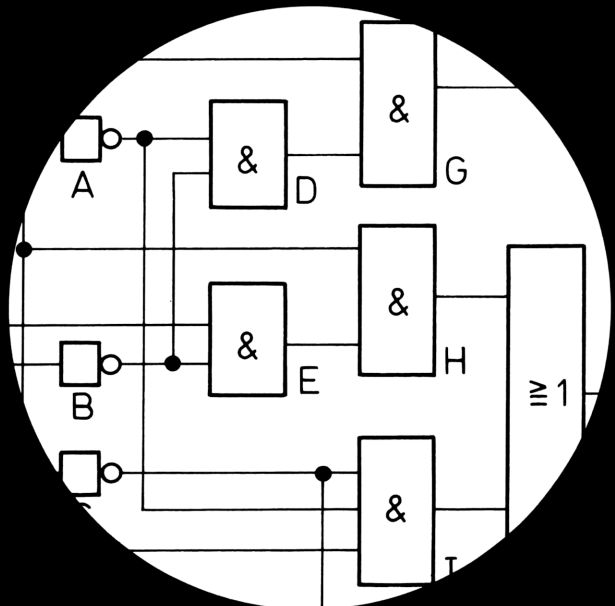


HP-41-Sammlung

**Strukturierte Programmierung,
Modell einer Datenverarbeitung,
Schaltalgebra, Logiknetzwerke,
Transistor-Verstärkerstufe,
Zahl π
Permutation, Transposition**



Vieweg Programmbibliothek
Mikrocomputer 23

HP-41-Sammlung

Vieweg Programmbibliothek

Mikrocomputer

Herausgegeben von Harald Schumny

Band 1
Graphik-Programme für TRS-80
und HP 9830

Band 2
Iterationen, Näherungsverfahren,
Sortiermethoden
BASIC-Programme für
CMB 3032, HP 9830, TRS-80,
Olivetti 6060

Band 3
BASIC und Pascal im Vergleich

Band 4
BASIC-Anwenderprogramme

Band 5
BASIC-Programme für den
PC-1211/1212

Band 6
Programme für den Einplatinen-
computer TM 990/189

Band 7
PC-1500-Sammlung I

Band 8
Programme für den PC-1251

Band 9
PC-1500-Sammlung II

Band 10
PC-1500-Sammlung III

Band 11
Anwenderprogramme zum ZX-81
und ZX-Spectrum

Band 12
17 Spiele für den PC-1500 A

Band 13
Ausgewählte BASIC-Computerspiele
(Atari 800)

Band 14
Lineares Optimieren
11 HP-41-Programme

Band 15
Dienstprogramme (Tool-Kit)
für den HP-41

Band 16
Geodätische Berechnungsmethoden
(Standard-BASIC)

Band 17
Gelenk-Getriebe für die
Handhabungs- und Robotertechnik
(HP-41 CV)

Band 18
Probleme der Festigkeitslehre
HP-41-Programme

Band 19
PC-1500-Sammlung IV

Band 20
Dienstprogramme (Tool Kit)
für den CBM 4032/8032

Band 21
HP-41 in der Praxis

Band 22
PC-1500-Sammlung V

Band 23
HP-41-Sammlung

Band 24
PC-1500-Sammlung VI

Band 25
Soziogramme

Vieweg Programmbibliothek
Mikrocomputer Band 23

Harald Schumny (Hrsg.)

HP-41-Sammlung

Strukturierte Programmierung
Modell einer Datenverarbeitung
Schaltalgebra, Logiknetzwerke
Transistor-Verstärkerstufe
Zahl π
Permutation, Transposition



Friedr. Vieweg & Sohn Braunschweig / Wiesbaden

CIP-Kurztitelaufnahme der Deutschen Bibliothek

[HP-einundvierzig-Sammlung]

HP-41-Sammlung: strukturierte Programmierung, Modell

e. Datenverarbeitung, Schaltalgebra, Logiknetzwerke,

Transistor-Verstärkerstufe, Zahl π , Permutation,

Transposition / Harald Schumny (Hrsg.). [Die Autoren

d. Bd. Eckard Gehrke . . .]. — Braunschweig;

Wiesbaden: Vieweg, 1985.

(Vieweg-Programmbibliothek Mikrocomputer; Bd. 23)

ISBN 3-528-04367-9

NE: Schumny, Harald [Hrsg.]; GT

Die Autoren des Bandes

Eckard Gehrke
Schlesierstraße 19
6704 Mutterstadt

Prof. Dr. Hartmut Ring
Am Kreuzgarten 3
5242 Kirchen-Freusberg

Karl Hackenberg
Kurt-Schumacher-Straße 12
3300 Braunschweig

Hans-Joachim Soyta
Brunecker Straße 66
8904 Friedberg

Herbert Hoffmann
Denkmalsweg 12
5900 Siegen 32

Das in dem Buch enthaltene Programm-Material ist mit keiner Verpflichtung oder Garantie irgendeiner Art verbunden. Der Autor übernimmt infolgedessen keine Verantwortung und wird keine daraus folgende oder sonstige Haftung übernehmen, die auf irgendeine Art aus der Benutzung dieses Programm-Materials oder Teilen davon entsteht.

