

Für den

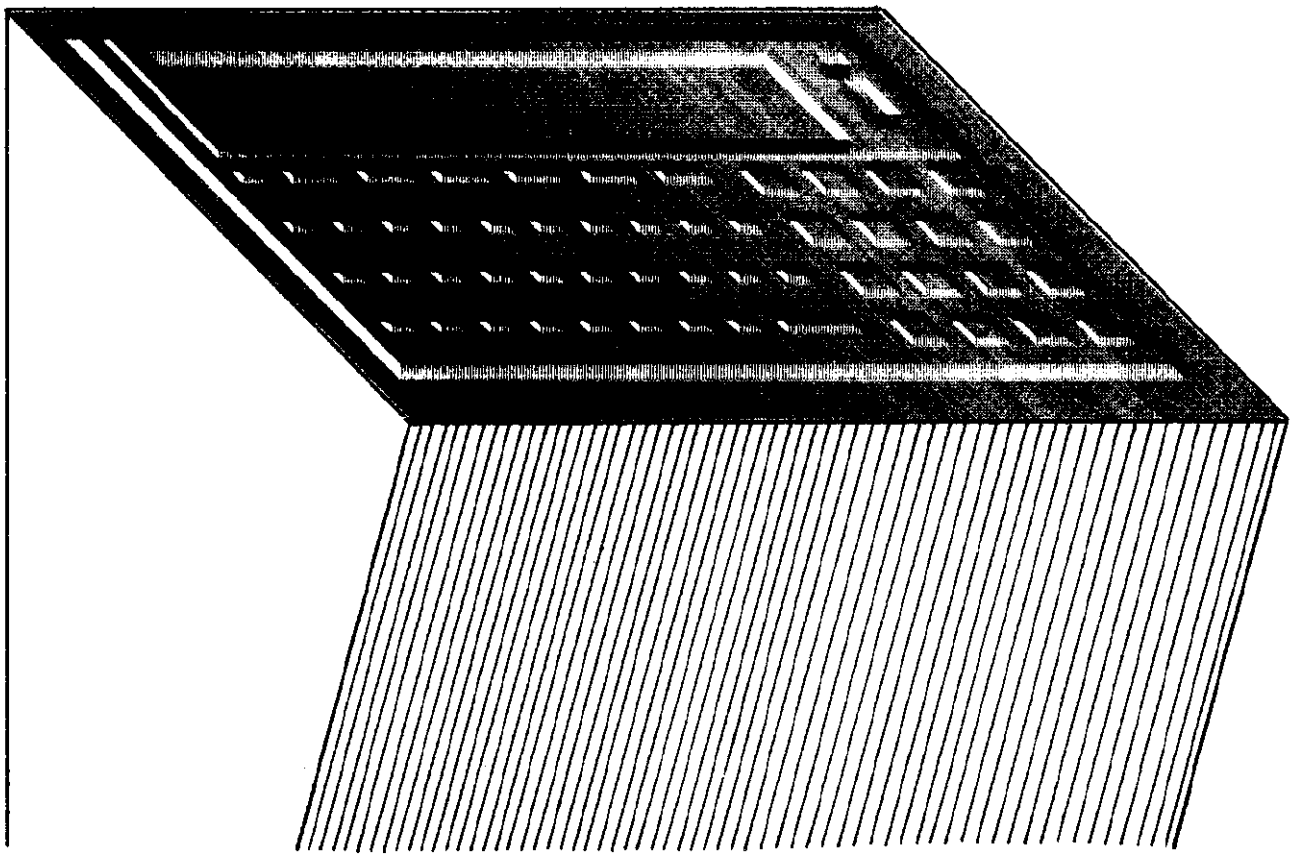
SHARP

PC-1260/61

Taschencomputer

Maschinensprache -

Handbuch



FISCHEL GmbH
ISBN 3-924327-29-7

Dr. J. STANGE

49. - 11 AB

MASCHINENSPRACHE-
 HANDBUCH

ZUM

SHARP PC-1260/61
 TASCHEPCOMPUTER

ISBN 3-924 327-29-7
 =====

AUTOR : Dr. J. STANGE

HERAUSGEBER : FISCHEL G.m.b.H.

CARTOONS : PETER LAWATSCH

DRUCK : GERHARD WEINERT

=====

© 1986 FISCHEL GmbH, Berlin

Alle Rechte vorbehalten. Ohne ausdrueckliche Genehmigung des Herausgebers ist es nicht gestattet, das Buch oder Teile daraus auf fotomechanischem (Fotokopie, Mikrokopie) oder sonstigem Wege zu vervielfaeltigen .
 Fuer etwaige Schaeden durch Anwendung der Anleitungen oder Programme dieses Buches uebernehmen wir keine Haftung .

=====

DURCH INFORMATION VORN !

F I S C H E L G.M.B.H.
 KAISER-FRIEDRICH-STR. 54A
 D-1000 B E R L I N 12

INHALTSVERZEICHNIS

E	Einleitung	3
I	Zahlensysteme, BCD-Code	5
II	Der Mikroprozessor SC 61860	7
III	Der Befehlssatz der ESR-H CPU	12
III.1	Transport-Befehle	17
III.2	Arithmetische Befehle	23
III.3	Logische Befehle	28
III.4	Shift-Befehle	30
III.5	CPU Steuerbefehle	31
III.6	Sprungbefehle	32
III.7	Unterprogramm-Befehle	34
III.8	Eingabe-/Ausgabe-Befehle	35
III.9	Unbekannte Befehle	38
III.10	Kurzbeschreibung der Befehle	41
IV	Die Basic-Befehle PEEK, POKE, CALL, CSAVEM, CLOADM	46
V	Basic-Maschinenprogramme	48
V.1	Disassembler-Anzeige/Ausdruck	50
V.2	Hexcode/Zeichen-Anzeige/Ausdruck	54
VI	Programmierbeispiele	55
VI.1	Renumber	56
VI.2	Breakpoint-Monitor	60
VII	Der Basic-Interpreter	65
VII.1	Untersuchung des System	66
VII.2	Die Tastenroutinen	68
VII.3	Die Anzeige- und Ausdruck-Routinen	76
VII.4	Systemadressen, Token-Tabelle	79
A	Anhang: Das ROM. Befehlslisting des Kern-ROM &0000 bis &1FFF	85
S	Stichwortverzeichnis	114

EINLEITUNG

Neben der Programmierung in Basic bietet der PC-1260/61 Pocket Computer auch die Möglichkeit der Programmierung in Maschinensprache. Dafür enthält der Basic-Interpreter die Befehle PEEK, POKE, CALL, CSAVEM und CLOADM mit denen Maschinenprogramme im Hexcode eingegeben, aufgerufen und auf Cassette gespeichert werden können.

Die Benutzung der Maschinensprache gibt auf der einen Seite die Möglichkeit, Programme oder Programmteile zu entwickeln, die bis zu tausend mal schneller ablaufen als Basicprogramme. Auf der anderen Seite gibt sie die Möglichkeit, sich mit dem Aufbau und der Funktion eines Mikroprozessors auseinanderzusetzen. Zwar unterscheiden sich die Befehle verschiedener Mikroprozessortypen im einzelnen. Der prinzipielle Aufbau eines Befehlssatzes folgt jedoch stets dem gleichen Schema.

Die ESR-H CPU (Central Processor Unit) des PC-1260/61 ist ein 8 Bit CMOS Mikroprozessor und trägt die Bezeichnung SC 61860. Der Befehlssatz enthält mehr als hundert Maschinenbefehle, die hier mit der von der Firma Sharp gegebenen Bezeichnung (Mnemonic) im einzelnen beschrieben werden. Als Grundlage diene das SHARP POCKET COMPUTER PC-1250/1251(1250A) MACHINE LANGUAGE REFERENCE MANUAL.

Mit der hier gegebenen Information können Maschinenprogramme aufgebaut werden. Stehen keine Programmierhilfsmittel zur Verfügung, so erfolgt die Eingabe der Code am besten über ein Basicprogramm mit dem POKE-Befehl. Für Änderungen stehen dann die Editiermöglichkeiten des Basic-Interpreters zur Verfügung. Im Text sind Beispiele für diese Eingabe mit POKE gegeben. Mit dem im Text beschriebenen Disassembler kann die Eingabe in Maschinensprache gelesen und so überprüft werden.

Da ein neu entwickeltes Maschinenprogramm, genauso wie ein Basicprogramm, im allgemeinen nicht sofort das tut,

was beabsichtigt war, ist eine Testmöglichkeit mit Single Step oder Breakpoint nützlich. Im Text wird ein Breakpoint-Monitor beschrieben, mit dem ein zu testendes Maschinenprogramm unterbrochen und der Inhalt aller Register angesehen werden kann.

Der Basic-Interpreter wurde untersucht und die wichtigsten Systemadressen angegeben. Die Unterprogramme für die Tastenerkennung, die Anzeige und den Ausdruck werden im einzelnen beschrieben.

Bei der Ausarbeitung des Buches wurde ein älterer PC-1260 und ein 1986 gelieferter PC-1261 benutzt. Diese beiden Pocket Computer besitzen verschiedene ROM-Versionen, die im weiteren als alte und neue Version bezeichnet werden. Der Unterschied zwischen beiden Versionen ist jedoch nicht schwerwiegend, so daß alle Basic-Befehle mit dem gleichen Namen angesprochen werden. Auch die Aufrufadressen der hier beschriebenen Unterprogrammen des Basic-Interpreters sind bei beiden Versionen gleich.

Sie stehen mit Ihrem SHARP-Computer doch nicht etwa auf dem Kriegspfad ???	
Lesen Sie einfach Alles für SHARP-Computer und dann wissen Sie was Sache ist !!!	
Fischel GmbH Kaiser-Friedrich-Str. 54a 1000 Berlin 12	

I ZAHLENSYSTEME, BCD-CODE

Im täglichen Leben rechnen wir mit dem Dezimalsystem. Dies ist jedoch nicht geeignet für die Durchführung von Rechenoperationen im Computer. Der Computer ist in seinem Kern von recht beschränkter Intelligenz und kann nur zwischen zwei Zuständen, niedrige Spannung/hohe Spannung, unterscheiden. Dies führt, wie wir später genauer sehen werden, zum Dualsystem. Dieses, auf den beiden Zahlen Null und Eins aufbauende Zahlensystem, wird jedoch nur im Kern des Computers, der CPU, bei den Grundrechenoperationen angewandt. Die Rechenergebnisse selbst können dann leicht per Programm in andere Zahlensysteme umgewandelt und so ausgegeben werden, so daß der Benutzer des Computers die Ergebnisse nicht in Dualzahlen erhält sondern im allgemeinen in Dezimalzahlen. Häufig werden die Ergebnisse jedoch auch in Oktalzahlen oder Hexadezimalzahlen ausgegeben. Diese beiden Zahlensysteme benutzt man, weil sich mit ihnen die "Computerworte" besser darstellen lassen. Z.B. läßt sich das bei Mikrocomputern häufig benutzte 8 Bit Wort am einfachsten und übersichtlichsten durch zwei Hexadezimalziffern darstellen.

Dezimalzahlen

$1234 = 4 \text{ mal } 1 + 3 \text{ mal } 10 + 2 \text{ mal } 100 + 1 \text{ mal } 1000$
oder

$1234 = 4 \text{ mal } 10^0 + 3 \text{ mal } 10^1 + 2 \text{ mal } 10^2 + 1 \text{ mal } 10^3$

Die Dezimalzahlen werden also in Vielfachen von Zehnerpotenzen dargestellt. Das gleiche Prinzip wird auch für die anderen Zahlensysteme angewandt.

Dualzahlen

Dualzahlen werden in Vielfachen von Zweierpotenzen dargestellt.

Dual 10101 = 1 mal 1 + 0 mal 2 + 1 mal 4 + 0 mal 8
+ 1 mal 16

oder

$$\begin{aligned}\text{Dual } 10101 &= 1 \text{ mal } 2^0 + 0 \text{ mal } 2^1 + 1 \text{ mal } 2^2 + 0 \text{ mal } 2^3 \\ &\quad + 1 \text{ mal } 2^4 \\ &= 21(\text{dezimal})\end{aligned}$$

Oktalzahlen

Oktalzahlen werden in Vielfachen von Achterpotenzen dargestellt.

$$\begin{aligned}\text{Oktal } 70342 &= 2 \text{ mal } 1 + 4 \text{ mal } 8 + 3 \text{ mal } 64 + 0 \text{ mal } 512 \\ &\quad + 7 \text{ mal } 4096\end{aligned}$$

oder

$$\begin{aligned}\text{Oktal } 70342 &= 2 \text{ mal } 8^0 + 4 \text{ mal } 8^1 + 3 \text{ mal } 8^2 + 0 \text{ mal } 8^3 \\ &\quad + 7 \text{ mal } 8^4 \\ &= 28898(\text{dezimal})\end{aligned}$$

Hexadezimalzahlen

Hexadezimalzahlen werden in Vielfachen von Sechzehnerpotenzen dargestellt. Für eine Stelle werden hierbei sechzehn Möglichkeiten benötigt. Dazu reichen die Zahlen von Null bis Neun nicht aus. Es wird darum gezählt:

0, 1, 2, 3, 4, 5, 6, 7, 8, 9, A, B, C, D, E, F. Hierbei ist A = 10(dezimal), B = 11(dezimal), C = 12(dezimal), D = 13(dezimal), E = 14(dezimal), F = 15(dezimal).

$$\begin{aligned}\text{Hexadezimal } 7E5B2 &= 2 \text{ mal } 1 + 11 \text{ mal } 16 + 5 \text{ mal } 256 \\ &\quad + 14 \text{ mal } 4096 + 7 \text{ mal } 65536\end{aligned}$$

oder

$$\begin{aligned}\text{Hexadezimal } 7E5B2 &= 2 \text{ mal } 16^0 + 11 \text{ mal } 16^1 + 5 \text{ mal } 16^2 \\ &\quad + 14 \text{ mal } 16^3 + 7 \text{ mal } 16^4 \\ &= 517554(\text{dezimal})\end{aligned}$$

Binary Coded Dezimals, BCD

Häufig werden Dezimalzahlen in Computerworten dargestellt. Es werden dann für eine Dezimalstelle 4 Bit (4 Dualstellen) benötigt. In einem 8 Bit Computerwort lassen sich also zwei Dezimalstellen unterbringen.

Beispiele: 23 = 00100011, 45 = 01000101, 99 = 10011001.

II DER MIKROPROZESSOR SC 61860

Der Mikroprozessor des Pocket Computers ist ein 8-Bit CMOS Prozessor und trägt die Bezeichnung SC 61860. Mit dem 8-Bit Datenbus können 8-Bit Worte verarbeitet werden. Der 16-Bit Adressbus kann auf 64 KByte Speicher zugreifen.

Bild 1 zeigt das Prinzipbild der ESR-H CPU. Neben der ALU (arithmetic logic unit) mit den Flags Carry und Zero gibt es den 16-Bit Programmcounter PC und den 16-Bit Datapointer DP, die beide mit dem 16-Bit Adressbus verbunden sind. Weiterhin gibt es in der CPU ein internes Memory (RAM), das die CPU-internen Register I, J, A, B, XL, XH, YL, YH, K, L,....., den System Stack und die Ports IA, IB, FO und OUTC enthält. Diese Register werden von den 7-Bit Adresspointer-Registern P, Q und R angesprochen. Das Register R fungiert als Stackpointer.

Alle Teile der CPU werden zusätzlich durch den Control-Bus, der hier nicht eingezeichnet ist, verbunden.

Die ALU kann arithmetische und logische Operationen mit zwei 8-Bit Argumenten ausführen. Das Ergebnis wird in ein Register, im allgemeinen den Akkumulator A, gegeben und kann zwei Flags beeinflussen. Das Zero Flag Z wird eins gesetzt, wenn das Ergebnis einer Operation Null war, anderenfalls ist $Z = 0$. Das Carry Flag wird bei einem Übertrag eins gesetzt, anderenfalls ist $C = 0$. Das d-Register ist ein Zähler für Blockoperationen. Es kann nicht direkt beschrieben werden sondern wird am Beginn einer Blockoperation mit dem Wert aus I oder J geladen.

Der 16-Bit Programmcounter PC enthält die Adresse des nächsten auszuführenden Befehls. Nur durch JUMP- oder CALL-Befehle kann der PC abweichend von der normalen Folge geändert werden.

Der 16-Bit Datapointer DP wird benutzt, um das externe Daten-Memory zu adressieren. Beim Lesen oder Beschreiben des externen Daten-Memory wird die Adresse stets

8 BIT DATEN BUS

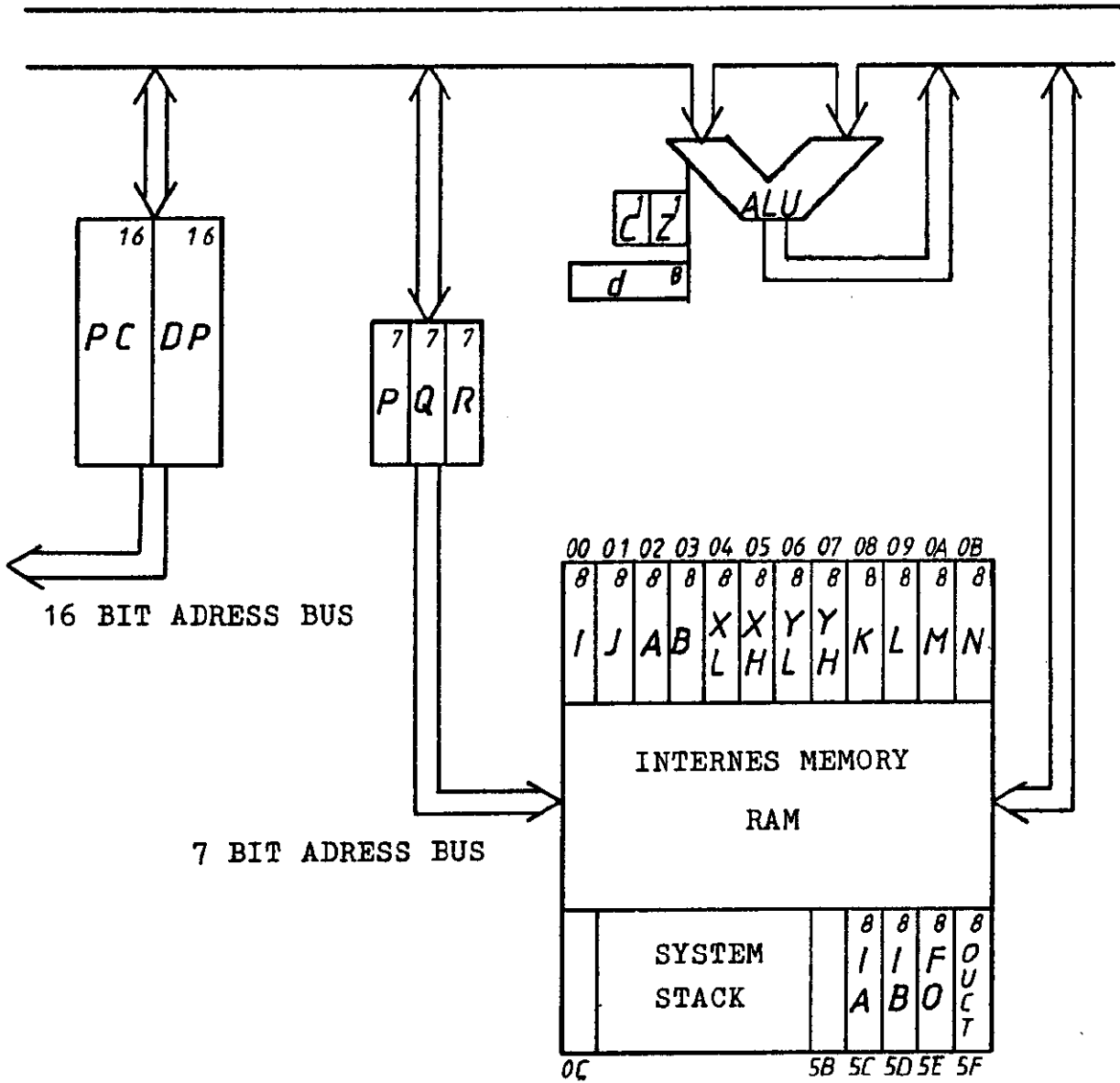


Bild 1: BLOCK DIAGRAMM DER ESR-H CPU

durch den Datapointer DP bestimmt. Er kann mit einer Adresse aus den Registern X oder Y oder durch Immediate-Werte geladen werden und kann inkrementiert oder dekrementiert werden.

Die 7-Bit Register P, Q und R werden benutzt, um die Register im internen Memory (RAM) zu adressieren. P ist das Hauptregister. R wird als Stackpointer auf den im internen Memory liegenden Systemstack benutzt. Diese Register sind 7-Bit Register, da das interne Memory nur 96 (dezimal) Plätze enthält.

Im CPU internen Memory befinden sich bei den Adressen 00 und 01 die Register I und J. Diese werden bei Block-Operationen verwendet. Ihr Inhalt wird vor den Block-Operationen in das d-Register gegeben und dieses dann auf &FF heruntergezählt.

Das J-Register ist bei Maschinenprogrammen mit Vorsicht zu behandeln. Für zahlreiche Routinen des Basic-Interpreters muß es den Wert 1 haben. Da das J-Register nach der Rückkehr von einem Maschinenprogramm-CALL nicht auf 1 gesetzt wird, muß dies vorher im Maschinenprogramm geschehen. Wenn möglich, sollte man das J-Register in Maschinenprogrammen nicht benutzen.

Die Register A und B haben die Adressen 02 und 03 und sind der Akkumulator, das Hauptregister, und der Reserve-Akkumulator. Der Akkumulator wird bei den meisten arithmetischen und logischen Operationen benutzt. Auch an vielen Transport-Operationen ist er beteiligt, z.B. bei LOAD und STORE.

Das X- und Y-register, die die Adressen 04 bis 07 haben, werden als Adresspointer benutzt. Sie sind 16-Bit Register, die je aus zwei Byte bestehen, wobei das low Byte dem high Byte vorangeht. Das X-Register wird im allgemeinen mit dem IXL-Befehl benutzt. Hierbei zeigt X auf die Adresse, deren Inhalt in den Akkumulator geladen wird. Dagegen wird das Y-Register beim IYS-Befehl benutzt.

Hierbei wird der Inhalt des Akkumulator in den Speicher gegeben.

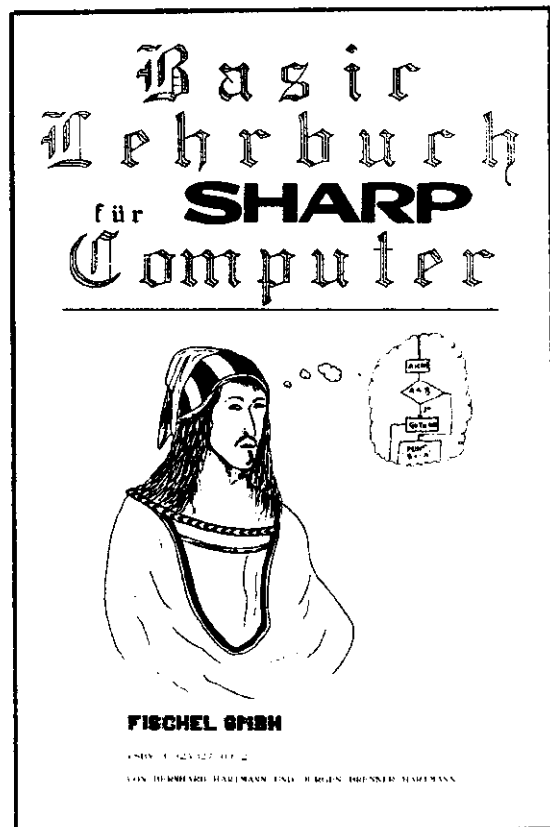
Das K- und L-Register, die die Adressen 08 und 09 haben, sind Ein-Byte Register für allgemeine Zwecke für die es Inkrement- und Dekrement-Befehle gibt.

Weiterhin gibt es im internen Memory (RAM) Input-/Output-Register mit den Adressen &5C, &5D, &5E und &5F.

Der System-Stack befindet sich im internen Memory (RAM). Das R-Register zeigt auf die obere Ebene des Stack. Der Stack beginnt bei &5B und wächst bei PUSH- und CALL-Befehlen abwärts in den Speicher. Beim POP- und RETURN-Befehl werden jeweils die zuletzt in den Stack gegebenen Werte wieder ausgelesen.

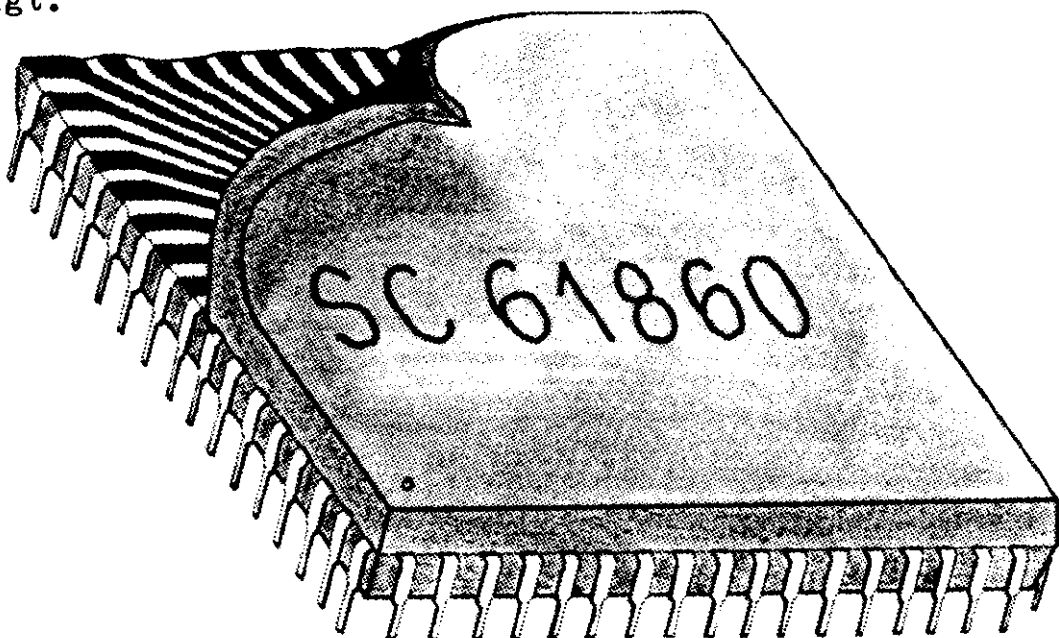
Internes Memory (RAM):

Adresse	Register/Port
00	I
01	J
02	A
03	B
04	XL
05	XH
06	YL
07	YH
08	K
09	L
..	..
..	..
5C	IA
5D	IB
5E	FO
5F	OUTC



Das RAM (beschreibbarer Speicherbereich) ist in das interne Memory (RAM) und den äußeren Daten-Speicher unterteilt. Ebenso gibt es zwei ROM-Bereiche. Der zur CPU gehörige Kern-ROM-Bereich hat 8 KByte und den Adressbereich &0000 bis &1FFF. Auf diesen Kern-ROM-Bereich läßt sich mit dem Basic-Befehl PEEK nicht zugreifen. Das äußere ROM von 32 KByte (&8000 bis &FFFF) kann mit PEEK gelesen werden.

Die Durchführung eines Befehls in der CPU geschieht in mehreren Schritten. Zuerst wird der Befehlscode aus der Adresse, auf die der Programmcounter zeigt, in das Instruktions-Register geholt. Danach wird der Befehl dekodiert und die für die Durchführung benötigten Signale erzeugt. Manche Befehle spielen sich nur in der CPU ab, andere greifen auch auf Speicherzellen zu. Bei den Mehrbyte-Befehlen muß weitere Information von den Speicherplätzen hinter dem Befehlscode geholt werden. So ergeben sich bei verschiedenen Befehlen recht unterschiedliche Ausführungszeiten, diese werden in Zykluszahlen angegeben. Bei der Durchführung eines Befehls wird der Programmcounter entsprechend der Bytezahl des Befehls erhöht, so daß er nach der Durchführung auf die Adresse des nächsten Befehls zeigt.



III DER BEFEHLSSATZ DER ESR-H CPU

Die CPU kann mehr als 100 Befehle ausführen, die aus einem oder mehreren Byte bestehen können. Die Maschinenbefehle von Mikroprozessoren werden üblicherweise in Gruppen eingeteilt. Zuerst werden diese Gruppen charakterisiert und dann die Befehle im einzelnen beschrieben.

Die Transport-Befehle umfassen hier auch die Increment- und Decrement-Befehle, da sie häufig miteinander verknüpft sind und sich daher nicht in verschiedene Gruppen einteilen lassen. Die Transport-Befehle transportieren den Inhalt von CPU-Registern oder Speicherzellen einzeln oder in Blöcken in andere CPU-Register oder Speicherzellen. Eine besondere Gruppe der Transport-Befehle sind die Lade-Immediate-Befehle. Hierbei wird das Immediate, das unmittelbar auf den Befehlscode folgende Byte oder Teilbyte, in ein Register geladen.

Die arithmetischen-Befehle geben die Möglichkeit der Addition und Subtraction (mit oder ohne Carry) und zwar von einem, zwei oder mehr Byte. Neben der binären ist auch die BCD-Addition oder -Subtraction möglich. Auch 4 Bit(BCD)-Shift ist möglich.

Die logischen-Befehle lassen die AND- und OR-Operationen zu. Die Vergleichs- und Bittest-Befehle sind ein Sonderfall der logischen Befehle. Hierbei werden die Register- oder Speicherinhalte selbst nicht verändert sondern nur die Flags beeinflusst.

Die Shift-Befehle verschieben einzelne Bit im Akkumulator und im Carry.

Die CPU-Control-Befehle kontrollieren das Carry und das Zero und Zeitverzögerungen der CPU.

Die Sprung-Befehle lassen absolute und programmcounterrelative Sprünge zu, die auch von Bedingungen abhängen können. Auch ein LOOP-Befehl ist vorhanden.

Die Unterprogramm-Befehle ermöglichen die direkten

Sprünge zu Unterprogrammen. Die Unterprogrammbefehle sind üblicherweise Dreibytebefehle. Hier gibt es jedoch auch Zweibyte-Unterprogrammbefehle. Der Return-Befehl bewirkt die Rückkehr von einer Unterroutine ins Hauptprogramm.

Die Eingabe/Ausgabe-Befehle ermöglichen das Einlesen, Auslesen oder Testen der Ports.

Unbekannte Befehle sind Befehle, über die keine genauen Informationen vorliegen. Der Begriff "Unbekannte Befehle" entstand im Zusammenhang mit dem 8085-Mikroprozessor. Dort wurden zu mehreren in der Befehlsliste unbesetzten Code Befehle gefunden und in einer Zeitschrift veröffentlicht. Die Stellungnahme des Herstellers war, daß diese Befehle ursprünglich einmal vorgesehen waren, dann aber nicht verwendet wurden und daß sie im Herstellungsprozeß auch nicht auf richtige Funktion geprüft werden.

Hier werden nur solche unbekanntem Befehle beschrieben, die entweder vom Hersteller in der Veröffentlichung über die CPU andeutungsweise erwähnt wurden oder die zumindest im ROM verwandt wurden.

In der Beschreibung der CPU der Firma Sharp wurden die Befehle DATA, READ, READM und WRIT mit Name und Code erwähnt, ohne daß sie näher beschrieben wurden. Außerdem wird dort auch von Mehrbyte-Befehlen (mehr als 3 Byte) gesprochen.

Für die Suche nach unbekanntem Befehlen gibt es verschiedene Möglichkeiten. Mit einem Breakpoint-Monitor läßt sich die Wirkung von unbekanntem Befehlen auf die Register beobachten. Auch die Bytezahl kann man dabei im allgemeinen feststellen. Setzt man nämlich einen Breakpoint hinter das erste Byte eines Mehrbyte-Befehls, so wird man im allgemeinen zur RESET-Taste geführt. Zusammenhänge mit der Systematik der Befehlsliste können Hinweise geben, z.B. bei den Increment- und Decrement-Befehlen. Schließlich kann man das ROM nach bestimmten Code durchsuchen (bzw. vom Computer durchsuchen lassen) und

untersuchen, in welchem Zusammenhang der Code dort auftritt.

Als Beispiel sei hier die Aufklärung des Table-CALL-Befehls beschrieben. Sucht man im Kernrom und im ROM nach dem unbekanntem Code &7A, so ergibt sich, daß stets auch der unbekanntem Code &69 vier Plätze später auftritt. (Es sei denn, der Code &7A befindet sich im Immediate eines Mehrbyte-Befehls.) Setzt man einen Breakpoint unmittelbar hinter &7A, so führt das zur RESET-Taste. Setzt man den Breakpoint jedoch auf den vier Plätze später folgenden Code &69, so zeigt sich, daß der Stackpointer um zwei vermindert ist und daß sich im Stack das zweite und dritte Immediate von &7A befindet. Interpretiert man diese beiden Byte als eine Adresse, so stellt man fest, daß diese in der Nähe liegt. Das erste Immediate ist meist eine kleine Zahl, z.B. 02 oder 07. Die Byte hinter dem Code &69 ergeben im allgemeinen keine sinnvolle Befehlsfolge. Statt dessen findet man dort in regelmäßiger Anordnung wieder in der Nähe liegende Adressen. Da sich im Stack eine Rückkehradresse befindet, liegt es nahe, diese Adressen als Einsprungpunkte zu Unterroutinen zu interpretieren. Verfolgt man die dortigen Befehlsfolgen, so stößt man auch stets bald auf einen Return-Befehl. Betrachtet man (z.B. bei dem später gegebenen Beispiel) die vorangehenden Befehle, so wird der Vergleich mit dem Akkumulator nahegelegt, da dort bereits der Wert des Akkus mit Immediate verglichen und über Verzweigungen entschieden wird.

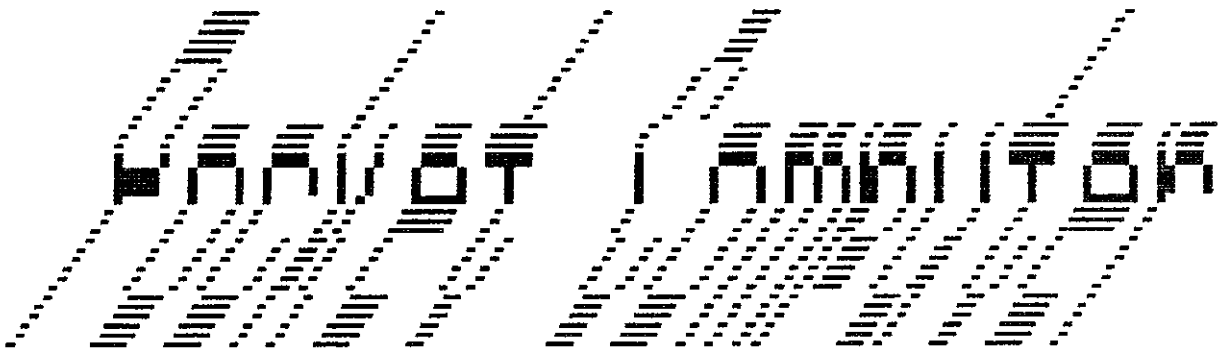
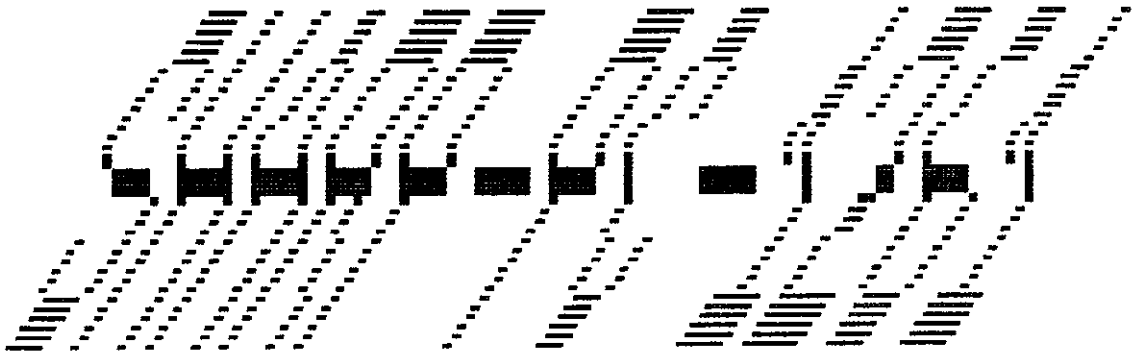
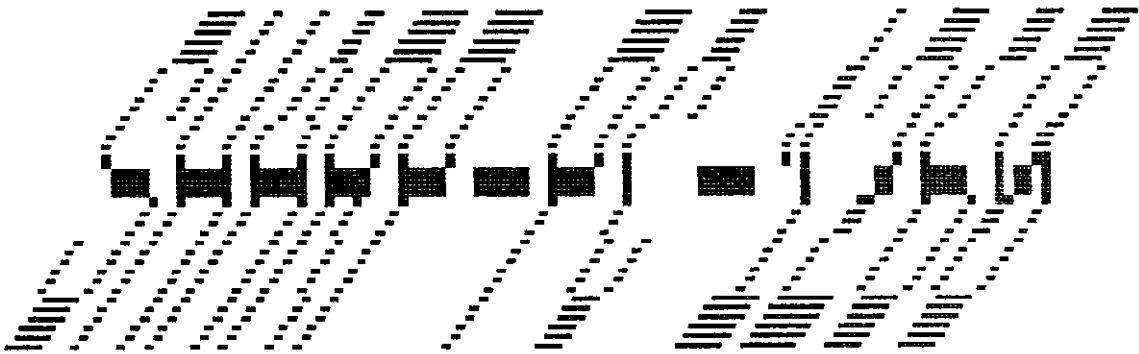
Abkürzungen bei der Beschreibung der Befehle

A	Akkumulator, 8 Bit
B	Extra-Akkumulator, 8 Bit
C	Carry Flag, 1 Bit
d	internes Block-Operation Register, 8 Bit
DP	Data Pointer, 16 Bit
DPH	höherwertige 8 Bit des DP
DPL	geringerwertige 8 Bit des DP
I	Block-Operation Register für W-Befehle, 8 Bit
J	Block-Operation Register für B-Befehle, 8 Bit Reserviert für den Basic-Interpreter
K	Allzweck-Register, 8 Bit
L	Allzweck-Register, 8 Bit
M	Allzweck-Register, 8 Bit
N	Allzweck-Register, 8 Bit
P	Adress Pointer im internen RAM, 7 Bit
PC	Programmcounter, 16 Bit
PCH	höherwertige 8 Bit des PC
PCL	geringerwertige 8 Bit des PC
Q	zweiter Adress Pointer im internen RAM, 7 Bit
R	Stack Pointer im internen RAM, 7 Bit
X	Adress Pointer auf das externe Memory, 16 Bit
XH	höherwertige 8 Bit von X
XL	geringerwertige 8 Bit von X
Y	Adress Pointer auf das externe Memory, 16 Bit
YH	höherwertige 8 Bit von Y
YL	geringerwertige 8 Bit von Y
Z	Zero Flag, 1 Bit
ℓ	6 Bit Operand beim LP-Befehl 5 Bit Operand beim CAL-Befehl
n	8 Bit Operand 7 Bit Operand für P- und Q-Register
nm	16 Bit Operand

(DP), (P), (Q), (R), (X), (Y) bedeutet jeweils den Inhalt der Speicherzelle, deren Adresse in DP, P, Q, R, X, Y ent-

halten ist.

Bei der Beschreibung der Befehle werden im allgemeinen Hexadezimalzahlen benutzt.



III.1 Transport-Befehle

Load Immediate register

Das unmittelbar auf den Befehlscode folgende Teilbyte, Byte oder Doppelbyte wird in das genannte Register geladen.

Code	Mnem.	Funktion	Flags
00n	LII	n → I	keine
01n	LIJ	n → J	keine
02n	LIA	n → A	keine
03n	LIB	n → B	keine
12n	LIP	n → P	keine
13n	LIQ	n → Q	keine
10nm	LIDP	n → DPH, m → DPL	keine
11n	LIDL	n → DPL	keine
80+e	LP	e → P, e = 6Bit (00 - 3F)	keine

Load accumulator from internal RAM pointer register

Der Inhalt eines 7Bit-internal-RAM-Register wird in den Akkumulator geladen.

20	LDP	P → A	keine
21	LDQ	Q → A	keine
22	LDR	R → A	keine

Store accumulator to internal RAM pointer register

Der Inhalt des Accumulator wird in ein 7Bit internal-RAM-Register gespeichert.

30	STP	A → P	keine
31	STQ	A → Q	keine
32	STR	A → R	keine

LoaD accumulator from internal RAM

Der Inhalt der Adresse im P-Register wird in den Akkumulator geladen.

Code	Mnem.	Funktion	Flags
59	LDM	(P) → A	keine

LoaD accumulator from external Data memory

Der Inhalt der Adresse im Data Pointer wird in den Akkumulator geladen.

57	LDD	(DP) → A	keine
----	-----	----------	-------

STore accumulator to Data pointer address

Der Inhalt des Akkumulator wird in die im Data Pointer enthaltene Adresse geschrieben.

52	STD	A → (DP)	keine
----	-----	----------	-------

Move data between Data pointer address and internal RAM address

Der Inhalt der Adresse im P-Register/Data Pointer DP wird in die Adresse des Data Pointer DP/P-Register geschrieben.

53	MVDM	(P) → (DP)	keine
55	MVMD	(DP) → (P)	keine

EXchange data between two register

Der Inhalt des Akkumulator wird mit dem B-Register oder dem Inhalt der Adresse im P-Register vertauscht

DA	EXAB	A ↔ B	keine
DB	EXAM	A ↔ (P)	keine

PUSH

Der Inhalt des Akkumulator wird in den Stack gegeben.

34	PUSH	R - 1 → R, A → (R)	keine
----	------	--------------------	-------

LEAVE

Es wird eine Null in den obersten Stackplatz gegeben.

Code	Mnem.	Funktion	Flags
D8	LEAVE	0 → (R)	keine

POP

Der Inhalt der Stackposition wird in den Akkumulator geholt.

5B	POP	(R) → A R + 1 → R	keine
----	-----	----------------------	-------

Block MoVe of data

Es werden d + 1 Bytes, beginnend mit (Q), in d + 1 Bytes, beginnend mit (P), gegeben. Wenn I/J Null ist, wird ein Byte transportiert.

08	MVW	I → d	keine
0A	MVB	J → d	keine

Wiederholt:
(Q) → (P)
und P + 1 → P, Q + 1 → Q, d - 1 → d
bis d = FF

Es werden d + 1 Bytes, beginnend mit (DP), in die d + 1 Bytes, beginnend mit (P), gegeben. Wenn I/J Null ist, wird ein Byte transportiert.

18	MVWD	I → d	keine
1A	MVBD	J → d	keine

Wiederholt:
(DP) → (P)
und DP + 1 → DP, P + 1 → P, d - 1 → d
bis d = FF

Block EXchange of data

Es werden $d + 1$ Bytes im internen RAM, beginnend bei (Q) und (P) vertauscht. Wenn I/J Null ist, wird ein Byte vertauscht,

Code	Mnem.	Funktion	Flags
09	EXW	$I \rightarrow d$	keine
0B	EXB	$J \rightarrow d$	keine
		Wiederholt: $(Q) \leftrightarrow (P)$ und $P + 1 \rightarrow P, Q + 1 \rightarrow Q, d - 1 \rightarrow d$ bis $d = FF$	

Es werden $d + 1$ Bytes, beginnend mit (DP), vertauscht mit $d + 1$ Bytes, beginnend mit (P). Wenn I/J Null ist, wird ein Byte vertauscht.

19	EXWD	$I \rightarrow d$	keine
1B	EXBD	$J \rightarrow d$	keine
		Wiederholt: $(DP) \leftrightarrow (P)$ und $DP + 1 \rightarrow DP, P + 1 \rightarrow P, d - 1 \rightarrow d$ bis $d = FF$	

INCrement or DECrement 7/8-Bit register

Es wird 1 zum/vom Inhalt eines 7/8-Bit-Register addiert/subtrahiert.

50	INCP	$P + 1 \rightarrow P$	keine
51	DECP	$P - 1 \rightarrow P$	keine
40	INCI	$I + 1 \rightarrow I, Q$ wird verändert	C, Z
41	DECI	$I - 1 \rightarrow I, Q$ wird verändert	C, Z
C0	INCJ	$J + 1 \rightarrow J, Q$ wird verändert	C, Z
C1	DECJ	$J - 1 \rightarrow J, Q$ wird verändert	C, Z
42	INCA	$A + 1 \rightarrow A, Q$ wird verändert	C, Z
43	DECA	$A - 1 \rightarrow A, Q$ wird verändert	C, Z
C2	INCB	$B + 1 \rightarrow B, Q$ wird verändert	C, Z
C3	DECB	$B - 1 \rightarrow B, Q$ wird verändert	C, Z

Code	Mnem.	Funktion	Flags
48	INCK	$K + 1 \rightarrow K$, Q wird verändert	C, Z
49	DECK	$K - 1 \rightarrow K$, Q wird verändert	C, Z
C8	INCL	$L + 1 \rightarrow L$, Q wird verändert	C, Z
C9	DECL	$L - 1 \rightarrow L$, Q wird verändert	C, Z

Q-Register wird verändert

Increment or Decrement 16-Bit register and move the address to DP register

X oder Y wird inkrementiert oder dekrementiert und der Inhalt von X oder Y nach DP gegeben.

