072  16.12.88 13.25     M_AS2    SCR   8128  16.12.88 13.24
HANDLING RTA    701  16.12.88 13.36     M_AS3    SCR   8128  16.12.88 13.24

    Copy file(s)            Display file            Format diskette

    Erase file              Print file

    Rename file             coMpare files           Quit

<ARROW>, <SPACE> or <Tab> selects, <CR> or <Command letter> executes.
View directory with <PgUp> and <PgDn>.
```

```
SAIA PCA ASSEMBLER - VERSION 19-07-88                                MAIN MENU
                 SAIA AG MURTEN ***** For demonstration only *****
Directory: C:\405\WORKASS                                   15.08.88 14.27

   Edit                        Text assembler              Compare programs

   Assemble                    Disassembler                Xref listing

   Link                        flow chart                  File handling

   Up/download                 Runtime analysis            Ms-dos command

   Online debug                Hex converter               Setup

   comms Interface             Programmer P16              Quit


Exit to MS-DOS (Y or N) ?
```

```
SAIA PCA ASSEMBLER - VERSION 19-07-88                                MAIN MENU
```

QUIT HILFSSEITE

Mit der Anwahl des Feldes 'Quit' verlassen Sie das PCA-ASSEMBLER Menü und kehren auf die DOS-Befehlsebene zurück. (Diese Anwahl muss mit ja 'Y' oder nein 'N' Bestätigt werden)
Alle Daten, die Sie in die Menüfelder eingegeben haben, werden in der Datei PCAMENU.DAT abgespeichert.
Beim nächsten Aufstarten des PCA-ASSEMBLERS wird die Datei PCAMENU.DAT gelesen und alle letzten Eintragungen in diese Menüfelder sind wieder present bzw. aktiv. Kann keine PCAMENU.DAT Datei gefunden werden oder wird ein Checksummenfehler festgestellt, so sind die Menüfelder leer bzw. es wird die Grundeinstellung ausgegeben.

Es ist ratsam, die Dateien für jedes Projekt in einem separaten Verzeichnis (oder Diskette) abzuspeichern. Damit wird auch die entsprechende PCAMENU.DAT Datei im zugehörigen Verzeichnis abgelegt und steht jederzeit zur Verfügung.

Details zur Dateiorganisation können dem 'README' entnommen werden.

Press any key to return to the menu