1985

Alle Rechte vorbehalten

© Friedr. Vieweg & Sohn Verlagsgesellschaft mbH, Braunschweig 1985

Die Vervielfältigung und Übertragung einzelner Textabschnitte, Zeichnungen oder Bilder, auch für Zwecke der Unterrichtsgestaltung, gestattet das Urheberrecht nur, wenn sie mit dem Verlag vorher vereinbart wurden. Im Einzelfall muß über die Zahlung einer Gebühr für die Nutzung fremden geistigen Eigentums entschieden werden. Das gilt für die Vervielfältigung durch alle Verfahren einschließlich Speicherung und jede Übertragung auf Papier, Transparente, Filme, Bänder, Platten und andere Medien. Dieser Vermerk umfaßt nicht die in den §§ 53 und 54 URG ausdrücklich erwähnten Ausnahmen.

Umschlaggestaltung: Peter Lenz, Wiesbaden

Druck und buchbinderische Verarbeitung: W. Langelüdecke, Braunschweig

Printed in Germany

ISBN 3-528-04367-9

Inhaltsverzeichnis

Einführung	1
<i>Hartmut Ring:</i>	
Ein Programmsystem für die Strukturierte Programmierung auf dem HP-41CV	3
<i>Karl Hackenberg:</i>	
Modell einer Datenverarbeitung	16
<i>Herbert Hoffmann:</i>	
Schaltalgebra und Logiknetzwerke – mit dem HP-41 getestet – Teil I	28
Teil II	39
<i>Hans-Joachim Soyta:</i>	
Berechnung einer Transistor-Verstärkerstufe (Emitterschaltung)	49
<i>Eckard Gehrke:</i>	
Die Berechnung der Zahl Pi (π)	58
<i>Karl Hackenberg:</i>	
Überführung einer gegebenen Permutation aus n Zahlenelementen durch Transposition in eine geordnete Folge steigender Wertigkeit	81

Einführung

Es gab einmal eine Rechnereinteilung in der folgenden Weise:

- Taschenrechner haben den verfügbaren Befehlsvorrat den Tasten zugeordnet, d.h. eine Taste entspricht einer Anweisung;
- Computer werden in einer "Sprache" programmiert, Anweisungen werden aus einer Folge von Tastenbetätigungen gebildet.

Diese Unterscheidung macht schon lange keinen Sinn mehr. Der HP-41 wird zu Recht Taschencomputer (oder gar "System") genannt. Er kann mit Hilfe der Schnittstelle HP-IL (Interface Loop) mit leistungsfähigen Peripheriegeräten und auch mit Tischrechnern zusammenarbeiten. In der Vieweg Programmbibliothek/Mikrocomputer haben wir darum auch bereits mehrere HP-41-Bände publiziert (Lineares Optimieren, Band 14; Dienstprogramme, Band 15; Gelenkgetriebe für die Handhabungs- und Robotertechnik, Band 17; Festigkeitslehre, Band 18; HP-41 in der Praxis, Band 21).

Der hiermit vorliegende Band 23 präsentiert sechs Arbeiten, die teilweise umfangreiche Programmpakete enthalten und durchweg ausgezeichnet dokumentiert sind. Hartmut Ring stellt sein "HP-41-Pascal-Programmiersystem" vor, mit dessen Hilfe tatsächlich auf dem HP-41 strukturiert programmiert werden kann. Der Autor schreibt: "Dieses System nimmt dem Anwender den fehleranfälligsten Teil der Programmentwicklung ab, nämlich die Umsetzung von strukturierten Verzweigungsstrukturen in entsprechende PTR-Verzweigungsstrukturen." Am Beispiel des Quicksort-Algorithmus wird die Handhabung des Systems demonstriert.

"Modell einer Datenverarbeitung" nennt Karl Hackenberg seinen Beitrag, mit dem er sich vorgenommen hat, "ein pragmatisches Datenverarbeitungsprogramm zu kreieren, das - stellvertretend

für viele ähnlich geartete Fälle - die Vielfalt des Rechners HP-41C/CV erkennen läßt". Als spezielles Beispiel wurde eine Betriebskostenabrechnung gewählt.

Norbert Hoffmann wendet sich danach der Schaltalgebra zu. Sein sehr ausführlich dokumentiertes Programmpaket verlangt vom Anwender einige Konzentration. Es ist aber mit dieser Software möglich, Logiknetzwerke zu testen und zu optimieren, die bis zu 24 integrierte Gatterschaltungen (ICs) mit je 9 Eingängen und 4 Ausgängen enthalten können.