04	IX	$X \rightarrow DP$, $DP + 1 \rightarrow DP \rightarrow X$	keine
05	DX	$X \rightarrow DP$, $DP - 1 \rightarrow DP \rightarrow X$	keine
06	IY	$Y \rightarrow DP$, $DP + 1 \rightarrow DP \rightarrow Y$	keine
07	DY	$Y \rightarrow DP$, $DP - 1 \rightarrow DP \rightarrow Y$	keine

Q-Register wird verändert

Increment or Decrement X-register and Load the content of the new address into the accumulator

Das X-Register wird inkrementiert oder dekrementiert, die neue Adresse wird in den Data Pointer DP gegeben und der Inhalt der neuen Adresse in den Accumulator geladen.

24	IXL	$X \rightarrow DP$ $DP + 1 \rightarrow DP \rightarrow X$ $(DP) \rightarrow A$ Q-Register wird verändert DP hat keinen Zugriff auf das Kern-ROM (0000 - 1FFF)	keine
25	DXL	$X \rightarrow DP$ $DP - 1 \rightarrow DP \rightarrow X$ $(DP) \rightarrow A$ Q-Register wird verändert DP hat keinen Zugriff auf das Kern-ROM (0000 - 1FFF)	keine

Increment or Decrement Y-register and Sore content of accumulator into the new address

Das Y-Register wird inkrementiert oder dekrementiert, die neue Adresse wird in den Data Pointer DP gegeben und der Inhalt des Akkumulator in die neue Adresse gegeben.

Code	Mnem.	Funktion	Flags
26	IYS	Y \rightarrow DP DP + 1 \rightarrow DP \rightarrow Y A \rightarrow (DP) Q-Register wird verändert	keine
27	DYS	Y \rightarrow DP DP - 1 \rightarrow DP \rightarrow Y A \rightarrow (DP) Q-Register wird verändert	keine

Fill a block internal or external memory with a single value

Der Inhalt des Akkumulator wird in d + 1 Byte, beginnend mit (P) oder (DP), gegeben. Wenn I/J Null ist, wird ein Byte gefüllt.

1E	FILM	I \rightarrow d Wiederholt: A \rightarrow (P) und P + 1 \rightarrow P, d - 1 \rightarrow d bis d = FF	keine
1F	FILD	I \rightarrow d Wiederholt: A \rightarrow (DP) und DP + 1 \rightarrow DP, d - 1 \rightarrow d bis d = FF	keine

III.2 Arithmetische Befehle

ADd/SuBtract Immediate to/from Accumulator/internal RAM

Das unmittelbar auf den Befehlscode folgende Byte wird zum/vom Akkumulator/internal RAM addiert/subtrahiert.

Code	Mnem.	Funktion	Flags
74n	ADIA	$A + n \rightarrow A$	C, Z
75n	SBIA	$A - n \rightarrow A$	C, Z
70n	ADIM	$(P) + n \rightarrow (P)$	C, Z
71n	SBIM	$(P) - n \rightarrow (P)$	C, Z

1 byte binary ADdition/SuBtraction without or with Carry of accumulator and internal RAM

Der Inhalt des Akkumulator wird ohne oder mit Carry zum/vom internen RAM addiert/subtrahiert.

44	ADM	$(P) + A \rightarrow (P)$	C, Z
45	SBM	$(P) - A \rightarrow (P)$	C, Z
C4	ADCM	$(P) + A + C \rightarrow (P)$	C, Z
C5	SBCM	$(P) - A - C \rightarrow (P)$	C, Z

ADdition/SuBtraction of two byte Binary

Der Inhalt der Adresse (P + 1,P) wird mit dem Inhalt der Adresse (B,A) addiert/subtrahiert.

14	ADB	$(P + 1,P) + (B,A) \rightarrow (P + 1,P)$	C, Z
15	SBB	$(P + 1,P) - (B,A) \rightarrow (P + 1,P)$	C, Z

Beispiele für ADB

1)	(05,04) : FFFE	
	(B,A) : 0001	
Nach P = 04, ADB:	(05,04) : FFFF, C = 0, Z = 0, P = 05	
2)	(05,04) : FFFE	
	(B,A) : 0002	
Nach P = 04, ADB:	(05,04) : 0000, C = 1, Z = 1, P = 05	

Beispiele für SBB

1) (05,04) : 1114
 (B,A) : 1113

 Nach P = 04, SBB: (05,04) : 0001, C = 0, Z = 0, P = 05

2) (05,04) : 1114
 (B,A) : 1114

 Nach P = 04, SBB: (05,04) : 0000, C = 0, Z = 1, P = 05

3) (05,04) : 1114
 (B,A) : 1115

 Nach P = 04, SBB: (05,04) : FFFF, C = 1, Z = 0, P = 05

Block BCD ADdition and SUbtraction

Es werden $d + 1$ Bytes, deren rechte Stelle durch P definiert wird, als eine Zahl betrachtet und hierzu/hiervon der Inhalt des Akkumulator BCD addiert/subtrahiert oder die $d + 1$ Bytes, deren rechte Stelle durch Q definiert wird, hierzu/hiervon addiert/subtrahiert.

ADd accumulator

Der Inhalt des Akkumulator wird zu dem Block von $d + 1$ Bytes, deren rechte Stelle durch P definiert ist, BCD addiert.

Code	Mnem.	Funktion	Flags
0C	ADN	I \rightarrow d Wiederholt (P) + A \rightarrow (P), BCD P - 1 \rightarrow P, d - 1 \rightarrow d bis d = FF	C, Z

Beispiele zu ADN

1) I = 03 (04) : 99999999
 A : 01

 Nach P = 07, ADN: (04) : 00000000, C = 1, Z = 1, P = 03

2) I = 03 (04) : 99999999
 A : 99

 Nach P = 07, ADN: (04) : 00000098, C = 1, Z = 0, P = 03

SuBtract accumulator

Der Inhalt des Akkumulator wird von dem Block von $d + 1$ Bytes, deren rechte Stelle durch P definiert ist, BCD subtrahiert.

Code	Mnem.	Funktion	Flags
OD	SBN	$I \rightarrow d$ Wiederholt $(P) - A \rightarrow (P)$, BCD $P - 1 \rightarrow P$, $d - 1 \rightarrow d$ bis $d = FF$	C, Z

Beispiele zu SBN

1)	$I = 03$	$(04) : 88888888$ <u> A : 99</u>	
Nach	$P = 07$, SBN:	$(04) : 88888789$, C = 0, Z = 0, P = 03	
2)	$I = 03$	$(04) : 00000088$ <u> A : 99</u>	
Nach	$P = 07$, SBN:	$(04) : 99999989$, C = 1, Z = 0, P = 03 99999989 = 100000000 - 11	

ADd tWo blocks

Die beiden Blöcke von $d + 1$ Bytes, deren rechte Stellen durch P und Q definiert sind, werden BCD addiert.

OE	ADW	$I \rightarrow d$ Wiederholt $(P) + (Q) \rightarrow (P)$, BCD $P - 1 \rightarrow P$, $Q - 1 \rightarrow Q$, $d - 1 \rightarrow d$ bis $d = FF$	C, Z
----	-----	---	------

Beispiele zu ADW

1)	$I = 03$	$(04) : 77777777$ <u> (08) : 22222222</u>	
Nach	$P = 07$, $Q = 0B$:	$(04) : 99999999$ C = 0, Z = 0, P = 03, Q = 06	

2) I = 03 (04) : 77777777
 (08) : 22222223
 Nach P = 07, Q = 0B, ADW: (04) : 00000000
 C = 1, Z = 1, P = 03
 Q = 06

3) I = 03 (04) : 77777777
 (08) : 22222224
 Nach P = 07, Q = 0B, ADW: (04) : 00000001
 C = 1, Z = 0, P = 03
 Q = 06

SuBtract two blocks

Die beiden Blöcke von d + 1 Bytes, deren rechte Stellen durch P und Q definiert sind, werden BCD subtrahiert.

Code	Mnem.	Funktion	Flags
OF	SBW	I → d	C, Z
		Wiederholt	
		(P) - (Q) → (P), BCD	
		P - 1 → P, Q - 1 → Q, d - 1 → d	
		bis d = FF	

Beispiele zu SBW

1) I = 03 (04) : 77777777
 (08) : 55555555
 Nach P = 07, Q = 0B, SBW: (04) : 22222222
 C = 0, Z = 0, P = 03
 Q = 06

2) I = 03 (04) : 77777777
 (08) : 77777777
 Nach P = 07, Q = 0B, SBW: (04) : 00000000
 C = 0, Z = 1, P = 03
 Q = 06

3) I = 03

(04) : 77777777

(08) : 77777799

Nach P = 07, Q = 0B, SBW:

(04) : 99999978 = 100000000 - 22

C = 1, Z = 0, P = 03

Q = 06

Shift 4 bits of a block

Es werden in einem Block von d + 1 Bytes 4 Bit nach rechts oder links verschoben.

Shift Right 4 bits

Es werden in einem Block von d + 1 Bytes, dessen erste Stelle durch P definiert ist, 4 Bit nach rechts verschoben. Von links wird eine Null eingeschoben.

Code	Mnem.	Funktion	Flags
1C	SRW	I → d Wiederholt Verschieben um 4 Bits in (P) P + 1 → P, d - 1 → d bis d = FF	keine

Beispiel für SRW

I = 03

(04) : 12345678

Nach P = 04, SRW:

(04) : 01234567, P = 08

Shift Left 4 bits

Es werden in einem Block von d + 1 Bytes, dessen letzte Stelle durch P definiert ist, 4 Bit nach links verschoben. Von rechts wird eine Null eingeschoben.

1D	SLW	I → d Wiederholt Verschieben um 4 Bits in (P) P - 1 → P, d - 1 → d bis d = FF	keine
----	-----	--	-------

Beispiel für SLW

I = 03

(04) : 12345678

Nach P = 07, SLW:

(04) : 23456780, P = 03

III.3 Logische Befehle

Logical ANd

Es werden zwei Byte (Akkumulator, internes RAM, externes Data Memory, Immediate) mit der UND-Operation verknüpft.

Code	Mnem.	Funktion	Flags
64n	ANIA	$A \wedge n \rightarrow A, Z$	Z
60n	ANIM	$(P) \wedge n \rightarrow (P), Z$	Z
D4n	ANID	$(DP) \wedge n \rightarrow (DP), Z$ R - 1 wird mitbenutzt	Z
46	ANMA	$(P) \wedge A \rightarrow (P), Z$	Z

Logical OR

Es werden zwei Byte (Akkumulator, internes RAM, externes Data Memory, Immediate) mit der ODER-Operation verknüpft.

65n	ORIA	$A \vee n \rightarrow A, Z$	Z
61n	ORIM	$(P) \vee n \rightarrow (P), Z$	Z
D5n	ORID	$(DP) \vee n \rightarrow (DP), Z$ R - 1 wird mitbenutzt	Z
47	ORMA	$(P) \vee A \rightarrow (P), Z$	Z

ComPare

Es werden zwei Byte miteinander verglichen, der Akkumulator mit einem Immediate, eine interne RAM-Zelle mit einem Immediate oder eine interne RAM-Zelle mit dem Akkumulator.

67n	CPIA	$A - n$ Für $A < n$: C = 1, Z = 0 A = n: C = 0, Z = 1 A > n: C = 0, Z = 0	C, Z
63n	CPIM	$(P) - n$ Für $(P) < n$: C = 1, Z = 0 (P) = n: C = 0, Z = 1 (P) > n: C = 0, Z = 0	C, Z

Code	Mnem.	Funktion	Flags
C7	CPMA	(P) - A	C, Z
		Für (P) < A: C = 1, Z = 0	
		(P) = A: C = 0, Z = 1	
		(P) > A: C = 0, Z = 0	

Bit TeSt Immediate

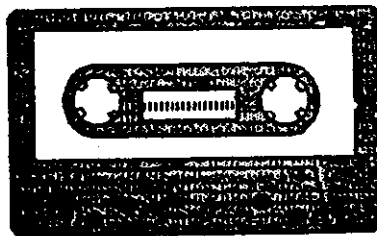
Zwei Byte werden mit dem AND-Operator verknüpft, der Akkumulator mit einem Immediate, eine interne RAM-Zelle mit einem Immediate oder eine externe Data-Zelle mit einem Immediate.

Die beiden Byte werden nicht verändert. Nur das Zero-Flag wird eins, falls das Ergebnis Null ist.

66n	TSIA	$A \wedge n \rightarrow Z$	Z
62n	TSIM	$(P) \wedge n \rightarrow Z$	Z
D6n	TSID	$(DP) \wedge n \rightarrow Z$	Z
		R - 1 wird mitbenutzt	

*und passend zu diesem
Buch :*

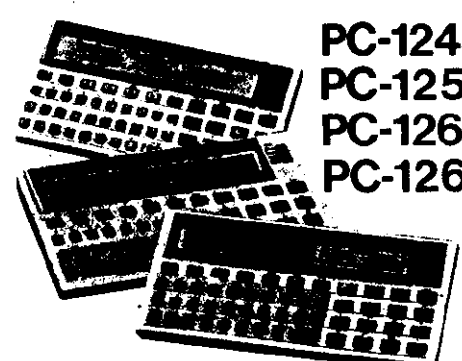
**Pocket -
SOFTWARE**



INGO LAUE

**Anwendungs-
Handbuch**

zum **SHARP** Taschencomputer




PC-1245

PC-1251

PC-1260

PC-1261

FISCHEL GmbH



INGO LAUE

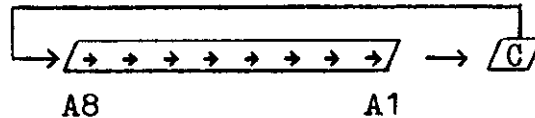
ISBN 3-824 327-14-9

III.4 Shift-Befehle

Shift Right accumulator

Der Inhalt des Akkumulator wird um eine Bit-Position nach rechts verschoben. Das ursprünglich geringstwertige Bit geht in das Carry-Flag. Der ursprüngliche Inhalt des Carry-Flag geht in das höchstwertige Bit des Akkumulator.

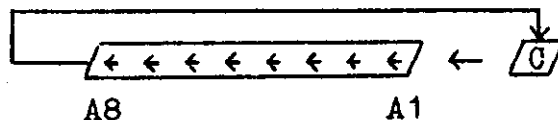
Code	Mnem.	Funktion	Flags
D2	SR		C



Shift Left accumulator

Der Inhalt des Akkumulator wird um eine Bit-Position nach links verschoben. Das ursprünglich höchstwertige Bit geht in das Carry-Flag. Der ursprüngliche Inhalt des Carry-Flag geht in das geringstwertige Bit.

5A	SL		C, Z
----	----	--	------

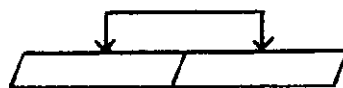


Bei SL wird, im Gegensatz zu SR, das Zero-Flag beeinflusst. Jedoch nicht nach den sonst üblichen Regeln.

SWaP

Die ersten 4 werden mit den letzten 4 Bit im Akkumulator vertauscht.

58	SWP		keine
----	-----	--	-------



Block-Shift-Befehle siehe unter arithmetische Befehle.

III.5 CPU-Steuerbefehle

Set Carry

Das Carry-Flag wird 1 gesetzt.

Auch das Zero-Flag wird 1 gesetzt.

Code	Mnem.	Funktion	Flags
DO	SC	1 → C, 1 → Z	C, Z

Reset Carry

Das Carry-Flag wird 0 gesetzt.

Das Zero-Flag wird 1 gesetzt.

D1	RC	0 → C, 1 → Z	C, Z
----	----	--------------	------

No Operation, 2 cycle

Die CPU wartet zwei Zyklen.

4D	NOPW	keine Funktion	keine
----	------	----------------	-------

No Operation, 3 cycle

Die CPU wartet drei Zyklen.

CE	NOPT	keine Funktion	keine
----	------	----------------	-------

WAIT

Die CPU wartet 6 + n Zyklen.

4En	WAIT	keine Funktion	keine
-----	------	----------------	-------

III.6 Sprungbefehle

JumP absolute without or with condition

Absoluter Sprung zu der in den auf den Befehlscode unmittelbar folgenden zwei Byte angegebenen Adresse. Das Sprungziel kann von einer Bedingung abhängen.

Code	Mnem.	Funktion	Flags
79nm	JP	$n \rightarrow PCH, m \rightarrow PCL$	keine
7Enm	JPZ	Falls $Z = 1: n \rightarrow PCH, m \rightarrow PCL$ $Z = 0: PC + 3 \rightarrow PC$	keine keine
7Cnm	JPNZ	Falls $Z = 0: n \rightarrow PCH, m \rightarrow PCL$ $Z = 1: PC + 3 \rightarrow PC$	keine keine
7Fnm	JPC	Falls $C = 1: n \rightarrow PCH, m \rightarrow PCL$ $C = 0: PC + 3 \rightarrow PC$	keine keine
7Dnm	JPNC	Falls $C = 0: n \rightarrow PCH, m \rightarrow PCL$ $C = 1: PC + 3 \rightarrow PC$	keine

JumP Relative without or with condition in Plus direction

Es wird relativ zum Programmcounter + 1 in Vorwärtsrichtung gesprungen. Die Sprungweite wird durch das unmittelbar auf den Befehlscode folgende Byte festgelegt. Das Sprungziel kann von einer Bedingung abhängen.

2Cn	JRP	$PC + 1 + n \rightarrow PC$	keine
38n	JRZP	Falls $Z = 1: PC + 1 + n \rightarrow PC$ $Z = 0: PC + 2 \rightarrow PC$	keine keine
28n	JRNZP	Falls $Z = 0: PC + 1 + n \rightarrow PC$ $Z = 1: PC + 2 \rightarrow PC$	keine keine
3An	JRCP	Falls $C = 1: PC + 1 + n \rightarrow PC$ $C = 0: PC + 2 \rightarrow PC$	keine keine
2An	JRNCP	Falls $C = 0: PC + 1 + n \rightarrow PC$ $C = 1: PC + 2 \rightarrow PC$	keine keine

JumP Relative without or with condition in Minus direction

Es wird relative zum Programmcounter + 1 in Rückwärtsrichtung gesprungen. Die Sprungweite wird durch das unmittelbar auf den Befehlscode folgende Byte festgelegt. Das Sprungziel kann von einer Bedingung abhängen.

Code	Mnem.	Funktion	Flags
2Dn	JRM	$PC + 1 - n \rightarrow PC$	keine
39n	JRZM	Falls $Z = 1$: $PC + 1 - n \rightarrow PC$ $Z = 0$: $PC + 2 \rightarrow PC$	keine keine
29n	JRNZM	Falls $Z = 0$: $PC + 1 - n \rightarrow PC$ $Z = 1$: $PC + 2 \rightarrow PC$	keine keine
3Bn	JRCM	Falls $C = 1$: $PC + 1 - n \rightarrow PC$ $C = 0$: $PC + 2 \rightarrow PC$	keine keine
2Bn	JRNCM	Falls $C = 0$: $PC + 1 - n \rightarrow PC$ $C = 1$: $PC + 2 \rightarrow PC$	keine keine

LOOP

Es wird der Inhalt der obersten Stackposition decremen- tiert. Wenn danach das Carry-Flag 0 ist, wird ein rela- tiver Sprung nach $PC + 1 - n$ ausgeführt. Falls das Carry-Flag 1 ist, wird der nächste Befehl durchgeführt. Die Anzahl der Schleifen muß vorangehend in den Stack gegeben werden. Nach Abschluß der Schleifen wird der Stackpointer automatisch um eins erhöht.

2Fn	LOOP	$(R) - 1 \rightarrow (R)$	keine
		Falls $C = 0$: $PC + 1 - n \rightarrow PC$	
		$C = 1$: $PC + 2 \rightarrow PC, R + 1 \rightarrow R$	

III.7 Unterprogramm-Befehle

CALL subroutine

Die Rückkehradresse hinter dem CALL-Befehl (PC + 3) wird in den Stack gegeben. Dann wird zu der Adresse, die durch die unmittelbar auf den Befehlscode folgenden zwei Byte gegeben ist, verzweigt.

Code	Mnem.	Funktion	Flags
78nm	CALL	PC + 3 \rightarrow (R - 1, R - 2) R - 2 \rightarrow R n \rightarrow PCH, m \rightarrow PCL	keine

CALl subroutine

Die Rückkehradresse hinter dem CAL-Befehl (PC + 2) wird in den Stack gegeben. Dann wird zu der durch ℓ n gegebenen Adresse verzweigt. ℓ ist durch die letzten 5 Bit des Befehlscode und n durch das unmittelbar auf den Befehlscode folgende Byte gegeben. Es können so nur die ersten 8 KByte des Speichers (0000 - 1FFF) adressiert werden.

E0ℓn	CAL	PC + 2 \rightarrow (R - 1, R - 2) R - 2 \rightarrow R ℓ \rightarrow PCH, 00 \leq ℓ \leq 1F n \rightarrow PCL	keine
------	-----	---	-------

ReTurN from subroutine

Die Rückkehradresse wird vom Stack in den Programmcounter geholt.

37	RTN	(R) \rightarrow PCL, (R + 1) \rightarrow PCH R + 2 \rightarrow R	keine
----	-----	---	-------

III.8 Eingabe-/Ausgabe-Befehle

IN port IA

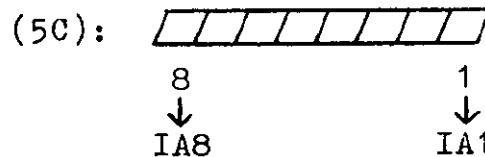
Einlesen des Port IA in den Akkumulator. Bevor das Port IA gelesen wird, muß es in den Input-Mode gesetzt werden. Dies ist bitweise möglich. Im internen RAM 5C werden alle die Bit, die eingelesen werden sollen, Null gesetzt. Dann wird der Inhalt von 5C mit dem Befehl OUTA herausgeschrieben. Danach kann Port IA in den Akkumulator gelesen werden.

Code	Mnem.	Funktion	Flags
4C	INA	IAport → A	Z

OUT port IA

Der Inhalt des internen RAM 5C wird in Port IA geschrieben.

5D	OUTA	(5C) → IAport	keine
----	------	---------------	-------



IN port IB

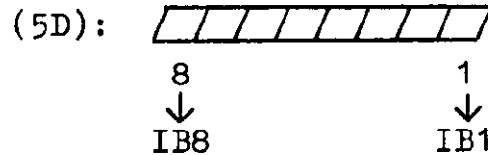
Einlesen des Port IB in den Akkumulator. Bevor das Port IB gelesen wird, muß es in den Input-Mode gesetzt werden. Dies ist bitweise möglich. Im internen RAM werden all die Bit, die eingelesen werden sollen, Null gesetzt. Dann wird der Inhalt von 5D mit dem Befehl OUTB herausgeschrieben. Danach kann Port IB in den Akkumulator gelesen werden.

CC	INB	IBport → A	Z
----	-----	------------	---

OUT port IB

Der Inhalt des internen RAM 5D wird in das Port IB geschrieben.

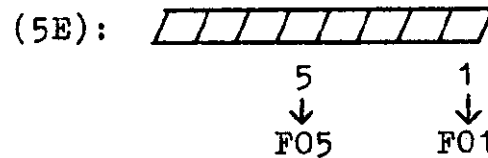
Code	Mnem.	Funktion	Flags
DD	OUTB	(5D) → IBport	keine



OUT port FO

Der Inhalt des internen RAM 5E (Bit 1 bis 5) wird in das Port FO geschrieben.

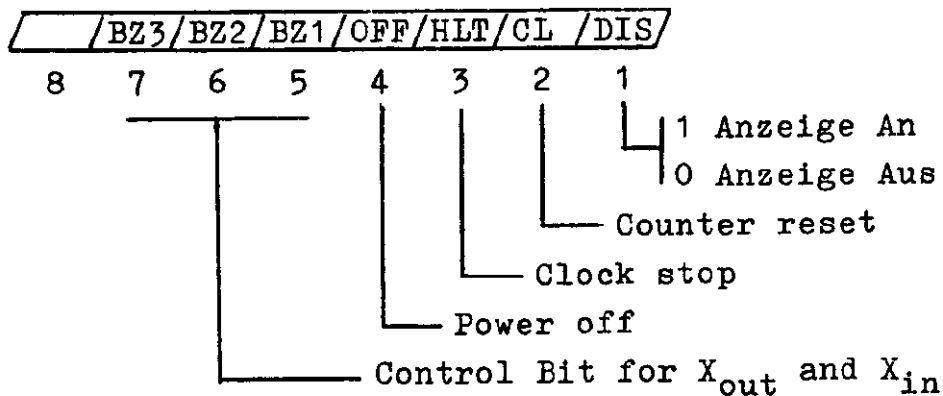
5F	OUTF	(5E) → portFO	keine
----	------	---------------	-------



OUT Control port

Der Inhalt des internen RAM 5F wird in das Control Port geschrieben. Das Bit 1 steuert die LCD-Anzeige. Ändert man Bit 1, so ist es wichtig, die anderen Bit zu erhalten.

DF	OUTC	(5F) → control port	keine
----	------	---------------------	-------



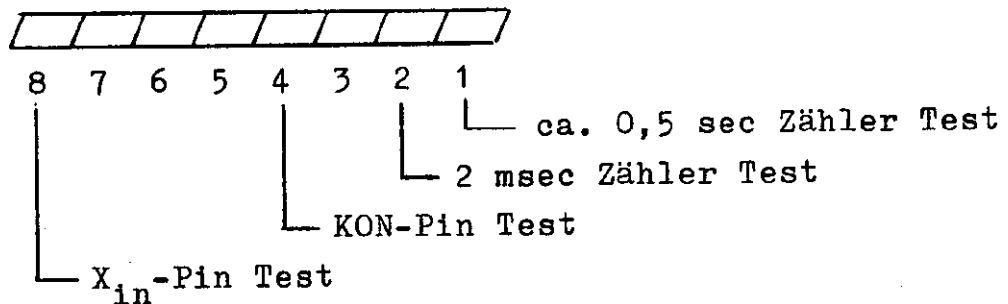
Bedeutung von BZ3 bis BZ1 auf der nächsten Seite

BZ3	BZ2	BZ1	X _{out}	X _{in}
0	0	0	0	inaktiv
0	0	1	1	inaktiv
0	1	0	2 kHz	inaktiv
0	1	1	4 kHz	inaktiv
1	0	0	0	aktiv
1	0	1	1	aktiv
1	1	x	X _{in} → X _{out}	aktiv

TEST pins or counter

Es wird das KON- oder X_{in}-Pin oder ein Counter getestet.

Code	Mnem.	Funktion	Flags
6Bn	TEST	n TEST → Z	Z



- TEST 80: X_{in} = 1/0 → Z = 0/1
- TEST 08: KON = 1/0 → Z = 0/1
- TEST 02: Zähler 1/0 → Z = 0/1
- TEST 01: Zähler 1/0 → Z = 0/1

III.9 Unbekannte Befehle

Erläuterungen zu den unbekanntem Befehlen finden sich am Anfang dieses Kapitels.

DATA from internal ROM to internal RAM

Es werden $d + 1$ Byte, beginnend bei der in B,A enthaltenen Adresse, in das interne RAM, beginnend bei (P), gelesen. Es wird auch auf das Kern-ROM (0000 - 1FFF) zugegriffen. Der Zugriff auf das Kern-ROM ist über den Datapointer DP nicht möglich.

Code	Mnem.	Funktion	Flags
35	DATA	I \rightarrow d (B,A) bis (B,A) + d + 1 \rightarrow (P) bis (P) + d + 1 Es kann auch P = 02 sein	keine

Beispiel

Im Kern-ROM ist 0002: 02, 01, 12, 5F

Nach I = 03, A = 02, B = 00, P = 04, DATA ergibt sich 04: 02, 01, 12, 5F, A = 02, B = 00, P = 08

Der DATA-Befehl wird in der Sharp-Veröffentlichung über die CPU mit Name und Code genannt, aber nicht näher beschrieben. Er wird im Kern-ROM mehrfach benutzt.

READ in accumulator or internal RAM

Ein Byte??? wird in den Akkumulator oder die Adresse im P-Register gelesen.

56	READ	Byte??? \rightarrow A	<i>LDA (PC+1)</i>	keine
54	READM	Byte??? \rightarrow (P)	<i>LDM (PC+1)</i>	keine

Diese beiden Befehle werden in der Sharp-Veröffentlichung über die CPU mit Name und Code genannt, aber nicht näher beschrieben. Es bleibt unklar, welches Byte gelesen wird.

WRITE

Es handelt sich möglicherweise um eine Umkehrung von READ?

Code	Mnem.	Funktion	Flags
D3	WRIT	???	

Dieser Befehl wird in der Sharp-Veröffentlichung über die CPU mit Name und Code genannt, aber nicht näher beschrieben.

Define Table Length and Return-Address

Es wird eine Tabellenlänge und eine Rückkehradresse definiert. Dies geschieht mit dem Code 7A. Danach wird eine Tabelle mit Vergleichsbyte und Subroutinensprungadressen aufgestellt. Dies geschieht mit dem Code 69.

7A	nm	DTLRA	$c \rightarrow$ Zähler	keine
			$nm \rightarrow (R - 1, R - 2)$	
			$R - 2 \rightarrow R$	

Compare and CALL

Es wird der Inhalt des Akkumulator mit den Tabellenwerten verglichen. Bei Übereinstimmung wird die Subroutine mit der dem Tabellenwert folgenden Adresse aufgerufen. Entspricht keiner der Tabellenwerte dem Inhalt des Akkumulator, so wird die letzte Adresse aufgerufen.

Code	Mnem.	Funktion	Flags	
69	$c_1 n_1 m_1$	CPCAL	$A - c_1 \rightarrow C, Z$	C, Z
	$c_2 n_2 m_2$		Falls: $Z = 1: n_1 \rightarrow PCH, m_1 \rightarrow PCL$	
			$Z = 0: \ell - 1 \rightarrow \ell \rightarrow Z$	
			da $Z = 0$ (für $\ell > 1$): weiter	
.....			
	$c_\ell n_\ell m_\ell$		$A - c_\ell \rightarrow C, Z$	
	$n_{\ell+1} m_{\ell+1}$		Falls $Z = 1: n_\ell \rightarrow PCH, m_\ell \rightarrow PCL$	
			$Z = 0: \ell - 1 \rightarrow \ell \rightarrow Z$	
			da $Z = 1: n_{\ell+1} \rightarrow PCH, m_{\ell+1} \rightarrow PCL$	

Diese Kombination 7A, Byte, Byte, Byte, 69,.... wird im Basic-Interpreter häufig benutzt.

Beispiel

```
OC76: CPIA AD
OC78: JRNCP 05 >OC7E
OC7A: CPIA AA
OC7C: JRNCP 06 >OC83
.....
.....
.....
OC83: 7A 02 0C 90
OC87: 69
OC88: AB 80 39
OC8B: AC 80 3C
OC8E: 80 3F
```

D.h. OC83: Zwei Tabellenwerte, Rückkehradresse OC90
OC88: Falls A = AB: CALL 8039, Rückkehr nach OC90
OC8B: Falls A = AC: CALL 803C, Rückkehr nach OC90
OC8E: Sonst : CALL 803F, Rückkehr nach OC90

INCrement or DECrement

Es wird der Inhalt eines Registers um eins erhöht oder erniedrigt. Für die Adressen OA, OB des internen RAM wurden die Namen M, N gewählt, da sie auf K, L folgen.

Code	Mnem.	Funktion	Flags
4A	INCM	(OA) + 1 → (OA)	C, Z
4B	DECM	(OA) - 1 → (OA)	C, Z
CA	INCN	(OB) + 1 → (OB)	C, Z
CB	DECN	(OB) - 1 → (OB)	C, Z

Diese Befehle kommen im Kern-ROM des PC-1260/1261 vor.

III.10 Kurzbeschreibung der Befehle

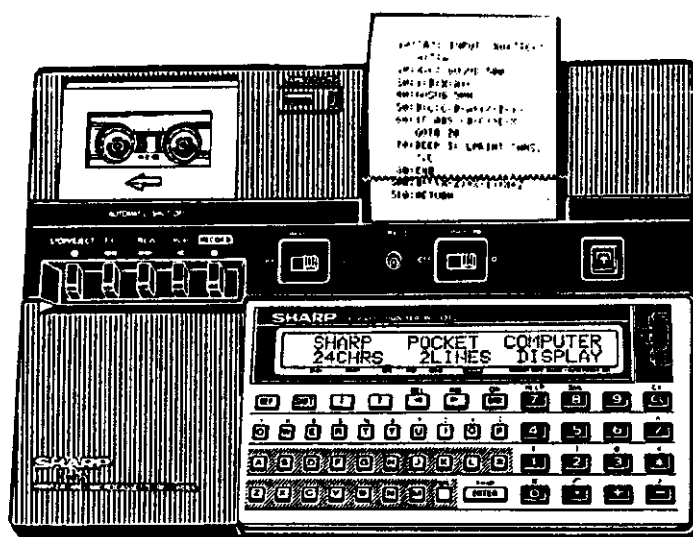
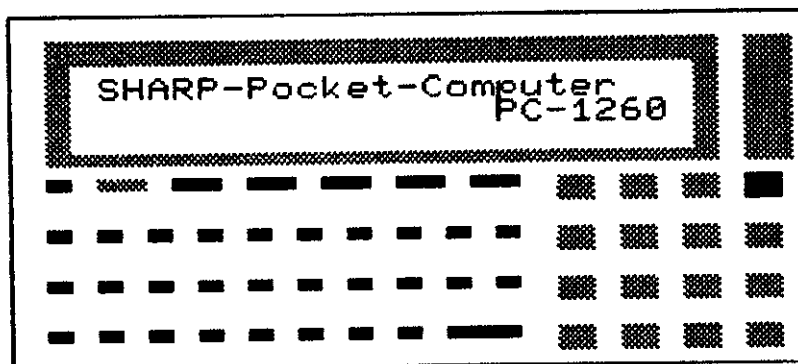
Mnem.	Funktion	Flags	Zyklen	Code	Byte
ADB	$(P+1, P) + (B, A) \rightarrow (P+1, P)$	C, Z	5	14	1
ADCM	$(P) + A + C \rightarrow (P)$	C, Z	3	C4	1
ADIAN	$A + n \rightarrow A$	C, Z	4	74	2
ADIMn	$(P) + n \rightarrow (P)$	C, Z	4	70	2
ADM	$(P) + A \rightarrow (P)$	C, Z	3	44	1
ADN	$(P) + A \rightarrow (P), (BCD)$	C, Z	7+3d	0C	1
ADW	$(P) + (Q) \rightarrow (P), (BCD)$	C, Z	7+3d	0E	1
ANIAN	$A \wedge n \rightarrow A$	-, Z	4	64	2
ANIDn	$(DP) \wedge n \rightarrow (DP)$	-, Z	6	D4	2
ANIMn	$(P) \wedge n \rightarrow (P)$	-, Z	4	60	2
ANMA	$(P) \wedge A \rightarrow (P)$	-, Z	3	46	1
CAL ℓ n	$PC + 2 \rightarrow (R-1, R-2),$ $R-2 \rightarrow R, \ell n \rightarrow PC$	-, -	7	EO+ ℓ	2
CALLnm	$PC + 3 \rightarrow (R-1, R-2),$ $R-2 \rightarrow R, nm \rightarrow PC$	-, -	8	78	3
CPCAL	siehe unbek. Befehle	C, Z	?	69	x
CPIAn	$A - n \rightarrow C, Z$	C, Z	4	67	2
CPIMn	$(P) - n \rightarrow C, Z$	C, Z	4	63	2
CPMA	$(P) - A \rightarrow C, Z$	C, Z	3	C7	1
DATA	siehe unbek. Befehle	-, -	?	35	1
DECA	$A - 1 \rightarrow A$	C, Z	4	43	1
DECB	$B - 1 \rightarrow B$	C, Z	4	C3	1
DECI	$I - 1 \rightarrow I$	C, Z	4	41	1
DECJ	$J - 1 \rightarrow J$	C, Z	4	C1	1
DECK	$K - 1 \rightarrow K$	C, Z	4	49	1
DECL	$L - 1 \rightarrow L$	C, Z	4	C9	1
? DECM ✓	$M - 1 \rightarrow M$	C, Z	4	4B	1
? DECN ✓	$N - 1 \rightarrow N$	C, Z	4	CB	1
DECP	$P - 1 \rightarrow P$	-, -	2	51	1
DTLRA	siehe unbek. Befehle	-, -	?	7A	4
DX	$X - 1 \rightarrow X, X \rightarrow DP$	-, -	6	05	1
DXL	$X - 1 \rightarrow X, X \rightarrow DP, (DP) \rightarrow A$	-, -	7	25	1
DY	$Y - 1 \rightarrow Y, Y \rightarrow DP$	-, -	6	07	1

Mnem.	Funktion	Flags	Zyklen	Code	Byte
DYS	$Y-1 \rightarrow Y, Y \rightarrow DP, A \rightarrow (DP)$	-, -	6	27	1
EXAB	$A \leftrightarrow B$	-, -	3	DA	1
EXAM	$A \leftrightarrow (P)$	-, -	3	DB	1
EXB	$(P) \leftrightarrow (Q)$	-, -	6+3d	0B	1
EXBD	$(P) \leftrightarrow (DP)$	-, -	7+6d	1B	1
EXW	$(P) \leftrightarrow (Q)$	-, -	6+3d	09	1
EXWD	$(P) \leftrightarrow (DP)$	-, -	7+6d	19	1
FIELD	$A \rightarrow (DP), DP+1 \rightarrow DP$	-, -	4+3d	1F	1
FILM	$A \rightarrow (P), P+1 \rightarrow P$	-, -	5+d	1E	1
INA	$IAport \rightarrow A$	-, Z	2	4C	1
INB	$IBport \rightarrow A$	-, Z	2	CC	1
INCA	$A+1 \rightarrow A$	C, Z	4	42	1
INCB	$B+1 \rightarrow B$	C, Z	4	C2	1
INCI	$I+1 \rightarrow I$	C, Z	4	40	1
INCJ	$J+1 \rightarrow J$	C, Z	4	CO	1
INCK	$K+1 \rightarrow K$	C, Z	4	48	1
INCL	$L+1 \rightarrow L$	C, Z	4	C8	1
INCM ^v	$M+1 \rightarrow M$	C, Z	4	4A	1
INCN ^v	$N+1 \rightarrow N$	C, Z	4	CA	1
INCP	$P+1 \rightarrow P$	-, -	2	50	1
IY	$Y+1 \rightarrow Y, Y \rightarrow DP$	-, -	6	06	1
IYS	$Y+1 \rightarrow Y, Y \rightarrow DP, A \rightarrow (DP)$	-, -	6	26	1
IX	$X+1 \rightarrow X, X \rightarrow DP$	-, -	6	04	1
IXL	$X+1 \rightarrow X, X \rightarrow DP, (DP) \rightarrow A$	-, -	7	24	1
JPnm	$n \rightarrow PCH, m \rightarrow PCL$	-, -	6	79	3
JPCnm	Falls C=1: $n \rightarrow PCH, m \rightarrow PCL$ Falls C=0: $PC+3 \rightarrow PC$	-, -	6	7F	3
JPNCnm	Falls C=0: $n \rightarrow PCH, m \rightarrow PCL$ Falls C=1: $PC+3 \rightarrow PC$	-, -	6	7D	3
JPNZnm	Falls Z=0: $n \rightarrow PCH, m \rightarrow PCL$ Falls Z=1: $PC+3 \rightarrow PC$	-, -	6	7C	3
JPZnm	Falls Z=1: $n \rightarrow PCH, m \rightarrow PCL$ Falls Z=0: $PC+3 \rightarrow PC$	-, -	6	7E	3
JRCMn	Falls C=1: $PC+1-n \rightarrow PC$ Falls C=0: $PC+2 \rightarrow PC$	-, -	7/4	3B	2

Mnem.	Funktion	Flags	Zyklen	Code	Byte
JRCPn	Falls C=1: PC+1+n→PC Falls C=0: PC+2→PC	-, -	7/4	3A	2
JRM	PC+1-n→PC	-, -	7	2D	2
JRNCMn	Falls C=0: PC+1-n→PC Falls C=1: PC+2→PC	-, -	7/4	2B	2
JRNCPn	Falls C=0: PC+1+n→PC Falls C=1: PC+2→PC	-, -	7/4	2A	2
JRNZMn	Falls Z=0: PC+1-n→PC Falls Z=1: PC+2→PC	-, -	7/4	29	2
JRNZPn	Falls Z=0: PC+1+n→PC Falls Z=1: PC+2→PC	-, -	7/4	28	2
JRPn	PC+1+n→PC	-, -	7	2C	2
JRZMn	Falls Z=1: PC+1-n→PC Falls Z=0: PC+2→PC	-, -	7/4	39	2
JRZPn	Falls Z=1: PC+1+n→PC Falls Z=0: PC+2→PC	-, -	7/4	38	2
LEAVE	O→(R)	-, -	2	D8	1
LDD	(DP)→A	-, -	3	57	1
LDM	(P)→A	-, -	2	59	1
LDP	P→A	-, -	2	20	1
LDQ	Q→A	-, -	2	21	1
LDR	R→A	-, -	2	22	1
LIAn	n→A	-, -	4	02	2
LIBn	n→B	-, -	4	03	2
LIDLn	n→DPL	-, -	5	11	2
LIDPnm	n→DPH, m→DPL	-, -	8	10	3
LIIn	n→I	-, -	4	00	2
LIJn	n→J	-, -	4	01	2
LIPn	n→P	-, -	4	12	2
LIQn	n→Q	-, -	4	13	2
LOOPn	(R)-1→(R) Falls C=0: PC+1-n→PC Falls C=1: PC+2→PC, R+1→R	C, Z	10/7	2F	2
LPℓ	ℓ→P	-, -	2	80+ℓ	1
MVB	(Q)→(P)	-, -	5+2d	0A	1

Mnem.	Funktion	Flags	Zyklen	Code	Byte
MVBD	(DP)→(P)	-, -	5+4d	1A	1
MVDM	(P)→(DP)	-, -	3	53	1
MVMD	(DP)→(P)	-, -	3	55	1
MVW	(Q)→(P)	-, -	5+2d	08	1
MVWD	(DP)→(P)	-, -	5+4d	18	1
NOPT	No operation	-, -	3	CE	1
NOPW	No operation	-, -	2	4D	1
ORIAN	A ∨ n→A	-, Z	4	65	2
ORIDn	(DP) ∨ n→(DP)	-, Z	6	D5	2
ORIMn	(P) ∨ n→(P)	-, Z	4	61	2
ORMA	(P) ∨ A→(P)	-, Z	3	47	1
OUTA	(5C)→IAport	-, -	3	5D	1
OUTB	(5D)→IBport	-, -	2	DD	1
OUTF	(5E)→FOport	-, -	3	5F	1
OUTC	(5F)→Control	-, -	2	DF	1
POP	(R)→A, R+1→R	-, -	2	5B	1
PUSH	R-1→R, A→(R)	-, -	3	34	1
RC	0→C, 1→Z	C, Z	2	D1	1
READ	siehe unbek. Befehle	-, -	? 3	56	1
READM	siehe unbek. Befehle	-, -	? 3	54	1
RTN	(R)→PCL, (R+1)→PCH R+2→R	-, -	4	37	1
SBB	(P+1, P)-(B, A)→(P+1, P)	C, Z	5	15	1
SBCM	(P)-A-C→(P)	C, Z	3	C5	1
SBIAn	A-n→A	C, Z	4	75	2
SBIMn	(P)-n→(P)	C, Z	4	71	2
SBM	(P)-A→(P)	C, Z	3	45	1
SBN	(P)-A→(P), (BCD)	C, Z	7+3d	0D	1
SBW	(P)-(Q)→(P), (BCD)	C, Z	7+3d	0F	1
SC	1→C, 1→Z	C, Z	2	D0	1
SL	1 bit shift left	C, Z	2	5A	1
SLW	4 bit shift left	-, -	5+d	1D	1
SR	1 bit shift right	C, -	2	D2	1
SRW	4 bit shift right	-, -	5+d	1C	1
STD	A→(DP)	-, -	2	52	1

Mnem.	Funktion	Flags	Zyklen	Code	Byte
STP	A→P	-, -	2	30	1
STQ	A→Q	-, -	2	31	1
STR	A→R	-, -	2	32	1
SWP	A1bisA4↔A5bisA8	-, -	2	58	1
TESTn	n→Test	-, Z	4	6B	2
TSIAN	A ∧ n→Z	-, Z	4	66	2
TSIDn	(DP) ∧ n→Z	-, Z	6	D6	2
TSIMn	(P) ∧ n→Z	-, Z	4	62	2
WAITn	No operation	-, -	6+n	4E	2
WRIT	siehe unbek. Befehle	-, -	? 1	D3	1



SHARP
Option CE-125

IV DIE BASIC-BEFEHLE PEEK, POKE, CSAVE M, CLOAD M, CALL

Die im Basic-Interpreter vorhandenen Befehle PEEK, POKE, CALL, CSAVE M und CLOAD M gestatten die Eingabe von Maschinenprogrammen, ihren Aufruf und ihre Speicherung auf Cassette.