Ebenfalls dem Bereich der "Elektronik" ist das Programm "Berechnung einer Transistor-Verstärkerstufe" von Hans-Joachim Soyta zuzuordnen. Emitterschaltungen können hiermit berechnet werden. Die verwendeten Formeln und Grundeinstellungen sind angegeben. Ergebnisse werden sauber lesbar ausgedruckt.

Es folgt ein Beitrag von Eckard Gehrke, der die Berechnung der Zahlen "Pi" und "e" zum Inhalt hat. Für Pi werden 800 Dezimale berechnet. Der Schwerpunkt der Arbeit liegt auf der Entwicklung des Algorithmus. Dabei werden die Prinzipien der strukturierten Programmierung beachtet (z.B. Modularisierung des Problems). Das Programm und die Bedienung sind beschrieben, Ergebnisse werden diskutiert und auf die e-Berechnung übertragen.

Das letzte Programm dieses Bandes wurde von Karl Hackenberg entwickelt. Er verwendet Werkzeuge der Kombinatorik, um aus beispielsweise Meßreihen geordnete Folgen zu erhalten. Ökonomen, Ingenieure und Naturwissenschaftler werden diese Software mit Gewinn anwenden können.

Die allgemein recht anspruchsvollen Programme belegen erneut die Beliebtheit und Leistungsfähigkeit des Taschencomputers HP-41. Eine Vielzahl von Einzelarbeiten und Programmpaketen liegt uns noch vor. Weitere HP-41-Bände innerhalb der Vieweg Programmbibliothek werden darum folgen.

Ein Programmsystem für die Strukturierte Programmierung auf dem HP-41 CV

von Hartmut Ring

Die Strukturierte Programmierung hat sich längst als Weg zu gut lesbaren und korrekten Programmen bewährt. Da aber der Stil der Programmierung weitgehend vom Konzept der verwendeten Sprache beeinflusst wird, ziehen meist nur Benutzer von modernen höheren Programmiersprachen Gewinn aus den Vorteilen des neuen Programmierstils.

Bei den heute verwendeten programmierbaren Taschenrechnern zeigt sich in dieser Hinsicht das krasse Gegenteil des Anzustrebenden: Die Möglichkeiten, die der PTR zur Strukturierung von Daten und Anweisungen bietet, sind derart beschränkt, daß es schon höchster Selbstdisziplin bedarf, dabei nicht in einen archaischen Spaghettistil zu verfallen.

Ein vollständiger Compiler für eine moderne Programmiersprache, wie etwa Pascal oder Ada, brächte keinen Ausweg aus dieser Situation: Erstens sind solche Compiler so umfangreich, daß sie nicht auf heutigen Taschenrechnern laufen können, und zweitens würde der erzeugte Programmcode zu verschwenderisch mit dem beschränkten Speicherplatz des PTR umgehen. Deshalb hat sich der Autor die Aufgabe gestellt, Komponenten eines Compilers zu implementieren, die im PTR selbst Platz finden und dennoch folgenden Anforderungen genügen:

- Es sollen alle wichtigen Programmstrukturierungsmöglichkeiten in praktisch beliebiger Verschachtelungstiefe verfügbar sein.
- Das vom Compiler erzeugte PTR-Programm soll so effektiv wie ein guter Programmierer den Speicherplatz des PTR ausnutzen.

Das Ergebnis dieser Bemühungen ist das "HP-41C-Pascal-Programmiersystem". Dieses System nimmt dem Anwender den fehleranfälligsten Teil der Programmentwicklung ab, nämlich die Umsetzung von strukturierten Verzweigungsstrukturen in entsprechende PTR-Verzweigungsstrukturen. Das System hat sich als echte Hilfe bei der Entwicklung und Dokumentation von strukturierten Benutzerprogrammen bewährt.

1 ÜBERBLICK ÜBER DAS HP-41C-PASCAL-PROGRAMMIERSYSTEM

1.1 Syntax

Das Syntaxdiagramm (Bild 1) zeigt den Sprachumfang des Programmiersystems.

Dabei werden die Zeichen A1, A2, A3,... als Symbole für Elementaranweisungen verwendet, d.h. Anweisungen, die stets linearen Programmstücken entsprechen. Entsprechend wird B1, B2, B3,... als Abkürzung für Bedingungen geschrieben.

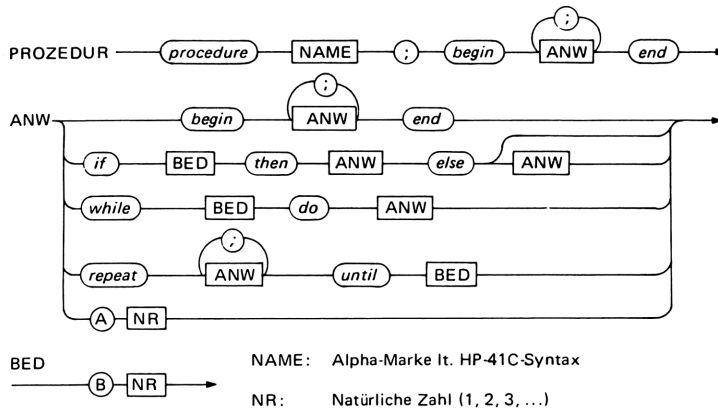
1.2 Eingabe von Pascal-Programmen

Pascal-Programme werden im User-Modus mit Hilfe der obersten 9 Tasten eingegeben. Die Tasten werden bezeichnet mit:

<i>if</i>	<i>repeat</i>	<i>while</i>	<i>begin</i>	ANW
<i>else</i>	<i>until</i>	;	<i>end</i>	

Einige Tasten erzeugen mehr als ihre Beschriftung zeigt:

<i>if</i>	erzeugt: IF B <NR> THEN
<i>while</i>	erzeugt: WHILE B <NR> DO
<i>until</i>	erzeugt: UNTIL B <NR>

Bild 1

Dabei durchläuft <NR> jeweils für Bedingungen und Anweisungen automatisch die Zahlen 1, 2, 3,.... Selbst größere Programme können also mit sehr wenigen Tastendrücken eingegeben werden.

1.3 Behandlung von Syntax-Fehlern

Syntax-Fehler werden bei der Eingabe sofort erkannt und durch eine Fehlermeldung (Piepton und Ausgabe einer Fehlernummer) angezeigt. Sie können sofort korrigiert werden.

1.4 Kontrollausdruck

Das eingegebene Pascal-Programm wird auf dem Drucker automatisch mit strukturierter Einrückung ausgegeben und dient zur Programmdokumentation.

1.5 Übersetzung

Anschließend übersetzt das System das eingegebene Pascal-Programm automatisch in eine entsprechende optimierte Folge von LBL- und GTO-Zeichen sowie Anweisungs- und Bedienungssymbolen. Um ein lauffähiges Programm zu erhalten, codiert nun der Benutzer noch die linearen Elementaranweisungen und die Bedingungen.

2 EIN BEISPIEL ZUM PRAKTISCHEN EINSATZ DES SYSTEMS

Der Weg von der Beschreibung eines Algorithmus bis zum fertigen PTR-Programm soll hier exemplarisch an einem bekannten Sortierverfahren beschrieben werden, das wegen seiner hohen Leistung den Namen "Quicksort" trägt (vgl. hierzu z.B. [1]).

2.1 Beschreibung des Quicksort-Algorithmus

Unser Ziel ist es, die Werte eines zusammenhängenden Registerbereichs so umzuordnen, daß sie in aufsteigender Reihenfolge angeordnet sind. Wir wollen die zu sortierenden Register zunächst mit $Z[1]$, $Z[2]$, ..., $Z[n]$ bezeichnen. Nun schreiben wir auf einen Zettel den Auftrag "Sortierbereich: $Z[1]$ bis $Z[n]$ " und legen ihn in einen (leeren) Zettelkasten. Jetzt beginnt die Erledigung des Sortierauftrags:

Wähle ein beliebiges Register aus dem Sortierbereich (z.B. das mittlere) und verwende seinen Inhalt als Vergleichswert. Suche nun von links anfangend die erste Zahl, die größer als der Vergleichswert ist und von rechts anfangend die erste Zahl, die kleiner als der Vergleichswert ist. Vertausche diese beiden Zahlen und wiederhole dieses Verfahren, bis sich die beiden Suchwege treffen.

In dem so entstandenen linken Teilbereich sind nun alle Zahlen kleiner als der Vergleichswert, im rechten Bereich größer (oder gleich). Damit ist das Sortierproblem zurückgeführt auf die Sortierung der beiden Teilbereiche. Wir schreiben also zwei neue Sortieraufträge und legen sie an Stelle des alten Auftrags in den Zettelkasten. (Falls ein entstandener Bereich nur aus einer Zahl besteht, brauchen wir natürlich keinen Sortierauftrag mehr!) Jetzt wird der nächste Sortierauftrag aus dem Zettelkasten erledigt. Dies wird wiederholt, bis der Kasten leer ist.

2.2 Umsetzung in eine Pascal-Prozedur

Der "Zettelkasten" wird in der Datenverarbeitung als Keller bezeichnet. Für seine Darstellung benötigen wir zunächst eine