PEEK

PEEK numerischer Ausdruck

Es wird der Inhalt der Speicherzelle aufgerufen, deren Adresse durch den numerischen Ausdruck gegeben ist.

POKE

POKE numerischer Ausdruck 1, numerischer Ausdruck 2, numerischer Ausdruck 3,.....

Beginnend bei der Speicheradresse, die durch den numerischen Ausdruck 1 gegeben ist, werden die folgenden numerischen Ausdrücke als Daten in den Speicher geschrieben.

CALL

CALL numerischer Ausdruck

Es wird das Maschinenprogramm gestartet, dessen Anfangsadresse durch den numerischen Ausdruck gegeben ist. Nach einem RTN-Befehl in dem Maschinenprogramm erfolgt die Rückkehr zum Basic-Interpreter.

CSAVE M

CSAVE M Textausdruck; numerischer Ausdruck 1, numerischer Ausdruck 2

Der Textausdruck ist nicht notwendig.

Das Maschinenprogramm, dessen Anfangsadresse durch den numerischen Ausdruck 1 und dessen Endadresse durch den numerischen Ausdruck 2 gegeben ist, wird unter dem durch den Textausdruck gegebenen Namen auf Cassette gespeichert.

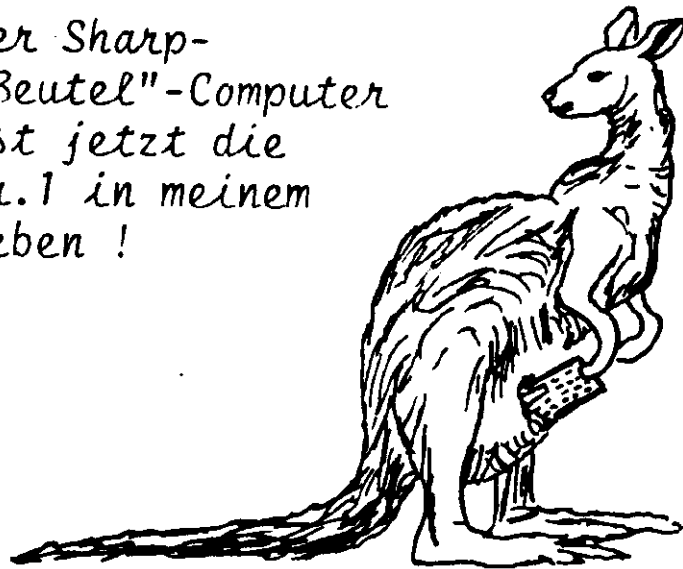
CLOAD M

CLOAD M Textausdruck; numerischer Ausdruck

Der Textausdruck ist nicht notwendig. D.h. es kann ein Programm, auch wenn es unter einem Namen auf Cassette abgespeichert wurde, ohne Angabe des Namen wieder in den Computer eingelesen werden. Wird jedoch ein anderer Name angegeben, so wird das Programm nicht eingelesen.

Der numerische Ausdruck ist nicht unbedingt notwendig. Soll das Programm in den gleichen Speicherbereich eingelesen werden, von dem aus es auf Cassette geschrieben worden ist, braucht der numerische Ausdruck nicht angegeben werden. Nur wenn in einen anderen Speicherbereich eingelesen werden soll, muß dessen Anfangsadresse angegeben werden.

Der Sharp-
"Beutel"-Computer
ist jetzt die
Nr.1 in meinem
Leben !



V BASIC-MASCHINENPROGRAMME

Es werden einige Programme beschrieben, die entweder ganz in Basic oder teils in Basic, teils in Maschinensprache geschrieben sind.

V.1 Disassembler

Für die von Sharp für die ESR-H CPU bekanntgegebene Maschinensprache wird hier ein Disassembler gegeben.

Er ist im wesentlichen in Maschinensprache geschrieben und hat somit auch Zugriff auf das mit der CPU verbundene Kern-ROM (&0000 bis &1FFF), was über den Basic-Befehl PEEK nicht möglich ist. Lediglich die Eingabe und die Ausgabe (Anzeige oder Ausdrucken) geschieht mit Basic-Befehlen. Dadurch ist dieser Teil leicht eigenen Wünschen anzupassen.

Die Ausgabe geschieht über die Standard-Variablen. Z\$ enthält die Hexadresse, Y\$ den Hexcode, M\$(Code) das Befehlswort (Mnemonic), X\$ das Einbyte- oder Zweibyte-Immediate und W\$ das Sprungziel bei relativen Sprungbefehlen. Das Ende des Programmes (laufende Adresse größer als die Endadresse) ist durch V = 1 und ein relativer Sprungbefehl (oder LOOP) durch U = 1 gekennzeichnet. Man kann also z.B. den Ausdruck des Hexcode verhindern, indem man Y\$ aus der Printliste streicht, oder die Sprungzieladresse eines relativen Sprungbefehls in einer zweiten Zeile ausdrucken lassen, indem man U abfragt.

Die Liste der Befehls Worte wird in der Feldvariablen M\$(255)*5 benötigt. Als erstes muß man also das später gegebene Mnemonic-Eingabe-Programm eingeben und mit DEF F starten. Für den PC-1261 lassen sich alle Mnemonic mit dem Basicprogramm (Zeile 3010 bis 3400) in einem Ablauf eingeben. Der Ablauf der Eingabe der Daten in M\$ ist auf der Anzeige zu verfolgen. Beim PC-1260 steht weniger Spei-

cher zur Verfügung, so daß das Programm etwas gekürzt werden muß. (Änderung der Basic-Zeilen 3020, 3350, 3400.)

```
3020: FOR N = 0 TO 252
```

```
3350: DATA "CAL18","CAL19","CAL1A","CAL1B","CAL1C":END
```

3400 entfällt.

Nach dem Start mit DEF F und dem Ablauf der Eingabe in M\$ gibt man die drei letzten Mnemonic über die Tastatur ein:

```
M$(253) = "CAL1D"
```

```
M$(254) = "CAL1E"
```

```
M$(255) = "CAL1F".
```

Nach Beendigung der Eingabe der M\$-Daten werden sie mit PRINT#M\$(*) auf Cassette geschrieben. Auch das Mnemonic-Eingabe-Programm sollte man sich mit CSAVE auf Cassette schreiben, um es später eventuell ändern zu können. Z.B. kann man einige unbekannte Befehle einfügen.

Nun kann man den später gegebenen Disassembler (Basic-Zeile 610 bis 990) eingeben. Startet man mit DEF M, so muß die Cassette mit den PRINT#M\$(*)-Daten zum Laden bereit stehen. Nachdem das M\$-Feld mit den Mnemonic gefüllt ist, kann man mit DEF D zur Anzeige und mit DEF L zum Ausdrucken starten.

SOFTWARE-RECHT

Die Bestimmungen des Urheber- und Wettbewerbsrechts für Computer-Programme

(Eine Pflichtlektüre für alle, die Software herstellen, kaufen oder verkaufen)

Berlin 1984

Dr. Roger Dorsch
Bernd Fischer

ISBN 3-924327-03-3

Disassembler-Tastenfunktionen

DEF M Start unter Bereitstellung des Variablenfeldes
M\$ auf Cassette.

DEF D Eingabe der Anfangsadresse. Anzeige von Hexadresse,
Hexcode des Befehls, Befehlswort, Immediate und
Sprungziel bei relativen Sprungbefehlen.

DEF L Eingabe von Anfangs- und Endadresse. Ausdruck von
Hexadresse, Hexcode, Mnemonic, Immediate und
Sprungziel bei relativen Sprungbefehlen.

Taste =

Durchlaufende Anzeige/Ausdruck

Taste -

Anhalten der Anzeige/Ausdruck

Taste +

Nächsten Befehl anzeigen/ausdrucken

Taste M

Abfrage einer neuen Anfangs- (Endadresse)

Taste .

Ende. Rückkehr zum Basic-Interpreter

Nach ca. 5 Minuten Wartestellung: END

MNEMONIC-EINGABE-
PROGRAMM FUER PC-1261.

FUER PC-1260 MUSS DAS
PROGRAMM ETWAS GEKUERZT
WERDEN. SIEHE TEXT.

3010:"F": CLEAR : DIM M
\$(255)*5: WAIT 0
3020:FOR N=0 TO 255
3030:READ M\$(N): PRINT
M\$(N)!"<": NEXT N
3040:DATA "LII ", "LIJ "
", "LIA ", "LIB "
", "IX", "DX", "IY", "D
Y"
3050:DATA "MVW", "EXW", "
MVB", "EXB", "ADN", "
SBM", "ADW", "SBW"
3060:DATA "LIDP ", "LIDL
", "LIP ", "LIQ "
", "ADB", "SBB", "16",
"17"
3070:DATA "MVWD", "EXWD"
", "MVBD", "EXBD", "SR
W", "SLW", "FILM", "F
ILD"
3080:DATA "LDP", "LDQ", "
LDR", "23", "IXL", "D
XL", "IYS", "DYS"
3090:DATA "JRNZP", "JRNZ
M", "JRNCP", "JRNCM"
", "JRP ", "JRM ", "
2E", "LOOP "
3100:DATA "STP", "STQ", "
STR", "33", "PUSH", "
DATA", "36", "RTN"
3110:DATA "JRZP ", "JRZM
", "JRCP ", "JRCM "
", "3C", "3D", "3E", "3
F"
3120:DATA "INCI", "DECI"
", "INCA", "DECA", "AD
M", "SBM", "ANMA", "O
RMA"
3130:DATA "INCK", "DECK"
", "4A", "4B", "INA", "
NOPW", "WAIT ", "4F"
3140:DATA "INCP", "DECP"
", "STD", "MVDM", "REA
DM", "MVMD", "READ", "
"LDD"
3150:DATA "SWP", "LDM", "
SL", "POP", "5C", "OU
TA", "5E", "OUTF"
3160:DATA "ANIM ", "ORIM
", "TSIM ", "CPIM "
", "ANIA ", "ORIA ", "
TSA ", "CPIA "

3170:DATA "68", "69", "6A
", "TEST ", "6C", "6D
", "6E", "6F"
3180:DATA "ADIM ", "SBIM
", "72", "73", "ADIA
", "SBIA ", "76", "7
7"
3190:DATA "CALL ", "JP
", "7A", "7B", "JPNZ
", "JPNC ", "JPZ "
", "JPC "
3200:DATA "LP00", "LP01"
", "LP02", "LP03", "LP
04", "LP05", "LP06", "
"LP07"
3210:DATA "LP08", "LP09"
", "LP0A", "LP0B", "LP
0C", "LP0D", "LP0E", "
"LP0F"
3220:DATA "LP10", "LP11"
", "LP12", "LP13", "LP
14", "LP15", "LP16", "
"LP17"
3230:DATA "LP18", "LP19"
", "LP1A", "LP1B", "LP
1C", "LP1D", "LP1E", "
"LP1F"
3240:DATA "LP20", "LP21"
", "LP22", "LP23", "LP
24", "LP25", "LP26", "
"LP27"
3250:DATA "LP28", "LP29"
", "LP2A", "LP2B", "LP
2C", "LP2D", "LP2E", "
"LP2F"
3260:DATA "LP30", "LP31"
", "LP32", "LP33", "LP
34", "LP35", "LP36", "
"LP37"
3270:DATA "LP38", "LP39"
", "LP3A", "LP3B", "LP
3C", "LP3D", "LP3E", "
"LP3F"
3280:DATA "INCJ", "DECJ"
", "INCB", "DECB", "AD
CM", "SBCM", "C6", "C
PMA"
3290:DATA "INCL", "DECL"
", "CA", "CB", "INB", "
CD", "NOPT", "CF"
3300:DATA "SC", "RC", "SR
", "WRIT", "ANID ", "
ORID ", "TSID ", "D7
"
3310:DATA "LEAVE", "D9", "
"EXAB", "EXAM", "DC"
", "OUTB", "DE", "OUTC
"

3320:DATA "CAL00", "CAL0
1", "CAL02", "CAL03"
", "CAL04", "CAL05", "
CAL06", "CAL07"
3330:DATA "CAL08", "CAL0
9", "CAL0A", "CAL0B"
", "CAL0C", "CAL0D", "
CAL0E", "CAL0F"
3340:DATA "CAL10", "CAL1
1", "CAL12", "CAL13"
", "CAL14", "CAL15", "
CAL16", "CAL17"
3350:DATA "CAL18", "CAL1
9", "CAL1A", "CAL1B"
", "CAL1C", "CAL1D", "
CAL1E", "CAL1F"
3400:END

```

PC-1260/61 DISASSEMBLER
610:"M": PRINT "LADE M$(
255)"
620: CLEAR : DIM M$(255)*
5: INPUT "M$(*)":
GOTO 640
630:"L": PRINT = LPRINT
:L=1:K=1:P=1: GOTO 6
50
640:"D": PRINT = PRINT :
L=0:K=0:P=1
650: INPUT "ANF-ADR " : A:E
=&FFFF
660: IF L=1 INPUT "END-AD
R " : E
670: Z$="ABCD": Y$="AB": X$
=Z$: W$=Z$: B= INT (A/
256): C=A-B*256: D=
INT (E/256): E=E-D*25
6
680: POKE &650C,C,B,E,D
700: IF P=0 GOTO 900
705: A=&5EE0: B=&65: C=&0C:
G= INT ((A+&B6)/256)
: H=A+&B6-G*256
710: POKE A,16,B,C+13,2,0
,0,22,31,17,C+5,0,3,
31,17,C,&82
720: POKE A+&10,24,17,C,2
4,&DA,&85,&C7,&DA,40
,3,&84,&C7,42,7,17,C
+22
730: POKE A+&20,2,16,82,5
5,&88,53,52,&DA,17,C
-11,&78,G,H,91,17,C-
9
740: POKE A+&30,&78,G,H,&
88,89,17,C-5,82,52,1
7,C-3,&78,G,H,&5B,52
750: POKE A+&40,&64,3,66,
&80,&DB,2,H,&84,&DB,
2,G,&85,&DB,91,&D1,&
D2
760: POKE A+&50,&D1,&D2,&
74,25,3,0,&84,20,36,
44,3,&D2,&D2,65,41,4
770: POKE A+&60,&64,3,6,&
67,0,56,12,52,6,&89,
89,16,B,C+5,&78,G
780: POKE A+&70,H,91,&67,
2,40,11,52,&8A,89,17
,C+7,&78,G,H,6,91
790: POKE A+&80,0,1,&86,1
6,B,C,25,&67,3,40,43
,6,2,32,3,62
800: POKE A+&90,16,B,C+13
,&82,27,&88,89,&D2,&
89,89,3,0,&86,58,4,2
0

```

```

810: POKE A+&A0,44,2,21,8
9,17,C+15,&78,G,H,&8
6,89,17,C+17,&78,G,H
820: POKE A+&B0,2,16,17,C
+30,82,55,0,2,52,&DA
,91,44,3,&DA,88,&64
830: POKE A+&C0,15,&67,10
,58,3,&74,7,&74,48,6
5,41,14,&82,64,25,55
840: POKE A+&D0,85,0,0,0,
86,0,0,0,0,0,&FF,&CF
,0,0,&FF,0,0,0,0,16,
0,0,0,0
850: POKE A+&E0,85,85,64,
0,5,5,10,&AA,0,0,0,0
,0,0,0,0,0,0,0,0,0,0
,0,0
860: POKE A+&100,0,0,0,0,
0,21,0,0,85,85,85,85
,85,85,85,85:P=0
900:A=&5EE0: WAIT 20
910: CALL A: IF V=1 END
920: PRINT Z$;" " : Y$;" "
: M$( PEEK &6507);" "
: X$;W$:T=0
930: I$= INKEY$ : J= ASC I
$: IF J=46 END
940: IF J=77 GOTO 650
950: IF J=43 LET K=0:
GOTO 910
960: IF J=45 LET K=0
970: IF J=61 LET K=1
980: IF K=1 GOTO 910
985: T=T+1: IF T>1260 END
990: GOTO 930

```

DAS MASCHINENPROGRAMM,
DAS IN DEN BASIC-ZEILEN
710 BIS 860 EINGEPOKT
WIRD, IST AUF DER FOLGEN
DEN SEITE ABGEDRUCKT.

MASCHINENPROGRAMM ZUM
PC-1260/61 DISASSEMBLER
&5EE0 BIS &5FEF

5EE0: 10 LIDP 6519
5EE3: 02 LIA 00
5EE5: 00 LII 16
5EE7: 1F FILD
5EE8: 11 LIDL 11
5EEA: 00 LII 03
5EEC: 1F FILD
5EED: 11 LIDL 0C
5EEF: 82 LP02
5EF0: 18 MVWD
5EF1: 11 LIDL 0C
5EF3: 18 MVWD
5EF4: 0A EXAB
5EF5: 85 LP05
5EF6: 07 CPMA
5EF7: 0A EXAB
5EF8: 28 JRNZP 03 >5EFC
5EFA: 84 LP04
5EFB: 07 CPMA
5EFC: 2A JRNCP 07 >5F04
5EFE: 11 LIDL 22
5F00: 02 LIA 10
5F02: 52 STD
5F03: 37 RTN
5F04: 88 LP08
5F05: 35 DATA
5F06: 34 PUSH
5F07: 0A EXAB
5F08: 11 LIDL 01
5F0A: 78 CALL 5F96
5F0D: 5B POP
5F0E: 11 LIDL 03
5F10: 78 CALL 5F96
5F13: 88 LP08
5F14: 59 LDM
5F15: 11 LIDL 07
5F17: 52 STD
5F18: 34 PUSH
5F19: 11 LIDL 09
5F1B: 78 CALL 5F96
5F1E: 5B POP
5F1F: 34 PUSH
5F20: 64 ANIA 03
5F22: 42 INCA
5F23: 80 LP00
5F24: 0B EXAM
5F25: 02 LIA 96
5F27: 84 LP04
5F28: 0B EXAM
5F29: 02 LIA 5F
5F2B: 85 LP05
5F2C: 0B EXAM
5F2D: 5B POP
5F2E: 01 RC
5F2F: 02 SR
5F30: 01 RC
5F31: 02 SR

5F32: 74 ADIA 19
5F34: 03 LIB 00
5F36: 84 LP04
5F37: 14 ADB
5F38: 24 IXL
5F39: 2C JRP 03 >5F3D
5F3B: 02 SR
5F3C: 02 SR
5F3D: 41 DECI
5F3E: 29 JRNZM 04 >5F3B
5F40: 64 ANIA 03
5F42: 06 IY
5F43: 67 CPIA 00
5F45: 38 JRZP 0C >5F52
5F47: 34 PUSH
5F48: 06 IY
5F49: 89 LP09
5F4A: 59 LDM
5F4B: 10 LIDP 6511
5F4E: 78 CALL 5F96
5F51: 5B POP
5F52: 67 CPIA 02
5F54: 28 JRNZP 0B >5F60
5F56: 34 PUSH
5F57: 8A LP0A
5F58: 59 LDM
5F59: 11 LIDL 13
5F5B: 78 CALL 5F96
5F5E: 06 IY
5F5F: 5B POP
5F60: 00 LII 01
5F62: 86 LP06
5F63: 10 LIDP 650C
5F66: 19 EXWD
5F67: 67 CPIA 03
5F69: 28 JRNZP 2B >5F95
5F6B: 06 IY
5F6C: 02 LIA 20
5F6E: 03 LIB 3E
5F70: 10 LIDP 6519
5F73: 82 LP02
5F74: 1B EXBD
5F75: 88 LP08
5F76: 59 LDM
5F77: 02 SR
5F78: 89 LP09
5F79: 59 LDM
5F7A: 03 LIB 00
5F7C: 86 LP06
5F7D: 3A JRCP 04 >5F82
5F7F: 14 ADB
5F80: 2C JRP 02 >5F83
5F82: 15 SBB
5F83: 59 LDM
5F84: 11 LIDL 1B
5F86: 78 CALL 5F96
5F89: 86 LP06
5F8A: 59 LDM
5F8B: 11 LIDL 1D
5F8D: 78 CALL 5F96
5F90: 02 LIA 10

5F92: 11 LIDL 2A
5F94: 52 STD
5F95: 37 RTN
5F96: 00 LII 02
5F98: 34 PUSH
5F99: 0A EXAB
5F9A: 5B POP
5F9B: 2C JRP 03 >5F9F
5F9D: 0A EXAB
5F9E: 58 SWP
5F9F: 64 ANIA 0F
5FA1: 67 CPIA 0A
5FA3: 3A JRCP 03 >5FA7
5FA5: 74 ADIA 07
5FA7: 74 ADIA 30
5FA9: 41 DECI
5FAA: 29 JRNZM 0E >5F9D
5FAC: 82 LP02
5FAD: 40 INCI
5FAE: 19 EXWD
5FAF: 37 RTN
5FB0: 55 MVMD
5FB1: 00 LII 00
5FB3: 00 LII 56
5FB5: 00 LII 00
5FB7: 00 LII 00
5FB9: 00 LII FF
5FBB: 0F CF
5FBC: 00 LII 00
5FBE: 0F CAL1F 00
5FC0: 00 LII 00
5FC2: 00 LII 10
5FC4: 00 LII 00
5FC6: 00 LII 00
5FC8: 55 MVMD
5FC9: 55 MVMD
5FCA: 40 INCI
5FCB: 00 LII 05
5FCD: 05 DX
5FCE: 0A MVB
5FCF: 0A LP2A
5FD0: 00 LII 00
5FD2: 00 LII 00
5FD4: 00 LII 00
5FD6: 00 LII 00
5FD8: 00 LII 00
5FDA: 00 LII 00
5FDC: 00 LII 00
5FDE: 00 LII 00
5FE0: 00 LII 00
5FE2: 00 LII 00
5FE4: 00 LII 15
5FE6: 00 LII 00
5FE8: 55 MVMD
5FE9: 55 MVMD
5FEA: 55 MVMD
5FEB: 55 MVMD
5FEC: 55 MVMD
5FED: 55 MVMD
5FEE: 55 MVMD
5FEF: 55 MVMD

V.2 Hexcode/Zeichen

Die Anzeige oder der Ausdruck von je vier Hexcode und der zugehörigen Zeichen geschieht mit einem reinen Basicprogramm über den PEEK-Befehl.

Tastenfunktionen

DEF H Eingabe der Anfangsadresse. Anzeige von 4 Hexcode und der zugehörigen Zeichen.

DEF J Eingabe der Anfangs- und Endadresse. Ausdruck von 4 Hexcode und der zugehörigen Zeichen.

Taste = Durchlaufende Anzeige/Ausdruck

Taste - Anhalten der Anzeige/Ausdruck

Taste + Nächste Zeile anzeigen/ausdrucken

Taste M Abfrage einer neuen Anfangs- (Endadresse)

Taste . Ende. Rückkehr zum Basic-Interpreter

Nach ca. 5 Minuten Wartestellung: END

PC-1260/61
HEXCODE/ZEICHEN

```
1010:"J":0=1: PRINT =          1050:H= PEEK (I+N-1):        1120:IF F=45 LET G=0
      LPRINT : GOTO 1014        GOSUB 1200:A$(N)=K          1130:IF G=1 LET I=I+4:
1012:"H":0=0: PRINT =          $+" "                      GOTO 1020
      PRINT                    1055:IF ((H)&F4) AND (H          1140:IF F=43 LET I=I+4:
1014:INPUT "ANF-ADR ":I        <&F9)> OR (H=&FB)          GOTO 1020
      :J=&FFFF:G=0: WAIT        OR (H=&FC) GOTO 10        1150:T=T+1: IF T>750
      0                          65                                END
1016:IF Q=1 INPUT "END-        1060:IF (H<&20) OR (H)&        1160:GOTO 1080
ADR ":J:G=1                    7A) LET H=32                1200:P= INT (H/16):Q=H-
1020:H= INT (I/256):          1065:M$=M$+ CHR$ H:          P*16: IF P>9 LET P
      GOSUB 1200:L$=K$:H        NEXT N                      =P+7
      =I-H*256: GOSUB 12        1070:PRINT L$:A$:B$:C$:        1210:IF Q>9 LET Q=Q+7
      00:L$=L$+K$+" ":T        D$:M$                        1220:P=P+48:Q=Q+48:K$=
      =0                          1080:N$= INKEY$ :F= ASC        CHR$ P+ CHR$ Q:
1030:M$="": IF I>J            N$                                RETURN
      PRINT = PRINT :          1090:IF F=77 GOTO 1014
      END
1040:FOR N=1 TO 4            1100:IF F=46 END
                                1110:IF F=61 LET G=1
```

VI PROGRAMMIERBEISPIELE

Das Basicprogramm beginnt beim PC-1260 bei &5880 und beim PC-1261 bei &4080, falls nicht mit EQU# n der Speicherbereich für das EASY SIMULATION PROGRAMMING erweitert wurde. Die Feldvariablen werden von &64FF an abwärts aufgebaut. Der Speicherbereich zwischen den Basicprogrammen und den Feldvariablen ist für Maschinenprogramme geeignet. Vom System wird gehalten die Basicprogrammmanfangsadresse in &66E2/1, die Basicprogrammendadresse in &66E4/3 und die Anfangsadresse der Feldvariablen in &66FD/C. Von dort kann man also mit PEEK die Information über die Speicherbelegung erhalten.

Für die Erstellung eines Maschinenprogrammes skizziert man am besten zuerst das Ziel und ganz grob den Weg, wie man es erreichen will. Danach wird das Flußdiagramm entwickelt, das in Blöcken die einzelnen Funktionen, die Abfragen und die Verbindungen zwischen den Programmteilen darstellt. Dann werden die einzelnen Programmteile aus Maschinenbefehlen aufgebaut. Hierbei muß man nicht unbedingt in der Reihenfolge des Flußdiagramm vorgehen, sondern kann zentrale Teile oder Unterroutinen zuerst entwickeln. Häufig stellt man dabei fest, daß das Flußdiagramm nicht präzise genug war und daß es im Verlauf der Programmentwicklung verfeinert werden muß.

Stehen keine Programmierhilfen zur Verfügung, so muß man den POKE-Befehl zur Eingabe des Maschinencode benutzen. Am günstigsten ist es, ein Basicprogramm mit POKE-Befehlen aufzubauen und dies auf Cassette zu speichern. Für eine Änderung des Maschinenprogrammes stehen so die Edierfunktionen des Basic-Interpreters zur Verfügung. Im folgenden sind Beispiele für diese Vorgehensweise gegeben.

VI.1 Renumber

Dieses Maschinenprogramm versieht ein Basicprogramm mit Zeilennummern, die gleichen Abstand haben. Der erste Befehl lädt z.B. 10(dezimal) und bestimmt die erste Zeilennummer und den Abstand der Zeilennummern.

Das Basicprogramm darf nicht auf Zeilennummern springen (GOTO, GOSUB) sondern muß hierzu Label benutzen. Anderenfalls ist anschließend das Programm durchzusehen und die Sprungadressen anzupassen.

Das Programm basiert auf folgenden Gegebenheiten:

Der Beginn des Basicprogramm wird vom System in den Speicherplätzen &66E2/1 gehalten. Im Basicprogrammspeicher steht am Beginn und Ende des Basicprogramm jeweils &FF. Hinter dem ersten &FF steht die Zeilennummer, für die jeweils 2 Byte zur Verfügung stehen. Danach ist in einem Byte die Länge der Basic-Zeile gegeben. Dahinter steht der codierte Zeileninhalt, der durch &OD abgeschlossen wird. Danach steht in den nächsten beiden Byte wieder die nächste Basic-Zeilennummer oder &FF, falls das Basicprogramm beendet ist.

Basicprogramm zur Eingabe von Renumber

```
4100:A=&5E00
```

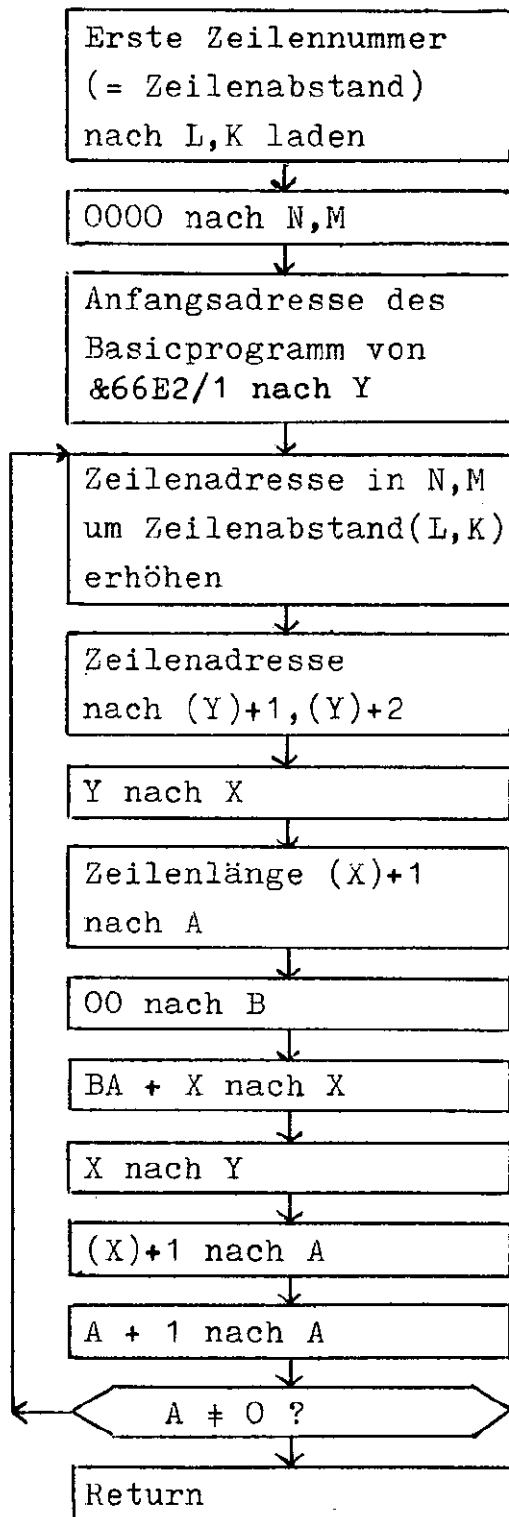
```
4110:POKE A,&02,&0A,&88,&DB,&02,&00,&89,&DB,&02,&00,&8A,  
      &00,&01,&1E,&10,&66
```

```
4120:POKE A+&10,&E1,&86,&18,&13,&08,&82,&08,&8A,&14,&13,  
      &0A,&82,&08,&DA,&26,&DA
```

```
4130:POKE A+&20,&26,&13,&06,&84,&08,&24,&03,&00,&84,&14,  
      &13,&04,&86,&08,&24,&42
```

```
4140:POKE A+&30,&29,&1E,&37:END
```

Flußdiagramm Renumber



Maschinenprogramm

```

5E00: 02 LIA 0A
5E02: 08 LP08
5E03: 0B EXAM
5E04: 02 LIA 00
5E06: 09 LP09
5E07: 0B EXAM
5E08: 02 LIA 00
5E0A: 0A LP0A
5E0B: 00 LII 01
5E0D: 1E FILM
5E0E: 10 LIDP 66E1
5E11: 06 LP06
5E12: 18 MVWD
5E13: 13 LIQ 08
5E15: 02 LP02
5E16: 08 MVW
5E17: 0A LP0A
5E18: 14 ADB
5E19: 13 LIQ 0A
5E1B: 02 LP02
5E1C: 08 MVW
5E1D: 0A EXAB
5E1E: 26 IYS
5E1F: 0A EXAB
5E20: 26 IYS
5E21: 13 LIQ 06
5E23: 04 LP04
5E24: 08 MVW
5E25: 24 IXL
5E26: 03 LIB 00
5E28: 04 LP04
5E29: 14 ADB
5E2A: 13 LIQ 04
5E2C: 06 LP06
5E2D: 08 MVW
5E2E: 24 IXL
5E2F: 42 INCA
5E30: 29 JRNZM 1E >5E13
5E32: 37 RTN
  
```

Ausbau von Renumber

Dieses einfache Renumberprogramm läßt sich natürlich verbessern.

Eine Verbesserung wäre, den Zeilenabstand und die erste Zeilennummer voneinander zu trennen, so daß beide unabhängig voneinander vorgegeben werden können. Dies läßt sich sicherlich leicht durchführen.

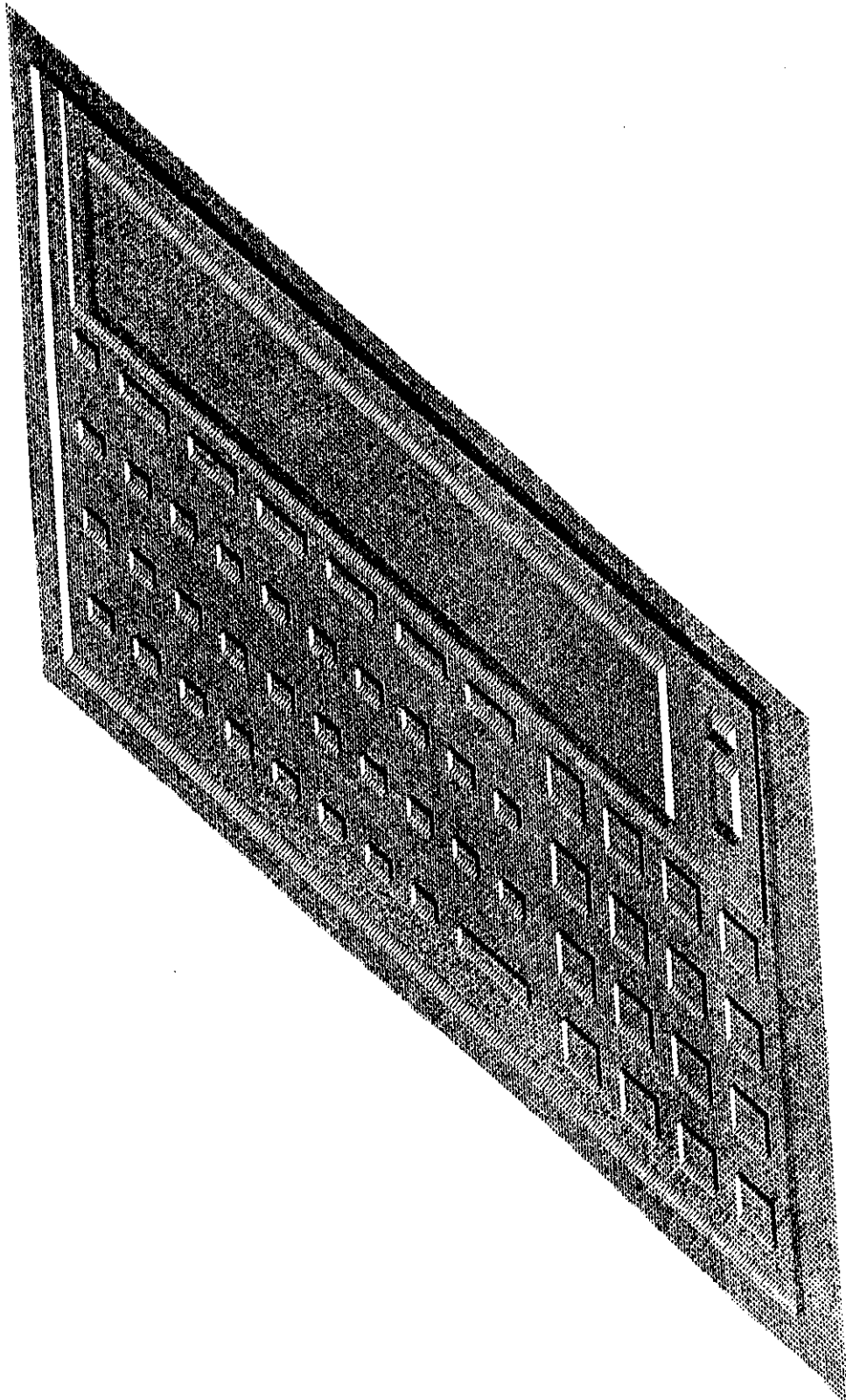
Eine weitere Verbesserung wäre, auch GOTO und GOSUB auf Zeilennummern vom Programm berücksichtigen zu lassen. Dies ist schon eine schwierigere Aufgabe.

Dies läßt sich durchführen, indem bei jeder Umrechnung einer Zeilennummer von alt auf neu jeweils das gesamte Basicprogramm durchgesehen wird nach GOTO/GOSUB auf die alte Zeilennummer und, falls vorhanden, auch bei diesem GOTO/GOSUB die Sprungadresse von alt auf neu umgeschrieben wird. Die Code für GOTO/GOSUB findet man in der Token-Tabelle.

Der Aufbau des Basicprogramm ist vorangehend beschrieben. Genauere Information erhält man, indem man das Hexcode/Zeichen-Programm des vorangehenden Kapitels lädt und es sich von Beginn an (Basicprogrammbeginn in &66E2/1) ansieht. Hier gibt es bereits in der ersten Zeile ein GOTO. Man stellt dabei fest, daß die Zeilennummer am Anfang der Zeile in zwei Byte gegeben ist, während die Zeilennummer hinter GOTO/GOSUB in ASCII-Zeichen (also Null = &30, Eins = &31, usw.) gegeben ist. Es ist also darauf zu achten, daß vom Renumber-Maschinenprogramm die Zeilennummer jeweils in der richtigen Form eingeschrieben wird. Weiterhin ist zu beachten, daß sich die Anzahl der ASCII-Zeichen hinter GOTO/GOSUB ändern kann, was wiederum einen Einfluß auf die Zeilenlänge (Byte hinter der Zeilennummer) hat.

Eine Teilaufgabe, die man durch ein Unterprogramm erledigen lassen kann, ist die Umwandlung von ASCII-Zeichen in Hexstellen oder umgekehrt. Die Umwandlung eines Byte in

die beiden zugehörigen ASCII-Hexzeichen wird auch im Disassembler benötigt. Das zugehörige Unterprogramm steht im Maschinensprache-Listing auf &5F96 bis &5FAF. Es wandelt das Byte aus dem Akkumulator um in zwei ASCII-Zeichen und schreibt diese nach DP, DP + 1.



VI.2 Breakpoint-Monitor

Ein Breakpoint-Monitor kann dazu benutzt werden, Maschinenprogramme auszutesten. Neben dem kommentierten Maschinenprogramm ist auch das Basic-Eingabe-Programm, das mit DEF = gestartet wird, gegeben.

Der Breakpoint-Monitor wird in dem zu testenden Maschinenprogramm an der Stelle, an der es unterbrochen werden soll, mit dem Maschinenbefehl CALL aufgerufen.

Vom Breakpoint-Monitor wird nach &5DF8/9 die Adresse gegeben, von der aus in dem zu testenden Maschinenprogramm der Monitor aufgerufen wurde. Danach folgen die Register:

```
5DF8: PH PL OO OO
FC: P Q R C,Z
5E00: I J A B
04: XL XH YL YH
08: K L M N
```

.....

.....

5E5C: Ports

Um den Inhalt der Register auf den Speicherplätzen &5DF8 bis &5E5F anzusehen, benötigt man also neben dem Breakpoint-Monitor und dem eigenen zu testenden Maschinenprogramm noch das 4 Hexcode/Zeichen-Programm des letzten Kapitels.

Beim PC-1261 steht genügend Speicher zur Verfügung, um auch den Disassembler zu laden, um das eigene zu testende Programm disassemblieren zu können. Hat man den Disassembler, das 4 Hexcode/Zeichen-Programm und das anschließend gegebene Basicprogramm zur Eingabe des Breakpoint-Monitor geladen, so ist dadurch z.B. der Speicherbereich von &4080 bis &4A00 belegt. Der Bereich von &5DF8 bis &64FF wird benötigt für den Register-Anzeige-Bereich, das Maschinenprogramm des Breakpoint-Monitor, das Maschinenprogramm des Disassemblers und die M\$(*)-Feldvariablen für

die Mnemonics des Disassemblers.

So verbleibt beim PC-1261 also der Bereich von &4A01 bis &5DF7 (4,9 KByte) für das Basicprogramm zum Einpoken des eigenen auszutestenden Maschinenprogramm und für dieses Maschinenprogramm selbst. Übrigens, das Ende des Basic-Bereiches wird vom System gehalten in &66E4/3.

Beim PC-1260 steht wesentlich weniger Speicherplatz zur Verfügung. So kann man hier den Disassembler nicht laden sondern nur das 4 Hexcode/Zeichen-Programm und das Basic-Programm zur Eingabe des Breakpoint-Monitor. Dadurch ist z.B. der Speicherbereich von &5880 bis &5C5B belegt.

Für ein Basicprogramm zum Einpoken eines eigenen auszutestenden Maschinenprogrammes steht also beim PC-1260 noch Platz von &5C5C bis &5DF7 zur Verfügung. Der Bereich von &5DF8 bis &5E5F wird zur Anzeige der Register benötigt und auf &5E60 bis &5ECC steht das Maschinenprogramm des Breakpoint-Monitor. So verbleibt für das eigene auszutestende Maschinenprogramm der Platz von &5EDO bis &64FF. Benötigt man mehr Platz für das Basicprogramm zum Einpoken des eigenen auszutestenden Maschinenprogramm, so muß man den Breakpoint-Monitor weiter nach hinten verlegen. Will man z.B. den Register-Anzeige-Bereich mit dem anschließenden Breakpoint-Maschinenprogramm von &5DF8 - &5ECC nach &61F8 - &62CC verlegen, so muß die Basiczeile 5000 geändert werden zu 5000: A = &6260: B = &62.

Basicprogramm zur Eingabe des Breakpoint-Monitor

```
5000: "=":A=&5E60:B=&5E
5010:POKE A,&34,&20,&10,B-1,&FC,&52,&21,&11,&FD,&52,&22,
      &34,&3A,9,2,1
5020:POKE A+16,&38,&B,2,0,&2C,7,2,&11,&38,3,2,&10,&11,
      &FF,&52,&11
5030:POKE A+32,&FE,&5B,&42,&42,&42,&52,&10,B,0,&80,&53,
      &11,1,&81,&53,&5B
5040:POKE A+48,&11,2,&82,&53,&11,3,&83,&53,&11,4,&84,&53,
      &11,5,&85,&53
5050:POKE A+64,2,5,&84,&DB,2,B,&85,&DB,2,&59,&34,4,&50,
      &53,&2F,4
5060:POKE A+80,&84,&5B,&DB,&85,&5B,&DB,5,5,5,&10,B-1,
      &F8,&53,&84,&11,&F9
5070:POKE A+96,&53,2,0,&11,&FA,&52,&11,&FB,&52,2,&56,&32,
      &37:END
```

Kommentiertes Maschinenprogramm zum Breakpoint-Monitor

```
5E60: 34 PUSH           A0 nach Stack
      61: 20 LDP           P nach A nach 5DFC
      62: 10 LIDP          5DFC
      65: 52 STD
      66: 21 LDQ           Q nach A nach 5DFD
      67: 11 LIDL          FD BF
      69: 52 STD
      6A: 22 LDR           R = R0 - 3 nach A nach Stack
      6B: 34 PUSH
      6C: 3A JRCP          09 >5E76 C,Z nach 5DFF
      6E: 02 LIA           01
      70: 38 JRZP          0B >5E7C
      72: 02 LIA           00
      74: 2C JRP           07 >5E7C
```

5E76: 02 LIA 11
78: 38 JRZP 03 >5E7C
7A: 02 LIA 10
7C: 11 LIDL FF 01
7E: 52 STD
7F: 11 LIDL FE 02 R₀ -3 nach A, R₀ nach 5DFE
81: 5B POP
82: 42 INCA
83: 42 INCA
84: 42 INCA
85: 52 STD
86: 10 LIDP 5E00 02 (00) = I nach 5E00
89: 80 LPO0
8A: 53 MVDM
8B: 11 LIDL 01 03 (01) = J nach 5E01
8D: 81 LPO1
8E: 53 MVDM
8F: 5B POP A₀ nach 5E02
90: 11 LIDL 02
92: 82 LPO2
93: 53 MVDM
94: 11 LIDL 03 01 (03) = B nach 5E03
96: 83 LPO3
97: 53 MVDM
98: 11 LIDL 04 02 (04) = XL nach 5E04
9A: 84 LPO4
9B: 53 MVDM
9C: 11 LIDL 05 01 (05) = XH nach 5E05
9E: 85 LPO5
9F: 53 MVDM
A0: 02 LIA 05 01 (06) bis (5F) nach 5E06 - 5E5F
A2: 84 LPO4
A3: DB EXAM
A4: 02 LIA 5E 01
A6: 85 LPO5
A7: DB EXAM

5EAB:	02	LIA	59	
AA:	34	PUSH		
AB:	04	IX		
AC:	50	INCF		
AD:	53	MVDM		
AE:	2F	LOOP	04	>5EAB
B0:	84	LPO4		Return-Adresse nach X
B1:	5B	POP		
B2:	DB	EXAM		
B3:	85	LPO5		
B4:	5B	POP		
B5:	DB	EXAM		
B6:	05	DX		X - 3 = Aufrufadresse nach
B7:	05	DX		5DF8/9
B8:	05	DX		
B9:	10	LIDP	5DF8	42 SA
BC:	53	MVDM		
BD:	84	LPO4		
BE:	11	LIDL	F9	23
CO:	53	MVDM		
C1:	02	LIA	00	00,00 nach 5DFA/B
C3:	11	LIDL	FA	2C
C5:	52	STD		
C6:	11	LIDL	FB	
C8:	52	STD		
C9:	02	LIA	56	56 nach R
CB:	32	STR		
5ECC:	37	RTN		Zurück zum Basic-Interpreter

VII DER BASIC-INTERPRETER

Der Basic-Interpreter befindet sich im ROM auf den Adressen &0000 bis &1FFF und &8000 bis &FFFF. Der Teil von &0000 bis &1FFF ist das mit der CPU verbundene Kern-ROM, auf das mit dem PEEK-Befehl nicht zugegriffen werden kann. Der Zugriff ist hier nur mit dem DATA-Maschinenbefehl möglich. Mit dem im Kapitel V gegebenen Disassembler kann das ROM (auch das Kern-ROM) angesehen oder ausgedruckt werden.