Variable k , die angibt, wieviele "Zettel im Kasten liegen". Auf den Zetteln steht jeweils die linke und rechte Grenze des Sortierbereichs. Deshalb führen wir zwei Vektoren $VON[1..m]$ und $BIS[1..m]$ ein. Der Index 1 zeigt auf den untersten "Zettel", k auf den obersten. Bei dem nun folgenden Algorithmus wird der größere Teilbereich gekellert und der kleinere gleich weiterverarbeitet. Es läßt sich zeigen, daß damit die Kellertiefe m für das Sortieren von mindestens $3 \cdot 2^m - 1$ Zahlen genügt. Da beim HP-41C maximal 319 Register zur Verfügung stehen, reicht also $m = 7$ aus.

```

procedure qsort;
  {n und Z[1..n] werden als globale Variablen aufgefaßt!}
  var i, j, k, von, bis: integer;
      vgl: real;
      VON, BIS: array [1..7] of integer;
A1, A2, A3   begin k := 1; VON[1] := 1; BIS[1] := n;
A4, A5, A6   repeat von := VON[k]; bis := BIS[k]; k := k - 1;
A7, A8, A9   repeat i := von; j := bis; vgl := Z[(von + bis) div 2];
              repeat
B1, A10       while Z[i] < vgl do i := i + 1;
B2, A11       while Z[j] > vgl do j := j - 1;
B3            if i ≤ j then begin
A12            vertausche Z[i] mit Z[j];
A13, A14       i := i + 1; j := j - 1 end
B4            until i > j;
B5            if j - von < bis - i then begin
B6            if i < bis then begin
A15, A16, A17   k := k + 1; VON[k] := i; BIS[k] := bis end;
A18           bis := j end
              else begin
B7            if von < j then begin
A19, A20, A21   k := k + 1; VON[k] := von; BIS[k] := j end;
A22           von := i end
B8            until von ≥ bis
B9            until k = 0
              end
end

```

2.3 Übersetzung der Pascal-Prozedur

Diese Prozedur wird nun mit Hilfe des HP-41C-Pascal-Programmierersystems in den Taschenrechner eingetragen und automatisch übersetzt. Bild 2 zeigt das Resultat dieses Schrittes.

(Die logische Negation wird im Zielprogramm mit dem Zeichen \neq dargestellt).

=====	*LBL *QSORT*
* HP-41C PASCAL *	A1
* PROGRAMMIERSYSTEM *	A2
=====	A3
PASCAL-QUELLPROGRAMM:	*LBL 15
	A4
PROCEDURE QSORT;	A5
BEGIN	A6
A1;	*LBL 16
A2;	A7
A3;	A8
REPEAT	A9
A4;	*LBL 17
A5;	*B1?
A6;	GTO 18
REPEAT	A10
A7;	GTO 17
A8;	*LBL 18
A9;	*B2?
REPEAT	GTO 19
WHILE B1 DO	A11
A10;	GTO 18
WHILE B2 DO	*LBL 19
A11;	*B3?
IF B3 THEN	GTO 20
BEGIN	A12
A12;	A13
A13;	A14
A14	*LBL 20
END	*B4?
UNTIL B4;	GTO 17
IF B5 THEN	*B5?
BEGIN	GTO 21
IF B6 THEN	*B6?
BEGIN	GTO 23
A15;	A15
A16;	A16
A17	A17
END;	*LBL 23
A18	A18
END	GTO 22
ELSE	*LBL 21
BEGIN	*B7?
IF B7 THEN	GTO 24
BEGIN	A19
A19;	A20
A20;	A21
A21	*LBL 24
END;	A22
A22	*LBL 22
END	*B8?
UNTIL B8	GTO 16
UNTIL B9	*B9?
END	GTO 15
=====	END
HP-41C-ZIELPROGRAMM:	=====

2.4 Umsetzung der Datenstrukturen

Da die Indizes der Vektor-(`array`-)Typen nicht immer mit den Registernummern des PTR übereinstimmen können, wählen wir für jeden Vektor ein eigenes Indexregister. In unserem Beispiel benötigen wir statt der Variablen `k` zwei Indexregister für `VON` und `BIS`.

Wir brauchen also: für `n`, `i`, `j`, `von`, `bis`, `vgl` je ein Register; für `k` zwei Register. Das sind zusammen 8 Register. Wenn wir hierfür `R00 .. R07` wählen, können wir `VON [1 .. 7]` nach `R08 .. R14` und `BIS [1 .. 7]` nach `R15 .. R21` legen. Ab `R22` kann dann der Sortierbereich `Z [1 .. n]` liegen. Wir treffen also folgende Registerzuordnung:

Register	00	01	02	03	04
Inhalt	$n + 21$	$k + 7$	$k + 14$	$i + 21$	$j + 21$

Register	05	06	07	08	..
Inhalt	$von + 21$	$bis + 21$	vgl	$VON[1] + 21$..

Register	14	15	..	21
Inhalt	$VON[7] + 21$	$BIS[1] + 21$..	$BIS[7] + 21$

Register	22	..	$n + 21$
Inhalt	$Z[1]$..	$Z[n]$

2.5 Codierung der Elementaranweisungen

Dieser Schritt ist nur noch Routinearbeit. Wir müssen lediglich darauf achten, daß jede Veränderung von `k` die entsprechende Veränderung von `R01` und `R02` zur Folge hat. Beispielsweise wird `A1` zu:

```

8      }
STO 01 } (aus k = 1 folgt: k + 7 = 8)
15     }
STO 02 } (aus k = 1 folgt: k + 14 = 15)
```

Das vollständige HP-41C-Programm zeigt `B i l d 3`.

A1	{	01*LBL "QSORT"	-B2?	{	36 RCL 07	A15	{	70 1
		02 8			37 RCL IND 04			71 ST+ 01
A2	{	03 STO 01	A11	{	38 X<=Y?	A16	{	72 ST+ 02
		04 15			39 GTO 19			73 RCL 03
A3	{	05 STO 02	-B3?	{	40 DSE 04	A17	{	74 STO IND 01
		06 22			41 GTO 18			75 RCL 06
A4	{	07 STO 08	A12	{	42*LBL 19	A18	{	76 STO IND 02
		08 RCL 00			43 RCL 04			77*LBL 23
A5	{	09 STO 15	-B4?	{	44 RCL 03	A19	{	78 RCL 04
		10*LBL 15			45 X>Y?			79 STO 06
A6	{	11 RCL IND 01	A13	{	46 GTO 20	-B7?	{	80 GTO 22
		12 STO 05			47 RCL IND 03			81*LBL 21
A7	{	13 RCL IND 02	A14	{	48 X<> IND 04	A20	{	82 RCL 05
		14 STO 06			49 STO IND 03			83 RCL 04
A8	{	15 DSE 01	-B5?	{	50 1	A21	{	84 X<=Y?
		16 DSE 02			51 ST+ 03			85 GTO 24
A9	{	17*LBL 16	A15	{	52 ST- 04	A22	{	86 1
		18 RCL 05			53*LBL 20			87 ST+ 01
A10	{	19 STO 03	-B6?	{	54 RCL 04	-B8?	{	88 ST+ 02
		20 RCL 06			55 RCL 03			89 RCL 05
A11	{	21 STO 04	A16	{	56 X<=Y?	A23	{	90 STO IND 01
		22 +			57 GTO 17			91 RCL 04
A12	{	23 2	-B7?	{	58 RCL 04	A24	{	92 STO IND 02
		24 /			59 RCL 05			93*LBL 24
A13	{	25 RCL IND X	A17	{	60 -	A25	{	94 RCL 03
		26 STO 07			61 RCL 06			95 STO 05
A14	{	27*LBL 17	-B8?	{	62 RCL 03	A26	{	96*LBL 22
		28 RCL IND 03			63 -			97 RCL 05
A15	{	29 RCL 07	A18	{	64 X<=Y?	A27	{	98 RCL 06
		30 X<=Y?			65 GTO 21			99 X>Y?
A16	{	31 GTO 18	-B9?	{	66 RCL 03	A28	{	100 GTO 16
		32 1			67 RCL 06			101 RCL 01
A17	{	33 ST+ 03	A19	{	68 X<=Y?	A29	{	102 8
		34 GTO 17			69 GTO 23			103 X<=Y?
A18	{	35*LBL 18	A20	{		A30	{	104 GTO 15
								105 END

Bild 3

3 BEDIENUNG DES HP-41C-PASCAL-PROGRAMMIERSYSTEMS

Das System besteht aus den Programmen SNTX (Syntaxanalyse, 713 Bytes) und SMTK (Semantik-Routinen, 857 Bytes). Es wird ein HP-41CV (oder HP-41C mit Quad Memory) mit einem X-Functions-Modul und ein Thermodrucker benötigt. Der Datenspeicherbereich ist mit SIZE 055 einzustellen. Die ersten 9 Tasten dürfen nicht mit User-Funktionen belegt sein.

Das System wird mit XEQ "SNTX" gestartet. Es meldet sich mit der Anzeige "PROGR.-NAME?". Hierauf ist der Programmname einzugeben und R/S zu drücken. Nun wird das Pascal-Programm mit Hilfe der reservierten Tasten eingegeben. Dabei ist nach jedem Tastendruck ein Piepton abzuwarten und die Anzeige zu beachten. Bei Syntaxfehlern erscheint eine Meldung "FEHLER mn". Die Bedeutung der Ziffern m und n zeigt folgende Tabelle:

m	erlaubte Eingabemöglichkeiten				
1	<i>begin</i>	<i>if</i>	<i>while</i>	<i>repeat</i>	ANW
2	<i>;</i>	<i>end</i>			
6	<i>;</i>	<i>until</i>			

n	0	1	2	3	4	5	6	7	8
falsche Eingabe:	<i>if</i>	<i>repeat</i>	<i>while</i>	<i>begin</i>	ANW	<i>else</i>	<i>until</i>	<i>;</i>	<i>end</i>

Der Fehler wird durch Eintasten der richtigen Anweisung korrigiert (einzige Ausnahme: wurde statt else versehentlich eine syntaktisch falsche Anweisung eingegeben, so wird die if-Anweisung ohne else-Teil abgeschlossen).