Im ROM des Basic-Interpreters befinden sich neben Befehlen auch zahlreiche Tabellen, z.B. die Listen mit denen die Tastencode in die ASCII-Code umgerechnet werden oder die Tokentabelle in der die Basic-Befehls Worte mit ihrem internen Code, den Sprungadressen und einem weiteren Code aufgelistet sind. Der letztgenannte Code enthält z.B. Informationen über den Befehlstyp und wird benutzt, um gegebenenfalls Fehlermeldungen zu erzeugen. Diese Tokentabelle ist im letzten Abschnitt dieses Kapitels gegeben.

Der Basic-Interpreter ist mit zahlreichen Unterprogrammen aufgebaut. Insbesondere die kurzen und häufig benutzten Unterprogramme befinden sich im allgemeinen im Kern-ROM. Zum im ROM (nur lesbarer Speicher) befindlichen Basic-Interpreter gehört noch ein RAM (lesbarer und beschreibbarer Speicher) in dem die Informationen über den augenblicklichen Zustand des Computers gehalten werden. Diese Adressen werden auch als Systemadressen bezeichnet.

VII.1 Untersuchung des System

Von den Unterprogrammen des Basic-Interpreters sind für die Programmierung in Maschinensprache besonders wichtig die Tastatur-, Anzeige- und Ausdruckroutinen.

Wie kann man an diese Information kommen, wenn nichts über das System, nicht einmal der Startpunkt bei RESET, bekannt ist.

Der Basic-Befehl CALL Adresse ruft ein Maschinenprogramm auf. Das Maschinenprogramm muß mit RTN abgeschlossen werden. Dieses RTN bewirkt unmittelbar den Rücksprung in den Basic-Interpreter. Die Rücksprungadresse muß sich also im Stack befinden. Läßt man ein Maschinenprogramm, das man über die Tastatur aufruft, den Wert des Stackpointer und den Inhalt des Stack in RAM-Plätze schreiben, so kann man sich diese Information mit dem PEEK-Befehl ansehen.

Für die neue Version, PC-1261, erhält man:

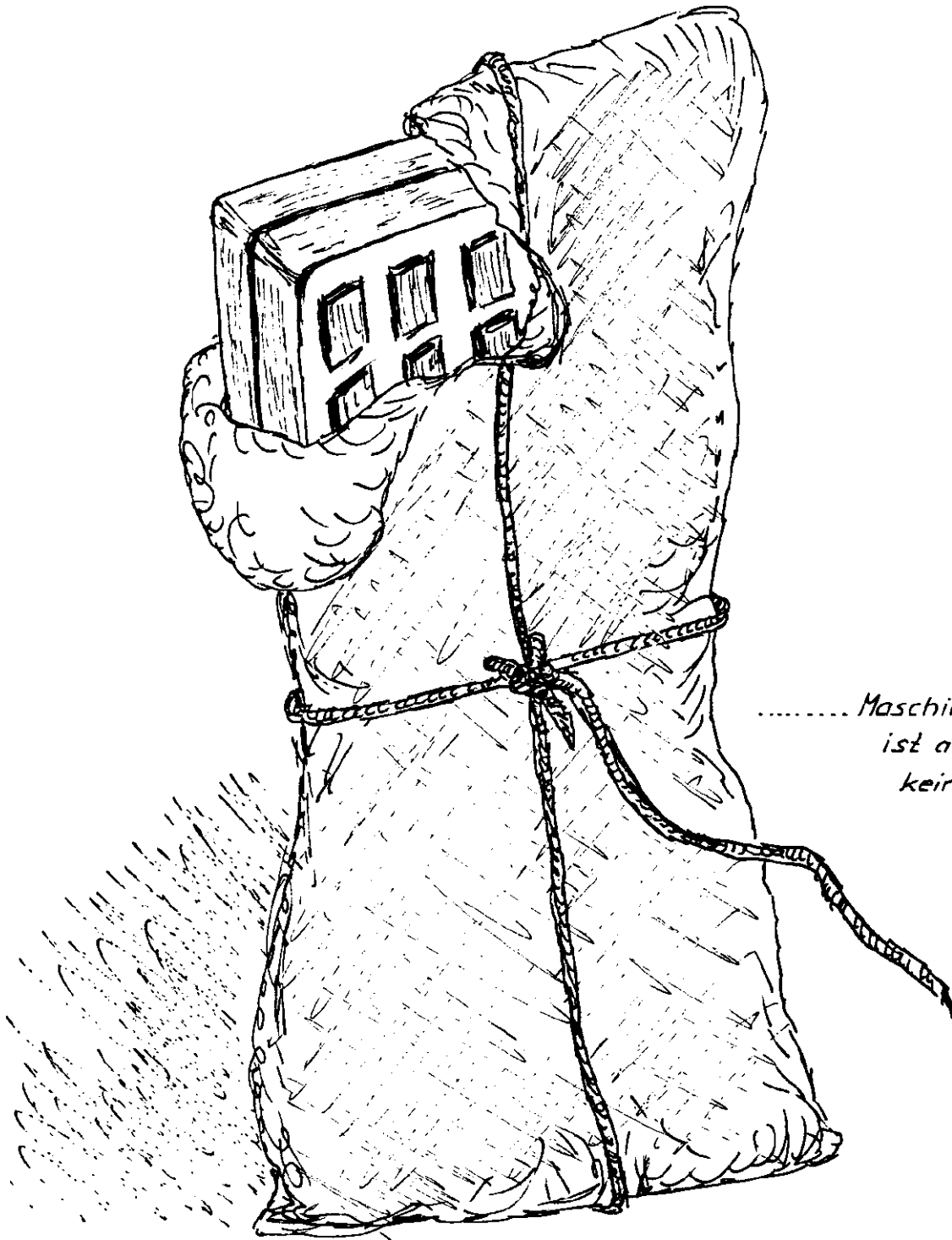
R = &56 und

&56: &A9, &C1, &AE, &B3, &1E, &AA.

Das letzte RTN eines Maschinenprogrammes spricht also die Adresse &C1A9 an. Weitere RTN würden die Adressen &B3AE und dann &AA1E ansprechen. (Bei der alten Version, PC-1260, sind es die drei Adressen &BF53, &B1A1, &A82C.)

Auf der Adresse &C1A9 wird das Carry-Flag zurückgesetzt und dann der Inhalt von &6635/6 nach X geladen. Danach folgt im Basic-Interpreter bereits ein RTN, also der Sprung nach &B3AE. Dort wird z.B. abhängig vom Inhalt von &35 und &36 (internes Memory) einiges organisiert. Bei der Unter-routine &D1D4 (&CFOF) trifft man auf eine Anzeigeroutine. Schließlich kommt wieder ein RTN im Basic-Interpreter und somit der Sprung nach &AA1E. Von hier wird gleich weitergesprungen nach &112E, wo der Tastenerkennungsteil beginnt. Nach der Tastenerkennung kommt schließlich die Tastenverarbeitung. Die so gefundenen Routinen werden im folgenden ausführlich beschrieben.

Diese Untersuchung des Basic-Interpreters durch die Ver-
folgung des Programm von einem Startpunkt ist recht müh-
sam und kann auch erst durchgeführt werden, wenn einige
Erfahrung in der Programmierung in Maschinensprache vor-
liegt. Auch die in der Tokentabelle gegebenen Sprunga-
dressen eignen sich oft als Ausgangspunkt für eine Unter-
suchung des System, wenn man sich für bestimmte Funktio-
nen des Rechners interessiert.



..... Maschinensprache
ist also doch
kein Geheimnis !

VII.2 Die Tastenroutinen

Im Tastenerkennungsteil, der bei &112E beginnt, befindet sich eine Unteroutine CAL03C1, die abfragt, ob gerade eine Taste betätigt wird. Falls nicht, wird mit A = &FF, C = 0 und Z = 0 zurückgekehrt. Falls gerade eine Taste betätigt wird, wird mit dem Tastencode im Akkumulator, Tabelle 1, und mit C = 1 und Z = 1 zurückgekehrt. Hier werden alle Tasten, also auch DEF, SHIFT oder SML, einzeln behandelt.

Der bei &112E beginnende Tastenerkennungsteil des Interpreters ist keine Unteroutine. Er schließt zwar mit einem RTN ab, aber es wird am Anfang (nach CAL1353, ANID FE = Lösche BUSY) die Adresse &8012 in den Stack gegeben. Außerdem wird bei DEF, Taste (A bis SPC) diese Stackadresse ausgetauscht gegen &8015. Um nun den Tastenerkennungsteil als Unteroutine zu benutzen, muß man bei einer späteren Adresse aufrufen, so daß die Eingabe von &8012 in den Stack umgangen wird. Man muß also CAL1138 aufrufen. Man umgeht dann zwar auch CAL1353, ANID FE. Dies bewirkt jedoch nur, daß BUSY in der Anzeige verschwindet und kann bei Bedarf vorangestellt werden.

Die Tastenerkennungsroutine CAL1138 wartet bis eine Taste betätigt wird (nach ca. 11 Minuten wird abgeschaltet) und kehrt dann mit dem Tastencode, Tabelle 2, im Akkumulator und einer Kennung in Carry und Zero zurück. SHIFT, Taste und SML, Taste werden zusammen behandelt. Bei DEF, Taste (A bis SPC) wird das entsprechende Programm aufgerufen. Die Routine setzt DEF, SHIFT und SML nicht selbsttätig zurück, darum empfiehlt es sich anschließend CAL1353, ANID 8F zu programmieren. Wird der Schalter auf PRO oder RSV gestellt, so erfolgt im allgemeinen eine Fehlermeldung.

Tabelle 1: Tastenabfrageroutine CAL03C1

Fragt nach momentaner Taste.

Falls keine Taste betätigt: A = &FF, C = 0, Z = 0.

Falls eine Taste betätigt: A = Code, C = 1, Z = 1.

Taste	Hexcode	Taste	Hexcode
DEF	2F	C	1C
SHIFT	27	V	00
↓	1F	B	07
↑	03	N	0D
◀	0A	M	12
▶	10	SPC	16
BRK	34	ENTER	19
Q	2E	7	06
W	26	8	32
E	1E	9	2A
R	02	CL	22
T	09	4	04
Y	0F	5	30
U	14	6	28
I	18	/	20
O	11	1	05
P	0B	2	31
A	2D	3	29
S	25	*	21
D	1D	0	1B
F	01	.	33
G	08	-	2B
H	0E	-	23
J	13		
K	17		
L	1A		
=	0C		
Z	2C		
X	24		

Tabelle 2: Tastenabfrageroutine CAL1138

Wartet auf Taste, auch SHIFT, Taste oder SML, Taste.

Nach ca. 11 Minuten wird der Computer abgeschaltet, wenn keine Taste betätigt wird.

Wird eine Taste betätigt, so erfolgt Rückkehr mit dem Tastencode im Akkumulator und einer Kennung in Carry und Zero.

Taste	Zeichen	Hexcode	C,Z
↓		05	1,0
DEF ↓		05	1,0
SHIFT ↓		05	1,0
SML ↓		05	1,0
↑		04	1,0
DEF ↑		04	1,0
SHIFT ↑		04	1,0
SML ↑		04	1,0
◀		0F	1,0
DEF ◀		0F	1,0
SHIFT ◀	DEL	0C	1,0
SML ◀		0F	1,0
▶		0E	1,0
DEF ▶		0E	1,0
SHIFT ▶	INS	0B	1,0
SML ▶		0E	1,0
BRK		07	1,0
DEF BRK		07	1,0
SHIFT BRK		07	1,0
SML BRK		07	1,0
Q	Q	51	0,1
DEF Q		51	0,1
SHIFT Q	!	21	1,0
SML Q	q	71	0,0
W	W	57	0,1
DEF W		57	0,1
SHIFT W	"	22	1,0
SML W	w	77	0,0

Taste	Zeichen	Hexcode	C,Z
E	E	45	0,1
DEF E		45	0,1
SHIFT E	#	23	1,0
SML E	e	65	0,0
R	R	52	0,1
DEF R		52	0,1
SHIFT R	\$	24	1,0
SML R	r	72	0,0
T	T	54	0,1
DEF T		54	0,1
SHIFT T	%	25	1,0
SML T	t	74	0,0
Y	Y	59	0,1
DEF Y		59	0,1
SHIFT Y	&	26	1,0
SML Y	y	79	0,0
U	U	55	0,1
DEF U		55	0,1
SHIFT U	?	3F	1,0
SML U	u	75	0,0
I	I	49	0,1
DEF I		49	0,1
SHIFT I	:	3A	1,0
SML I	i	69	0,0
O	O	4F	0,1
DEF O		4F	0,1
SHIFT O	,	2C	1,0
SML O	o	6F	0,0
P	P	50	0,1
DEF P		50	0,1
SHIFT P	;	3B	1,0
SML P	p	70	0,0
A	A	41	0,1
SHIFT A		81	0,0
SML A	a	61	0,0

Taste	Zeichen	Hexcode	C,Z
S	S	53	0,1
SHIFT S		F3	0,0
SML S	s	73	0,0
D	D	44	0,1
SHIFT D		84	0,0
SML D	d	64	0,0
F	F	46	0,1
SHIFT F		86	0,0
SML F	f	66	0,0
G	G	47	0,1
SHIFT G		87	0,0
SML G	g	67	0,0
H	H	48	0,1
SHIFT H		88	0,0
SML H	h	68	0,0
J	J	4A	0,1
SHIFT J		8A	0,0
SML J	j	6A	0,0
K	K	4B	0,1
SHIFT K		8B	0,0
SML K	k	6B	0,0
L	L	4C	0,1
SHIFT L		8C	0,0
SML L	l	6C	0,0
=	=	3D	0,1
SHIFT =		F4	0,0
SML =		3D	0,1
Z	Z	5A	0,1
SHIFT Z		FA	0,0
SML Z	z	7A	0,0
X	X	58	0,1
SHIFT X		F8	0,0
SML X	x	78	0,0
C	C	43	0,1
SHIFT C		83	0,0
SML C	c	63	0,0

Taste	Zeichen	Hexcode	C,Z
V	V	56	0,1
SHIFT V		F6	0,0
SML V	v	76	0,0
B	B	42	0,1
SHIFT B		82	0,0
SML	b	62	0,0
N	N	4E	0,1
SHIFT N		8E	0,0
SML N	n	6E	0,0
M	M	4D	0,1
SHIFT M		8D	0,0
SML M	m	6D	0,0
SPC		20	0,1
SHIFT SPC		F1	0,0
SML SPC		20	0,1
ENTER		0D	1,0
SHIFT ENTER		06	1,0
SML ENTER		0D	1,0
7	7	37	1,0
DEF 7		37	1,0
SHIFT 7		14	1,0
SML 7		37	1,0
8	8	38	1,0
DEF 8		38	1,0
SML 8		38	1,0
9	9	39	1,0
DEF 9		39	1,0
SHIFT 9		39	1,0
SML 9		39	1,0
CL		02	1,0
DEF CL		02	1,0
SHIFT CL		03	1,0
SML CL		02	1,0
4	4	34	1,0
DEF 4		34	1,0
SHIFT 4		34	1,0

Taste	Zeichen	Hexcode	C,Z
SML 4		34	1,0
5	5	35	1,0
DEF 5		35	1,0
SHIFT 5		35	1,0
SML 5		35	1,0
6	6	36	1,0
DEF		36	1,0
SHIFT 6		36	1,0
SML 6		36	1,0
/	/	2F	1,0
DEF /		2F	1,0
SHIFT /	^	5E	0,0
SML /		2F	1,0
1	1	31	1,0
DEF 1		31	1,0
SHIFT 1	(28	1,0
SML 1		31	1,0
2	2	32	1,0
DEF 2		32	1,0
SHIFT 2)	29	1,0
SML 2		32	1,0
3	3	33	1,0
DEF 3		33	1,0
SHIFT 3	@	40	1,0
SML 3		33	1,0
*	*	2A	1,0
DEF *		2A	1,0
SHIFT *	<	3C	1,0
SML *		2A	1,0
0	0	30	1,0
DEF 0		30	1,0
SHIFT 0	π	FB	0,0
SML 0		30	0,0
.	.	2E	1,0
DEF .		2E	1,0

Taste	Zeichen	Hexcode	C, Z
SHIFT .	√	FC	0,0
SML .		2E	1,0
+	+	2B	1,0
DEF +		2B	1,0
SHIFT +		2B	1,0
SML +		2B	1,0
-	-	2D	1,0
DEF -		2D	1,0
SHIFT -	>	3E	1,0
SML -		2D	1,0

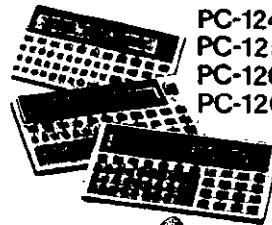
Basic Lehrbuch für SHARP Computer



FISCHEL GbH

Anwendungs- Handbuch

von SHARP Taschencomputer



PC-1245
PC-1251
PC-1260
PC-1261

FISCHEL GmbH INGO LAUER



FISCHEL GmbH
Kaiser Friedrich Str. 54a 1000 Berlin 12

Tel. 030/3236029

VII.3 Die Anzeige- und Ausdruck-Routinen

Die Anzeige-Routine kann sowohl mit CALL 800F als auch mit CAL107D aufgerufen werden. Sie bringt den Inhalt der RAM-Plätze &6650 bis &667F auf die der Anzeige zugeordneten Systemplätze &2000 ff. Damit der Text auch auf der zweizeiligen Anzeige sichtbar wird, muß diese noch mit der Routine CAL043B (Display ON) eingeschaltet werden. Das letztere ist nicht nötig, wenn anschließend die Tastenabfrage-Routine CAL1138 aufgerufen wird, da diese das Display einschaltet. Die Routine CAL106D bringt das Prompt-Zeichen auf die Displayplätze. Die Routine CAL104D tut das gleiche, zusätzlich initialisiert sie jedoch noch einige Systemplätze im Basic-Interpreter. Die Routine CAL1B2F bringt den Code &60 (space) auf die Plätze &6650 bis &667F.

Die Ausdruck-Routinen werden mit CALL 8021, CAL 1B92 oder CAL 1B90 aufgerufen. CALL 8021 bringt den Inhalt der internen Memory-Plätze &10 bis &27 in eine Ausdruckzeile des Druckers CE-126P. CAL 1B92 tut das gleiche wie CALL 8021 und setzt zusätzlich den Inhalt der Speicherplätze &66B8 bis &66CF gleich Null. CAL 1B90 bringt den Inhalt der Speicherplätze &66B8 bis &66CF in eine Druckerzeile. Anschließend wird der Inhalt dieser Speicherplätze gleich Null gesetzt.

In Tabelle 3 ist die Zuordnung zwischen den Hexcode und den durch den Zeichengenerator des Druckers erzeugten Zeichen gegeben. Die Code &08, &0A und &0D ergeben keine Zeichen, sie besitzen Steuerfunktionen für die Ausdruckroutine. Durch Austesten wurden die in Tabelle 4 gegebenen Steuerfunktionen festgestellt.

Tabelle 3: Zuordnung zwischen Hexcode und Zeichen
für den Drucker CE-126P

00		34	4	67	g	99	1	DE	#
01	*	35	5	68	h	9A	*	DF	?
02	2	36	6	69	i	9B	2	DD	3
03	C	37	7	6A	j	9C	C	D1	4
04	P	38	8	6B	k	9D	P	D2	*
05	E	39	9	6C	l	9E	E	D3	6
06	~	3A	:	6D	m	9F	~	D4	7
07	~	3B	:	6E	n	A0	~	D5	1
08	~	3C	<	6F	o	A1	.	D6	3
09	II	3D	=	70	p	A2	II	D7	5
0A	~	3E	>	71	q	A3	~	D8	9
0B	~	3F	?	72	r	A4	~	D9	II
0C	~	40	@	73	s	A5	.	DA	V
0D	~	41	A	74	t	A6	~	DB	0
0E	0	42	B	75	u	A7	0	DC	7
0F	+	43	C	76	v	A8	+	DD	0
10	+	44	D	77	w	A9	+	DE	*
11	-	45	E	78	x	AA	-	DF	*
12	x	46	F	79	y	AB	x	E0	~
13	+	47	G	7A	z	AC	+	E1	E
14	=	48	H	7B	{	AD	=	E2	U
15	0	49	I	7C		AE	0	E3	i
16	2	4A	J	7D	}	AF	2	E4	0
17	-	4B	K	7E	~	B0	-	E5	2
18	f	4C	L	7F	~	B1	f	E6	E
19	II	4D	M	80	0	B2	II	E7	0
1A	4	4E	N	81	1	B3	4	E8	~
1B	+	4F	O	82	2	B4	+	E9	~
1C	+	50	P	83	3	B5	+	EA	0
1D	↑	51	Q	84	4	B6	↑	EB	~
1E	↓	52	R	85	5	B7	↓	EC	0
1F	0	53	S	86	6	B8	0	ED	~
20		54	T	87	7	B9		EE	~
21	!	55	U	88	8	BA	!	EF	~
22	~	56	V	89	9	BB	~	F0	0
23	~	57	W	8A	II	BC	~	F1	~
24	~	58	X	8B	II	BD	~	F2	~
25	~	59	Y	8C	II	BE	~	F3	~
26	~	5A	Z	8D	II	BF	~	F4	~
27		5B		8E	II	C0		F5	~
28	(5C	¥	8F	II	C1	(F6	~
29)	5D		90	+	C2)	F7	~
2A	*	5E	^	91	-	C3	*	F8	~
2B	+	5F	-	92	x	C4	+	F9	~
2C	~	60		93	+	C5	~	FA	
2D	-	61	a	94	/	C6	-	FB	~
2E	~	62	b	95	-	C7	~	FC	~
2F	/	63	c	96	-	C8	/	FD	~
30	0	64	d	97	-	C9	0	FE	~
31	1	65	e	98	-	CA	1	FF	~
32	2	66	f			CB	2		
33	3					CC	3		
						CD			

Tabelle 4: Sonderzeichen mit Steuerfunktion für die
Ausdruckroutine CALL 8021

Befindet sich im Ausdruck-Speicher, &10 bis &27, einer der Hexcode &08, &0A oder &0D, so bewirken diese folgende Steuerfunktionen:

- &08: Die Zeichen vor &08 und &08 selbst werden als space ausgedruckt.
Die hinter &08 stehenden Zeichen werden unverändert ausgedruckt.
Steht &08 auf dem letzten Platz (&27), so wird die Zeile nicht ausgedruckt.
- &0A: Wird wie &08 behandelt. Jedoch zusätzlich wird vorangehend eine Leerzeile ausgedruckt.
- &0D: Die Zeichen vor &0D werden rechtsbündig in einer Zeile ausgedruckt.
Die Zeichen nach &0D werden rechtsbündig in eine zweite Zeile ausgedruckt.
Steht &0D auf dem ersten Platz (&10), so wird keine erste Zeile ausgedruckt.
Steht &0D auf dem letzten Platz (&27), so wird keine zweite Zeile ausgedruckt.

VII.4 System-Adressen, Token-Tabelle

ROM-Speicher, Basic-Interpreter

&0000 - &1FFF Kern-ROM

&8000 - &FFFF ROM

RAM-Speicher

&2000 - &203B 1. - 12. Displaystelle

&203D Sonder-Anzeigesegmente



DEF SHIFT SMALL Print Busy

&2040 - &207B 13. - 24. Displaystelle

&207C Sonder-Anzeigesegmente



- Error Grad Rad Deg

↑ Balken zwischen ERROR und MEMORY SAFE GUARD

&2800 - &283B 25. - 36. Displaystelle

&2840 - &287B 37. - 48. Displaystelle

&4000 - &407F PC-1261 Grundbereich für EASY SIMULATION PROGRAM. Kann mit EQU# n um n mal &80 erweitert werden.

&4080 - &64FF PC-1261 Basicprogramm-Bereich und Feldvariable. Bei erweitertem EASY SIMULATION PROGRAM späterer Beginn.

&5800 - &587F PC-1260 Grundbereich für EASY SIMULATION PROGRAM. Kann mit EQU# n um n mal &80 erweitert werden.

&5880 - &64FF PC-1260 Basicprogramm-Bereich und Feldvariable. Bei erweitertem EASY SIMULATION PROGRAM späterer Beginn.

&6500 - &65CF Standardvariable

&6500 - &6507 Z = A(26)

.....

&6508 - &65CF A = A(1)

Wenn 1. Stelle = &F5: 7 stellige Textvariable.

&65D0 - &65FF Reservespeicher

&6650 - &667F	Anzeigebuffer
(&66E2/1)	Beginn des Basicprogramm
(&66E4/3)	Ende des Basicprogramm
&66B8 - &66CF	Ausdruckbuffer
(&66FD/C)	Beginn der Feldvariablen
&67B0 - &67FF	Eingabebuffer

Kurze Kern-ROM-Routinen

&0222	BA → Y
&0225	BA → X
&022E	BA → X, X - 1 → X, DP
&0242	BA → Y, Y - 1 → Y, DP
&0437	Display OFF
&043B	Display ON
&0F65	X ↔ (&38,&39)
&133B	(&38,&39) → X
&1344	X → Y
&1349	Y → X
&134E	X → (&38,&39)
&1353	&203D → DP
&1B2F	&60 → (&6650-&667F)
&1BB0	&60 → (&10-&27)
&1BCD	&20 → (&6650-&6667)
&1BD6	&20 → (&6668-&667F)

ROM-Unterprogramme

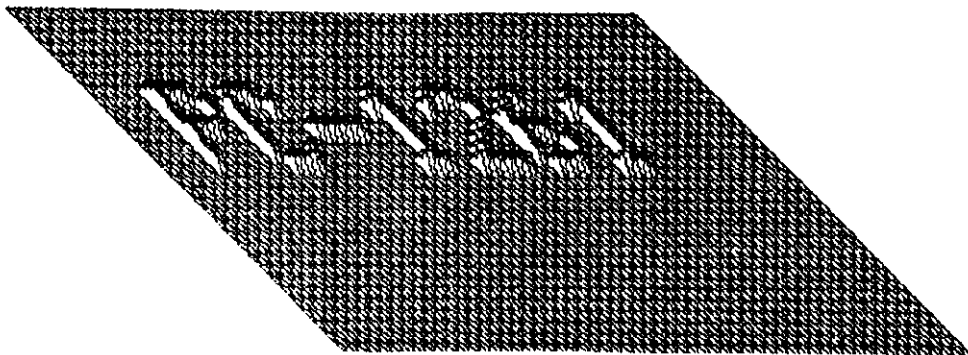
&03C1	Abfrage ob gerade eine Taste betätigt. Siehe VII.2 Beeinflußt : P,Q,A,B,&8,&9,&A Unbeeinflußt: I,J=1,X,Y,&B-&F
&1138	Wartet auf Taste. Siehe VII.2 Beeinflußt : P,Q,I,A,B,X,Y,&8,&9,&A Unbeeinflußt: J=1,&B-&F
&107D oder &800F	Anzeigeroutine, &6650 - &6667 → 1. Zeile Siehe VII.3 &6668 - &667F → 2. Zeile Beeinflußt : P,Q,A,B,X,Y,&9,&B,&C,&D Unbeeinflußt: I,J=1,&8,&A,&E,&F

&106D Prompt-Zeichen auf die Anzeige.
 Siehe VII.3
 Beeinflußt :P,Q,I,A,B,X,Y,&9,&B,&C,&D
 Unbeeinflußt:J=1,&8,&A,&E,&F

&8021 Ausdruck-Routine, &10 - &27 → CE-126P
 Siehe VII.3
 Beeinflußt :P,Q,A,B,&8,&9
 Unbeeinflußt:I,J=1,X,Y,&A bis &F

&1B92 Ausdruckroutine, &10 - &27 → CE-126P
 Siehe VII.3 und 0 → &66B8 bis &66CF
 Beeinflußt :P,Q,I,A,B,&8,&9
 Unbeeinflußt:J=1,X,Y,&A bis &F

&1B90 Ausdruckroutine, &66B8 - &66CF → CE-126P
 Siehe VII.3 und 0 → &66B8 bis &66CF
 Beeinflußt :P,Q,I,A,B,&8,&9
 Unbeeinflußt:J=1,X,Y,&A bis &F



Token-tabelle

Die anschließend gegebenen Token-tabellen (für die neue und die alte Version) geben an:

Den Namen des Basic-Befehls, den Code unter dem der Befehl im Basicprogramm codiert wird, die Sprungadresse und den Code, der zusätzliche Information über den Befehl enthält. Dieser wird z.B. benutzt, um gegebenenfalls eine Fehlermeldung zu erzeugen.

Sollten sich noch weitere ROM-Versionen auf dem Markt befinden, bei denen die Token-tabelle an einer anderen Stelle liegt, so läßt sie sich durch das folgende Basicprogramm, das nach dem Beginn der Token-tabelle (ARE) sucht, finden. Man muß sich dabei etwas Zeit lassen, da der Ablauf von &8000 = 32768 bis zu &A000 = 40960 etwa eine halbe Stunde dauert. Hat man die Anzeige verpaßt und hat sich der Rechner bereits ausgeschaltet, so kann man die Anfangsadresse der Token-tabelle nach dem Wiedereinschalten bei der Variablen A abfragen.

SUCHE NACH ARE(AD)

```
1:WAIT 0:A=&A000
2:PRINT A: IF PEEK A=&
  41 GOTO 4
3:A=A+1: GOTO 2
4:A=A+1: IF PEEK A=&52
  GOTO 6
5:GOTO 2
6:A=A+1: IF PEEK A=&45
  LET A=A-2: WAIT :
  PRINT A: END
7:GOTO 2
```

PC-1261 NEUE VERSION
 TOKENTABELLE

A43C:	41 52 45 41	AREA	A51C:	E0 BF FA A4	A5FC:	B0 BD 07 C6		
A440:	44 E1 C4 1F	D	A520:	47 52 41 44	GRAD	A600:	52 45 54 55	RETU
A444:	A3 41 4E 44	AND	A524:	C3 C3 B8 F5	♣	A604:	52 4E E3 C2	RN
A448:	A1 87 81 A3		A528:	49 4E 50 55	INPU	A608:	D6 A4 52 45	RE
A44C:	41 42 53 99	ABS	A52C:	54 DF C4 E5	T	A60C:	41 44 DB C1	AD
A450:	87 81 A3 41	A	A530:	E2 49 46 D4	IF	A610:	EF A7 52 45	RE
A454:	54 4E 9F 87	TN	A534:	C2 9D A3 49	I	A614:	53 54 4F 52	STOR
A458:	81 A3 41 53	AS	A538:	4E 54 98 87	NT	A618:	45 E4 C2 6D	E m
A45C:	4E 9D 87 81	N	A53C:	81 A6 49 4E	IN	A61C:	A3 52 4E 44	RND
A460:	A3 41 43 53	ACS	A540:	4B 45 59 24	KEY\$	A620:	A0 87 81 A6	
A464:	9E 87 81 A3		A544:	AD 87 81 D4		A624:	52 41 4E 44	RAND
A468:	41 53 43 A4	ASC	A548:	4C 49 53 54	LIST	A628:	4F 4D C0 C0	OM
A46C:	87 81 B4 42	B	A54C:	B4 C5 96 C5		A62C:	F7 E6 52 49	♣ RI
A470:	45 45 50 C4	ECP	A550:	4C 4C 49 53	LLIS	A630:	47 48 54 24	GHT\$
A474:	C1 47 F4 43	G C	A554:	54 B5 C7 D6	T	A634:	AC 87 81 C6	
A478:	4F 4E 54 B2	ONT	A558:	C6 4C 50 52	LPR	A638:	52 41 44 49	RADI
A47C:	C4 C4 C5 43	C	A55C:	49 4E 54 E2	INT	A63C:	41 4E C2 C3	AN
A480:	4C 45 41 52	LEAR	A560:	C7 86 A3 4C	L	A640:	B0 E3 52 45	RE
A484:	C9 C1 7B E5		A564:	4F 47 92 87	OG	A644:	4D D7 C2 CF	M
A488:	43 4C 4F 41	CLOA	A568:	81 A2 4C 4E	LN	A648:	B4 53 54 4F	STO
A48C:	44 B7 96 95	D	A56C:	91 87 81 A3		A64C:	50 DA C3 3C	P <
A490:	C5 43 53 41	CSA	A570:	4C 45 54 D6	LET	A650:	A3 53 51 52	SQR
A494:	56 45 B6 95	VE	A574:	C0 66 A3 4C	f L	A654:	94 87 81 A3	
A498:	13 E3 43 4F	CO	A578:	45 4E A6 87	EN	A658:	53 49 4E 95	SIN
A49C:	53 96 87 81	S	A57C:	81 85 4C 45	LE	A65C:	87 81 A3 53	S
A4A0:	A4 43 48 52	CHR	A580:	46 54 24 AB	FT\$	A660:	47 4E 9A 87	GN
A4A4:	24 A8 87 81	\$	A584:	87 81 D3 4D	M	A664:	81 A4 53 54	ST
A4A8:	A4 43 41 4C	CAL	A588:	45 4D AF 87	EM	A668:	52 24 A9 87	R\$
A4AC:	4C CC C1 91	L	A58C:	81 C4 4D 45	ME	A66C:	81 A4 53 54	ST
A4B0:	E3 43 4C 53	CLS	A590:	4D 23 BA 87	M#	A670:	45 50 D1 0F	EP
A4B4:	CE C0 FA A5		A594:	81 C4 4D 49	MI	A674:	C1 F4 54 48	TH
A4B8:	43 48 41 49	CHAI	A598:	44 24 AA 87	D\$	A678:	45 4E D2 0F	EN
A4BC:	4E E5 96 3C	N <	A59C:	81 C5 4D 45	ME	A67C:	C1 E3 54 41	TA
A4C0:	A6 43 55 52	CUR	A5A0:	52 47 45 B8	RGE	A680:	4E 97 87 81	N
A4C4:	53 4F 52 CF	SOR	A5A4:	96 BD D4 4E	N	A684:	A4 54 52 4F	TRO
A4C8:	C1 0C B3 44	D	A5A8:	45 58 54 D9	EXT	A688:	4E C7 C3 40	N @
A4CC:	49 4D CB BD	IM	A5AC:	BE B2 A3 4E	N	A68C:	E5 54 52 4F	TRO
A4D0:	7B E6 44 45	DE	A5B0:	4F 54 A3 87	OT	A690:	46 46 C8 C3	FF
A4D4:	47 52 45 45	GREE	A5B4:	81 A3 4E 45	NE	A694:	44 E2 54 4F	D TO
A4D8:	C1 C3 A8 E3		A5B8:	57 B1 BC 72	W r	A698:	D0 0F C1 F5	♣
A4DC:	44 45 47 9B	DEG	A5BC:	F2 4F 4E D3	ON	A69C:	55 53 49 4E	USIN
A4E0:	89 37 A3 44	7 D	A5C0:	C3 C0 A2 4F	O	A6A0:	47 CA C5 F8	G ♣
A4E4:	4D 53 9C 89	MS	A5C4:	52 A2 87 81	R	A6A4:	F3 56 41 4C	VAL
A4E8:	3E A4 44 41	> DA	A5C8:	B5 50 52 49	PRI	A6A8:	A5 87 81 94	
A4EC:	54 41 DC C4	TA	A5CC:	4E 54 DE C6	NT	A6AC:	57 41 49 54	WAIT
A4F0:	1F B3 45 4E	EN	A5D0:	E1 E4 50 41	PA	A6B0:	C5 C3 84 F0	
A4F4:	44 D8 C3 24	D \$	A5D4:	53 53 B3 C4	SS			
A4F8:	A3 45 58 50	EXP	A5D8:	37 C2 50 49	7 PI			
A4FC:	93 87 81 84		A5DC:	AE 87 81 C4				
A500:	45 51 55 23	EQU#	A5E0:	50 45 45 4B	PEEK			
A504:	B9 C6 24 D3	\$	A5E4:	A7 87 81 A4				
A508:	46 4F 52 D5	FOR	A5E8:	50 4F 4B 45	POKE			
A50C:	BE 0F B4 47	G	A5EC:	CD C1 AE E5				
A510:	4F 54 4F C6	OTO	A5F0:	50 41 55 53	PAUS			
A514:	BF C4 E5 47	G	A5F4:	45 DD C3 48	E H			
A518:	4F 53 55 42	OSUB	A5F8:	B3 52 55 4E	RUN			

PC-1260 ALTE VERSION
 TOKENTABELLE

A254:	D2 B5 41 52	AR	A334:	55 42 E0 BD UB	A414:	55 4E B0 BA UN	
A258:	45 41 44 E1 EAD		A338:	A4 A4 47 52 GR	A418:	B1 C6 52 45 RE	
A25C:	C1 96 A3 41	A	A33C:	41 44 C3 C1 AD	A41C:	54 55 52 4E TURN	
A260:	4E 44 A1 0A ND		A340:	2F F5 49 4E /IN	A420:	E3 C0 80 A4	
A264:	7F A3 41 42	AB	A344:	50 55 54 DF PUT	A424:	52 45 41 44 READ	
A268:	53 99 0A 7F S		A348:	C2 5C E2 49	I	A428:	DB BF 99 A7
A26C:	A3 41 54 4E	ATN	A34C:	46 D4 C0 47 F G		A42C:	52 45 53 54 REST
A270:	9F 0A 7F A3		A350:	A3 49 4E 54	INT	A430:	4F 52 45 E4 ORE
A274:	41 53 4E 9D	ASN	A354:	98 0A 7F A6		A434:	C0 17 A3 52 R
A278:	0A 7F A3 41	A	A358:	49 4E 4B 45	INKE	A438:	4E 44 A0 0A ND
A27C:	43 53 9E 0A	CS	A35C:	59 24 AD 0A Y\$		A43C:	7F A6 52 41 RA
A280:	7F A3 41 53	AS	A360:	7F D4 4C 49	LI	A440:	4E 44 4F 4D NDOM
A284:	43 A4 0A 7F	C	A364:	53 54 B4 C3	ST	A444:	C0 BE A1 E6
A288:	B4 42 45 45	BEE	A368:	08 C5 4C 4C	LL	A448:	52 49 47 48 RIGH
A290:	50 C4 BE F1	P	A36C:	49 53 54 B5	IST	A44C:	54 24 AC 0A T\$
A294:	F4 43 4F 4E	CON	A370:	C5 45 C6 4C	E L	A450:	7F C6 52 41 RA
A298:	54 B2 C2 3B	T ;	A374:	50 52 49 4E	PRIN	A454:	44 49 41 4E DIAN
A29C:	C5 43 4C 45	CLE	A378:	54 E2 C4 F8	T *	A458:	C2 C1 27 E3
A2A0:	41 52 C9 BF	AR	A37C:	A3 4C 4F 47	LOG	A45C:	52 45 4D D7 REM
A2A4:	25 E5 43 4C	% CL	A380:	92 0A 7F A2		A460:	C0 79 B4 53 y S
A2A8:	4F 41 44 B7	OAD	A384:	4C 4E 91 0A	LN	A464:	54 4F 50 DA TOP
A2A8:	95 7C C5 43	C	A388:	7F A3 4C 45	LE	A468:	C0 B3 A3 53 S
A2AC:	53 41 56 45	SAVE	A38C:	54 D6 BE 10	T	A46C:	51 52 94 0A OR
A2B0:	B6 94 08 E3		A390:	A3 4C 45 4E	LEN	A470:	7F A3 53 49 SI
A2B4:	43 4F 53 96	COS	A394:	A6 0A 7F 85		A474:	4E 95 0A 7F N
A2B8:	0A 7F A4 43	C	A398:	4C 45 46 54	LEFT	A478:	A3 53 47 4E SGN
A2BC:	48 52 24 A8	HR\$	A39C:	24 AB 0A 7F	\$	A47C:	9A 0A 7F A4
A2C0:	0A 7F A4 43	C	A3A0:	D3 4D 45 4D	MEM	A480:	53 54 52 24 STR\$
A2C4:	41 4C 4C CC	ALL	A3A4:	AF 0A 7F C4		A484:	A9 0A 7F A4
A2C8:	BF 3B E3 43	; C	A3A8:	4D 45 4D 23	MEM\$	A488:	53 54 45 50 STEP
A2CC:	4C 53 CE BE	LS	A3AC:	BA 0A 7F C4		A48C:	D1 0F C1 F4
A2D0:	A4 A5 43 48	CH	A3B0:	4D 49 44 24	MID\$	A490:	54 48 45 4E THEN
A2D4:	41 49 4E E5	AIN	A3B4:	AA 0A 7F C5		A494:	D2 0F C1 E3
A2D8:	95 31 A6 43	1 C	A3B8:	4D 45 52 47	MERG	A498:	54 41 4E 97 TAN
A2DC:	55 52 53 4F	URSO	A3BC:	45 B8 95 A2	E	A49C:	0A 7F A4 54 T
A2E0:	52 CF BE B6	R	A3C0:	D4 4E 45 58	NEX	A4A0:	52 4F 4E C7 RON
A2E4:	B3 44 49 4D	DIM	A3C4:	54 D9 BC 5C	T	A4A4:	C0 B7 E5 54 T
A2E8:	CB BB 25 E6	%	A3C8:	A3 4E 4F 54	NOT	A4A8:	52 4F 46 46 ROFF
A2EC:	44 45 47 52	DEGR	A3CC:	A3 0A 7F A3		A4AC:	C8 C0 BB E2
A2F0:	45 45 C1 C1	EE	A3D0:	4E 45 57 B1	NEW	A4B0:	54 4F D0 0F TO
A2F4:	1F E3 44 45	DE	A3D4:	BA 1C F2 4F	0	A4B4:	C1 F5 55 53 #US
A2F8:	47 9B 88 53	G S	A3D8:	4E D3 C1 37	N 7	A4B8:	49 4E 47 CA ING
A2FC:	A3 44 4D 53	DMS	A3DC:	A2 4F 52 A2	OR	A4BC:	C3 6A F3 56 j V
A300:	9C 88 5A A4	Z	A3E0:	0A 7F B5 50	P	A4C0:	41 4C A5 0A AL
A304:	44 41 54 41	DATA	A3E4:	52 49 4E 54	RINT	A4C4:	7F 94 57 41 WA
A308:	DC C1 96 B3		A3E8:	DE C4 53 E4	S	A4C8:	49 54 C5 C0 IT
A30C:	45 4E 44 D8	END	A3EC:	50 41 53 53	PASS	A4CC:	FB F0 FD 20 X
A310:	C0 9B A3 45	E	A3F0:	B3 C1 AE C2			
A314:	58 50 93 0A	XP	A3F4:	50 49 AE 0A	PI		
A318:	7F 84 45 51	EQ	A3F8:	7F C4 50 45	PE		
A31C:	55 23 B9 C3	U\$	A3FC:	45 4B A7 0A	EK		
A320:	96 D3 46 4F	FO	A400:	7F A4 50 4F	PO		
A324:	52 D5 BB B9	R	A404:	4B 45 CD BF	KE		
A328:	B4 47 4F 54	GOT	A408:	58 E5 50 41	X PA		
A32C:	4F C6 BD 6E	0 n	A40C:	55 53 45 DD	USE		
A330:	E5 47 4F 53	GOS	A410:	C0 BF B3 52	R		

Anhang Das ROM. Befehlslisting des Kern-ROM, 0 bis &1FFF

Das ROM des PC-1260/61 besteht aus zwei Teilen, dem Kern-ROM von &0000 bis &1FFF und dem Haupt-ROM von &8000 bis &FFFF.

Das Befehlslisting des Kern-ROM des PC-1260/61 ist anschließend ausgedruckt. Der Ausdruck erfolgte mit dem im Kapitel V gegebenen Disassembler. Bei einem Byte für Byte Vergleich der Kern-ROM der beiden in der Einleitung beschriebenen Versionen ergab sich, daß beide übereinstimmten.

Das Haupt-ROM ist bei beiden Versionen unterschiedlich. Ein Beispiel hierfür sind die beiden gegebenen Token-Tabellen. Die Tokentabellen unterscheiden sich jedoch nur durch ihre Positionen im ROM und ihre Sprungadressen. Ansonsten sind sie gleich, insbesondere auch die Namen der Befehle, so daß die beiden Versionen von außen nicht zu unterscheiden sind.

Der ROM-Beginn von &8000 bis &8044 enthält JUMP-Befehle zu bestimmten Programmteilen des Haupt-ROM oder des Kern-ROM. Die Positionen sind offensichtlich bei beiden Versionen gleich, nur die Sprungziele sind verschieden, wenn sie ins Haupt-ROM führen.

	ROM-BEGINN ALTE VERSION	ROM-BEGINN NEUE VERSION
Z.B. wird im Kern-ROM		
auf &107D die Anzeige-	8000: 79 JP 884C	8000: 79 JP 8930
Routine (CALL 800F)	8003: 79 JP 8C4B	8003: 79 JP 8D2F
aufgerufen. Bei der	8006: 79 JP AA93	8006: 79 JP AC85
alten Version steht	8009: 79 JP 86B9	8009: 79 JP 879D
dort &800F: JP CFOF,	800C: 79 JP D348	800C: 79 JP D690
bei der neuen Version	800F: 79 JP CFOF	800F: 79 JP D1D4
&800F: JP D1D4. D.h.	8012: 79 JP A73A	8012: 79 JP A92C
bezüglich des Kern-	8015: 79 JP A8EA	8015: 79 JP AACD
ROM und des ROM-Be-	8018: 79 JP D3BA	8018: 79 JP D702
ginn sind beide Ver-	801B: 79 JP 8F7B	801B: 79 JP 9062
sionen gleich.	801E: 79 JP 8ED6	801E: 79 JP 8FBD
	8021: 79 JP A11F	8021: 79 JP A305
	8024: 79 JP 04B4	8024: 79 JP 04B4
	8027: 79 JP 04B6	8027: 79 JP 04B6
	802A: 79 JP 0516	802A: 79 JP 0516
	802D: 79 JP 0518	802D: 79 JP 0518
	8030: 79 JP 054E	8030: 79 JP 054E
	8033: 79 JP 0550	8033: 79 JP 0550
	8036: 79 JP 990C	8036: 79 JP 9AF2
	8039: 79 JP 8B1B	8039: 79 JP 8BFF
	803C: 79 JP 8B9F	803C: 79 JP 8C83
	803F: 79 JP 8B4E	803F: 79 JP 8C32
	8042: 79 JP CCC4	8042: 79 JP CF55

PC-1260/1261
 KERNROM-AUSDRUCK
 %0000 - %1FFF

0000: 4E WAIT E0
 0002: 02 LIA 01
 0004: 12 LIP 5F
 0006: DB EXAM
 0007: DF OUTC
 0008: 4E WAIT E0
 000A: 60 ANIM 00
 000C: DF OUTC
 000D: 6B TEST 40
 000F: 7E JPZ 8003
 0012: 4E WAIT E0
 0014: 6B TEST 40
 0016: 39 JRZM 0F >0008
 0018: 02 LIA 5C
 001A: 32 STR
 001B: 4C INA
 001C: 67 CPIA 01
 001E: 38 JRZP 2C >004B
 0020: 12 LIP 5F
 0022: 10 LIDP 0000
 0025: 57 LDD
 0026: 67 CPIA 02
 0028: 3A JRCP 15 >003E
 002A: 38 JRZP 1A >0045
 002C: 66 TSIA F3
 002E: 38 JRZP 18 >0047
 0030: 02 LIA 11
 0032: DB EXAM
 0033: 00 LII 02
 0035: 02 LIA FF
 0037: 12 LIP 5C
 0039: 1E FILM
 003A: 5D OUTA
 003B: DD OUTB
 003C: 5F OUTF
 003D: DF OUTC
 003E: 42 INCA
 003F: 6B TEST 40
 0041: 29 JRNZM 04 >003E
 0043: 2D JRM 44 >0000
 0045: 02 LIA 60
 0047: DB EXAM
 0048: DF OUTC
 0049: 2D JRM 0C >003E
 004B: 10 LIDP 6000
 004E: 00 LII BF
 0050: 02 LIA 00
 0052: 1F FILD
 0053: 02 LIA 01
 0055: E4 CAL04 3D
 0057: 88 LP08
 0058: 00 LII 4F
 005A: 02 LIA 96
 005C: 1E FILM
 005D: 88 LP08
 005E: 00 LII 4F

0060: 1C SRW
 0061: 00 LII 23
 0063: 02 LIA 03
 0065: 34 PUSH
 0066: AF LP2F
 0067: 13 LIQ 53
 0069: 0E ADW
 006A: 2F LOOP 05 >0066
 006C: 34 PUSH
 006D: AF LP2F
 006E: 13 LIQ 53
 0070: 0F SBW
 0071: 2F LOOP 05 >006D
 0073: 00 LII 3F
 0075: 10 LIDP 6000
 0078: 90 LP10
 0079: 19 EXWD
 007A: 11 LIDL 00
 007C: 90 LP10
 007D: 18 MYWD
 007E: 6B TEST 40
 0080: 39 JRZM 81 >0000
 0082: 6B TEST 01
 0084: 39 JRZM 07 >007E
 0086: 01 LIJ 01
 0088: E3 CAL03 C1
 008A: 2B JRNCM 0D >007E
 008C: 82 LP02
 008D: 63 CPIM 2F
 008F: 29 JRNZM 12 >007E
 0091: 12 LIP 5F
 0093: 59 LDM
 0094: 61 ORIM 31
 0096: 66 TSIA 10
 0098: 38 JRZP 03 >009C
 009A: 60 ANIM 21
 009C: DF OUTC
 009D: 2D JRM 41 >005D
 009F: A9 LP29
 00A0: 2C JRP 02 >00A3
 00A2: 91 LP11
 00A3: 00 LII 06
 00A5: 1C SRW
 00A6: 20 LDP
 00A7: 75 SBIA 07
 00A9: 30 STP
 00AA: 59 LDM
 00AB: 58 SWP
 00AC: DB EXAM
 00AD: 37 RTN
 00AE: AA LP2A
 00AF: 2C JRP 02 >00B2
 00B1: 92 LP12
 00B2: 59 LDM
 00B3: 58 SWP
 00B4: 64 ANIA 0F
 00B6: 51 DECP
 00B7: 60 ANIM F0
 00B9: 47 ORMA
 00BA: 20 LDP

00BB: 74 ADIA 06
 00BD: 30 STP
 00BE: 00 LII 05
 00C0: 1D SLW
 00C1: 37 RTN
 00C2: 90 LP10
 00C3: 00 LII 07
 00C5: 1C SRW
 00C6: 37 RTN
 00C7: 91 LP11
 00C8: 62 TSIM 0F
 00CA: 28 JRNZP 0E >00D9
 00CC: 92 LP12
 00CD: 02 LIA 05
 00CF: 63 CPIM 00
 00D1: 28 JRNZP 07 >00D9
 00D3: 50 INCP
 00D4: 43 DECA
 00D5: 2B JRNCM 07 >00CF
 00D7: D0 SC
 00D8: 37 RTN
 00D9: D1 RC
 00DA: 37 RTN
 00DB: B1 LP31
 00DC: 60 ANIM FD
 00DE: 98 LP18
 00DF: 2C JRP 08 >00E8
 00E1: B1 LP31
 00E2: 60 ANIM FB
 00E4: B1 LP31
 00E5: 60 ANIM FE
 00E7: 90 LP10
 00E8: 00 LII 07
 00EA: 02 LIA 00
 00EC: 1E FILM
 00ED: 37 RTN
 00EE: 91 LP11
 00EF: 2C JRP 01 >00F1
 00F1: 00 LII 01
 00F3: 02 LIA 10
 00F5: 0C ADN
 00F6: D1 RC
 00F7: 37 RTN
 00F8: B1 LP31
 00F9: 60 ANIM FD
 00FB: 98 LP18
 00FC: 00 LII 07
 00FE: 02 LIA 00
 0100: 1E FILM
 0101: 9A LP1A
 0102: 61 ORIM 10
 0104: 37 RTN
 0105: 99 LP19
 0106: 2C JRP 05 >010C
 0108: A9 LP29
 0109: 2C JRP 02 >010C
 010B: 91 LP11
 010C: 60 ANIM F0
 010E: 50 INCP
 010F: 02 LIA 00

0111:	00	LII	05	016F:	60	ANIM	FD	01CA:	47	ORMA	
0113:	1E	FILM		0171:	62	TSIM	01	01CB:	37	RTN	
0114:	37	RTN		0173:	38	JRZP	03 >0177	01CC:	91	LP11	
0115:	99	LP19		0175:	61	ORIM	02	01CD:	59	LDM	
0116:	2C	JRP	02 >0119	0177:	98	LP18		01CE:	64	ANIA	F0
0118:	91	LP11		0178:	2C	JRP	28 >01A1	01D0:	A9	LP29	
0119:	00	LII	01	017A:	B1	LP31		01D1:	60	ANIM	0F
011B:	02	LIA	10	017B:	60	ANIM	FE	01D3:	47	ORMA	
011D:	0D	SBM		017D:	62	TSIM	04	01D4:	90	LP10	
011E:	37	RTN		017F:	38	JRZP	03 >0183	01D5:	59	LDM	
011F:	98	LP18		0181:	61	ORIM	01	01D6:	A8	LP28	
0120:	2C	JRP	02 >0123	0183:	90	LP10		01D7:	DB	EXAM	
0122:	90	LP10		0184:	13	LIQ	20	01D8:	37	RTN	
0123:	60	ANIM	00	0186:	2C	JRP	1C >01A3	01D9:	99	LP19	
0125:	50	INCP		0188:	A0	LP20		01DA:	59	LDM	
0126:	60	ANIM	0F	0189:	2D	JRM	2E >015C	01DB:	64	ANIA	F0
0128:	37	RTN		018B:	B1	LP31		01DD:	91	LP11	
0129:	91	LP11		018C:	60	ANIM	FB	01DE:	60	ANIM	0F
012A:	59	LDM		018E:	62	TSIM	01	01E0:	47	ORMA	
012B:	64	ANIA	0F	0190:	38	JRZP	03 >0194	01E1:	98	LP18	
012D:	60	ANIM	F0	0192:	61	ORIM	04	01E2:	59	LDM	
012F:	99	LP19		0194:	A0	LP20		01E3:	90	LP10	
0130:	13	LIQ	11	0195:	2C	JRP	0B >01A1	01E4:	DB	EXAM	
0132:	00	LII	01	0197:	B1	LP31		01E5:	37	RTN	
0134:	0F	SBW		0198:	60	ANIM	F7	01E6:	91	LP11	
0135:	91	LP11		019A:	62	TSIM	01	01E7:	59	LDM	
0136:	47	ORMA		019C:	38	JRZP	03 >01A0	01E8:	64	ANIA	F0
0137:	37	RTN		019E:	61	ORIM	08	01EA:	99	LP19	
0138:	B1	LP31		01A0:	A8	LP28		01EB:	60	ANIM	0F
0139:	60	ANIM	F7	01A1:	13	LIQ	10	01ED:	47	ORMA	
013B:	62	TSIM	02	01A3:	00	LII	07	01EE:	90	LP10	
013D:	38	JRZP	03 >0141	01A5:	08	MYW		01EF:	59	LDM	
013F:	61	ORIM	08	01A6:	37	RTN		01F0:	98	LP18	
0141:	A0	LP28		01A7:	B1	LP31		01F1:	DB	EXAM	
0142:	2C	JRP	19 >015C	01A8:	59	LDM		01F2:	37	RTN	
0144:	B1	LP31		01A9:	D2	SR		01F3:	99	LP19	
0145:	60	ANIM	FD	01AA:	60	ANIM	FC	01F4:	59	LDM	
0147:	62	TSIM	08	01AC:	64	ANIA	01	01F5:	13	LIQ	11
0149:	38	JRZP	03 >014D	01AE:	47	ORMA		01F7:	00	LII	06
014B:	61	ORIM	02	01AF:	5A	SL		01F9:	08	MYW	
014D:	98	LP18		01B0:	5A	SL		01FA:	99	LP19	
014E:	13	LIQ	28	01B1:	64	ANIA	02	01FB:	60	ANIM	0F
0150:	2C	JRP	52 >01A3	01B3:	47	ORMA		01FD:	64	ANIA	F0
0152:	B1	LP31		01B4:	90	LP10		01FF:	47	ORMA	
0153:	60	ANIM	FE	01B5:	13	LIQ	18	0200:	37	RTN	
0155:	62	TSIM	02	01B7:	00	LII	07	0201:	91	LP11	
0157:	38	JRZP	03 >015B	01B9:	09	EXW		0202:	59	LDM	
0159:	61	ORIM	01	01BA:	37	RTN		0203:	13	LIQ	19
015B:	90	LP10		01BB:	B1	LP31		0205:	00	LII	06
015C:	13	LIQ	18	01BC:	59	LDM		0207:	08	MYW	
015E:	2C	JRP	44 >01A3	01BD:	60	ANIM	FA	0208:	91	LP11	
0160:	B1	LP31		01BF:	34	PUSH		0209:	60	ANIM	0F
0161:	60	ANIM	FE	01C0:	D2	SR		020B:	64	ANIA	F0
0163:	62	TSIM	08	01C1:	D2	SR		020D:	47	ORMA	
0165:	38	JRZP	03 >0169	01C2:	64	ANIA	01	020E:	37	RTN	
0167:	61	ORIM	01	01C4:	47	ORMA		020F:	99	LP19	
0169:	90	LP10		01C5:	5B	POP		0210:	59	LDM	
016A:	13	LIQ	28	01C6:	5A	SL		0211:	60	ANIM	0F
016C:	2C	JRP	36 >01A3	01C7:	5A	SL		0213:	64	ANIA	F0
016E:	B1	LP31		01C8:	64	ANIA	04	0215:	DA	EXAB	

0216:	51	DECP				0270:	99	LP19			02CA:	D1	RC	
0217:	02	LIA	00			0271:	59	LDM			02CB:	47	ORMA	
0219:	DB	EXAM				0272:	60	ANIM	0F		02CC:	01	LIJ	01
021A:	50	INCP				0274:	64	ANIA	F0		02CE:	37	RTN	
021B:	00	LII	01			0276:	58	SWP			02CF:	D1	RC	
021D:	13	LIQ	03			0277:	8A	LP0A			02D0:	B1	LP31	
021F:	0F	SBW				0278:	47	ORMA			02D1:	59	LDM	
0220:	D1	RC				0279:	37	RTN			02D2:	D2	SR	
0221:	37	RTN				027A:	8A	LP0A			02D3:	64	ANIA	01
0222:	06	LP06				027B:	59	LDM			02D5:	03	LIB	00
0223:	2C	JRP				027C:	64	ANIA	F0		02D7:	83	LP03	
0225:	84	LP04				027E:	91	LP11			02D8:	C4	ADCM	
0226:	13	LIQ	02			027F:	60	ANIM	0F		02D9:	59	LDM	
0228:	0A	MYB				0281:	47	ORMA			02DA:	64	ANIA	01
0229:	37	RTN				0282:	8A	LP0A			02DC:	B1	LP31	
022A:	02	LIA	B0			0283:	59	LDM			02DD:	60	ANIM	FE
022C:	03	LIB	67			0284:	58	SWP			02DF:	47	ORMA	
022E:	E2	CAL02	25			0285:	64	ANIA	F0		02E0:	37	RTN	
0230:	05	DX				0287:	99	LP19			02E1:	91	LP11	
0231:	37	RTN				0288:	60	ANIM	0F		02E2:	59	LDM	
0232:	02	LIA	50			028A:	47	ORMA			02E3:	64	ANIA	F0
0234:	03	LIB	66			028B:	37	RTN			02E5:	97	LP17	
0236:	2D	JRM	09			028C:	B1	LP31			02E6:	13	LIQ	1F
0238:	02	LIA	80			028D:	60	ANIM	00		02E8:	2C	JRP	08 >02F1
023A:	03	LIB	20			028F:	99	LP19			02EA:	99	LP19	
023C:	2D	JRM	0F			0290:	62	TSIM	08		02EB:	59	LDM	
023E:	02	LIA	B0			0292:	38	JRZP	06 >0299		02EC:	64	ANIA	F0
0240:	03	LIB	67			0294:	60	ANIM	F7		02EE:	9F	LP1F	
0242:	E2	CAL02	22			0296:	B1	LP31			02EF:	13	LIQ	17
0244:	07	DY				0297:	61	ORIM	02		02F1:	00	LII	06
0245:	37	RTN				0299:	2C	JRP	04 >029E		02F3:	0E	ADM	
0246:	02	LIA	80			029B:	B1	LP31			02F4:	50	INCP	
0248:	03	LIB	20			029C:	60	ANIM	00		02F5:	60	ANIM	0F
024A:	2D	JRM	09			029E:	91	LP11			02F7:	47	ORMA	
024C:	A1	LP21				029F:	62	TSIM	08		02F8:	D1	RC	
024D:	59	LDM				02A1:	38	JRZP	06 >02A8		02F9:	37	RTN	
024E:	13	LIQ	19			02A3:	60	ANIM	F7		02FA:	10	LIDP	663D
0250:	00	LII	06			02A5:	B1	LP31			02FD:	D6	TSID	01
0252:	08	MYW				02A6:	61	ORIM	01		02FF:	38	JRZP	05 >0305
0253:	A1	LP21				02A8:	22	LDR			0301:	D4	ANID	FE
0254:	60	ANIM	0F			02A9:	74	ADIA	02		0303:	2C	JRP	0C >0310
0256:	64	ANIA	F0			02AB:	10	LIDP	663C		0305:	97	LP17	
0258:	47	ORMA				02AE:	52	STD			0306:	00	LII	06
0259:	37	RTN				02AF:	B0	LP30			0308:	02	LIA	50
025A:	A1	LP21				02B0:	60	ANIM	00		030A:	0C	ADM	
025B:	59	LDM				02B2:	37	RTN			030B:	97	LP17	
025C:	13	LIQ	11			02B3:	99	LP19			030C:	60	ANIM	00
025E:	00	LII	06			02B4:	60	ANIM	F7		030E:	E4	CAL04	70
0260:	08	MYW				02B6:	B1	LP31			0310:	97	LP17	
0261:	A1	LP21				02B7:	62	TSIM	02		0311:	60	ANIM	00
0262:	60	ANIM	0F			02B9:	38	JRZP	04 >02BE		0313:	37	RTN	
0264:	64	ANIA	F0			02BB:	99	LP19			0314:	B1	LP31	
0266:	47	ORMA				02BC:	61	ORIM	08		0315:	62	TSIM	01
0267:	37	RTN				02BE:	02	LIA	00		0317:	28	JRNZP	04 >031C
0268:	91	LP11				02C0:	B1	LP31			0319:	61	ORIM	01
0269:	59	LDM				02C1:	62	TSIM	01		031B:	37	RTN	
026A:	60	ANIM	0F			02C3:	38	JRZP	03 >02C7		031C:	60	ANIM	FE
026C:	64	ANIA	F0			02C5:	02	LIA	08		031E:	37	RTN	
026E:	8A	LP0A				02C7:	91	LP11			031F:	99	LP19	
026F:	DB	EXAM				02C8:	60	ANIM	F7		0320:	59	LDM	

0321:	60	ANIM	0F	037E:	E3	CAL03	6B	03DE:	4E	WAIT	25
0323:	64	ANIA	F0	0380:	10	LIDP	28A0	03E0:	59	LDM	
0325:	34	PUSH		0383:	90	LP10		03E1:	D1	RC	
0326:	91	LP11		0384:	00	LII	07	03E2:	5A	SL	
0327:	59	LDM		0386:	18	MVWD		03E3:	DB	EXAM	
0328:	64	ANIA	F0	0387:	B1	LP31		03E4:	8A	LP0A	
032A:	99	LP19		0388:	60	ANIM	FE	03E5:	4C	INA	
032B:	47	ORMA		038A:	91	LP11		03E6:	5D	OUTA	
032C:	9F	LP1F		038B:	62	TSIM	08	03E7:	4E	WAIT	0D
032D:	13	LIO	17	038D:	38	JRZP	06 >0394	03E9:	DB	EXAM	
032F:	2C	JRP	01 >0331	038F:	60	ANIM	F7	03EA:	D1	RC	
0331:	00	LII	06	0391:	B1	LP31		03EB:	5A	SL	
0333:	0F	SBW		0392:	61	ORIM	01	03EC:	46	ANMA	
0334:	5B	POP		0394:	37	RTN		03ED:	DB	EXAM	
0335:	50	INCP		0395:	10	LIDP	28A8	03EE:	38	JRZP	03 >03F2
0336:	60	ANIM	0F	0398:	E3	CAL03	59	03F0:	E4	CAL04	25
0338:	47	ORMA		039A:	10	LIDP	28A8	03F2:	89	LP09	
0339:	37	RTN		039D:	2C	JRP	01 >039F	03F3:	59	LDM	
033A:	95	LP15		039F:	98	LP18		03F4:	88	LP08	
033B:	63	CPIM	99	03A0:	00	LII	07	03F5:	44	ADM	
033D:	28	JRNZP	11 >034F	03A2:	18	MVWD		03F6:	C9	DECL	
033F:	96	LP16		03A3:	B1	LP31		03F7:	12	LIP	5C
0340:	63	CPIM	99	03A4:	60	ANIM	FD	03F9:	29	JRNZM	1A >03E0
0342:	28	JRNZP	0B >034E	03A6:	99	LP19		03FB:	60	ANIM	00
0344:	02	LIA	01	03A7:	62	TSIM	08	03FD:	5D	OUTA	
0346:	00	LII	05	03A9:	38	JRZP	06 >03B0	03FE:	50	INCP	
0348:	0C	ADM		03AB:	60	ANIM	F7	03FF:	61	ORIM	01
0349:	97	LP17		03AD:	B1	LP31		0401:	89	LP09	
034A:	60	ANIM	00	03AE:	61	ORIM	02	0402:	02	LIA	03
034C:	E4	CAL04	70	03B0:	37	RTN		0404:	DB	EXAM	
034E:	37	RTN		03B1:	84	LP04		0405:	12	LIP	5D
034F:	63	CPIM	00	03B2:	10	LIDP	6627	0407:	DD	OUTB	
0351:	29	JRNZM	04 >034E	03B5:	1A	MVBD		0408:	59	LDM	
0353:	96	LP16		03B6:	00	LII	01	0409:	64	ANIA	03
0354:	63	CPIM	00	03B8:	85	LP05		040B:	44	ADM	
0356:	39	JRZM	0E >0349	03B9:	02	LIA	01	040C:	4E	WAIT	25
0358:	37	RTN		03BB:	0C	ADM		040E:	4C	INA	
0359:	99	LP19		03BC:	11	LIDL	27	040F:	38	JRZP	03 >0413
035A:	60	ANIM	F7	03BE:	84	LP04		0411:	E4	CAL04	25
035C:	B1	LP31		03BF:	1B	EXBD		0413:	88	LP08	
035D:	62	TSIM	02	03C0:	37	RTN		0414:	70	ADIM	08
035F:	38	JRZP	06 >0366	03C1:	02	LIA	34 INC 011	0416:	C9	DECL	
0361:	60	ANIM	FD	03C3:	6B	TEST	08	0417:	29	JRNZM	13 >0405
0363:	99	LP19		03C5:	28	JRNZP	5D >0423	0419:	12	LIP	5D
0364:	61	ORIM	08	03C7:	88	LP08		041B:	60	ANIM	F0
0366:	98	LP18		03C8:	60	ANIM	00	041D:	DD	OUTB	
0367:	00	LII	07	03CA:	89	LP09		041E:	DA	EXAB	
0369:	19	EXWD		03CB:	02	LIA	07	041F:	67	CPIA	36
036A:	37	RTN		03CD:	DB	EXAM		0421:	2A	JRNCP	02 >0424
036B:	91	LP11		03CE:	8A	LP0A		0423:	D0	SC	
036C:	60	ANIM	F7	03CF:	61	ORIM	FF	0424:	37	RTN	
036E:	B1	LP31		03D1:	03	LIB	FF	0425:	C2	INCB	
036F:	62	TSIM	01	03D3:	12	LIP	5C	0426:	2A	JRNCP	08 >042F
0371:	38	JRZP	06 >0378	03D5:	02	LIA	01	0428:	D1	RC	
0373:	60	ANIM	FE	03D7:	DB	EXAM		0429:	5A	SL	
0375:	91	LP11		03D8:	5D	OUTA		042A:	3A	JRCP	04 >042F
0376:	61	ORIM	08	03D9:	50	INCP		042C:	48	INCK	
0378:	90	LP10		03DA:	60	ANIM	F0	042D:	2D	JRM	06 >0428
0379:	2D	JRM	13 >0367	03DC:	DD	OUTB		042F:	02	LIA	35
037B:	10	LIDP	28A0	03DD:	51	DECP		0431:	28	JRNZP	03 >0435

0433:	88	LP08				0490:	7E	JPZ	0FB6			04F4:	91	LP11			
0434:	59	LDM				0493:	67	CPIA	80			04F5:	2D	JRM	28	>04CE	
0435:	DA	EXAB				0495:	38	JRZP	0A	>04A0		04F7:	E2	CAL02	CF		
0436:	37	RTN				0497:	62	TSIM	0F			04F9:	B1	LP31			
0437:	02	LIA	00	VCP	0FF	0499:	28	JRNZP	08	>04A2		04FA:	62	TSIM	01		
0439:	2C	JRP	03	>043D		049B:	50	INCP				04FC:	28	JRNZP	09	>0506	
043B:	02	LIA	01	VO	0FF	049C:	62	TSIM	F0			04FE:	E2	CAL02	EA		
043D:	12	LIP	5F	CCC	A	049E:	28	JRNZP	03	>04A2		0500:	E2	CAL02	01		
043F:	DB	EXAM				04A0:	E0	CAL00	E4			0502:	E2	CAL02	CF		
0440:	DF	OUTC				04A2:	D1	RC				0504:	2D	JRM	30	>04D5	
0441:	37	RTN				04A3:	37	RTN				0506:	E3	CAL03	1F		
0442:	02	LIA	A5			04A4:	92	LP12				0508:	3A	JRCP	05	>050E	
0444:	A0	LP20				04A5:	62	TSIM	F0			050A:	E3	CAL03	14		
0445:	00	LII	07			04A7:	29	JRNZM	22	>0486		050C:	2D	JRM	0D	>0500	
0447:	1E	FILM				04A9:	97	LP17				050E:	E2	CAL02	01		
0448:	37	RTN				04AA:	00	LII	05			0510:	E1	CAL01	05		
0449:	E4	CAL04	42	Point	0FF	04AC:	1D	SLW				0512:	E3	CAL03	1F		
044B:	A0	LP20				04AD:	00	LII	01			0514:	2D	JRM	15	>0500	
044C:	02	LIA	5A			04AF:	02	LIA	10			0516:	E1	CAL01	6E		
044E:	DB	EXAM				04B1:	0D	SNB				0518:	E2	CAL02	CF		
044F:	02	LIA	08			04B2:	2D	JRM	0F	>04A4		051A:	91	LP11			
0451:	E4	CAL04	3D			04B4:	E3	CAL03	14			051B:	13	LIQ	19		
0453:	02	LIA	08			04B6:	92	LP12				051D:	00	LII	01		
0455:	E4	CAL04	3D			04B7:	62	TSIM	F0			051F:	0E	ADW			
0457:	37	RTN				04B9:	38	JRZP	19	>04D3		0520:	A1	LP21			
0458:	E4	CAL04	42	Point	0FF	04BB:	9A	LP1A				0521:	13	LIQ	11		
045A:	6B	TEST	08			04BC:	62	TSIM	F0			0523:	00	LII	06		
045C:	28	JRNZP	03	>0460		04BE:	38	JRZP	16	>04D5		0525:	08	MVW			
045E:	E4	CAL04	4F			04C0:	E1	CAL01	29			0526:	E2	CAL02	68		
0460:	37	RTN				04C2:	02	LIA	10			0528:	91	LP11			
0461:	12	LIP	5C			04C4:	98	LP18				0529:	02	LIA	00		
0463:	60	ANIM	00			04C5:	62	TSIM	80			052B:	1E	FILM			
0465:	5D	OUTA				04C7:	38	JRZP	22	>04EA		052C:	89	LP09			
0466:	50	INCP				04C9:	9A	LP1A				052D:	02	LIA	0B		
0467:	60	ANIM	C0			04CA:	00	LII	05			052F:	DB	EXAM			
0469:	DD	OUTB				04CC:	1C	SRW				0530:	A7	LP27			
046A:	4E	WAIT	30			04CD:	99	LP19				0531:	59	LDM			
046C:	CC	INB				04CE:	00	LII	01			0532:	64	ANIA	0F		
046D:	66	TSIA	01			04D0:	0C	ADN				0534:	A2	LP22			
046F:	37	RTN				04D1:	2D	JRM	0E	>04C4		0535:	00	LII	05		
0470:	E0	CAL00	C7			04D3:	E1	CAL01	52			0537:	1C	SRW			
0472:	3A	JRCP	2D	>04A0		04D5:	91	LP11				0538:	40	INCI			
0474:	91	LP11				04D6:	62	TSIM	0F			0539:	43	DECA			
0475:	62	TSIM	0F			04D8:	28	JRNZP	0E	>04E7		053A:	3A	JRCP	07	>0542	
0477:	38	JRZP	2C	>04A4		04DA:	00	LII	04			053C:	97	LP17			
0479:	02	LIA	10			04DC:	50	INCP				053D:	13	LIQ	1F		
047B:	00	LII	01			04DD:	63	CPIM	00			053F:	0E	ADW			
047D:	0C	ADN				04DF:	28	JRNZP	07	>04E7		0540:	2D	JRM	08	>0539	
047E:	91	LP11				04E1:	41	DECI				0542:	C9	DECL			
047F:	00	LII	06			04E2:	2B	JRNCM	07	>04DC		0543:	3A	JRCP	05	>0549	
0481:	1C	SRW				04E4:	50	INCP				0545:	91	LP11			
0482:	91	LP11				04E5:	60	ANIM	00			0546:	1C	SRW			
0483:	59	LDM				04E7:	E4	CAL04	70			0547:	2D	JRM	18	>0530	
0484:	58	SWP				04E9:	37	RTN				0549:	E2	CAL02	7A		
0485:	DB	EXAM				04EA:	99	LP19				054B:	E4	CAL04	70		
0486:	90	LP10				04EB:	00	LII	01			054D:	37	RTN			
0487:	62	TSIM	F0			04ED:	0D	SNB				054E:	E0	CAL00	F8		
0489:	38	JRZP	18	>04A2		04EE:	3A	JRCP	08	>04F7		0550:	E2	CAL02	CF		
048B:	59	LDM				04F0:	92	LP12				0552:	E0	CAL00	EE		
048C:	64	ANIA	F0			04F1:	00	LII	05			0554:	99	LP19			
048E:	67	CPIA	10			04F3:	1C	SRW									

0555:	13	LIQ	11	05B2:	51	DECP		060A:	09	EXW	
0557:	00	LII	01	05B3:	61	ORIM	90	060B:	20	JRP	15 >0621
0559:	0F	SBW		05B5:	51	DECP		060D:	67	CPIA	10
055A:	90	LP10		05B6:	61	ORIM	99	060F:	38	JRZP	11 >0621
055B:	13	LIQ	18	05B8:	37	RTN		0611:	62	TSIM	0F
055D:	08	MYW		05B9:	B0	LP30		0613:	28	JRNZP	0D >0621
055E:	E2	CAL02	68	05BA:	60	ANIM	FE	0615:	91	LP11	
0560:	A1	LP21		05BC:	D1	RC		0616:	59	LDM	
0561:	00	LII	06	05BD:	D2	SR		0617:	64	ANIA	F0
0563:	02	LIA	00	05BE:	2A	JRNCP	03 >05C2	0619:	58	SWP	
0565:	1E	FILM		05C0:	61	ORIM	01	061A:	03	LIB	09
0566:	89	LP09		05C2:	80	LP00		061C:	83	LP03	
0567:	02	LIA	0C	05C3:	DB	EXAM		061D:	45	SBM	
0569:	DB	EXAM		05C4:	59	LDM		061E:	59	LDM	
056A:	A7	LP27		05C5:	01	LIJ	17	061F:	E5	CAL05	B9
056B:	1D	SLW		05C7:	81	LP01		0621:	E1	CAL01	D9
056C:	9F	LP1F		05C8:	45	SBM		0623:	E4	CAL04	70
056D:	13	LIQ	17	05C9:	59	LDM		0625:	37	RTN	
056F:	0F	SBW		05CA:	31	STQ		0626:	E0	CAL00	DB
0570:	3A	JRCP	0E >057F	05CB:	74	ADIA	08	0628:	10	LIDP	207C
0572:	A7	LP27		05CD:	34	PUSH		062B:	D6	TSID	02
0573:	70	ADIM	01	05CE:	B0	LP30		062D:	7E	JPZ	00F6
0575:	59	LDM		05CF:	01	LIJ	01	0630:	D6	TSID	10
0576:	64	ANIA	0F	05D1:	62	TSIM	01	0632:	28	JRNZP	05 >0638
0578:	67	CPIA	0A	05D3:	30	STP		0634:	E7	CAL07	12
057A:	29	JRNZM	0F >056C	05D4:	38	JRZP	04 >05D9	0636:	D0	SC	
057C:	79	JP	0FB1	05D6:	09	EXW		0637:	37	RTN	
057F:	9F	LP1F		05D7:	5B	POP		0638:	98	LP18	
0580:	13	LIQ	17	05D8:	37	RTN		0639:	61	ORIM	99
0582:	0E	ADW		05D9:	59	LDM		063B:	50	INCP	
0583:	9F	LP1F		05DA:	DA	EXAB		063C:	61	ORIM	90
0584:	1D	SLW		05DB:	60	ANIM	0F	063E:	50	INCP	
0585:	C9	DECL		05DD:	09	EXW		063F:	61	ORIM	90
0586:	2B	JRNCM	1D >056A	05DE:	5B	POP		0641:	D0	SC	
0588:	91	LP11		05DF:	30	STP		0642:	37	RTN	
0589:	13	LIQ	21	05E0:	DA	EXAB		0643:	98	LP18	
058B:	08	MYW		05E1:	64	ANIA	F0	0644:	62	TSIM	F0
058C:	2D	JRM	44 >0549	05E3:	DB	EXAM		0646:	28	JRNZP	1B >0662
058E:	10	LIDP	28B8	05E4:	37	RTN		0648:	59	LDM	
0591:	90	LP10		05E5:	E1	CAL01	6E	0649:	99	LP19	
0592:	00	LII	07	05E7:	E1	CAL01	18	064A:	67	CPIA	01
0594:	19	EXWD		05E9:	2C	JRP	05 >05EF	064C:	28	JRNZP	0C >0659
0595:	10	LIDP	28B8	05EB:	E1	CAL01	6E	064E:	59	LDM	
0598:	90	LP10		05ED:	E0	CAL00	EE	064F:	64	ANIA	F0
0599:	18	MYWD		05EF:	E1	CAL01	0B	0651:	67	CPIA	30
059A:	90	LP10		05F1:	90	LP10		0653:	2A	JRNCP	0E >0662
059B:	02	LIA	99	05F2:	59	LDM		0655:	58	SWP	
059D:	DB	EXAM		05F3:	64	ANIA	F0	0656:	74	ADIA	0A
059E:	91	LP11		05F5:	67	CPIA	90	0658:	37	RTN	
059F:	02	LIA	90	05F7:	28	JRNZP	15 >060D	0659:	67	CPIA	00
05A1:	DB	EXAM		05F9:	E0	CAL00	EE	065B:	28	JRNZP	06 >0662
05A2:	37	RTN		05FB:	90	LP10		065D:	59	LDM	
05A3:	E0	CAL00	E4	05FC:	59	LDM		065E:	58	SWP	
05A5:	10	LIDP	6627	05FD:	64	ANIA	F0	065F:	64	ANIA	0F
05A8:	92	LP12		05FF:	67	CPIA	90	0661:	37	RTN	
05A9:	00	LII	01	0601:	02	LIA	0A	0662:	02	LIA	0E
05AB:	2D	JRM	13 >0599	0603:	28	JRNZP	1B >061F	0664:	37	RTN	
05AD:	E0	CAL00	DB	0605:	92	LP12		0665:	98	LP18	
05AF:	9A	LP1A		0606:	13	LIQ	1A	0666:	62	TSIM	FF
05B0:	61	ORIM	60	0608:	00	LII	05	0668:	28	JRNZP	0B >0674

066A:	99	LP19				0712:	03	LIB	07
066B:	59	LDM				0714:	02	LIA	98
066C:	64	ANIA	F0			0716:	2D	JRM	0D >070A
066E:	67	CPIA	80			0718:	30	STP	
0670:	2A	JRNCP	03	>0674		0719:	10	LIDP	2999
0672:	58	SWP				071C:	56	READ	
0673:	37	RTN				071D:	64	ANIA	41
0674:	02	LIA	07			071F:	39	JRZM	26 >06FA
0676:	37	RTN				0721:	85	LP05	
0677:	8D	LP0D				0722:	15	SBB	
0678:	60	ANIM	00			0723:	82	LP02	
067A:	E1	CAL01	08			0724:	43	DECA	
067C:	E1	CAL01	05			0725:	21	LDO	
067E:	37	RTN				0726:	37	RTN	
067F:	80	LP30				0727:	37	RTN	
0680:	61	ORIM	08			0728:	82	LP02	
0682:	2C	JRP	04	>0687		0729:	64	ANIA	43
0684:	80	LP30				072B:	40	INCI	
0685:	60	ANIM	F7			072C:	77	77	
0687:	A1	LP21				072D:	47	ORMA	
0688:	59	LDM				072E:	93	LP13	
0689:	60	ANIM	0F			072F:	19	EXWD	
068B:	64	ANIA	F0			0730:	43	DECA	
068D:	34	PUSH				0731:	42	INCA	
068E:	A0	LP20				0732:	72	72	
068F:	00	LII	06			0733:	76	76	
0691:	62	TSIM	10			0734:	86	LP06	
0693:	38	JRZP	03	>0697		0735:	27	DYS	
0695:	A1	LP21				0736:	43	DECA	
0696:	1C	SRW				0737:	42	INCA	
0697:	A0	LP20				0738:	92	LP12	
0698:	59	LDM				0739:	31	STQ	
0699:	58	SWP				073A:	04	IX	
069A:	64	ANIA	0F			073B:	45	SBM	
069C:	D1	RC				073C:	43	DECA	
069D:	D2	SR				073D:	42	INCA	
069E:	80	LP00				073E:	94	LP14	
069F:	45	SBM				073F:	26	IYS	
06A0:	59	LDM				0740:	47	ORMA	
06A1:	74	ADIA	21			0741:	56	READ	
06A3:	31	STQ				0742:	43	DECA	
06A4:	99	LP19				0743:	42	INCA	
06A5:	59	LDM				0744:	94	LP14	
06A6:	64	ANIA	F0			0745:	46	ANMA	
06A8:	34	PUSH				0746:	01	LIJ	89
06A9:	80	LP30				0748:	43	DECA	
06AA:	62	TSIM	08			0749:	42	INCA	
06AC:	9F	LP1F				074A:	94	LP14	
06AD:	38	JRZP	21	>06CF		074B:	47	ORMA	
06AF:	0E	ADM				074C:	97	LP17	
06B0:	2A	JRNCP	15	>06C6		074D:	32	STR	
06B2:	20	LDP				074E:	43	DECA	
06B3:	75	SBI A	19			074F:	42	INCA	
06B5:	80	LP00				0750:	94	LP14	
06B6:	DB	EXAM				0751:	48	INCK	
06B7:	59	LDM				0752:	16	16	
06B8:	74	ADIA	19			0753:	86	LP06	
06BA:	80	LP30				0754:	43	DECA	
06BB:	62	TSIM	08			0755:	42	INCA	
06BD:	30	STP				0756:	94	LP14	
06BE:	02	LIA	01						
06C0:	38	JRZP	04	>06C5					
06C2:	0C	ADM							
06C3:	2C	JRP	02	>06C6					
06C5:	0D	SBN							
06C6:	5B	POP							
06C7:	99	LP19							
06C8:	60	ANIM	0F						
06CA:	47	ORMA							
06CB:	5B	POP							
06CC:	A1	LP21							
06CD:	47	ORMA							
06CE:	37	RTN							
06CF:	0F	SBW							
06D0:	2D	JRM	21	>06B0					
06D2:	74	ADIA	0D						
06D4:	67	CPIA	14						
06D6:	3A	JRCP	0B	>06E2					
06D8:	02	LIA	13						
06DA:	2C	JRP	07	>06E2					
06DC:	67	CPIA	0D						
06DE:	3A	JRCP	03	>06E2					
06E0:	02	LIA	0C						
06E2:	34	PUSH							
06E3:	85	LP05							
06E4:	02	LIA	07						
06E6:	DB	EXAM							
06E7:	84	LP04							
06E8:	02	LIA	18						
06EA:	DB	EXAM							
06EB:	5B	POP							
06EC:	D1	RC							
06ED:	5A	SL							
06EE:	DA	EXAB							
06EF:	83	LP03							
06F0:	59	LDM							
06F1:	5A	SL							
06F2:	44	ADM							
06F3:	59	LDM							
06F4:	03	LIB	00						
06F6:	84	LP04							
06F7:	14	ADB							
06F8:	13	LIQ	04						
06FA:	82	LP02							
06FB:	01	LIJ	01						
06FD:	0A	MYB							
06FE:	9A	LP1A							
06FF:	00	LII	05						
0701:	35	DATA							
0702:	99	LP19							
0703:	60	ANIM	F0						
0705:	37	RTN							
0706:	03	LIB	07						
0708:	02	LIA	90						
070A:	98	LP18							
070B:	00	LII	07						
070D:	35	DATA							
070E:	81	LP31							
070F:	60	ANIM	FD						
0711:	37	RTN							

0757: 48 INCK
 0758: 18 MVWD
 0759: 82 LP02
 075A: 43 DECA
 075B: 42 INCA
 075C: 94 LP14
 075D: 48 INCK
 075E: 19 EXWD
 075F: 01 LIJ 43
 0761: 42 INCA
 0762: 94 LP14
 0763: 48 INCK
 0764: 19 EXWD
 0765: 03 LIB 45
 0767: 00 LII 00
 0769: 00 LII 00
 076B: 00 LII 57
 076D: 10 LIDP 5931
 0770: 37 RTN
 0771: 50 INCP
 0772: 57 LDD
 0773: 29 JRNZM 38 >0730
 0775: 69 69
 0776: 76 76
 0777: 83 LP03
 0778: 57 LDD
 0779: 29 JRNZM 57 >0723
 077B: 60 ANIM 41
 077D: 45 SBM
 077E: 57 LDD
 077F: 29 JRNZM 57 >0729
 0781: 79 JP 3221
 0784: 57 LDD
 0785: 29 JRNZM 57 >072F
 0787: 79 JP 5111
 078A: 57 LDD
 078B: 29 JRNZM 57 >0735
 078D: 79 JP 5131
 0790: 99 LP19
 0791: 90 LP10
 0792: 43 DECA
 0793: 42 INCA
 0794: 94 LP14
 0795: 48 INCK
 0796: 19 EXWD
 0797: 03 LIB 00
 0799: 10 LIDP 5729
 079C: 57 LDD
 079D: 79 JP 5131
 07A0: 00 LII 00
 07A2: 31 STQ
 07A3: 41 DECI
 07A4: 59 LDM
 07A5: 26 IYS
 07A6: 53 MVDM
 07A7: 59 LDM
 07A8: F3 CAL13 DD
 07AA: 3A JRCP 08 >07B3
 07AC: 8E LP0E
 07AD: 01 LIJ 01