Nach Beendigung der Programmeingabe beginnt die Übersetzung, die bei größeren Programmen längere Zeit dauern kann (beim Beispielprogramm Quicksort etwa 10 Minuten). Zur Schonung der Batterien kann der Drucker nach dem Druck der ersten Zeile des Zielprogramms ausgeschaltet werden. Der Rechner meldet sich dann wieder mit einer Aufforderung zum Einschalten des Druckers. Dann wird das übersetzte Programm ausgedruckt.

4 ANWEISUNGSLISTEN

01 LBL "SNTX"	11 STO 06	21 FC?C 25	31 PRA
02 11	12 OCT	22 CRFLAS	32 XEQ "Z"
03 STO 00	13 STO 11	23 APPREC	33 "PASCAL-"
04 1	14 FIX 0	24 APPREC	34 "QUELLPROGRAMM:"
05 STO 02	15 CF 29	25 XEQ "Z"	35 PRA
06 STO 03	16 SF 27	26 "*" HP-41C "	36 ADV
07 2	17 "C"	27 "HPASCAL"	37 AON
08 STO 04	18 124	28 PRA	38 "PROGR.-NAME?"
09 30	19 SF 25	29 "*" PROGRAMMIER"	39 TONE 7
10 STO 01	20 CLFL	30 "SYSTEM"	40 PROMPT

41 AOFF	92 10	143*LBL 19	194*LBL 00
42 ASTO 10	93 *	144 RCL Z	195 *IF B*
43 *PROCEDURE *	94 RCL 05	145 RCL IND 01	196 ARCL 03
44 ARCL 10	95 +	146 XTOA	197 *IF THEN*
45 59	96 ASTO 08	147 1	198 XEQ 16
46 XTOA	97 ASHF	148 ST+ 01	199 RCL 03
47 ACA	98 ASTO 09	149 DSE Z	200 STO IND 01
48 XEQ 03	99 *FEHLER *	150 GTO 15	201 1
49*LBL 20	100 ARCL X	151 RCL 07	202 ST+ 00
50 TONE 7	101 CF 21	152 ST- 01	203 ST+ 01
51 STOP	102 AVIEW	153 RCL 04	204 ST+ 03
52*LBL *Z*	103 SF 21	154 200	205 32
53 *-----*	104 CLA	155 +	206 STO IND 00
54 ACA	105 ARCL 08	156 STO IND 01	207 RTN
55 ACA	106 ARCL 09	157 1	208*LBL 01
56 ACA	107 BEEP	158 ST+ 01	209 *REPEAT*
57 PRBUF	108 STOP	159 ST+ 04	210 XEQ 16
58 RTN	109*LBL 16	160 APPREC	211 1
59*LBL A	110 PRBUF	161 CLA	212 ST+ 00
60 0	111 RCL 00	162 ARCL 08	213 35.001
61 GTO 00	112 11	163 ARCL 09	214 STO IND 00
62*LBL B	113 -	164*LBL 18	215 RTN
63 1	114 SKPCHR	165 RCL IND 00	216*LBL 02
64 GTO 00	115 ACA	166 STO 06	217 *WHILE B*
65*LBL C	116 CF 21	167 1	218 ARCL 03
66 2	117 AVIEW	168 ST- 00	219 *IF DO*
67 GTO 00	118 SF 21	169 RTN	220 XEQ 16
68*LBL D	119 RTN	170*LBL 29	221 RCL 03
69 3	120*LBL 17	171 59	222 STO IND 01
70 GTO 00	121 1	172 ACCHR	223 1
71*LBL E	122 RCL 07	173 XTOA	224 ST+ 00
72 4	123 XYY?	174 CF 21	225 ST+ 01
73 GTO 00	124 GTO 00	175 AVIEW	226 ST+ 03
74*LBL F	125 RCL 06	176 SF 21	227 34
75 5	126 FRC	177 1	228 STO IND 00
76 GTO 00	127 1 E3	178 ST+ 00	229 RTN
77*LBL G	128 *	179 .1	230*LBL 03
78 6	129 +	180 %	231 *BEGIN*
79 GTO 00	130 STO 07	181 RCL 06	232 XEQ 16
80*LBL H	131*LBL 00	182 +	233 1
81 7	132 ST- 01	183 STO IND 00	234 ST+ 00
82 GTO 00	133 ASTO 08	184 30	235 31.001
83*LBL I	134 ASHF	185 STO 06	236 STO IND 00
84 8	135 ASTO 09	186 RTN	237 RTN
85*LBL 06	136 CLA	187*LBL 30	238*LBL 04
86 STO 05	137 RCL 06	188 4	239 *A*
87 GTO IND 06	138 30	189 RCL 05	240 ARCL 02
88*LBL 15	139 -	190 XYY?	241 XEQ 16
89 RCL 05	140 XTOA	191 GTO 15	242 RCL 02
90 29	141 RCL 07	192 XEQ IND X	243 STO IND 01
91 -	142 XTOA	193 GTO 20	244 XEQ 18