07AF: 13 LIO 18
 07B1: 0A MYB
 07B2: D1 RC
 07B3: 37 RTN
 07B4: 10 LIDP 6630
 07B7: 80 LP30
 07B8: 1B EXBD
 07B9: 37 RTN
 07BA: 10 LIDP 6630
 07BD: 80 LP30
 07BE: 1A MVBD
 07BF: 37 RTN
 07C0: 10 LIDP 66E1
 07C3: 86 LP06
 07C4: 1A MVBD
 07C5: 06 IY
 07C6: 10 LIDP 66E3
 07C9: 86 LP06
 07CA: 1B EXBD
 07CB: 11 LIDL E3
 07CD: 86 LP06
 07CE: 1A MVBD
 07CF: 10 LIDP 66D9
 07D2: D4 ANID DF
 07D4: 37 RTN
 07D5: 24 IXL
 07D6: 67 CPIA 0D
 07D8: 38 JRZP 0C >07E5
 07DA: B6 LP36
 07DB: 62 TSIM 02
 07DD: 28 JRNZP 05 >07E3
 07DF: 67 CPIA 3A
 07E1: 38 JRZP 03 >07E5
 07E3: D0 SC
 07E4: 37 RTN
 07E5: 05 DX
 07E6: E8 CAL08 ED
 07E8: D1 RC
 07E9: 37 RTN
 07EA: FC CAL1C 2B
 07EC: B6 LP36
 07ED: 60 ANIM FB
 07EF: F0 CAL10 4D
 07F1: FB CAL1B A7
 07F3: B5 LP35
 07F4: 60 ANIM 00
 07F6: B7 LP37
 07F7: 60 ANIM 3F
 07F9: F3 CAL13 57
 07FB: D4 ANID E1
 07FD: FB CAL1B B7
 07FF: D4 ANID 00
 0801: 11 LIDL 2C
 0803: D4 ANID 00
 0805: F3 CAL13 53
 0807: D4 ANID 9A
 0809: 11 LIDL 7C
 080B: D4 ANID 07
 080D: 10 LIDP 66D7
 0810: D4 ANID F3

0812: 11 LIDL D8
 0814: D4 ANID CE
 0816: D5 ORID 06
 0818: 11 LIDL D9
 081A: D4 ANID F9
 081C: 11 LIDL DA
 081E: D4 ANID BF
 0820: 78 CALL 0018
 0823: F3 CAL13 B7
 0825: 04 IX
 0826: 04 IX
 0827: 04 IX
 0828: 10 LIDP 66FE
 082B: 84 LP04
 082C: 1B EXBD
 082D: 37 RTN
 082E: D2 SR *Reeper*
 082F: 12 LIP 5F
 0831: 60 ANIM 01
 0833: 2A JRNCP 05 >0839
 0835: 61 ORIM 30
 0837: 2C JRP 03 >083B
 0839: 61 ORIM 20
 083B: 6B TEST 02
 083D: 39 JRZM 03 >083B
 083F: DF OUTC
 0840: 37 RTN
 0841: 8B LP0B
 0842: 59 LDM
 0843: 58 SWP
 0844: 03 LIB 01
 0846: D1 RC
 0847: E8 CAL08 2F
 0849: E8 CAL08 2E
 084B: E8 CAL08 2E
 084D: E8 CAL08 2E
 084F: E8 CAL08 2E
 0851: D0 SC
 0852: E8 CAL08 2F
 0854: C3 DECB
 0855: 2B JRNCM 10 >0846
 0857: 37 RTN
 0858: 34 PUSH
 0859: 8B LP0B
 085A: 64 ANIA 0F
 085C: 44 ADM
 085D: 5B POP
 085E: 34 PUSH
 085F: 58 SWP
 0860: 64 ANIA 0F
 0862: C4 ADCM
 0863: 5B POP
 0864: 37 RTN
 0865: 03 LIB 03
 0867: 2C JRP 03 >086B
 0869: 03 LIB 0F
 086B: 12 LIP 5F
 086D: 60 ANIM 01
 086F: 61 ORIM 30
 0871: DF OUTC

0872:	02	LIA	F9		
0874:	6B	TEST	02		
0876:	39	JRZM	03	>0874	
0878:	43	DECA			
0879:	2B	JRNCM	06	>0874	
087B:	03	DECB			
087C:	3A	JRCP	07	>0884	
087E:	6B	TEST	08		
0880:	39	JRZM	0F	>0872	
0882:	2C	JRP	33	>0886	
0884:	37	RTN			
0885:	22	LDR			
0886:	DA	EXAB			
0887:	02	LIA	20		
0889:	32	STR			
088A:	98	LP18			
088B:	5B	POP			
088C:	07	CPMA			
088D:	28	JRNZP	0B	>0899	
088F:	50	INCP			
0890:	20	LDP			
0891:	67	CPIA	20		
0893:	29	JRNZM	09	>088B	
0895:	D0	SC			
0896:	DA	EXAB			
0897:	32	STR			
0898:	37	RTN			
0899:	D1	RC			
089A:	2D	JRM	05	>0896	
089C:	84	LP04			
089D:	59	LDM			
089E:	8E	LP0E			
089F:	07	CPMA			
08A0:	28	JRNZP	09	>08AA	
08A2:	85	LP05			
08A3:	59	LDM			
08A4:	8F	LP0F			
08A5:	07	CPMA			
08A6:	28	JRNZP	03	>08AA	
08A8:	D0	SC			
08A9:	37	RTN			
08AA:	D1	RC			
08AB:	37	RTN			
08AC:	A0	LP20			
08AD:	00	LII	07		
08AF:	02	LIA	00		
08B1:	1E	FILM			
08B2:	A0	LP20			
08B3:	61	ORIM	F5		
08B5:	37	RTN			
08B6:	78	CALL	8036		
08B9:	10	LIDP	6637		
08BC:	57	LDD			
08BD:	32	STR			
08BE:	E8	CAL08	D8		
08C0:	B5	LP35			
08C1:	61	ORIM	20		
08C3:	D1	RC			
08C4:	E8	CAL08	FA		
08C6:	11	LIDL	D7		
08C8:	D4	ANID	DB		
08CA:	37	RTN			
08CB:	6B	TEST	02		
08CD:	39	JRZM	03	>08CB	
08CF:	6B	TEST	02		
08D1:	39	JRZM	03	>08CF	
08D3:	6B	TEST	02		
08D5:	39	JRZM	03	>08D3	
08D7:	37	RTN			
08D8:	10	LIDP	66E3		
08DB:	86	LP06			
08DC:	1A	MVBD			
08DD:	07	DY			
08DE:	02	LIA	FF		
08E0:	26	IYS			
08E1:	DA	EXAB			
08E2:	37	RTN			
08E3:	E8	CAL08	ED		
08E5:	22	LDR			
08E6:	74	ADIA	02		
08E8:	10	LIDP	6637		
08EB:	52	STD			
08EC:	37	RTN			
08ED:	84	LP04			
08EE:	59	LDM			
08EF:	10	LIDP	6639		
08F2:	52	STD			
08F3:	85	LP05			
08F4:	59	LDM			
08F5:	10	LIDP	663A		
08F8:	52	STD			
08F9:	37	RTN			
08FA:	10	LIDP	6639		
08FD:	57	LDD			
08FE:	84	LP04			
08FF:	DB	EXAM			
0900:	10	LIDP	663A		
0903:	57	LDD			
0904:	85	LP05			
0905:	DB	EXAM			
0906:	01	LIJ	01		
0908:	37	RTN			
0909:	10	LIDP	66D7		
090C:	D4	ANID	FB		
090E:	37	RTN			
090F:	02	LIA	07		
0911:	88	LP08			
0912:	DB	EXAM			
0913:	8B	LP0B			
0914:	60	ANIM	00		
0916:	37	RTN			
0917:	84	LP04			
0918:	10	LIDP	66FC		
091B:	1B	EXBD			
091C:	84	LP04			
091D:	10	LIDP	66FC		
0920:	1A	MVBD			
0921:	37	RTN			
0922:	02	LIA	19		
0924:	89	LP09			
0925:	DB	EXAM			
0926:	98	LP18			
0927:	59	LDM			
0928:	89	LP09			
0929:	45	SRM			
092A:	37	RTN			
092B:	E9	CAL09	0F		
092D:	25	DXL			
092E:	E8	CAL08	58		
0930:	E8	CAL08	44		
0932:	E8	CAL08	CB		
0934:	49	DECK			
0935:	2B	JRNCM	09	>092D	
0937:	E8	CAL08	41		
0939:	37	RTN			
093A:	E9	CAL09	41		
093C:	E9	CAL09	57		
093E:	3B	JRCM	05	>093A	
0940:	37	RTN			
0941:	6B	TEST	08		
0943:	38	JRZP	1F	>0963	
0945:	4E	WAIT	05		
0947:	6B	TEST	08		
0949:	38	JRZP	19	>0963	
094B:	E9	CAL09	A3		
094D:	79	JP	08B9		
0950:	E9	CAL09	41		
0952:	E9	CAL09	67		
0954:	3B	JRCM	05	>0950	
0956:	37	RTN			
0957:	CC	INB			
0958:	66	TSIA	40		
095A:	28	JRNZP	0A	>0965	
095C:	4E	WAIT	00		
095E:	CC	INB			
095F:	66	TSIA	40		
0961:	28	JRNZP	03	>0965	
0963:	D1	RC			
0964:	37	RTN			
0965:	D0	SC			
0966:	37	RTN			
0967:	CC	INB			
0968:	66	TSIA	40		
096A:	39	JRZM	06	>0965	
096C:	4E	WAIT	00		
096E:	CC	INB			
096F:	66	TSIA	40		
0971:	39	JRZM	0D	>0965	
0973:	D1	RC			
0974:	37	RTN			
0975:	12	LIP	5D		
0977:	60	ANIM	0F		
0979:	61	ORIM	20		
097B:	DD	OUTB			
097C:	4E	WAIT	DC		
097E:	37	RTN			
097F:	12	LIP	5D		
0981:	60	ANIM	0F		
0983:	2D	JRM	09	>097B	

0985:	4E	WAIT	08	09EF:	2A	JRNCP	52	>0A42	0A46:	67	CPIA	AF
0987:	12	LIP	5E	09F1:	67	CPIA	60		0A48:	38	JRZP	0B >0A54
0989:	60	ANIM	FC	09F3:	2A	JRNCP	4A	>0A3E	0A4A:	2A	JRNCP	0D >0A58
098B:	5F	OUTF		09F5:	67	CPIA	5B		0A4C:	EA	CAL0A	7F
098C:	4E	WAIT	FF	09F7:	2A	JRNCP	0D	>0A05	0A4E:	2C	JRP	17 >0A66
098E:	4E	WAIT	FF	09F9:	67	CPIA	41		0A50:	EA	CAL0A	DA
0990:	4E	WAIT	FF	09FB:	2A	JRNCP	64	>0A60	0A52:	2C	JRP	13 >0A66
0992:	4E	WAIT	FF	09FD:	67	CPIA	3A		0A54:	EA	CAL0A	EE
0994:	51	DECP		09FF:	2A	JRNCP	05	>0A05	0A56:	2C	JRP	0F >0A66
0995:	60	ANIM	0F	0A01:	67	CPIA	30		0A58:	EB	CAL0B	C7
0997:	61	ORIM	20	0A03:	2A	JRNCP	60	>0A64	0A5A:	2C	JRP	0B >0A66
0999:	DD	OUTB		0A05:	7A	7A			0A5C:	EB	CAL0B	0F
099A:	61	ORIM	30	0A06:	10	LIDP	0A3C		0A5E:	2C	JRP	07 >0A66
099C:	DD	OUTB		0A09:	69	69			0A60:	EB	CAL0B	50
099D:	4E	WAIT	FF	0A0A:	2E	2E			0A62:	2C	JRP	03 >0A66
099F:	37	RTN		0A0B:	0B	EXB			0A64:	EB	CAL0B	B3
09A0:	78	CALL	8036	0A0C:	B3	LP33			0A66:	3A	JRCP	17 >0A7E
09A3:	12	LIP	5E	0A0D:	2B	JRNCP	0B	>0A03	0A68:	B3	LP33	
09A5:	60	ANIM	FC	0A0F:	1E	FILM			0A69:	62	TSIM	01
09A7:	5F	OUTF		0A10:	2D	JRM	0B	>0A06	0A6B:	39	JRZM	0C >09E0
09A8:	50	INCP		0A12:	1E	FILM			0A6D:	F3	CAL13	40
09A9:	60	ANIM	01	0A13:	2A	JRNCP	0A	>0A1E	0A6F:	2A	JRNCP	09 >0A79
09AB:	DF	OUTC		0A15:	96	LP16			0A71:	E2	CAL02	9B
09AC:	E9	CAL09	75	0A16:	2F	LOOP	0A	>0A0D	0A73:	E2	CAL02	FA
09AE:	2D	JRM	30 >097F	0A18:	96	LP16			0A75:	E2	CAL02	BE
09B0:	02	LIA	31	0A19:	5E	5E			0A77:	3A	JRCP	06 >0A7E
09B2:	E8	CAL08	CF	0A1A:	0A	MYB			0A79:	05	DX	
09B4:	43	DECA		0A1B:	96	LP16			0A7A:	D1	RC	
09B5:	2B	JRNCP	04 >09B2	0A1C:	3C	3C			0A7B:	B7	LP37	
09B7:	37	RTN		0A1D:	0B	EXB			0A7C:	60	ANIM	3F
09B8:	B7	LP37		0A1E:	81	LP01			0A7E:	37	RTN	
09B9:	60	ANIM	C0	0A1F:	3D	3D			0A7F:	67	CPIA	A1
09BB:	61	ORIM	01	0A20:	0B	EXB			0A81:	38	JRZP	14 >0A96
09BD:	62	TSIM	40	0A21:	81	LP01			0A83:	67	CPIA	A2
09BF:	28	JRNZP	07 >09C7	0A22:	3E	3E			0A85:	38	JRZP	10 >0A96
09C1:	10	LIDP	66F7	0A23:	0B	EXB			0A87:	67	CPIA	AD
09C4:	02	LIA	60	0A24:	81	LP01			0A89:	38	JRZP	17 >0AA1
09C6:	52	STD		0A25:	2C	JRP	0B	>0A31	0A8B:	EF	CAL0F	9C
09C7:	10	LIDP	66F5	0A27:	E7	CAL07	28		0A8D:	3A	JRCP	07 >0A95
09CA:	02	LIA	80	0A29:	0C	ADN			0A8F:	EF	CAL0F	F0
09CC:	52	STD		0A2A:	11	LIDL	29		0A91:	61	ORIM	10
09CD:	11	LIDL	F6	0A2C:	0C	ADN			0A93:	EF	CAL0F	06
09CF:	02	LIA	80	0A2D:	1C	SRW			0A95:	37	RTN	
09D1:	52	STD		0A2E:	22	LDR			0A96:	EF	CAL0F	AB
09D2:	11	LIDL	D6	0A2F:	0D	SBN			0A98:	3A	JRCP	1D >0AB6
09D4:	02	LIA	90	0A30:	33	33			0A9A:	05	DX	
09D6:	52	STD		0A31:	FC	CAL1C	0B		0A9B:	EF	CAL0F	EB
09D7:	B3	LP33		0A33:	C2	INCB			0A9D:	24	IXL	
09D8:	60	ANIM	FE	0A34:	FB	CAL1B	0B		0A9E:	79	JP	0EA9
09DA:	10	LIDP	66F2	0A36:	0F	SBN			0AA1:	EF	CAL0F	A5
09DD:	02	LIA	01	0A37:	26	IYS			0AA3:	3A	JRCP	12 >0AB6
09DF:	52	STD		0A38:	0D	SBN			0AA5:	EF	CAL0F	6A
09E0:	24	IXL		0A39:	5C	5C			0AA7:	EF	CAL0F	65
09E1:	67	CPIA	AE	0A3A:	0B	EXB			0AA9:	E3	CAL03	C1
09E3:	38	JRZP	78 >0A5C	0A3B:	C7	CPMA			0AAB:	3A	JRCP	0B >0AB7
09E5:	67	CPIA	FE	0A3C:	2C	JRP	29	>0A66	0AAD:	EF	CAL0F	65
09E7:	38	JRZP	56 >0A3E	0A3E:	EF	CAL0F	C1		0AAF:	02	LIA	00
09E9:	67	CPIA	FB	0A40:	2C	JRP	3D	>0A7E	0AB1:	97	LP17	
09EB:	2A	JRNCP	19 >0A05	0A42:	67	CPIA	BA		0AB2:	DB	EXAM	
09ED:	67	CPIA	91	0A44:	38	JRZP	0B	>0A50	0AB3:	EF	CAL0F	E6

0AB5:	D1	RC	
0AB6:	37	RTN	
0AB7:	74	ADIA	A0
0AB9:	03	LIB	83
0ABB:	2A	JRNCP	02 >0ABE
0ABD:	C2	INCB	
0ABE:	E2	CAL02	2E
0AC0:	24	IXL	
0AC1:	EF	CAL0F	65
0AC3:	67	CPIA	60
0AC5:	2B	JRNCM	17 >0AAF
0AC7:	67	CPIA	20
0AC9:	3B	JRCM	1B >0AAF
0ACB:	86	LP06	
0ACC:	63	CPIM	AF
0ACE:	7D	JPNC	0FC0
0AD1:	26	IYS	
0AD2:	02	LIA	01
0AD4:	88	LP08	
0AD5:	DB	EXAM	
0AD6:	ED	CAL0D	4F
0AD8:	D1	RC	
0AD9:	37	RTN	
0ADA:	EF	CAL0F	9C
0ADC:	3A	JRCP	31 >0B0E
0ADE:	10	LIDP	66E1
0AE1:	8A	LP0A	
0AE2:	1A	MVBD	
0AE3:	02	LIA	02
0AE5:	03	LIB	00
0AE7:	8A	LP0A	
0AE8:	14	ADB	
0AE9:	10	LIDP	6623
0AEC:	2C	JRP	0D >0AFA
0AEE:	EF	CAL0F	9C
0AF0:	3A	JRCP	1D >0B0E
0AF2:	10	LIDP	66FC
0AF5:	8A	LP0A	
0AF6:	1A	MVBD	
0AF7:	10	LIDP	66E3
0AFA:	82	LP02	
0AFB:	1A	MVBD	
0AFC:	8A	LP0A	
0AFD:	15	SBB	
0AFE:	82	LP02	
0AFF:	13	LIQ	0A
0B01:	0A	MVB	
0B02:	43	DECA	
0B03:	2A	JRNCP	02 >0B06
0B05:	C3	DECB	
0B06:	98	LP18	
0B07:	13	LIQ	02
0B09:	0A	MVB	
0B0A:	F0	CAL10	E0
0B0C:	EF	CAL0F	E6
0B0E:	37	RTN	
0B0F:	EF	CAL0F	9C
0B11:	3A	JRCP	0B >0B1D
0B13:	02	LIA	A0
0B15:	03	LIB	07
0B17:	90	LP10	
0B18:	00	LII	07
0B1A:	35	DATA	
0B1B:	EF	CAL0F	E6
0B1D:	37	RTN	
0B1E:	EF	CAL0F	AB
0B20:	05	DX	
0B21:	2B	JRNCM	87 >0A9B
0B23:	B7	LP37	
0B24:	62	TSIM	04
0B26:	38	JRZP	20 >0B47
0B28:	10	LIDP	66F6
0B2B:	57	LDD	
0B2C:	43	DECA	
0B2D:	52	STD	
0B2E:	03	LIB	28
0B30:	E2	CAL02	42
0B32:	06	IY	
0B33:	57	LDD	
0B34:	67	CPIA	65
0B36:	28	JRNZP	10 >0B47
0B38:	24	IXL	
0B39:	67	CPIA	2D
0B3B:	28	JRNZP	10 >0B4C
0B3D:	02	LIA	64
0B3F:	EF	CAL0F	06
0B41:	EF	CAL0F	F0
0B43:	61	ORIM	04
0B45:	D1	RC	
0B46:	37	RTN	
0B47:	24	IXL	
0B48:	67	CPIA	2B
0B4A:	39	JRZM	0E >0B3D
0B4C:	02	LIA	65
0B4E:	2D	JRM	10 >0B3F
0B50:	EF	CAL0F	9C
0B52:	3A	JRCP	15 >0B68
0B54:	05	DX	
0B55:	F7	CAL17	32
0B57:	3A	JRCP	10 >0B68
0B59:	B3	LP33	
0B5A:	62	TSIM	08
0B5C:	28	JRNZP	0C >0B69
0B5E:	60	ANIM	9D
0B60:	61	ORIM	20
0B62:	F7	CAL17	D5
0B64:	3A	JRCP	03 >0B68
0B66:	EF	CAL0F	E6
0B68:	37	RTN	
0B69:	EF	CAL0F	08
0B6B:	3A	JRCP	12 >0B7E
0B6D:	10	LIDP	66D6
0B70:	57	LDD	
0B71:	03	LIB	28
0B73:	E2	CAL02	42
0B75:	8B	LP0B	
0B76:	59	LDM	
0B77:	26	IYS	
0B78:	06	IY	
0B79:	86	LP06	
0B7A:	10	LIDP	66D6
0B7D:	53	MVDM	
0B7E:	EF	CAL0F	E1
0B80:	37	RTN	
0B81:	EF	CAL0F	AB
0B83:	3A	JRCP	10 >0B94
0B85:	67	CPIA	3C
0B87:	28	JRNZP	0D >0B95
0B89:	24	IXL	
0B8A:	67	CPIA	3D
0B8C:	28	JRNZP	19 >0BA6
0B8E:	02	LIA	62
0B90:	EF	CAL0F	EB
0B92:	EE	CAL0E	A9
0B94:	37	RTN	
0B95:	67	CPIA	3E
0B97:	29	JRNZM	08 >0B90
0B99:	24	IXL	
0B9A:	67	CPIA	3D
0B9C:	38	JRZP	05 >0BA2
0B9E:	02	LIA	3E
0BA0:	2C	JRP	0F >0BB0
0BA2:	02	LIA	61
0BA4:	2D	JRM	15 >0B90
0BA6:	67	CPIA	3E
0BA8:	28	JRNZP	05 >0BAE
0BAA:	02	LIA	63
0BAC:	2D	JRM	1D >0B90
0BAE:	02	LIA	3C
0BB0:	05	DX	
0BB1:	2D	JRM	22 >0B90
0BB3:	EF	CAL0F	9C
0BB5:	3A	JRCP	0B >0BC1
0BB7:	05	DX	
0BB8:	B3	LP33	
0BB9:	60	ANIM	FD
0BBB:	ED	CAL0D	A7
0BBD:	3A	JRCP	03 >0BC1
0BBF:	EF	CAL0F	E6
0BC1:	37	RTN	
0BC2:	02	LIA	94
0BC4:	79	JP	0A7F
0BC7:	EF	CAL0F	AB
0BC9:	3A	JRCP	0D >0BD7
0BCB:	EF	CAL0F	FD
0BCD:	28	JRNZP	0A >0BD8
0BCF:	EF	CAL0F	F4
0BD1:	7C	JPNZ	0FC1
0BD4:	B3	LP33	
0BD5:	61	ORIM	01
0BD7:	37	RTN	
0BD8:	F0	CAL10	04
0BDA:	7D	JPNC	0FC1
0BDD:	EF	CAL0F	65
0BDF:	78	CALL	8009
0BE2:	EF	CAL0F	65
0BE4:	2B	JRNCM	1A >0BCB
0BE6:	37	RTN	
0BE7:	EF	CAL0F	AB
0BE9:	3A	JRCP	15 >0BFF

0BE8:	EF CAL0F	FD	
0BED:	38 JRZP	12	>0C00
0BEF:	F0 CAL10	04	
0BF1:	3A JRCP	15	>0C07
0BF3:	05 DX		
0BF4:	24 IXL		
0BF5:	EF CAL0F	06	
0BF7:	3A JRCP	07	>0BFF
0BF9:	EE CAL0E	BA	
0BFB:	3A JRCP	03	>0BFF
0BFD:	EF CAL0F	E1	
0BFF:	37 RTN		
0C00:	B7 LP37		
0C01:	62 TSIM	80	
0C03:	29 JRNZM	11	>0BF3
0C05:	2D JRM	37	>0BCF
0C07:	EF CAL0F	65	
0C09:	78 CALL	8009	
0C0C:	EF CAL0F	65	
0C0E:	2B JRNCM	24	>0BEB
0C10:	37 RTN		
0C11:	EF CAL0F	9C	
0C13:	3A JRCP	07	>0C1B
0C15:	EF CAL0F	06	
0C17:	3A JRCP	03	>0C1B
0C19:	EF CAL0F	E1	
0C1B:	37 RTN		
0C1C:	EF CAL0F	AB	
0C1E:	3A JRCP	16	>0C35
0C20:	EF CAL0F	F0	
0C22:	61 ORIM	20	
0C24:	EF CAL0F	FD	
0C26:	38 JRZP	0F	>0C36
0C28:	F0 CAL10	04	
0C2A:	2A JRNCP	13	>0C3E
0C2C:	EF CAL0F	65	
0C2E:	78 CALL	8009	
0C31:	EF CAL0F	65	
0C33:	2B JRNCM	10	>0C24
0C35:	37 RTN		
0C36:	B7 LP37		
0C37:	62 TSIM	80	
0C39:	29 JRNZM	6B	>0BCF
0C3B:	EF CAL0F	C1	
0C3D:	37 RTN		
0C3E:	57 LDD		
0C3F:	67 CPIA	2C	
0C41:	38 JRZP	08	>0C4A
0C43:	67 CPIA	28	
0C45:	28 JRNZP	A5	>0CEB
0C47:	F0 CAL10	0E	
0C49:	37 RTN		
0C4A:	86 LP06		
0C4B:	13 LIQ	3A	
0C4D:	0B EXB		
0C4E:	86 LP06		
0C4F:	13 LIQ	3A	
0C51:	0B EXB		
0C52:	02 LIA	01	
0C54:	88 LP08		
0C55:	DB EXAM		
0C56:	86 LP06		
0C57:	63 CPIM	80	
0C59:	38 JRZP	3F	>0C99
0C5B:	48 INCK		
0C5C:	07 DY		
0C5D:	57 LDD		
0C5E:	67 CPIA	2C	
0C60:	39 JRZM	0B	>0C56
0C62:	88 LP08		
0C63:	10 LIDP	66F2	
0C66:	53 MYDM		
0C67:	67 CPIA	28	
0C69:	28 JRNZP	40	>0CAA
0C6B:	86 LP06		
0C6C:	63 CPIM	80	
0C6E:	39 JRZM	34	>0C3B
0C70:	07 DY		
0C71:	57 LDD		
0C72:	F0 CAL10	0E	
0C74:	EF CAL0F	65	
0C76:	67 CPIA	AD	
0C78:	2A JRNCP	05	>0C7E
0C7A:	67 CPIA	AA	
0C7C:	2A JRNCP	06	>0C83
0C7E:	EF CAL0F	DD	
0C80:	EF CAL0F	65	
0C82:	37 RTN		
0C83:	7A 7A		
0C84:	02 LIA	0C	
0C86:	90 LP10		
0C87:	69 69		
0C88:	AB LP2B		
0C89:	80 LP00		
0C8A:	39 JRZM	AC	>0BDF
0C8C:	80 LP00		
0C8D:	3C 3C		
0C8E:	80 LP00		
0C8F:	3F 3F		
0C90:	02 LIA	01	
0C92:	10 LIDP	66F2	
0C95:	52 STD		
0C96:	EF CAL0F	65	
0C98:	37 RTN		
0C99:	B7 LP37		
0C9A:	62 TSIM	80	
0C9C:	39 JRZM	CE	>0BCF
0C9E:	10 LIDP	66F5	
0CA1:	57 LDD		
0CA2:	67 CPIA	88	
0CA4:	29 JRNZM	6A	>0C3B
0CA6:	B3 LP33		
0CA7:	61 ORIM	01	
0CA9:	37 RTN		
0CAA:	8A LP0A		
0CAB:	DB EXAM		
0CAC:	63 CPIM	41	
0CAE:	3B JRCM	74	>0C3B
0CB0:	63 CPIM	5B	
0CB2:	2B JRNCM	78	>0C3B
0CB4:	F0 CAL10	0E	
0CB6:	ED CAL0D	12	
0CB8:	88 LP08		
0CB9:	63 CPIM	03	
0CBB:	7D JPNC	0FC5	
0CBE:	F5 CAL15	36	
0CC0:	3B JRCM	8C	>0C35
0CC2:	10 LIDP	66D0	
0CC5:	53 MYDM		
0CC6:	10 LIDP	66F5	
0CC9:	57 LDD		
0CCA:	67 CPIA	80	
0CCC:	39 JRZM	92	>0C3B
0CCE:	75 SBIA	08	
0CD0:	52 STD		
0CD1:	83 LIB	20	
0CD3:	E2 CAL02	42	
0CD5:	06 IY		
0CD6:	90 LP10		
0CD7:	00 LII	07	
0CD9:	18 MYWD		
0CDA:	F5 CAL15	36	
0CDC:	3B JRCM	A8	>0C35
0CDE:	10 LIDP		66D1
0CE1:	53 MYDM		
0CE2:	B3 LP33		
0CE3:	60 ANIM	2D	
0CE5:	61 ORIM	28	
0CE7:	EF CAL0F	65	
0CE9:	2C JRP	24	>0D0E
0CEB:	B3 LP33		
0CEC:	60 ANIM	29	
0CEE:	61 ORIM	28	
0CF0:	F0 CAL10	0E	
0CF2:	8A LP0A		
0CF3:	DB EXAM		
0CF4:	ED CAL0D	12	
0CF6:	EF CAL0F	65	
0CF8:	8A LP0A		
0CF9:	63 CPIM	40	
0CFB:	28 JRNZP	09	>0D05
0CFD:	B3 LP33		
0CFE:	61 ORIM	80	
0D00:	F6 CAL16	6D	
0D02:	EF CAL0F	65	
0D04:	37 RTN		
0D05:	D1 RC		
0D06:	F5 CAL15	36	
0D08:	3B JRCM	07	>0D02
0D0A:	10 LIDP	66D0	
0D0D:	1B EXBD		
0D0E:	F5 CAL15	75	
0D10:	2D JRM	0F	>0D02
0D12:	10 LIDP	66D6	
0D15:	57 LDD		
0D16:	03 LIB	28	
0D18:	E2 CAL02	42	
0D1A:	57 LDD		
0D1B:	86 LP06		
0D1C:	10 LIDP	66D6	

0D1F:	53	MVDM		
0D20:	8B	LP0B		
0D21:	DB	EXAM		
0D22:	63	CPIM	F0	
0D24:	2A	JRNCP	09	>0D2E
0D26:	63	CPIM	1B	
0D28:	3A	JRCP	05	>0D2E
0D2A:	63	CPIM	A0	
0D2C:	28	JRNZP	04	>0D31
0D2E:	B3	LP33		
0D2F:	61	ORIM	04	
0D31:	D1	RC		
0D32:	37	RTN		
0D33:	EF	CAL0F	A5	
0D35:	3A	JRCP	25	>0D5B
0D37:	EF	CAL0F	6A	
0D39:	86	LP06		
0D3A:	24	IXL		
0D3B:	67	CPIA	22	
0D3D:	38	JRZP	11	>0D4F
0D3F:	67	CPIA	0D	
0D41:	38	JRZP	0C	>0D4E
0D43:	63	CPIM	AF	
0D45:	2A	JRNCP	05	>0D4B
0D47:	26	IYS		
0D48:	48	INCK		
0D49:	2B	JRNCM	10	>0D3A
0D4B:	EF	CAL0F	CD	
0D4D:	37	RTN		
0D4E:	05	DX		
0D4F:	88	LP08		
0D50:	59	LDM		
0D51:	97	LP17		
0D52:	DB	EXAM		
0D53:	06	IY		
0D54:	86	LP06		
0D55:	10	LIDP	66F7	
0D58:	53	MVDM		
0D59:	EF	CAL0F	E6	
0D5B:	37	RTN		
0D5C:	EF	CAL0F	9C	
0D5E:	3A	JRCP	36	>0D95
0D60:	24	IXL		
0D61:	05	DX		
0D62:	ED	CAL0D	96	
0D64:	67	CPIA	3A	
0D66:	2A	JRNCP	13	>0D7A
0D68:	67	CPIA	30	
0D6A:	3A	JRCP	1B	>0D86
0D6C:	64	ANIA	0F	
0D6E:	96	LP16		
0D6F:	1D	SLW		
0D70:	70	ADIM	10	
0D72:	96	LP16		
0D73:	47	ORMA		
0D74:	24	IXL		
0D75:	49	DECK		
0D76:	29	JRNZM	13	>0D64
0D78:	2C	JRP	12	>0D8B
0D7A:	67	CPIA	41	
0D7C:	3A	JRCP	09	>0D86
0D7E:	67	CPIA	47	
0D80:	2A	JRNCP	05	>0D86
0D82:	75	SBIA	37	
0D84:	2D	JRM	17	>0D6E
0D86:	96	LP16		
0D87:	1D	SLW		
0D88:	49	DECK		
0D89:	29	JRNZM	04	>0D86
0D8B:	EF	CAL0F	65	
0D8D:	78	CALL	8000	
0D90:	EF	CAL0F	65	
0D92:	EF	CAL0F	E6	
0D94:	05	DX		
0D95:	37	RTN		
0D96:	E0	CAL00	E4	
0D98:	91	LP11		
0D99:	61	ORIM	F0	
0D9B:	00	LII	04	
0D9D:	88	LP08		
0D9E:	02	LIA	0A	
0DA0:	DB	EXAM		
0DA1:	24	IXL		
0DA2:	67	CPIA	30	
0DA4:	39	JRZM	04	>0DA1
0DA6:	37	RTN		
0DA7:	B3	LP33		
0DA8:	61	ORIM	10	
0DAA:	60	ANIM	F3	
0DAC:	E0	CAL00	E7	
0DAE:	02	LIA	00	
0DB0:	DA	EXAB		
0DB1:	88	LP08		
0DB2:	02	LIA	99	
0DB4:	00	LII	01	
0DB6:	1E	FILM		
0DB7:	B3	LP33		
0DB8:	24	IXL		
0DB9:	67	CPIA	30	
0DBB:	28	JRNZP	05	>0DC1
0DBD:	60	ANIM	EF	
0DBF:	2D	JRM	08	>0DB8
0DC1:	67	CPIA	20	
0DC3:	38	JRZP	2E	>0DF2
0DC5:	67	CPIA	45	
0DC7:	38	JRZP	4C	>0E14
0DC9:	67	CPIA	2E	
0DCB:	38	JRZP	29	>0DF5
0DCD:	67	CPIA	30	
0DCF:	3A	JRCP	A6	>0E76
0DD1:	67	CPIA	3A	
0DD3:	2A	JRNCP	A2	>0E76
0DD5:	B3	LP33		
0DD6:	60	ANIM	EF	
0DD8:	62	TSIM	04	
0DDA:	28	JRNZP	09	>0DE4
0DDC:	34	PUSH		
0DDD:	89	LP09		
0DDE:	02	LIA	01	
0DE0:	00	LII	01	
0DE2:	0C	ADN		
0DE3:	5B	POP		
0DE4:	C2	INCB		
0DE5:	83	LP03		
0DE6:	63	CPIM	0B	
0DE8:	2A	JRNCP	09	>0DF2
0DEA:	96	LP16		
0DEB:	00	LII	04	
0DED:	1D	SLW		
0DEE:	96	LP16		
0DEF:	64	ANIA	0F	
0DF1:	47	ORMA		
0DF2:	24	IXL		
0DF3:	2D	JRM	33	>0DC1
0DF5:	B3	LP33		
0DF6:	62	TSIM	02	
0DF8:	28	JRNZP	7D	>0E76
0DFA:	60	ANIM	EF	
0DFC:	62	TSIM	04	
0DFE:	29	JRNZM	0D	>0DF2
0E00:	61	ORIM	04	
0E02:	83	LP03		
0E03:	63	CPIM	00	
0E05:	29	JRNZM	14	>0DF2
0E07:	00	LII	01	
0E09:	24	IXL		
0E0A:	67	CPIA	30	
0E0C:	29	JRNZM	4C	>0DC1
0E0E:	89	LP09		
0E0F:	02	LIA	01	
0E11:	0D	SBN		
0E12:	2D	JRM	0A	>0E09
0E14:	B3	LP33		
0E15:	62	TSIM	02	
0E17:	28	JRNZP	5E	>0E76
0E19:	EE	CAL0E	9C	
0E1B:	F3	CAL13	9D	
0E1D:	67	CPIA	2E	
0E1F:	38	JRZP	11	>0E31
0E21:	67	CPIA	30	
0E23:	3A	JRCP	5B	>0E7F
0E25:	67	CPIA	3A	
0E27:	2A	JRNCP	6A	>0E92
0E29:	64	ANIA	0F	
0E2B:	91	LP11		
0E2C:	DB	EXAM		
0E2D:	58	SWP		
0E2E:	64	ANIA	F0	
0E30:	47	ORMA		
0E31:	F3	CAL13	9D	
0E33:	67	CPIA	2E	
0E35:	39	JRZM	05	>0E31
0E37:	67	CPIA	45	
0E39:	39	JRZM	09	>0E31
0E3B:	67	CPIA	30	
0E3D:	3A	JRCP	05	>0E43
0E3F:	67	CPIA	3A	
0E41:	3B	JRCM	19	>0E29
0E43:	90	LP10		
0E44:	02	LIA	00	
0E46:	DB	EXAM		
0E47:	59	LDM		

0E48:	B3	LP33	
0E49:	62	TSM	08
0E4B:	00	LII	01
0E4D:	38	JRZP	04 >0E52
0E4F:	91	LP11	
0E50:	DB	EXAM	
0E51:	0D	SPN	
0E52:	91	LP11	
0E53:	13	LIQ	09
0E55:	0E	ADM	
0E56:	90	LP10	
0E57:	63	CPIM	90
0E59:	3A	JRCP	0A >0E64
0E5B:	63	CPIM	99
0E5D:	38	JRZP	0D >0E6B
0E5F:	E0	CAL00	E7
0E61:	D1	RC	
0E62:	05	DX	
0E63:	37	RTN	
0E64:	63	CPIM	00
0E66:	38	JRZP	09 >0E70
0E68:	EF	CAL0F	DD
0E6A:	37	RTN	
0E6B:	50	INCP	
0E6C:	63	CPIM	00
0E6E:	39	JRZM	10 >0E5F
0E70:	91	LP11	
0E71:	00	LII	01
0E73:	1D	SLW	
0E74:	2D	JRM	14 >0E61
0E76:	83	LP03	
0E77:	63	CPIM	00
0E79:	39	JRZM	19 >0E61
0E7B:	EE	CAL0E	9C
0E7D:	2C	JRP	18 >0E96
0E7F:	67	CPIA	2B
0E81:	39	JRZM	67 >0E1B
0E83:	67	CPIA	2D
0E85:	28	JRNZP	10 >0E96
0E87:	B3	LP33	
0E88:	59	LDM	
0E89:	64	ANIA	08
0E8B:	74	ADIA	08
0E8D:	60	ANIM	F7
0E8F:	47	ORMA	
0E90:	2D	JRM	76 >0E1B
0E92:	67	CPIA	45
0E94:	39	JRZM	7A >0E1B
0E96:	90	LP10	
0E97:	13	LIQ	08
0E99:	0A	MVB	
0E9A:	2D	JRM	2B >0E70
0E9C:	00	LII	04
0E9E:	C2	INCB	
0E9F:	83	LP03	
0EA0:	63	CPIM	0B
0EA2:	2A	JRNCP	05 >0EA8
0EA4:	96	LP16	
0EA5:	1D	SLW	
0EA6:	3B	JRCM	09 >0E9E
0EA8:	37	RTN	
0EA9:	8A	LP0A	
0EAA:	DB	EXAM	
0EAB:	EF	CAL0F	FD
0EAD:	03	LIB	28
0EAF:	E2	CAL02	42
0EB1:	28	JRNZP	28 >0EDA
0EB3:	67	CPIA	90
0EB5:	7E	JPZ	0FCD
0EB8:	EF	CAL0F	14
0EBA:	10	LIDP	66F5
0EBD:	57	LDD	
0EBE:	67	CPIA	83
0EC0:	7E	JPZ	0FCD
0EC3:	10	LIDP	66F5
0EC6:	57	LDD	
0EC7:	03	LIB	20
0EC9:	E2	CAL02	42
0ECB:	06	IY	
0ECC:	00	LII	07
0ECE:	90	LP10	
0ECF:	19	EXWD	
0ED0:	86	LP06	
0ED1:	59	LDM	
0ED2:	74	ADIA	08
0ED4:	10	LIDP	66F5
0ED7:	52	STD	
0ED8:	D1	RC	
0ED9:	37	RTN	
0EDA:	8A	LP0A	
0EDB:	59	LDM	
0EDC:	EF	CAL0F	1F
0EDE:	8B	LP0B	
0EDF:	DB	EXAM	
0EE0:	57	LDD	
0EE1:	EF	CAL0F	1F
0EE3:	8B	LP0B	
0EE4:	C7	CPMA	
0EE5:	3A	JRCP	1C >0F02
0EE7:	28	JRNZP	0A >0EF2
0EE9:	8A	LP0A	
0EEA:	63	CPIM	5E
0EEC:	38	JRZP	15 >0F02
0EEE:	63	CPIM	83
0EF0:	38	JRZP	11 >0F02
0EF2:	8A	LP0A	
0EF3:	59	LDM	
0EF4:	34	PUSH	
0EF5:	EF	CAL0F	65
0EF7:	78	CALL	8009
0EFA:	EF	CAL0F	65
0EFC:	5B	POP	
0EFD:	8A	LP0A	
0EFE:	DB	EXAM	
0EFF:	2B	JRNCM	55 >0EAB
0F01:	37	RTN	
0F02:	EF	CAL0F	08
0F04:	2D	JRM	4B >0EBA
0F06:	8A	LP0A	
0F07:	DB	EXAM	
0F08:	10	LIDP	66F6
0F0B:	57	LDD	
0F0C:	67	CPIA	90
0F0E:	38	JRZP	BE >0FCD
0F10:	03	LIB	28
0F12:	E2	CAL02	42
0F14:	8A	LP0A	
0F15:	59	LDM	
0F16:	26	IYS	
0F17:	06	IY	
0F18:	86	LP06	
0F19:	10	LIDP	66F6
0F1C:	53	MVDM	
0F1D:	D1	RC	
0F1E:	37	RTN	
0F1F:	67	CPIA	40
0F21:	2A	JRNCP	16 >0F38
0F23:	67	CPIA	30
0F25:	2A	JRNCP	0F >0F35
0F27:	75	SBIA	28
0F29:	74	ADIA	5D
0F2B:	03	LIB	0F
0F2D:	2A	JRNCP	02 >0F30
0F2F:	C2	INCB	
0F30:	00	LII	00
0F32:	82	LP02	
0F33:	35	DATA	
0F34:	37	RTN	
0F35:	02	LIA	07
0F37:	37	RTN	
0F38:	67	CPIA	90
0F3A:	2A	JRNCP	14 >0F4F
0F3C:	67	CPIA	5E
0F3E:	38	JRZP	0D >0F4C
0F40:	3A	JRCP	08 >0F49
0F42:	67	CPIA	64
0F44:	3B	JRCM	10 >0F35
0F46:	02	LIA	04
0F48:	37	RTN	
0F49:	02	LIA	09
0F4B:	37	RTN	
0F4C:	02	LIA	03
0F4E:	37	RTN	
0F4F:	67	CPIA	A1
0F51:	38	JRZP	08 >0F5A
0F53:	67	CPIA	A2
0F55:	38	JRZP	04 >0F5A
0F57:	02	LIA	02
0F59:	37	RTN	
0F5A:	02	LIA	08
0F5C:	37	RTN	
0F5D:	0A	MVB	
0F5E:	09	EXW	
0F5F:	05	DX	
0F60:	06	IY	
0F61:	09	EXW	
0F62:	06	IY	
0F63:	00	LII	05
0F65:	84	LP04	
0F66:	13	LIQ	38
0F68:	0B	EXB	

0F69: 37 RTN		0FC0: 37 RTN		1025: 28 JRNZP 03 >1029
0F6A: 02 LIA 00		0FC1: 02 LIA 01		1027: D0 SC
0F6C: 88 LP08		0FC3: 2D JRM 07 >0FBD		1028: 37 RTN
0F6D: DB EXAM		0FC5: 02 LIA 03		1029: 5B POP
0F6E: 02 LIA D0		0FC7: 2D JRM 0B >0FBD		102A: 5B POP
0F70: 94 LP14		0FC9: 02 LIA 04		102B: EF CAL0F DD
0F71: DB EXAM		0FCB: 2D JRM 0F >0FBD		102D: 37 RTN
0F72: 10 LIDP 66F7		0FCD: 02 LIA 05		102E: 22 LDR ; POS# X
0F75: 57 LDD		0FCF: 2D JRM 13 >0FBD		102F: 43 DECA
0F76: 03 LIB 67		0FD1: 02 LIA 06		1030: 43 DECA
0F78: E2 CAL02 42		0FD3: 2D JRM 17 >0FBD		1031: 30 STP
0F7A: 06 IY		0FD5: 02 LIA 07		1032: 13 LIQ 04
0F7B: 95 LP15		0FD7: 2D JRM 1B >0FBD		1034: 0A MYB
0F7C: 13 LIQ 06		0FD9: 02 LIA 08		1035: 31 STQ
0F7E: 0A MYB		0FDB: 2D JRM 1F >0FBD		1036: 0B EXB
0F7F: 07 IY		0FDD: 02 LIA 09		1037: 32 STR
0F80: 37 RTN		0FDF: 2D JRM 23 >0FBD		1038: 37 RTN
0F81: 03 LIB 18		0FE1: EF CAL0F F0		1039: 22 LDR ; POS# X
0F83: 10 LIDP 66F5		0FE3: 61 ORIM 01		103A: 42 INCA
0F86: 57 LDD		0FE5: 37 RTN		103B: 42 INCA
0F87: 75 SBIA 08		0FE6: EF CAL0F F0		103C: 84 LP04
0F89: 52 STD		0FE8: 61 ORIM 02		103D: 31 STQ
0F8A: 43 DECA		0FEA: 37 RTN		103E: 0A MYB
0F8B: 84 LP04		0FEB: EF CAL0F F0		103F: 30 STP
0F8C: DB EXAM		0FED: 61 ORIM 08		1040: DA EXAB
0F8D: 02 LIA 20		0FEF: 37 RTN		1041: 22 LDR
0F8F: 95 LP05		0FF0: B7 LP37		1042: 31 STQ
0F90: DB EXAM		0FF1: 60 ANIM C0		1043: 0A MYB
0F91: 04 IX		0FF3: 37 RTN		1044: DA EXAB
0F92: 00 LII 07		0FF4: 10 LIDP 66F5		1045: 32 STR
0F94: DA EXAB		0FF7: DA EXAB		1046: 37 RTN
0F95: 30 STP		0FF8: 57 LDD		1047: 24 IXL
0F96: 18 MYWD		0FF9: 67 CPIA 80		1048: 26 IYS
0F97: 37 RTN		0FFB: DA EXAB		1049: C3 DECB
0F98: 03 LIB 10		0FFC: 37 RTN		104A: 29 JRNZM 04 >1047
0F9A: 2D JRM 18 >0F83		0FFD: 10 LIDP 66F6		104C: 37 RTN
0F9C: B7 LP37		1000: 57 LDD		104D: F3 CAL13 5B
0F9D: D1 RC		1001: 67 CPIA 80		104F: D4 ANID C0
0F9E: 62 TSIM 22		1003: 37 RTN		1051: F3 CAL13 57
0FA0: 38 JRZP 03 >0FA4		1004: 03 LIB 28		1053: D4 ANID E1
0FA2: EF CAL0F C1		1006: E2 CAL02 42		1055: F9 CAL19 E9
0FA4: 37 RTN		1008: 57 LDD		1057: 11 LIDL 7F
0FA5: B7 LP37		1009: EF CAL0F 1F		1059: D4 ANID 00
0FA6: D1 RC		100B: 67 CPIA 09		105B: B5 LP35
0FA7: 62 TSIM 26		100D: 37 RTN		105C: 60 ANIM BF
0FA9: 2D JRM 0A >0FA0		100E: 86 LP06		105E: 62 TSIM 80
0FAB: B7 LP37		100F: 10 LIDP 66F6		1060: 28 JRNZP 06 >1067
0FAC: D1 RC		1012: 53 MYDM		1062: 10 LIDP 661E
0FAD: 62 TSIM 1D		1013: D1 RC		1065: D4 ANID 00
0FAF: 2D JRM 10 >0FA0		1014: 37 RTN		1067: FB CAL1B B7
0FB1: EF CAL0F 65		1015: EF CAL0F 81		1069: D4 ANID 00
0FB3: 05 DX		1017: 9C LP1C		106B: FC CAL1C 2B
0FB4: EF CAL0F 65		1018: 63 CPIM D0		106D: 90 LP10
0FB6: 10 LIDP 663C		101A: 94 LP14		106E: 02 LIA 3E
0FB9: 57 LDD		101B: 38 JRZP 07 >1023		1070: DB EXAM
0FBA: 32 STR		101D: 63 CPIM D0		1071: 91 LP11
0FBB: 02 LIA 02		101F: 38 JRZP 09 >1029		1072: 02 LIA 20
0FBD: B4 LP34		1021: D1 RC		1074: 00 LII 16
0FBE: DB EXAM		1022: 37 RTN		1076: 1E FILM
0FBF: D0 SC		1023: 63 CPIM D0		1077: FB CAL1B 52

1079:	FB CAL1B	B0	
107B:	FB CAL1B	5E	
107D:	78 CALL	800F	
1080:	37 RTN		
1081:	D6 TSID	10	
1083:	38 JRZP	21 >10A5	
1085:	D4 ANID	EF	
1087:	10 LIDP	66E8	
108A:	57 LDD		
108B:	F3 CAL13	5F	
108D:	10 LIDP	66E9	
1090:	57 LDD		
1091:	D6 TSID	80	
1093:	03 LIB	20	
1095:	38 JRZP	05 >109B	
1097:	75 SBIA	80	
1099:	03 LIB	28	
109B:	E2 CAL02	42	
109D:	03 LIB	05	
109F:	25 DXL		
10A0:	26 IYS		
10A1:	C3 DECB		
10A2:	29 JRNZM	04 >109F	
10A4:	37 RTN		
10A5:	D5 ORID	10	
10A7:	02 LIA	F9	
10A9:	2D JRM	1F >108B	
10AB:	8E LP0E		
10AC:	59 LDM		
10AD:	D2 SR		
10AE:	DB EXAM		
10AF:	2A JRNCP	03 >10B3	
10B1:	F0 CAL10	CF	
10B3:	F0 CAL10	C1	
10B5:	F0 CAL10	C8	
10B7:	F0 CAL10	CF	
10B9:	F0 CAL10	C8	
10BB:	F0 CAL10	C1	
10BD:	49 DECK		
10BE:	2B JRNCM	14 >10AB	
10C0:	37 RTN		
10C1:	92 LP12		
10C2:	13 LIQ	22	
10C4:	00 LII	02	
10C6:	09 EXW		
10C7:	37 RTN		
10C8:	9A LP1A		
10C9:	13 LIQ	12	
10CB:	00 LII	03	
10CD:	08 MVW		
10CE:	37 RTN		
10CF:	94 LP14		
10D0:	13 LIQ	1C	
10D2:	00 LII	02	
10D4:	0E ADW		
10D5:	37 RTN		
10D6:	13 LIQ	1D	
10D8:	2C JRP	03 >10DC	
10DA:	13 LIQ	15	
10DC:	84 LP04		
10DD:	0A MVB		
10DE:	05 DX		
10DF:	37 RTN		
10E0:	80 LP30		
10E1:	61 ORIM	02	
10E3:	E0 CAL00	E4	
10E5:	2C JRP	18 >10FE	
10E7:	E0 CAL00	E4	
10E9:	80 LP30		
10EA:	60 ANIM	FC	
10EC:	99 LP19		
10ED:	62 TSIM	80	
10EF:	38 JRZP	0E >10FE	
10F1:	80 LP30		
10F2:	61 ORIM	01	
10F4:	02 LIA	00	
10F6:	99 LP19		
10F7:	DB EXAM		
10F8:	DA EXAB		
10F9:	02 LIA	00	
10FB:	98 LP18		
10FC:	DB EXAM		
10FD:	15 SBB		
10FE:	8E LP0E		
10FF:	13 LIQ	18	
1101:	0A MVB		
1102:	88 LP08		
1103:	02 LIA	07	
1105:	DB EXAM		
1106:	E0 CAL00	DB	
1108:	9C LP1C		
1109:	61 ORIM	10	
110B:	E1 CAL01	88	
110D:	F0 CAL10	AB	
110F:	02 LIA	07	
1111:	88 LP08		
1112:	DB EXAM		
1113:	8F LP0F		
1114:	59 LDM		
1115:	8E LP0E		
1116:	DB EXAM		
1117:	F0 CAL10	AB	
1119:	91 LP11		
111A:	61 ORIM	40	
111C:	E4 CAL04	70	
111E:	80 LP30		
111F:	62 TSIM	02	
1121:	28 JRNZP	0A >112C	
1123:	62 TSIM	01	
1125:	38 JRZP	06 >112C	
1127:	60 ANIM	FE	
1129:	91 LP11		
112A:	61 ORIM	08	
112C:	D1 RC		
112D:	37 RTN		
112E:	F3 CAL13	53	
1130:	D4 ANID	FE	
1132:	02 LIA	80	
1134:	34 PUSH		
1135:	02 LIA	12	
1137:	34 PUSH		
1138:	02 LIA	03	
113A:	E4 CAL04	3D	
113C:	6B TEST	01	
113E:	02 LIA	80	
1140:	03 LIB	66	
1142:	E2 CAL02	42	
1144:	02 LIA	00	
1146:	26 IYS		
1147:	02 LIA	FB	
1149:	26 IYS		
114A:	00 LII	FB	
114C:	F8 CAL18	DA	
114E:	38 JRZP	1C >116B	
1150:	10 LIDP	661E	
1153:	D4 ANID	00	
1155:	11 LIDL	1F	
1157:	D4 ANID	00	
1159:	10 LIDP	203C	
115C:	57 LDD		
115D:	64 ANIA	F0	
115F:	47 ORMA		
1160:	59 LDM		
1161:	52 STD		
1162:	D6 TSID	04	
1164:	02 LIA	08	
1166:	38 JRZP	03 >116A	
1168:	02 LIA	0A	
116A:	37 RTN		
116B:	E3 CAL03	C1	
116D:	7D JPNC	12B6	
1170:	34 PUSH		
1171:	E3 CAL03	C1	
1173:	DA EXAB		
1174:	5B POP		
1175:	7D JPNC	12B6	
1178:	83 LP03		
1179:	C7 CPMA		
117A:	7C JPNZ	12B6	
117D:	F3 CAL13	53	
117F:	D6 TSID	20	
1181:	28 JRNZP	42 >11C4	
1183:	D6 TSID	08	
1185:	28 JRNZP	38 >118E	
1187:	D6 TSID	10	
1189:	28 JRNZP	2E >1188	
118B:	74 ADIA	A0	
118D:	03 LIB	83	
118F:	2A JRNCP	02 >1192	
1191:	C2 INCB		
1192:	E2 CAL02	2E	
1194:	24 IXL		
1195:	F3 CAL13	57	
1197:	D6 TSID	01	
1199:	28 JRNZP	DB >1275	
119B:	D5 ORID	01	
119D:	11 LIDL	2C	
119F:	52 STD		
11A0:	67 CPIA	02	
11A2:	3B JRCM	65 >113E	

11A4:	67	CPIA	20				
11A6:	2A	JRNCP	85	>122C			
11A8:	67	CPIA	10				
11AA:	3B	JRCM	41	>116A			
11AC:	F3	CAL13	53				
11AE:	28	JRNZP	3A	>11E9			
11B0:	D6	TSID	20				
11B2:	38	JRZP	44	>11F7			
11B4:	D4	ANID	DF				
11B6:	2D	JRM	79	>113E			
11B8:	74	ADIA	78				
11BA:	03	LIB	84				
11BC:	2D	JRM	2E	>118F			
11BE:	74	ADIA	D6				
11C0:	03	LIB	83				
11C2:	2D	JRM	34	>118F			
11C4:	D6	TSID	08				
11C6:	28	JRNZP	1C	>11E3			
11C8:	D6	TSID	10				
11CA:	28	JRNZP	12	>11DD			
11CC:	10	LIDP	203C				
11CF:	D6	TSID	80				
11D1:	28	JRNZP	05	>11D7			
11D3:	67	CPIA	28				
11D5:	39	JRZM	4B	>118B			
11D7:	74	ADIA	0C				
11D9:	03	LIB	84				
11DB:	2D	JRM	4D	>118F			
11DD:	74	ADIA	AE				
11DF:	03	LIB	84				
11E1:	2D	JRM	53	>118F			
11E3:	74	ADIA	42				
11E5:	03	LIB	84				
11E7:	2D	JRM	59	>118F			
11E9:	67	CPIA	12				
11EB:	28	JRNZP	11	>11FD			
11ED:	D6	TSID	40				
11EF:	28	JRNZP	09	>11F9			
11F1:	D5	ORID	40				
11F3:	D4	ANID	DF				
11F5:	2D	JRM	B8	>113E			
11F7:	D5	ORID	20				
11F9:	D4	ANID	BF				
11FB:	2D	JRM	BE	>113E			
11FD:	67	CPIA	15				
11FF:	28	JRNZP	74	>1274			
1201:	F3	CAL13	5B				
1203:	D6	TSID	02				
1205:	F3	CAL13	53				
1207:	28	JRNZP	10	>1218			
1209:	10	LIDP	203C				
120C:	D6	TSID	80				
120E:	F3	CAL13	53				
1210:	38	JRZP	0B	>121C			
1212:	D6	TSID	08				
1214:	38	JRZP	13	>1228			
1216:	D4	ANID	F7				
1218:	D4	ANID	9F				
121A:	2D	JRM	DD	>113E			
121C:	D6	TSID	10				
121E:	38	JRZP	05	>1224			
1220:	D4	ANID	EF				
1222:	2D	JRM	0B	>1218			
1224:	D5	ORID	10				
1226:	2D	JRM	0F	>1218			
1228:	D5	ORID	08				
122A:	2D	JRM	13	>1218			
122C:	F3	CAL13	53				
122E:	38	JRZP	05	>1234			
1230:	67	CPIA	3D				
1232:	28	JRNZP	10	>1243			
1234:	D6	TSID	40				
1236:	38	JRZP	3D	>1274			
1238:	10	LIDP	203C				
123B:	D6	TSID	08				
123D:	38	JRZP	36	>1274			
123F:	65	ORIA	80				
1241:	2C	JRP	28	>126A			
1243:	67	CPIA	5B				
1245:	2A	JRNCP	2E	>1274			
1247:	67	CPIA	41				
1249:	3A	JRCP	2A	>1274			
124B:	D6	TSID	40				
124D:	38	JRZP	26	>1274			
124F:	10	LIDP	203C				
1252:	D6	TSID	08				
1254:	38	JRZP	1F	>1274			
1256:	FC	CAL1C	51				
1258:	28	JRNZP	1B	>1274			
125A:	75	SBIA	41				
125C:	74	ADIA	E4				
125E:	03	LIB	84				
1260:	2A	JRNCP	02	>1263			
1262:	C2	INCB					
1263:	E2	CAL02	2E				
1265:	24	IXL					
1266:	66	TSIA	80				
1268:	38	JRZP	0B	>1274			
126A:	DA	EXAB					
126B:	5B	POP					
126C:	5B	POP					
126D:	02	LIA	80				
126F:	34	PUSH					
1270:	02	LIA	15				
1272:	34	PUSH					
1273:	DA	EXAB					
1274:	37	RTN					
1275:	D6	TSID	02				
1277:	38	JRZP	48	>12C0			
1279:	10	LIDP	662C				
127C:	55	MYMD					
127D:	C7	CPMA					
127E:	28	JRNZP	41	>12C0			
1280:	F3	CAL13	57				
1282:	D6	TSID	08				
1284:	28	JRNZP	15	>129A			
1286:	34	PUSH					
1287:	10	LIDP	662E				
128A:	82	LP02					
128B:	1A	MYBD					
128C:	67	CPIA	00				
128E:	28	JRNZP	1A	>12A9			
1290:	DA	EXAB					
1291:	67	CPIA	00				
1293:	DA	EXAB					
1294:	28	JRNZP	14	>12A9			
1296:	F4	CAL14	DE				
1298:	5B	POP					
1299:	37	RTN					
129A:	6B	TEST	01				
129C:	7E	JPZ	113E				
129F:	D6	TSID	04				
12A1:	7E	JPZ	113E				
12A4:	F0	CAL10	81				
12A6:	79	JP	113E				
12A9:	42	INCA					
12AA:	2A	JRNCP	02	>12AD			
12AC:	C2	INCB					
12AD:	82	LP02					
12AE:	10	LIDP	662E				
12B1:	1B	EXBD					
12B2:	5B	POP					
12B3:	79	JP	113E				
12B6:	F3	CAL13	57				
12B8:	D6	TSID	01				
12BA:	38	JRZP	43	>12FE			
12BC:	D6	TSID	02				
12BE:	38	JRZP	39	>12F8			
12C0:	11	LIDL	2B				
12C2:	D4	ANID	FD				
12C4:	00	LII	FB				
12C6:	6B	TEST	01				
12C8:	38	JRZP	53	>131C			
12CA:	02	LIA	80				
12CC:	03	LIB	66				
12CE:	E2	CAL02	2E				
12D0:	03	LIB	02				
12D2:	24	IXL					
12D3:	66	TSIA	0F				
12D5:	28	JRNZP	0D	>12E3			
12D7:	34	PUSH					
12D8:	02	LIA	00				
12DA:	E4	CAL04	3D				
12DC:	4E	WAIT	E0				
12DE:	02	LIA	01				
12E0:	E4	CAL04	3D				
12E2:	5B	POP					
12E3:	42	INCA					
12E4:	52	STD					
12E5:	2A	JRNCP	30	>1316			
12E7:	C3	DECB					
12E8:	29	JRNZM	17	>12D2			
12EA:	02	LIA	FF				
12EC:	03	LIB	02				
12EE:	52	STD					
12EF:	05	DX					
12F0:	C3	DECB					
12F1:	29	JRNZM	04	>12EE			
12F3:	E4	CAL04	58				
12F5:	79	JP	114A				
12F8:	40	INCI					

12F9:	7D	JPNC	114C	1366:	5B	POP		13C4:	2C	JRP	1A	>13DF	
12FC:	D4	ANID	FE	1367:	03	LIB	00	13C6:	02	LIA	0E		
12FE:	E3	CAL03	B1	1369:	67	CPIA	20	13C8:	2C	JRP	16	>13DF	
1300:	F3	CAL13	57	1368:	3A	JRCP	19	>1385	13CA:	02	LIA	04	
1302:	D6	TSID	00	136D:	67	CPIA	7B	13CC:	2C	JRP	12	>13DF	
1304:	39	JRZM	3F	>12C6	136F:	3A	JRCP	17	>1387	13CE:	02	LIA	0C
1306:	10	LIDP	66F1	1371:	67	CPIA	A0	13D0:	2C	JRP	0E	>13DF	
1309:	02	LIA	B0	1373:	3A	JRCP	11	>1385	13D2:	02	LIA	0A	
130B:	52	STD		1375:	75	SBIA	25	13D4:	2C	JRP	0A	>13DF	
130C:	78	CALL	8006	1377:	67	CPIA	BB	13D6:	B2	LP32			
130F:	F3	CAL13	57	1379:	3A	JRCP	0D	>1387	13D7:	60	ANIM	00	
1311:	D4	ANID	E1	137B:	67	CPIA	CB	13D9:	61	ORIM	01		
1313:	79	JP	114A	137D:	3A	JRCP	07	>1385	13DB:	2C	JRP	0F	>13EB
1316:	F3	CAL13	57	137F:	75	SBIA	10	13DD:	02	LIA	08		
1318:	D6	TSID	04	1381:	67	CPIA	C9	13DF:	B2	LP32			
131A:	28	JRNZP	16	>1331	1383:	3A	JRCP	03	>1387	13E0:	60	ANIM	F0
131C:	F3	CAL13	57	1385:	02	LIA	20	13E2:	47	ORMA			
131E:	D6	TSID	01	1387:	75	SBIA	1F	13E3:	F3	CAL13	40		
1320:	7C	JPNZ	114C	1389:	D1	RC		13E5:	7D	JPNC	0FDD		
1323:	12	LIP	5D	138A:	8B	LP08		13E8:	B2	LP32			
1325:	61	ORIM	07	138B:	DB	EXAM		13E9:	60	ANIM	7F		
1327:	DD	OUTB		138C:	59	LDM		13EB:	91	LP11			
1328:	4E	WAIT	1C	138D:	5A	SL		13EC:	62	TSIM	08		
132A:	02	LIA	05	138E:	DA	EXAB		13EE:	28	JRNZP	44	>1433	
132C:	E4	CAL04	3D	138F:	5A	SL		13F0:	90	LP10			
132E:	79	JP	114C	1390:	DA	EXAB		13F1:	63	CPIM	90		
1331:	F0	CAL10	81	1391:	5A	SL		13F3:	2A	JRNCP	4B	>143F	
1333:	79	JP	114C	1392:	DA	EXAB		13F5:	63	CPIM	00		
1336:	10	LIDP	6702	1393:	5A	SL		13F7:	28	JRNZP	38	>1430	
1339:	B8	LP38		1394:	DA	EXAB		13F9:	50	INCP			
133A:	1A	MVBD		1395:	44	ADM		13FA:	02	LIA	00		
133B:	84	LP04		1396:	59	LDM		13FC:	DB	EXAM			
133C:	13	LIQ	38	1397:	2A	JRNCP	02	>139A	13FD:	58	SWP		
133E:	0A	MVB		1399:	C2	INCB		13FE:	64	ANIA	0F		
133F:	37	RTN		139A:	84	LP04		1400:	67	CPIA	05		
1340:	94	LP14		139B:	14	ADB		1402:	2A	JRNCP	2D	>1430	
1341:	63	CPIM	D0	139C:	37	RTN		1404:	88	LP08			
1343:	37	RTN		139D:	24	IXL		1405:	DB	EXAM			
1344:	86	LP06		139E:	67	CPIA	20	1406:	48	INCK			
1345:	13	LIQ	04	13A0:	39	JRZM	04	>139D	1407:	02	LIA	00	
1347:	0A	MVB		13A2:	37	RTN		1409:	03	LIB	00		
1348:	37	RTN		13A3:	84	LP04		140B:	00	LII	03		
1349:	84	LP04		13A4:	13	LIQ	0C	140D:	94	LP14			
134A:	13	LIQ	06	13A6:	0B	EXB		140E:	1D	SLW			
134C:	0A	MVB		13A7:	37	RTN		140F:	91	LP11			
134D:	37	RTN		13A8:	10	LIDP	203C	1410:	60	ANIM	0F		
134E:	B8	LP38		13AB:	D6	TSID	01	1412:	44	ADM			
134F:	13	LIQ	04	13AD:	38	JRZP	09	>13B7	1413:	59	LDM		
1351:	0A	MVB		13AF:	84	LP04		1414:	2A	JRNCP	04	>1419	
1352:	37	RTN		13B0:	10	LIDP	283E	1416:	C2	INCB			
1353:	10	LIDP	203D	13B3:	01	LIJ	01	1417:	3A	JRCP	18	>1430	
1356:	37	RTN		13B5:	1A	MVBD		1419:	49	DECK			
1357:	10	LIDP	662B	13B6:	37	RTN		141A:	38	JRZP	2C	>1447	
135A:	37	RTN		13B7:	10	LIDP	661B	141C:	F5	CAL15	59		
135B:	10	LIDP	66DA	13BA:	84	LP04		141E:	98	LP18			
135E:	37	RTN		13BB:	1A	MVBD		141F:	13	LIQ	02		
135F:	34	PUSH		13BC:	37	RTN		1421:	0A	MVB			
1360:	02	LIA	45	13BD:	10	LIDP	66E1	1422:	F5	CAL15	59		
1362:	03	LIB	80	13C0:	2D	JRM	07	>13BA	1424:	F5	CAL15	59	
1364:	E2	CAL02	25	13C2:	02	LIA	06	1426:	3A	JRCP	09	>1430	

1428:	98 LP18		
1429:	14 ADB		
142A:	82 LP02		
142B:	13 LIQ	18	
142D:	0A MVB		
142E:	2B JRNCM	22 >140D	
1430:	EF CAL0F	C5	
1432:	37 RTN		
1433:	B2 LP32		
1434:	59 LDM		
1435:	64 ANIA	0F	
1437:	67 CPIA	0A	
1439:	29 JRNZM	0A >1430	
143B:	61 ORIM	80	
143D:	2D JRM	4E >13F0	
143F:	98 LP18		
1440:	02 LIA	00	
1442:	00 LII	01	
1444:	1E FILM		
1445:	2C JRP	05 >144B	
1447:	98 LP18		
1448:	13 LIQ	02	
144A:	0A MVB		
144B:	B2 LP32		
144C:	59 LDM		
144D:	64 ANIA	07	
144F:	38 JRZP	1D >146D	
1451:	99 LP19		
1452:	67 CPIA	02	
1454:	3A JRCP	4E >14A3	
1456:	67 CPIA	04	
1458:	3A JRCP	24 >147D	
145A:	63 CPIM	00	
145C:	29 JRNZM	2D >1430	
145E:	98 LP18		
145F:	67 CPIA	06	
1461:	3A JRCP	0B >146D	
1463:	38 JRZP	05 >1469	
1465:	63 CPIM	30	
1467:	2C JRP	03 >146B	
1469:	63 CPIM	51	
146B:	2B JRNCM	3C >1430	
146D:	99 LP19		
146E:	63 CPIM	00	
1470:	28 JRNZP	0B >147C	
1472:	51 DECP		
1473:	63 CPIM	00	
1475:	28 JRNZP	06 >147C	
1477:	B2 LP32		
1478:	62 TSIM	00	
147A:	39 JRZM	4B >1430	
147C:	37 RTN		
147D:	62 TSIM	00	
147F:	28 JRNZP	12 >1492	
1481:	B2 LP32		
1482:	62 TSIM	00	
1484:	39 JRZM	18 >146D	
1486:	98 LP18		
1487:	02 LIA	00	
1489:	03 LIB	00	
148B:	13 LIQ	02	
148D:	0B EXB		
148E:	98 LP18		
148F:	15 SBB		
1490:	2D JRM	24 >146D	
1492:	B2 LP32		
1493:	62 TSIM	80	
1495:	39 JRZM	66 >1430	
1497:	99 LP19		
1498:	63 CPIM	80	
149A:	29 JRNZM	6B >1430	
149C:	51 DECP		
149D:	63 CPIM	00	
149F:	29 JRNZM	70 >1430	
14A1:	2D JRM	35 >146D	
14A3:	63 CPIM	FF	
14A5:	39 JRZM	76 >1430	
14A7:	2D JRM	3B >146D	
14A9:	00 LII	05	
14AB:	9E LP1E		
14AC:	F4 CAL14	EC	
14AE:	29 JRNZM	04 >14AB	
14B0:	37 RTN		
14B1:	86 LP06		
14B2:	59 LDM		
14B3:	8A LP0A		
14B4:	DB EXAM		
14B5:	37 RTN		
14B6:	86 LP06		
14B7:	59 LDM		
14B8:	8B LP0B		
14B9:	DB EXAM		
14BA:	59 LDM		
14BB:	37 RTN		
14BC:	91 LP11		
14BD:	62 TSIM	00	
14BF:	28 JRNZP	1A >14DA	
14C1:	8F LP0F		
14C2:	63 CPIM	00	
14C4:	28 JRNZP	13 >14D8	
14C6:	FB CAL1B	B7	
14C8:	D6 TSID	06	
14CA:	28 JRNZP	06 >14D1	
14CC:	B3 LP33		
14CD:	62 TSIM	04	
14CF:	38 JRZP	02 >14D2	
14D1:	06 IY		
14D2:	86 LP06		
14D3:	59 LDM		
14D4:	8A LP0A		
14D5:	DB EXAM		
14D6:	28 JRNZP	02 >14D9	
14D8:	06 IY		
14D9:	37 RTN		
14DA:	02 LIA	2D	
14DC:	26 IYS		
14DD:	37 RTN		
14DE:	02 LIA	FE	
14E0:	2C JRP	03 >14E4	
14E2:	02 LIA	CE	
14E4:	03 LIB	FF	
14E6:	10 LIDP	662E	
14E9:	82 LP02		
14EA:	1B EXBD		
14EB:	37 RTN		
14EC:	1D SLW		
14ED:	50 INCP		
14EE:	02 LIA	00	
14F0:	DB EXAM		
14F1:	65 ORIA	30	
14F3:	26 IYS		
14F4:	C3 DECB		
14F5:	37 RTN		
14F6:	10 LIDP	66D7	
14F9:	D6 TSID	04	
14FB:	37 RTN		
14FC:	10 LIDP	66FC	
14FF:	84 LP04		
1500:	1A MVBD		
1501:	05 DX		
1502:	02 LIA	FF	
1504:	03 LIB	64	
1506:	82 LP02		
1507:	13 LIQ	04	
1509:	0B EXB		
150A:	84 LP04		
150B:	15 SBB		
150C:	82 LP02		
150D:	13 LIQ	04	
150F:	0B EXB		
1510:	38 JRZP	12 >1523	
1512:	24 IXL		
1513:	8A LP0A		
1514:	C7 CPMA		
1515:	24 IXL		
1516:	28 JRNZP	05 >151C	
1518:	8B LP0B		
1519:	C7 CPMA		
151A:	38 JRZP	09 >1524	
151C:	24 IXL		
151D:	DA EXAB		
151E:	24 IXL		
151F:	84 LP04		
1520:	14 ADB		
1521:	2B JRNCM	20 >1502	
1523:	D0 SC		
1524:	37 RTN		
1525:	86 LP06		
1526:	10 LIDP	66FC	
1529:	1B EXBD		
152A:	86 LP06		
152B:	11 LIDL	FC	
152D:	1A MVBD		
152E:	07 DY		
152F:	8A LP0A		
1530:	59 LDM		
1531:	26 IYS		
1532:	8B LP0B		
1533:	59 LDM		
1534:	26 IYS		

1535:	37	RTN	
1536:	F3	CAL13	CE
1538:	98	LP18	
1539:	37	RTN	
153A:	88	LP08	
153B:	13	LIQ	02
153D:	0A	MVB	
153E:	10	LIDP	66E3
1541:	82	LP02	
1542:	1A	MVBD	
1543:	82	LP02	
1544:	13	LIQ	06
1546:	0B	EXB	
1547:	86	LP06	
1548:	15	SBB	
1549:	82	LP02	
154A:	13	LIQ	06
154C:	0B	EXB	
154D:	37	RTN	
154E:	98	LP18	
154F:	59	LDM	
1550:	75	SBIA	1A
1552:	03	LIB	00
1554:	D1	RC	
1555:	F5	CAL15	59
1557:	F5	CAL15	59
1559:	D1	RC	
155A:	5A	SL	
155B:	DA	EXAB	
155C:	5A	SL	
155D:	DA	EXAB	
155E:	37	RTN	
155F:	07	DY	
1560:	02	LIA	40
1562:	26	IYS	
1563:	02	LIA	00
1565:	26	IYS	
1566:	37	RTN	
1567:	10	LIDP	66FC
156A:	F5	CAL15	6E
156C:	11	LIDL	3E
156E:	8C	LP0C	
156F:	13	LIQ	06
1571:	0A	MVB	
1572:	8C	LP0C	
1573:	1B	EXBD	
1574:	37	RTN	
1575:	10	LIDP	66D0
1578:	8C	LP0C	
1579:	1A	MVBD	
157A:	F4	CAL14	FC
157C:	3A	JRCP	3D >15BA
157E:	B3	LP33	
157F:	62	TSIM	40
1581:	7C	JPNZ	0FC5
1584:	04	IX	
1585:	04	IX	
1586:	24	IXL	
1587:	8C	LP0C	
1588:	DB	EXAM	
1589:	07	CPMA	
158A:	DB	EXAM	
158B:	DA	EXAB	
158C:	7F	JPC	0FC5
158F:	24	IXL	
1590:	8D	LP0D	
1591:	DB	EXAM	
1592:	C7	CPMA	
1593:	DB	EXAM	
1594:	7F	JPC	0FC5
1597:	59	LDM	
1598:	8A	LP0A	
1599:	DB	EXAM	
159A:	02	LIA	00
159C:	8B	LP0B	
159D:	DB	EXAM	
159E:	59	LDM	
159F:	C2	INCB	
15A0:	5A	SL	
15A1:	DA	EXAB	
15A2:	F5	CAL15	C2
15A4:	8C	LP0C	
15A5:	59	LDM	
15A6:	03	LIB	00
15A8:	8A	LP0A	
15A9:	14	ADB	
15AA:	24	IXL	
15AB:	34	PUSH	
15AC:	03	LIB	00
15AE:	F5	CAL15	C2
15B0:	82	LP02	
15B1:	13	LIQ	0A
15B3:	0A	MVB	
15B4:	84	LP04	
15B5:	14	ADB	
15B6:	F5	CAL15	F6
15B8:	5B	POP	
15B9:	37	RTN	
15BA:	B3	LP33	
15BB:	62	TSIM	40
15BD:	7E	JPZ	0FC5
15C0:	D1	RC	
15C1:	37	RTN	
15C2:	B0	LP30	
15C3:	13	LIQ	02
15C5:	0A	MVB	
15C6:	02	LIA	00
15C8:	03	LIB	00
15CA:	B0	LP30	
15CB:	13	LIQ	02
15CD:	0B	EXB	
15CE:	67	CPIA	00
15D0:	28	JRNZP	07 >15D8
15D2:	DA	EXAB	
15D3:	67	CPIA	00
15D5:	38	JRZP	1B >15F1
15D7:	DA	EXAB	
15D8:	D1	RC	
15D9:	DA	EXAB	
15DA:	D2	SR	
15DB:	DA	EXAB	
15DC:	D2	SR	
15DD:	82	LP02	
15DE:	13	LIQ	0A
15E0:	0B	EXB	
15E1:	2A	JRNCP	05 >15E7
15E3:	B0	LP30	
15E4:	14	ADB	
15E5:	3A	JRCP	0F >15F5
15E7:	F5	CAL15	59
15E9:	3A	JRCP	0B >15F5
15EB:	82	LP02	
15EC:	13	LIQ	0A
15EE:	0B	EXB	
15EF:	2B	JRNCM	22 >15CE
15F1:	8A	LP0A	
15F2:	13	LIQ	30
15F4:	0B	EXB	
15F5:	37	RTN	
15F6:	F0	CAL10	2E
15F8:	B3	LP33	
15F9:	62	TSIM	20
15FB:	38	JRZP	40 >163C
15FD:	62	TSIM	04
15FF:	38	JRZP	5E >165E
1601:	02	LIA	D0
1603:	94	LP14	
1604:	DB	EXAM	
1605:	10	LIDP	66F7
1608:	57	LDD	
1609:	03	LIB	67
160B:	95	LP15	
160C:	13	LIQ	02
160E:	0A	MVB	
160F:	86	LP06	
1610:	13	LIQ	02
1612:	0A	MVB	
1613:	07	DY	
1614:	89	LP09	
1615:	60	ANIM	00
1617:	B3	LP33	
1618:	62	TSIM	90
161A:	28	JRNZP	27 >1642
161C:	22	LDR	
161D:	74	ADIA	04
161F:	30	STP	
1620:	59	LDM	
1621:	88	LP08	
1622:	DB	EXAM	
1623:	86	LP06	
1624:	63	CPIM	AF
1626:	2A	JRNCP	2C >1653
1628:	24	IXL	
1629:	67	CPIA	00
162B:	38	JRZP	06 >1632
162D:	26	IYS	
162E:	C8	INCL	
162F:	49	DECK	
1630:	29	JRNZM	0E >1623
1632:	89	LP09	

1633:	59	LDM		
1634:	97	LP17		
1635:	DB	EXAM		
1636:	06	IY		
1637:	86	LP06		
1638:	10	LIDP	66F7	
163B:	53	MVDM		
163C:	F0	CAL10	39	
163E:	D1	RC		
163F:	F3	CAL13	44	
1641:	37	RTN		
1642:	02	LIA	07	
1644:	88	LP08		
1645:	DB	EXAM		
1646:	24	IXL		
1647:	67	CPIA	F5	
1649:	39	JRZM	27 >1623	
164B:	04	IX		
164C:	24	IXL		
164D:	67	CPIA	00	
164F:	39	JRZM	1E >1632	
1651:	2C	JRP	16 >1668	
1653:	24	IXL		
1654:	67	CPIA	00	
1656:	39	JRZM	25 >1632	
1658:	F0	CAL10	39	
165A:	EF	CAL0F	CD	
165C:	2D	JRM	1E >163F	
165E:	04	IX		
165F:	90	LP10		
1660:	00	LII	07	
1662:	18	MVWD		
1663:	90	LP10		
1664:	63	CPIM	F5	
1666:	29	JRNZM	2B >163C	
1668:	F0	CAL10	39	
166A:	EF	CAL0F	DD	
166C:	37	RTN		
166D:	B3	LP33		
166E:	62	TSIM	40	
1670:	28	JRNZP	1E >168F	
1672:	F5	CAL15	36	
1674:	3A	JRCP	1C >1691	
1676:	63	CPIM	1B	
1678:	2A	JRNCP	19 >1692	
167A:	63	CPIM	00	
167C:	38	JRZP	12 >168F	
167E:	02	LIA	01	
1680:	45	SBM		
1681:	02	LIA	C7	
1683:	03	LIB	65	
1685:	E2	CAL02	25	
1687:	98	LP18		
1688:	59	LDM		
1689:	F5	CAL15	52	
168B:	84	LP04		
168C:	15	SBB		
168D:	2D	JRM	98 >15F6	
168F:	EF	CAL0F	C5	
1691:	37	RTN		
1692:	10	LIDP	66D8	
1695:	D6	TSID	08	
1697:	28	JRNZP	26 >16BE	
1699:	10	LIDP	66FC	
169C:	84	LP04		
169D:	1A	MVBD		
169E:	02	LIA	07	
16A0:	03	LIB	00	
16A2:	84	LP04		
16A3:	15	SBB		
16A4:	F5	CAL15	4E	
16A6:	84	LP04		
16A7:	15	SBB		
16A8:	F3	CAL13	44	
16AA:	F5	CAL15	3A	
16AC:	7D	JPNC	0FD1	
16AF:	10	LIDP	66D8	
16B2:	D5	ORID	08	
16B4:	F5	CAL15	67	
16B6:	F5	CAL15	5F	
16B8:	02	LIA	03	
16BA:	03	LIB	00	
16BC:	2C	JRP	4D >170A	
16BE:	11	LIDL	DE	
16C0:	57	LDD		
16C1:	98	LP18		
16C2:	C7	CPMA		
16C3:	38	JRZP	03 >16C7	
16C5:	2A	JRNCP	18 >16DE	
16C7:	8B	LP0B		
16C8:	60	ANIM	00	
16CA:	8A	LP0A		
16CB:	02	LIA	40	
16CD:	DB	EXAM		
16CE:	F4	CAL14	FC	
16D0:	7F	JPC	0FC5	
16D3:	24	IXL		
16D4:	DA	EXAB		
16D5:	24	IXL		
16D6:	84	LP04		
16D7:	14	ADB		
16D8:	F5	CAL15	4E	
16DA:	84	LP04		
16DB:	15	SBB		
16DC:	2D	JRM	E7 >15F6	
16DE:	DA	EXAB		
16DF:	10	LIDP	66FC	
16E2:	84	LP04		
16E3:	1A	MVBD		
16E4:	05	DX		
16E5:	24	IXL		
16E6:	67	CPIA	40	
16E8:	7C	JPNZ	0FC5	
16EB:	F3	CAL13	44	
16ED:	04	IX		
16EE:	98	LP18		
16EF:	59	LDM		
16F0:	DA	EXAB		
16F1:	83	LP03		
16F2:	45	SBM		
16F3:	02	LIA	00	
16F5:	DA	EXAB		
16F6:	F5	CAL15	59	
16F8:	F5	CAL15	59	
16FA:	F5	CAL15	59	
16FC:	86	LP06		
16FD:	15	SBB		
16FE:	F5	CAL15	3A	
1700:	7D	JPNC	0FD1	
1703:	F5	CAL15	67	
1705:	F5	CAL15	5F	
1707:	24	IXL		
1708:	DA	EXAB		
1709:	24	IXL		
170A:	88	LP08		
170B:	13	LIQ	02	
170D:	0B	EXB		
170E:	88	LP08		
170F:	14	ADB		
1710:	88	LP08		
1711:	13	LIQ	02	
1713:	0B	EXB		
1714:	DA	EXAB		
1715:	26	IYS		
1716:	DA	EXAB		
1717:	26	IYS		
1718:	06	IY		
1719:	06	IY		
171A:	02	LIA	08	
171C:	26	IYS		
171D:	F3	CAL13	49	
171F:	02	LIA	00	
1721:	03	LIB	00	
1723:	82	LP02		
1724:	13	LIQ	08	
1726:	0B	EXB		
1727:	78	CALL	800C	
172A:	98	LP18		
172B:	10	LIDP	66DE	
172E:	53	MVDM		
172F:	79	JP	15F6	
1732:	B3	LP33		
1733:	60	ANIM	63	
1735:	8B	LP0B		
1736:	60	ANIM	00	
1738:	24	IXL		
1739:	8A	LP0A		
173A:	DB	EXAM		
173B:	59	LDM		
173C:	67	CPIA	5B	
173E:	7D	JPNC	0FC1	
1741:	67	CPIA	41	
1743:	7F	JPC	0FC1	
1746:	24	IXL		
1747:	67	CPIA	28	
1749:	28	JRNZP	2C >1776	
174B:	8B	LP0B		
174C:	61	ORIM	80	
174E:	B3	LP33		
174F:	61	ORIM	08	

1751:	8A	LP0A		
1752:	63	CPIM	41	
1754:	28	JRNZP	11	>1766
1756:	10	LIDP	66D8	
1759:	D6	TSID	40	
175B:	28	JRNZP	14	>1770
175D:	D5	ORID	40	
175F:	B3	LP33		
1760:	62	TSIM	40	
1762:	38	JRZP	05	>1768
1764:	D5	ORID	90	
1766:	D1	RC		
1767:	37	RTN		
1768:	B3	LP33		
1769:	61	ORIM	90	
176B:	8A	LP0A		
176C:	60	ANIM	40	
176E:	2D	JRM	09	>1766
1770:	D6	TSID	80	
1772:	39	JRZM	0B	>1768
1774:	2D	JRM	0F	>1766
1776:	67	CPIA	24	
1778:	28	JRNZP	14	>178D
177A:	8B	LP0B		
177B:	61	ORIM	20	
177D:	B3	LP33		
177E:	61	ORIM	04	
1780:	24	IXL		
1781:	67	CPIA	28	
1783:	28	JRNZP	03	>1787
1785:	2D	JRM	3B	>174B
1787:	B3	LP33		
1788:	61	ORIM	10	
178A:	05	DX		
178B:	D1	RC		
178C:	37	RTN		
178D:	67	CPIA	5B	
178F:	2B	JRNCM	09	>1787
1791:	67	CPIA	41	
1793:	2A	JRNCP	09	>179D
1795:	67	CPIA	3A	
1797:	2B	JRNCM	11	>1787
1799:	67	CPIA	30	
179B:	3B	JRCM	15	>1787
179D:	8B	LP0B		
179E:	DB	EXAM		
179F:	24	IXL		
17A0:	67	CPIA	28	
17A2:	28	JRNZP	1C	>17BF
17A4:	B3	LP33		
17A5:	61	ORIM	08	
17A7:	02	LIA	80	
17A9:	8B	LP0B		
17AA:	44	ADM		
17AB:	D1	RC		
17AC:	37	RTN		
17AD:	67	CPIA	5B	
17AF:	2B	JRNCM	26	>178A
17B1:	67	CPIA	41	
17B3:	2B	JRNCM	15	>179F
17B5:	67	CPIA	3A	
17B7:	2B	JRNCM	2E	>178A
17B9:	67	CPIA	30	
17BB:	3B	JRCM	32	>178A
17BD:	2D	JRM	1F	>179F
17BF:	67	CPIA	24	
17C1:	29	JRNZM	15	>17AD
17C3:	B3	LP33		
17C4:	61	ORIM	04	
17C6:	02	LIA	40	
17C8:	8B	LP0B		
17C9:	44	ADM		
17CA:	24	IXL		
17CB:	67	CPIA	28	
17CD:	29	JRNZM	44	>178A
17CF:	2D	JRM	2C	>17A4
17D1:	F7	CAL17	32	
17D3:	3A	JRCP	1E	>17F2
17D5:	B3	LP33		
17D6:	62	TSIM	80	
17D8:	38	JRZP	1A	>17F3
17DA:	59	LDM		
17DB:	34	PUSH		
17DC:	B7	LP37		
17DD:	61	ORIM	80	
17DF:	E9	CAL09	88	
17E1:	5B	POP		
17E2:	B3	LP33		
17E3:	DB	EXAM		
17E4:	3A	JRCP	0D	>17F2
17E6:	24	IXL		
17E7:	67	CPIA	29	
17E9:	7C	JPNZ	0FC1	
17EC:	EF	CAL0F	65	
17EE:	F6	CAL16	6D	
17F0:	EF	CAL0F	65	
17F2:	37	RTN		
17F3:	62	TSIM	10	
17F5:	38	JRZP	12	>1808
17F7:	62	TSIM	40	
17F9:	7C	JPNZ	0FDD	
17FC:	8A	LP0A		
17FD:	59	LDM		
17FE:	75	SBIA	41	
1800:	98	LP18		
1801:	DB	EXAM		
1802:	EF	CAL0F	65	
1804:	F6	CAL16	81	
1806:	2D	JRM	17	>17F0
1808:	62	TSIM	08	
180A:	38	JRZP	55	>1860
180C:	59	LDM		
180D:	34	PUSH		
180E:	8A	LP0A		
180F:	10	LIDP	6629	
1812:	1B	EXBD		
1813:	B7	LP37		
1814:	61	ORIM	80	
1816:	E9	CAL09	88	
1818:	3A	JRCP	45	>185E
181A:	F5	CAL15	36	
181C:	3A	JRCP	41	>185E
181E:	10	LIDP	66D0	
1821:	53	MVDM		
1822:	10	LIDP	66F5	
1825:	57	LDD		
1826:	67	CPIA	80	
1828:	38	JRZP	17	>1840
182A:	10	LIDP	2080	
182D:	90	LP10		
182E:	00	LII	07	
1830:	18	MVMD		
1831:	F5	CAL15	36	
1833:	3A	JRCP	2A	>185E
1835:	10	LIDP	66D1	
1838:	53	MVDM		
1839:	24	IXL		
183A:	67	CPIA	29	
183C:	28	JRNZP	1D	>185A
183E:	2C	JRP	0C	>184B
1840:	24	IXL		
1841:	67	CPIA	29	
1843:	28	JRNZP	16	>185A
1845:	10	LIDP	66D1	
1848:	02	LIA	00	
184A:	52	STD		
184B:	EF	CAL0F	65	
184D:	5B	POP		
184E:	B3	LP33		
184F:	DB	EXAM		
1850:	8A	LP0A		
1851:	10	LIDP	6629	
1854:	1A	MYBD		
1855:	F5	CAL15	75	
1857:	EF	CAL0F	65	
1859:	37	RTN		
185A:	5B	POP		
185B:	EF	CAL0F	C1	
185D:	37	RTN		
185E:	5B	POP		
185F:	37	RTN		
1860:	62	TSIM	40	
1862:	7C	JPNZ	0FC1	
1865:	EF	CAL0F	65	
1867:	F4	CAL14	FC	
1869:	3A	JRCP	0D	>1877
186B:	04	IX		
186C:	04	IX		
186D:	04	IX		
186E:	04	IX		
186F:	24	IXL		
1870:	34	PUSH		
1871:	F5	CAL15	F6	
1873:	5B	POP		
1874:	EF	CAL0F	65	
1876:	37	RTN		
1877:	B3	LP33		
1878:	62	TSIM	20	
187A:	38	JRZP	0E	>1889
187C:	E0	CAL00	E4	

19A0:	F9 CAL19	85	
19A2:	2C JRP	03	>19A6
19A4:	F9 CAL19	8C	
19A6:	02 LIA	08	
19A8:	44 ADM		
19A9:	04 IX		
19AA:	00 LII	07	
19AC:	90 LP10		
19AD:	18 MVWD		
19AE:	37 RTN		
19AF:	10 LIDP	6689	
19B2:	2C JRP	04	>19B7
19B4:	10 LIDP	6688	
19B7:	57 LDD		
19B8:	67 CPIA	09	
19BA:	2A JRNCP	05	>19C0
19BC:	42 INCA		
19BD:	52 STD		
19BE:	D1 RC		
19BF:	37 RTN		
19C0:	D0 SC		
19C1:	37 RTN		
19C2:	10 LIDP	207E	
19C5:	37 RTN		
19C6:	10 LIDP	207F	
19C9:	37 RTN		
19CA:	84 LP04		
19CB:	10 LIDP	6685	
19CE:	1B EXBD		
19CF:	84 LP04		
19D0:	10 LIDP	6685	
19D3:	1A MVBD		
19D4:	37 RTN		
19D5:	86 LP06		
19D6:	10 LIDP	6621	
19D9:	1A MVBD		
19DA:	06 IY		
19DB:	57 LDD		
19DC:	67 CPIA	FF	
19DE:	37 RTN		
19DF:	10 LIDP	6688	
19E2:	2C JRP	08	>19EB
19E4:	10 LIDP	6689	
19E7:	2C JRP	03	>19EB
19E9:	F9 CAL19	C2	
19EB:	D4 ANID	00	
19ED:	37 RTN		
19EE:	F3 CAL13	57	
19F0:	D6 TSID	02	
19F2:	28 JRNZP	05	>19F8
19F4:	D5 ORID	02	
19F6:	F4 CAL14	E2	
19F8:	37 RTN		
19F9:	86 LP06		
19FA:	10 LIDP	6621	
19FD:	1A MVBD		
19FE:	06 IY		
19FF:	86 LP06		
1A00:	10 LIDP	6685	
1A03:	1B EXBD		
1A04:	8A LP0A		
1A05:	10 LIDP	6621	
1A08:	1A MVBD		
1A09:	4A 4A		
1A0A:	8A LP3A		
1A0B:	13 LIQ	0A	
1A0D:	0A MYB		
1A0E:	8A LP3A		
1A0F:	10 LIDP	6625	
1A12:	1B EXBD		
1A13:	8A LP3A		
1A14:	10 LIDP	6625	
1A17:	1A MVBD		
1A18:	37 RTN		
1A19:	84 LP04		
1A1A:	13 LIQ	3A	
1A1C:	0A MYB		
1A1D:	37 RTN		
1A1E:	8A LP0A		
1A1F:	02 LIA	07	
1A21:	DB EXAM		
1A22:	88 LP08		
1A23:	02 LIA	10	
1A25:	DB EXAM		
1A26:	8A LP3A		
1A27:	13 LIQ	04	
1A29:	0A MYB		
1A2A:	37 RTN		
1A2B:	10 LIDP	668C	
1A2E:	57 LDD		
1A2F:	67 CPIA	7F	
1A31:	37 RTN		
1A32:	67 CPIA	41	
1A34:	3A JRCP	13	>1A48
1A36:	67 CPIA	5B	
1A38:	3A JRCP	0D	>1A46
1A3A:	67 CPIA	61	
1A3C:	3A JRCP	0B	>1A48
1A3E:	67 CPIA	7B	
1A40:	3A JRCP	05	>1A46
1A42:	67 CPIA	FE	
1A44:	28 JRNZP	03	>1A48
1A46:	D1 RC		
1A47:	37 RTN		
1A48:	D0 SC		
1A49:	37 RTN		
1A4A:	E0 CAL00	E7	
1A4C:	88 LP08		
1A4D:	02 LIA	10	
1A4F:	DB EXAM		
1A50:	05 DX		
1A51:	24 IXL		
1A52:	67 CPIA	0D	
1A54:	38 JRZP	19	>1A6E
1A56:	67 CPIA	FE	
1A58:	38 JRZP	17	>1A70
1A5A:	67 CPIA	30	
1A5C:	3A JRCP	16	>1A73
1A5E:	67 CPIA	7B	
1A60:	2A JRNCP	0D	>1A6E
1A62:	67 CPIA	3A	
1A64:	38 JRZP	09	>1A6E
1A66:	67 CPIA	3D	
1A68:	38 JRZP	05	>1A6E
1A6A:	67 CPIA	5E	
1A6C:	28 JRNZP	12	>1A7F
1A6E:	05 DX		
1A6F:	37 RTN		
1A70:	24 IXL		
1A71:	2C JRP	0D	>1A7F
1A73:	67 CPIA	23	
1A75:	39 JRZM	08	>1A6E
1A77:	67 CPIA	2C	
1A79:	38 JRZP	05	>1A7F
1A7B:	67 CPIA	28	
1A7D:	2B JRNCM	10	>1A6E
1A7F:	88 LP08		
1A80:	63 CPIM	17	
1A82:	2B JRNCM	32	>1A51
1A84:	34 PUSH		
1A85:	59 LDM		
1A86:	30 STP		
1A87:	5B POP		
1A88:	DB EXAM		
1A89:	48 INCK		
1A8A:	2D JRM	3A	>1A51
1A8C:	F9 CAL19	53	
1A8E:	F9 CAL19	66	
1A90:	8B LP0B		
1A91:	02 LIA	0A	
1A93:	DB EXAM		
1A94:	04 IX		
1A95:	FA CAL1A	1E	
1A97:	05 DX		
1A98:	24 IXL		
1A99:	66 TSIA	80	
1A9B:	38 JRZP	1B	>1A87
1A9D:	88 LP08		
1A9E:	59 LDM		
1A9F:	30 STP		
1AA0:	24 IXL		
1AA1:	C7 CPMA		
1AA2:	28 JRNZP	0C	>1AAF
1AA4:	67 CPIA	00	
1AA6:	38 JRZP	05	>1AAC
1AA8:	48 INCK		
1AA9:	4B 4B		
1AAA:	29 JRNZM	0E	>1A9D
1AAC:	D1 RC		
1AAD:	2C JRP	0B	>1AB9
1AAF:	FA CAL1A	19	
1AB1:	84 LP04		
1AB2:	70 ADIM	0F	
1AB4:	CB CB		
1AB5:	29 JRNZM	22	>1A94
1AB7:	EF CAL0F	C9	
1AB9:	F9 CAL19	53	
1ABB:	37 RTN		
1ABC:	F9 CAL19	53	
1ABE:	84 LP04		

1ABF: 10 LIDP 6621	1B19: 98 LP18	1B73: 18 EXBD
1AC2: 1A MVBD	1B1A: 59 LDM	1B74: 37 RTN
1AC3: 24 IXL	1B1B: 88 LP08	1B75: 10 LIDP 66B8
1AC4: 67 CPIA FF	1B1C: 07 CPMA	1B78: 00 LII 17
1AC6: 38 JRZP 2B >1AF2	1B1D: 37 RTN	1B7A: 90 LP10
1AC8: FA CAL1A 1E	1B1E: 03 LIB 01	1B7B: 18 MVWD
1ACA: 88 LP08	1B20: F4 CAL14 A9	1B7C: 11 LIDL 10
1ACB: 59 LDM	1B22: 24 IXL	1B7E: 57 LDD
1ACC: 30 STP	1B23: 67 CPIA 2E	1B7F: 74 ADIA 10
1ACD: 24 IXL	1B25: 29 JRNZM 08 >1B1E	1B81: 88 LP08
1ACE: 67 CPIA 3D	1B27: 06 IY	1B82: DB EXAM
1AD0: 28 JRNZP 07 >1AD8	1B28: 37 RTN	1B83: 37 RTN
1AD2: 63 CPIM 00	1B29: DA EXAB	1B84: 10 LIDP 66B8
1AD4: 28 JRNZP 12 >1AE7	1B2A: 8A LP0A	1B87: 90 LP10
1AD6: 2C JRP 0D >1AE4	1B2B: 59 LDM	1B88: 00 LII 17
1AD8: 67 CPIA FE	1B2C: 83 LP03	1B8A: 19 EXWD
1ADA: 28 JRNZP 02 >1ADD	1B2D: 45 SBM	1B8B: 02 LIA 20
1ADC: 24 IXL	1B2E: 37 RTN	1B8D: 90 LP10
1ADD: 07 CPMA	1B2F: 02 LIA 60	1B8E: 1E FILM
1ADE: 28 JRNZP 08 >1AE7	1B31: 10 LIDP 6650	1B8F: 37 RTN
1AE0: 48 INCK	1B34: 00 LII 2F	1B90: FB CAL1B 75
1AE1: 4B 4B	1B36: 1F FILD	1B92: 78 CALL 8021
1AE2: 29 JRNZM 19 >1ACA	1B37: 37 RTN	1B95: 10 LIDP 66B8
1AE4: D1 RC	1B38: 00 LII 17	1B98: 00 LII 17
1AE5: 2C JRP 10 >1AF6	1B3A: 90 LP10	1B9A: 02 LIA 00
1AE7: 67 CPIA FF	1B3B: 10 LIDP 6650	1B9C: 1F FILD
1AE9: 38 JRZP 0A >1AF4	1B3E: 18 MVWD	1B9D: 88 LP08
1AEB: 67 CPIA 23	1B3F: 11 LIDL 1E	1B9E: 02 LIA 10
1AED: 39 JRZM 26 >1AC8	1B41: 57 LDD	1BA0: DB EXAM
1AEF: 24 IXL	1B42: 74 ADIA 10	1BA1: 10 LIDP 661D
1AF0: 2D JRM 0A >1AE7	1B44: 88 LP08	1BA4: D4 ANID 00
1AF2: FA CAL1A 26	1B45: DB EXAM	1BA6: 37 RTN
1AF4: EF CAL0F C9	1B46: 37 RTN	1BA7: 02 LIA 0D
1AF6: F9 CAL19 53	1B47: 00 LII 17	1BA9: 10 LIDP 67B0
1AF8: 37 RTN	1B49: 90 LP10	1BAC: 00 LII 4F
1AF9: 59 LDM	1B4A: 10 LIDP 6668	1BAE: 1F FILD
1AFA: 89 LP09	1B4D: 18 MVWD	1BAF: 37 RTN
1AFB: DB EXAM	1B4E: 75 SBIA 18	1BB0: 02 LIA 60
1AFC: 59 LDM	1B50: 2D JRM 0F >1B42	1BB2: 00 LII 17
1AFD: 67 CPIA 00	1B52: 90 LP10	1BB4: 90 LP10
1AFF: 38 JRZP 14 >1B14	1B53: 10 LIDP 6650	1BB5: 1E FILM
1B01: 8A LP0A	1B56: 00 LII 17	1BB6: 37 RTN
1B02: DB EXAM	1B58: 19 EXWD	1BB7: 10 LIDP 661F
1B03: 88 LP0B	1B59: 90 LP10	1BBA: 37 RTN
1B04: 60 ANIM 00	1B5A: 11 LIDL 50	1BBB: 10 LIDP 663D
1B06: 24 IXL	1B5C: 18 MVWD	1BBE: D6 TSID 04
1B07: 67 CPIA FE	1B5D: 37 RTN	1BC0: 37 RTN
1B09: 28 JRNZP 05 >1B0F	1B5E: 90 LP10	1BC1: E2 CAL02 2A
1B0B: C8 INCL	1B5F: 10 LIDP 6668	1BC3: 02 LIA 10
1B0C: CA CA	1B62: 00 LII 17	1BC5: 88 LP08
1B0D: 2D JRM 08 >1B06	1B64: 19 EXWD	1BC6: DB EXAM
1B0F: 4B 4B	1B65: 90 LP10	1BC7: 78 CALL 8042
1B10: 29 JRNZM 0B >1B06	1B66: 11 LIDL 68	1BCA: FB CAL1B 52
1B12: 89 LP09	1B68: 18 MVWD	1BCC: 37 RTN
1B13: 59 LDM	1B69: 37 RTN	1BCD: 10 LIDP 6650
1B14: 37 RTN	1B6A: 10 LIDP 6619	1BD0: 02 LIA 20
1B15: 97 LP17	1B6D: 8A LP0A	1BD2: 00 LII 17
1B16: 59 LDM	1B6E: 1A MVBD	1BD4: 1F FILD
1B17: 88 LP08	1B6F: 8A LP0A	1BD5: 37 RTN
1B18: DB EXAM	1B70: 10 LIDP 661B	1BD6: 10 LIDP 6668

1BD9: 2D JRM 0A >1BD0	1C40: 28 JRNZP 03 >1C44	1CA7: 38 JRZP 0D >1CB5
1BDB: FB CAL1B CD 045	1C42: 70 ADIM 01	1CA9: 67 CPIA 60
1BDD: FB CAL1B D6	1C44: 60 ANIM F8	1CAB: 38 JRZP 09 >1CB5
1BDF: 37 RTN	1C46: 70 ADIM 08	1CAD: 67 CPIA 5C
1BE0: 02 LIA 4F	1C48: 86 LP06	1CAF: 28 JRNZP 07 >1CB7
1BE2: 03 LIB 66	1C49: 63 CPIM 80	1CB1: D6 TSID 80
1BE4: E2 CAL02 22	1C4B: 2A JRNCP 03 >1C4F	1CB3: 28 JRNZP 03 >1CB7
1BE6: 37 RTN	1C4D: D1 RC	1CB5: 02 LIA 20
1BE7: 02 LIA 67	1C4E: 37 RTN	1CB7: 67 CPIA 20
1BE9: 2D JRM 08 >1BE2	1C4F: D0 SC	1CB9: 37 RTN
1BEB: 10 LIDP 6616	1C50: 37 RTN	1CBA: 67 CPIA A1
1BEE: 84 LP04	1C51: 10 LIDP 6612	1CBC: 3A JRCP 1F >1CDC
1BEF: 1B EXBD	1C54: D6 TSID 80	1CBE: D6 TSID 80
1BF0: 10 LIDP 6616	1C56: 37 RTN	1CC0: 38 JRZP 17 >1CD8
1BF3: 84 LP04	1C57: FC CAL1C 51	1CC2: 67 CPIA E0
1BF4: 1A MVBD	1C59: 38 JRZP 14 >1C6E	1CC4: 3A JRCP 11 >1CD6
1BF5: 37 RTN	1C5B: D6 TSID 02	1CC6: 67 CPIA F0
1BF6: 10 LIDP 6614	1C5D: 28 JRNZP 12 >1C70	1CC8: 3A JRCP 13 >1CDC
1BF9: 84 LP04	1C5F: D6 TSID 04	1CCA: 67 CPIA F9
1BFA: 1B EXBD	1C61: 28 JRNZP 0C >1C6E	1CCC: 3A JRCP 09 >1CD6
1BFB: 10 LIDP 6614	1C63: 10 LIDP 6611	1CCE: 67 CPIA FB
1BFE: 84 LP04	1C66: D6 TSID 10	1CD0: 3A JRCP 0B >1CDC
1BFF: 1A MVBD	1C68: 28 JRNZP 07 >1C70	1CD2: 67 CPIA FD
1C00: 37 RTN	1C6A: D6 TSID 08	1CD4: 2A JRNCP 07 >1CDC
1C01: 02 LIA 67	1C6C: 38 JRZP 03 >1C70	1CD6: D1 RC
1C03: 2C JRP 03 >1C07	1C6E: D1 RC	1CD7: 37 RTN
1C05: 02 LIA 4F	1C6F: 37 RTN	1CD8: 67 CPIA F5
1C07: 03 LIB 66	1C70: D0 SC	1CDA: 2B JRNCP 11 >1CCA
1C09: E2 CAL02 25	1C71: 37 RTN	1CDC: D0 SC
1C0B: 37 RTN	1C72: FB CAL1B 38	1CDD: 37 RTN
1C0C: FC CAL1C 01	1C74: 10 LIDP 2080	1CDE: 10 LIDP 66B8
1C0E: FB CAL1B E0	1C77: 90 LP10	1CE1: 00 LII 17
1C10: 2C JRP 05 >1C16	1C78: 19 EXWD	1CE3: 02 LIA 00
1C12: FC CAL1C 05	1C79: FB CAL1B 47	1CE5: 1F FILD
1C14: FB CAL1B E7	1C7B: 11 LIDL 98	1CE6: 37 RTN
1C16: 03 LIB 18	1C7D: 90 LP10	1CE7: DA EXAB
1C18: 24 IXL	1C7E: 19 EXWD	1CE8: 10 LIDP 660D
1C19: 26 IYS	1C7F: 37 RTN	1CEB: 57 LDD
1C1A: C3 DECB	1C80: 10 LIDP 2080	1CEC: 83 LP03
1C1B: 29 JRNZM 04 >1C18	1C83: 00 LII 17	1CED: 44 ADM
1C1D: 37 RTN	1C85: 90 LP10	1CEE: DA EXAB
1C1E: 24 IXL	1C86: 18 MVWD	1CEF: 37 RTN
1C1F: 67 CPIA E0	1C87: FB CAL1B 52	1CF0: DA EXAB
1C21: 2A JRNCP 04 >1C26	1C89: 11 LIDL 98	1CF1: 10 LIDP 660E
1C23: 26 IYS	1C8B: 90 LP10	1CF4: 2D JRM 0A >1CEB
1C24: 2D JRM 07 >1C1E	1C8C: 18 MVWD	1CF6: DA EXAB
1C26: 66 TSIA 08	1C8D: FB CAL1B 5E	1CF7: 10 LIDP 660F
1C28: 29 JRNZM 06 >1C23	1C8F: 37 RTN	1CFA: 2D JRM 10 >1CEB
1C2A: 37 RTN	1C90: 24 IXL	1CFC: 00 LII FF
1C2B: 10 LIDP 6611	1C91: 67 CPIA 7B	1CFE: 00 LII FF
1C2E: D4 ANID 00	1C93: 3A JRCP 0A >1C9E	1D00: 87 LP37
1C30: 10 LIDP 6612	1C95: 67 CPIA F0	1D01: 02 LIA 00
1C33: D4 ANID 70	1C97: 3A JRCP 44 >1CDC	1D03: FF CALIF 00
1C35: 37 RTN	1C99: 67 CPIA FE	1D05: FF CALIF 00
1C36: 86 LP06	1C9B: 28 JRNZP 02 >1C9E	1D07: FF CALIF 00
1C37: 59 LDM	1C9D: 24 IXL	1D09: FF CALIF 00
1C38: 64 ANIA 0F	1C9E: 10 LIDP 203C	1D0B: FF CALIF 00
1C3A: 67 CPIA 0F	1CA1: 67 CPIA 7B	1D0D: FF CALIF 00
1C3C: 38 JRZP 05 >1C42	1CA3: 2A JRNCP 16 >1CBA	1D0F: FF CALIF 00
1C3E: 67 CPIA 07	1CA5: 67 CPIA 00	1D11: FD CALID 00

1D13:	FF	CAL1F	00	1D89:	BF	LP3F		1DF9:	FF	CAL1F	00
1D15:	FF	CAL1F	80	1D8A:	00	LII	FF	1DFB:	FF	CAL1F	00
1D17:	FF	CAL1F	00	1D8C:	00	LII	FF	1DFD:	FF	CAL1F	00
1D19:	FF	CAL1F	00	1D8E:	00	LII	FF	1DFF:	FF	CAL1F	BA
1D1B:	FF	CAL1F	00	1D90:	81	LP01		1E01:	71	SBIM	00
1D1D:	FF	CAL1F	00	1D91:	FF	CAL1F	00	1E03:	FF	CAL1F	00
1D1F:	FF	CAL1F	00	1D93:	FF	CAL1F	00	1E05:	FF	CAL1F	00
1D21:	FF	CAL1F	00	1D95:	FF	CAL1F	00	1E07:	FF	CAL1F	00
1D23:	FF	CAL1F	00	1D97:	FF	CAL1F	00	1E09:	FF	CAL1F	00
1D25:	FF	CAL1F	01	1D99:	FF	CAL1F	00	1E0B:	FF	CAL1F	00
1D27:	FF	CAL1F	00	1D9B:	FF	CAL1F	00	1E0D:	FF	CAL1F	00
1D29:	FF	CAL1F	00	1D9D:	FF	CAL1F	00	1E0F:	FF	CAL1F	00
1D2B:	FF	CAL1F	00	1D9F:	BF	LP3F		1E11:	FF	CAL1F	00
1D2D:	FF	CAL1F	00	1DA0:	00	LII	FF	1E13:	FF	CAL1F	00
1D2F:	FF	CAL1F	80	1DA2:	00	LII	FF	1E15:	FF	CAL1F	00
1D31:	BF	LP3F		1DA4:	00	LII	FF	1E17:	FF	CAL1F	00
1D32:	00	LII	FF	1DA6:	00	LII	FF	1E19:	FF	CAL1F	00
1D34:	00	LII	FF	1DA8:	00	LII	FF	1E1B:	FF	CAL1F	00
1D36:	00	LII	FF	1DAA:	00	LII	FF	1E1D:	FF	CAL1F	00
1D38:	00	LII	BF	1DAC:	00	LII	FF	1E1F:	FF	CAL1F	06
1D3A:	00	LII	FF	1DAE:	80	LP00		1E21:	FF	CAL1F	00
1D3C:	00	LII	FF	1DAF:	FF	CAL1F	00	1E23:	FF	CAL1F	00
1D3E:	00	LII	FF	1DB1:	7F	JPC	00FF	1E25:	FF	CAL1F	00
1D40:	00	LII	FF	1DB4:	00	LII	FF	1E27:	FF	CAL1F	00
1D42:	00	LII	FF	1DB6:	00	LII	FF	1E29:	FF	CAL1F	00
1D44:	00	LII	FF	1DB8:	00	LII	FF	1E2B:	FF	CAL1F	00
1D46:	00	LII	FF	1DBA:	00	LII	FF	1E2D:	FF	CAL1F	00
1D48:	00	LII	FF	1DBC:	00	LII	FF	1E2F:	BF	LP3F	
1D4A:	00	LII	FF	1DBE:	20	LDP		1E30:	00	LII	FF
1D4C:	00	LII	FF	1DBF:	FF	CAL1F	20	1E32:	00	LII	FF
1D4E:	04	IX		1DC1:	FF	CAL1F	00	1E34:	00	LII	FF
1D4F:	FD	CAL1D	40	1DC3:	FF	CAL1F	00	1E36:	00	LII	FF
1D51:	FF	CAL1F	00	1DC5:	FF	CAL1F	20	1E38:	00	LII	FF
1D53:	FF	CAL1F	00	1DC7:	FF	CAL1F	01	1E3A:	00	LII	FF
1D55:	FF	CAL1F	00	1DC9:	FF	CAL1F	00	1E3C:	00	LII	FF
1D57:	FF	CAL1F	00	1DCB:	FF	CAL1F	00	1E3E:	00	LII	FF
1D59:	FF	CAL1F	00	1DCD:	FF	CAL1F	10	1E40:	00	LII	FF
1D5B:	FF	CAL1F	00	1DCF:	FF	CAL1F	00	1E42:	00	LII	FF
1D5D:	FF	CAL1F	40	1DD1:	DF	OUTC		1E44:	00	LII	FF
1D5F:	FE	CAL1E	00	1DD2:	00	LII	FF	1E46:	00	LII	FF
1D61:	FF	CAL1F	00	1DD4:	00	LII	FF	1E48:	00	LII	FF
1D63:	FF	CAL1F	00	1DD6:	00	LII	FF	1E4A:	00	LII	FF
1D65:	FF	CAL1F	00	1DD8:	00	LII	FF	1E4C:	00	LII	FF
1D67:	FF	CAL1F	00	1DDA:	00	LII	FF	1E4E:	00	LII	FF
1D69:	FF	CAL1F	00	1DDC:	00	LII	FF	1E50:	00	LII	BF
1D6B:	FF	CAL1F	00	1DDE:	00	LII	FF	1E52:	00	LII	FF
1D6D:	FF	CAL1F	00	1DE0:	00	LII	FF	1E54:	00	LII	FF
1D6F:	FF	CAL1F	00	1DE2:	00	LII	FF	1E56:	40	INCI	
1D71:	FF	CAL1F	00	1DE4:	00	LII	FF	1E57:	FF	CAL1F	00
1D73:	FF	CAL1F	00	1DE6:	00	LII	FF	1E59:	FF	CAL1F	00
1D75:	FF	CAL1F	00	1DE8:	00	LII	FF	1E5B:	FF	CAL1F	00
1D77:	FF	CAL1F	00	1DEA:	00	LII	FF	1E5D:	FF	CAL1F	00
1D79:	FF	CAL1F	00	1DEC:	00	LII	FF	1E5F:	FD	CAL1D	02
1D7B:	FF	CAL1F	00	1DEE:	40	INCI		1E61:	FF	CAL1F	00
1D7D:	FF	CAL1F	00	1DEF:	5F	OUTF		1E63:	FF	CAL1F	00
1D7F:	FF	CAL1F	00	1DF0:	40	INCI		1E65:	FF	CAL1F	00
1D81:	FF	CAL1F	00	1DF1:	FF	CAL1F	00	1E67:	FF	CAL1F	00
1D83:	FF	CAL1F	00	1DF3:	FF	CAL1F	00	1E69:	FF	CAL1F	00
1D85:	FF	CAL1F	00	1DF5:	FF	CAL1F	00	1E6B:	FF	CAL1F	00
1D87:	FF	CAL1F	01	1DF7:	FF	CAL1F	00	1E6D:	FF	CAL1F	00

1E6F: FF CAL1F 00	1EE3: FF CAL1F 00	1F5A: 00 LII FF	1FCF: 00 LII 00
1E71: FF CAL1F 00	1EE5: FF CAL1F 00	1F5C: 00 LII FF	1FD1: 00 LII 00
1E73: FF CAL1F 00	1EE7: EF CAL0F 00	1F5E: 00 LII 00	1FD3: FF CAL1F 00
1E75: FF CAL1F 00	1EE9: FF CAL1F 00	1F60: 00 LII 00	1FD5: FF CAL1F 00
1E77: FF CAL1F 00	1EEB: FF CAL1F 00	1F62: 00 LII FF	1FD7: 81 LP01
1E79: FE CAL1E 00	1EED: FF CAL1F 00	1F64: 00 LII FF	1FD8: 00 LII 00
1E7B: FF CAL1F 00	1EEF: FF CAL1F 88	1F66: 00 LII 00	1FDA: 00 LII FF
1E7D: FF CAL1F 02	1EF1: FF CAL1F 00	1F68: 00 LII 00	1FDC: 00 LII FF
1E7F: FF CAL1F 00	1EF3: FF CAL1F 00	1F6A: 00 LII FF	1FDE: 00 LII 00
1E81: FF CAL1F 00	1EF5: FF CAL1F 20	1F6C: 00 LII FF	1FE0: 00 LII 00
1E83: FF CAL1F 00	1EF7: FF CAL1F 00	1F6E: 00 LII 00	1FE2: 00 LII FF
1E85: FF CAL1F 00	1EF9: FD CAL1D 00	1F70: 00 LII 00	1FE4: 00 LII FF
1E87: FF CAL1F 00	1EFB: FF CAL1F 00	1F72: 00 LII FF	1FE6: 00 LII 00
1E89: FF CAL1F 00	1EFD: FF CAL1F 02	1F74: 00 LII FF	1FE8: 00 LII 00
1E8B: FF CAL1F 00	1EFF: 7F JPC DE86	1F76: 00 LII 00	1FEA: 00 LII FF
1E8D: FF CAL1F 00	1F02: 00 LII FF	1F78: 00 LII 10	1FEC: 00 LII FF
1E8F: FF CAL1F 00	1F04: 00 LII FF	1F7A: 00 LII FF	1FEE: 00 LII 00
1E91: FF CAL1F 00	1F06: 00 LII 00	1F7C: 00 LII FF	1FF0: D1 RC
1E93: FF CAL1F 00	1F08: 00 LII 08	1F7E: 00 LII 00	1FF1: 42 INCA
1E95: FF CAL1F 00	1F0A: 00 LII FF	1F80: 00 LII 00	1FF2: 5B POP
1E97: FF CAL1F 00	1F0C: 00 LII FF	1F82: 00 LII FF	1FF3: 5B POP
1E99: FF CAL1F 00	1F0E: 00 LII 00	1F84: 00 LII FF	1FF4: 37 RTN
1E9B: FF CAL1F 00	1F10: 00 LII 00	1F86: 00 LII 00	1FF5: FF CAL1F 00
1E9D: FF CAL1F 00	1F12: 00 LII FF	1F88: 00 LII 00	1FF7: 00 LII 00
1E9F: FF CAL1F 00	1F14: 00 LII FF	1F8A: 00 LII FF	1FF9: 00 LII 00
1EA1: FF CAL1F 00	1F16: 00 LII 10	1F8C: 00 LII FF	1FFB: FF CAL1F 00
1EA3: FF CAL1F 00	1F18: 00 LII 14	1F8E: 00 LII 00	1FFD: FF CAL1F 10
1EA5: FF CAL1F 00	1F1A: 00 LII FF	1F90: 40 INCI	1FFF: 00 LII 08
1EA7: FF CAL1F 20	1F1C: 00 LII FF	1F91: 00 LII 00	
1EA9: B7 LP37	1F1E: 00 LII 00	1F93: FF CAL1F 00	
1EAA: 00 LII FF	1F20: 00 LII 00	1F95: FF CAL1F 00	
1EAC: 00 LII FF	1F22: 00 LII FF	1F97: 00 LII 00	
1EAE: 02 LIA FF	1F24: 00 LII FF	1F99: 00 LII 00	
1EB0: 00 LII FF	1F26: 00 LII 10	1F9B: FF CAL1F 00	
1EB2: 00 LII FF	1F28: 02 LIA 00	1F9D: FF CAL1F 00	
1EB4: 00 LII FF	1F2A: 00 LII FF	1F9F: 00 LII 00	
1EB6: 00 LII FF	1F2C: 00 LII FF	1FA1: 00 LII 00	
1EB8: 00 LII FF	1F2E: 00 LII 00	1FA3: FF CAL1F 00	
1EBA: 00 LII FF	1F30: 02 LIA 00	1FA5: FF CAL1F 02	
1EBC: 00 LII FF	1F32: 00 LII FF	1FA7: 00 LII 00	
1EBE: 05 DX	1F34: 00 LII FF	1FA9: 00 LII 00	
1EBF: FF CAL1F 00	1F36: 00 LII 00	1FAB: FF CAL1F 00	
1EC1: FD CAL1D 00	1F38: 90 LP10	1FAD: FF CAL1F 02	
1EC3: FF CAL1F 00	1F39: 00 LII 00	1FAF: 00 LII 00	
1EC5: FF CAL1F 00	1F3B: FF CAL1F 00	1FB1: 00 LII 00	
1EC7: FF CAL1F 00	1F3D: FF CAL1F 80	1FB3: FF CAL1F 00	
1EC9: FF CAL1F 00	1F3F: 08 MVW	1FB5: FF CAL1F 00	
1ECB: FF CAL1F 00	1F40: 40 INCI	1FB7: 00 LII 00	
1ECD: FF CAL1F 00	1F41: 00 LII 00	1FB9: 00 LII 00	
1ECF: FF CAL1F 00	1F43: FF CAL1F 00	1FBB: FF CAL1F 00	
1ED1: DF OUTC	1F45: FF CAL1F 44	1FBD: FF CAL1F 00	
1ED2: 00 LII FF	1F47: 00 LII 00	1FBF: 08 MVW	
1ED4: 00 LII FF	1F49: 00 LII 00	1FC0: 00 LII 00	
1ED6: 00 LII FF	1F4B: FF CAL1F 00	1FC2: 00 LII FF	
1ED8: 00 LII FF	1F4D: FF CAL1F 00	1FC4: 00 LII FF	
1EDA: 00 LII FF	1F4F: 10 LIDP 0000	1FC6: 00 LII 00	
1EDC: 00 LII FF	1F52: 00 LII FF	1FC8: 00 LII 00	
1EDE: 00 LII FF	1F54: 00 LII FF	1FCA: 00 LII FF	
1EE0: 20 LDP	1F56: 00 LII 04	1FCC: 00 LII FF	
1EE1: FF CAL1F 00	1F58: 00 LII 00	1FCE: 08 MVW	

Stichwortverzeichnis

Abkürzungen	III-4
ADB	III.2-1
ADCM	III.2-1
ADIA	III.2-1
ADIM	III.2-1
ADM	III.2-1
ADN	III.2-2
Adressbus	II-1
Adress-Pointer	II-1, III
ADW	III.2-3
Akkumulator	II, III
Alte Version	E-2
ALU	II-1
ANIA	III.3-1
ANID	III.3-1
ANIM	III.3-1
Anzeigeroutine	VII.3-1
Arithmetische Befehle	III-1, III.2-1
Ausdruckroutine	VII.3
Ausgabe-Befehle	III-2, III.8
Basic-Interpreter	E, IV, VII
Basicprogramm Speicher	VII.4
BCD	I-2, III-1
Befehlssatz	E-1, III
Befehlslisting des Kern-ROM	A
Beginn des Basicprogramm	VII.4-2
Beginn der Feldvariablen	VII.4-2
Bit	I-2, II-1
Bittest-Befehle	III-1, III.3
Blockoperationen	II-1
Breakpoint	E-2
Breapoint-Monitor	E-2, III-2, VI.2
Carry-Flag	II, III
CAL	III.7-1
CALL	E-1, III.7-1, IV

CLOADM	E-1, IV
Computerwort	I
Controlbus	II-1
CPCAL	III.9-2
CPIA	III.3-1
CPIM	III.3-1
CPMA	III.3-2
CPU	E-1, II-1, III
CPU-Register	II, III
CPU-Steuerbefehle	III-1, III.5
CSAVEM	E-1, IV
DATA	III.9-1
Datapointer	II-1, III
Datenbus	II
Daten Memory	II
DECA	III.1-4
DECB	III.1-4
DECI	III.1-4
DECJ	III.1-4
DECK	III.1-5
DECL	III.1-5
DECM	III.9-3
DECN	III.9-3
DECP	III.1-4
Dekrement-Befehle	II-4, III.1-4, IV.1
Dezimalsystem	I-1
Disassembler	E-1, V.1
Dreibyte Befehle	III-2
Drucker CE-126P	VII.3
DTLRA	III.9-2
Dualzahlen	I-1
DX	III.1-5
DXL	III.1-5
DY	III.1-5
DYS	III.1-5
Easy Simulation Programming	VI-1, VII.4-1

Eingabe-Befehle	III-2, III.8
Ende des Basicprogramm	VII.4-2
EXAB	III.1-2
EXAM	III.1-2
EXB	III.1-4
EXBD	III.1-4
Externes Memory	II, III
EXW	III.1-4
EXWD	III.1-4
Feldvariable	VI-1
FILD	III.1-6
FILM	III.1-6
Flag	II, III
Flußdiagramm	VI-1
Haupt-ROM	A-1
Hexadezimalzahlen	I
Hexcode/Zeichen	V.2
INA	III.8-1
INB	III.8-1
INCA	III.1-4
INCB	III.1-4
INCI	III.1-4
INCJ	III.1-4
INCK	III.1-5
INCL	III.1-5
INCM	III.9-3
INCN	III.9-3
INCP	III.1-4
Inkrement-Befehle	II-4, III-1, III.1-4
Input-Register	II-4
Instructionregister	II-5
Internes Memory	II, III
Immediate	II-3, III
IX	III.1-5
IXL	III.1-5
IY	III.1-5

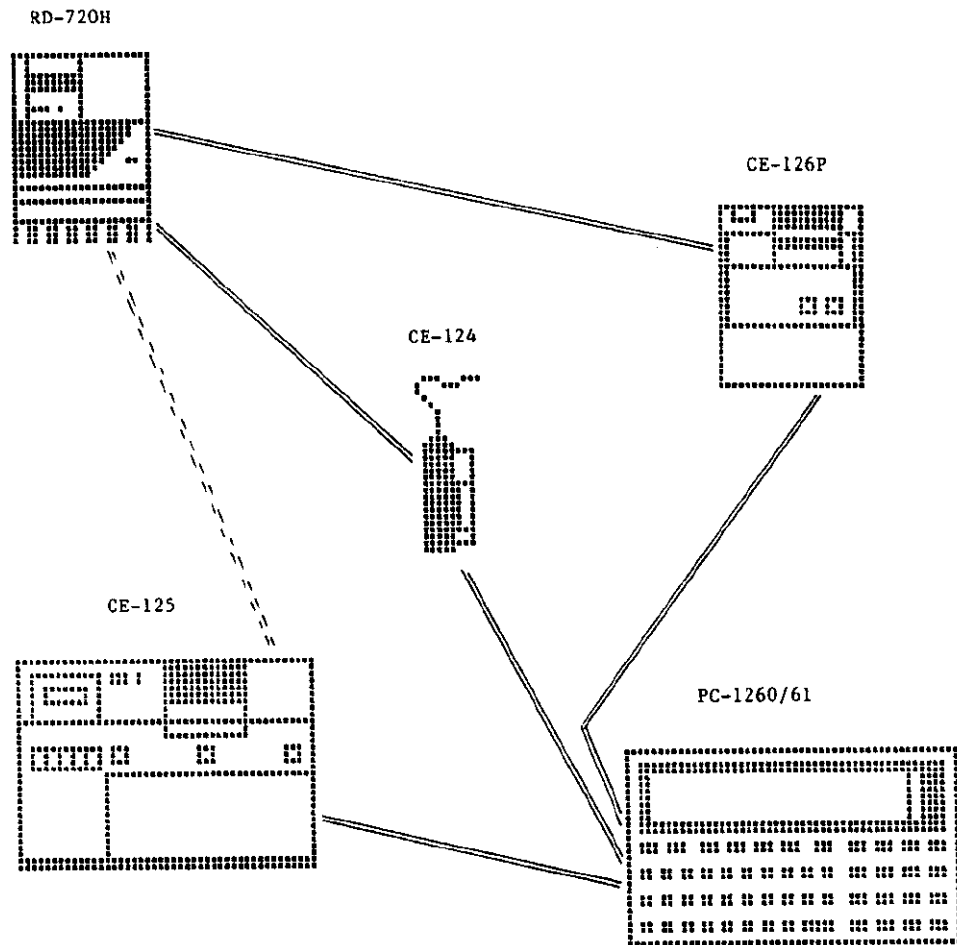
IYS	III.1-5
JP	III.6-1
JPC	III.6-1
JPNC	III.6-1
JPNZ	III.6-1
JPZ	III.6-1
JRCM	III.6-2
JRCP	III.6-1
JRNCM	III.6-2
JRNCP	III.6-1
JRNZM	III.6-2
JRNZP	III.6-1
JRM	III.6-2
JRP	III.6-1
JRZM	III.6-2
JRZP	III.6-1
Kern-ROM	II-5, VII, A-1
Kern-ROM-Ausdruck	A
Kurzbeschreibung der Befehle	III.10
LDD	III.1-2
LDM	III.1-2
LDP	III.1-1
LDQ	III.1-1
LDR	III.1-1
LEAVE	III.1-3
LIA	III.1-1
LIB	III.1-1
LIDL	III.1-1
LIDP	III.1-1
LII	III.1-1
LIJ	III.1-1
LIP	III.1-1
LIQ	III.1-1
Logische Befehle	III-1, III.3
LOOP	III.6-2
LP <i>e</i>	III.1-1

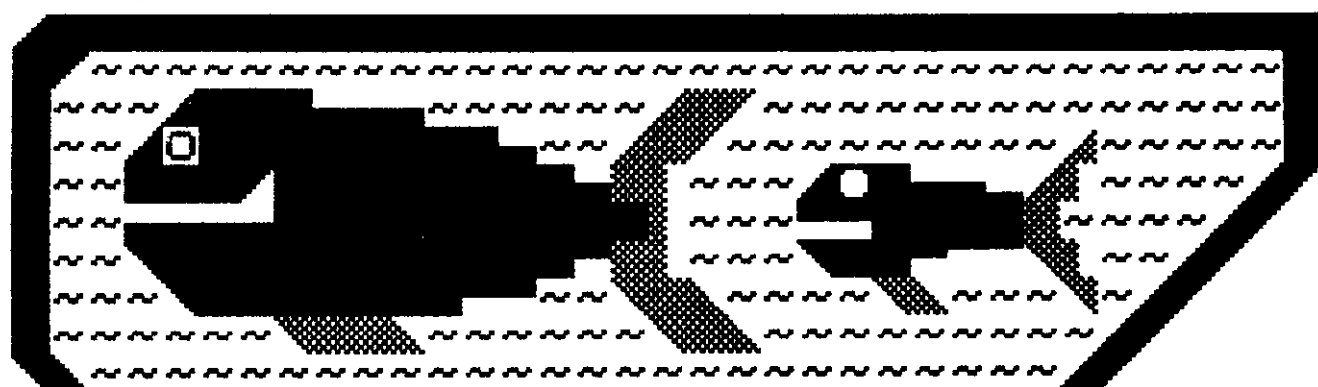
Mikroprozessor	E-1, II, III-1
Mnemonic	E-1
MVB	III.1-3
MVDM	III.1-2
MVMD	III.1-2
MVW	III.1-3
Neue Version	E-2
NOPT	III.5-1
NOPW	III.5-1
Oktalzahlen	I-1
ORIA	III.3-1
ORID	III.3-1
ORIM	III.3-1
ORMA	III.3-1
OUTA	III.8-1
OUTB	III.8-2
OUTC	III.8-2
OUTF	III.8-2
Output-Register	II-4
PEEK	E-1, IV-1
POKE	E-1, IV-1
POP	III.1-3
Ports	II
Programmcounter	II
PUSH	III.1-2
RAM	II-5, VII
RC	III.5-1
READ	III.9-1
READM	III.9-1
Register	II, III
Renumber	VI.1
Reserveakkumulator	II-3
Return-Befehl	III-2, III.7
ROM	II, VII, A-1
ROM-Beginn	A-1
ROM-Version	E-2

RTN	III.7-1
SBB	III.2-1
SBCM	III.2-1
SBIA	III.2-1
SBIM	III.2-1
SBM	III.2-1
SBN	III.2-3
SBW	III.2-4
SC	III.5-1
Shift-Befehle	III-1, III.4
Single Step	E-2
SL	III.4-1
SLW	III.2-5
Sprungbefehle	III-1, III.6
SR	III.4-1
SRW	III.2-5
Stack	II
Stackpointer	II, III
Standardvariable	V.1-1, VII.4-1
STD	III.1-2
STP	III.1-1
STQ	III.1-1
STR	III.1-1
SWP	III.4-1
System	VII
Systemadressen	E-2, VII.4
System Stack	II
Tastenroutinen	VII.2
Transport-Befehle	III-1, III.1
Token-tabelle	VII.4
TSIA	III.3-2
TSID	III.3-2
TSIM	III.3-2
Unbekannte Befehle	III-2, III.9
Unterprogramm-Befehle	III-1, III.7

Vergleichs-Befehle
 WAIT
 WRIT
 Zahlensysteme
 Zero-Flag
 Zweibyte-Befehle
 Zyklus

III-1, III.3
 III.5-1
 III.9-2
 I-1
 II, III
 III-2
 II-5, III.10





Alles fuer

SHARP-Computer

FISCHER

GmbH

Kaiser - Friedrich - Strasse 54a

1000 Berlin 12

durch

Information!

vorn