245 ST+ 01	296 2	21 RCLPT	72 DELCHR
246 ST+ 02	297 STO 07	22 INT	73 SEEKPT
247 RTN	298 XEQ 17	23 200	74 DELCHR
248+LBL 31	299 GTO IND 06	24 +	75 2
249 8	300+LBL 35	25 STO 12	76 SEEKPT
250 RCL 05	301 6	26 15	77 SF 25
251 X=Y?	302 RCL 05	27 STO 01	78+LBL 17
252 GTO 00	303 X=Y?	28+LBL 20	79 DELREC
253 7	304 GTO 00	29 -200	80 FS? 25
254 X*Y?	305 7	30 RCL IND 00	81 GTO 17
255 GTO 15	306 X*Y?	31 DSE 00	82 "LL"
256 XEQ 29	307 GTO 15	32 X=0?	83 ASTO 08
257 GTO 20	308 XEQ 29	33 GTO 16	84 XEQ 40
258+LBL 00	309 GTO 20	34 +	85 "LG"
259 "END"	310+LBL 00	35 X<0?	86 ASTO 08
260 XEQ 16	311 "UNTIL B"	36 GTO 00	87 XEQ 40
261 CLX	312 ARCL 03	37 STO 04	88+LBL 41
262 STO 07	313 XEQ 16	38 SEEKPT	89 1
263 XEQ 17	314 RCL 03	39 GETREC	90 SEEKPT
264 10	315 STO IND 01	40 ATOX	91 "GG"
265 RCL 00	316 1	41 STO 05	92 POSFL
266 X>Y?	317 ST+ 01	42 ATOX	93 X<0?
267 GTO 20	318 ST+ 03	43 STO 06	94 GTO 01
268 PRBUF	319 STO 07	44 ATOX	95 1 E-3
269 XEQ "Z"	320 XEQ 17	45 STO 07	96 +
270 GTO "SMTK"	321 GTO 20	46 ATOX	97 SEEKPT
271+LBL 32	322 END	47 STO 08	98 1
272 RCL 05		48 ATOX	99 DELCHR
273 5		49 STO 09	100 -
274 X*Y?		50 RCL 05	101 SEEKPT
275 GTO 00		51 30	102 1
276 "ELSE"	01+LBL "SMTK"	52 +	103 DELCHR
277 XEQ 16	02 "HP-41C-ZIEL"	53 GTO IND X	104 GTO 41
278 1	03 "PROGRAMM:"	54+LBL 00	105+LBL 40
279 ST+ 00	04 PRA	55 "A"	106 1
280 33	05 ADV	56 1	107 SEEKPT
281 STO IND 00	06 CLA	57 SEEKPT	108 CLA
282 30	07 XEQ 02	58 APPCHR	109 ARCL 08
283 STO 06	08 34	59 CLA	110 POSFL
284 GTO 20	09 XTOA	60 RCL L	111 X<0?
285+LBL 00	10 ARCL 10	61 RCL 10	112 RTN
286 2	11 XTOA	62 +	113 STO 00
287 STO 07	12 ACA	63 XTOA	114 1
288 XEQ 17	13 PRBUF	64 CLX	115 DELCHR
289 GTO IND 06	14 64	65 SEEKPT	116 -
290+LBL 33	15 STO 10	66 APPCHR	117 SEEKPT
291 3	16 CF 26	67 GTO 20	118 GETREC
292 STO 07	17 12	68+LBL 16	119 ATOX
293 XEQ 17	18 STO 00	69 CLX	120 RCL 10
294 GTO IND 06	19 CLX	70 SEEKPT	121 MOD
295+LBL 34	20 STO 11	71 1	122 STO 01

123 ATOX	174 RCL 10	225 ARCL 02	276 FC?C 05
124 RCL 10	175 *	226 ACA	277 SF 05
125 MOD	176 RCL 07	227 *-	278 SF 06
126 STO 02	177 +	228 FS?C 05	279*LBL 24
127 RCL 00	178 XTOA	229 ACA	280 FC?C 05
128 1	179 DSE 04	230 PRBUF	281 SF 05
129 -	180 GTO 29	231 CF 12	282*LBL 25
130 SEEKPT	181 FS?C 01	232 ISG 00	283 *-L
131 LASTX	182 FS? 17	233 CLX	284 FC? 06
132 DELCHR	183 SF 02	234 GTO 18	285 *-G
133 CLX	184 FS? 02	235*LBL 00	286 !
134 STO 03	185 INSCR	236 7	287 SEEKPT
135 SEEKPT	186 FC?C 02	237 SKPCOL	288 APPCHR
136 SF 01	187 INSREC	238 SF 12	289 CLA
137*LBL 28	188 RCLPT	239 *-B	290 RCL 02
138 GETREC	189 STO 03	240 SF 05	291 FC?C 05
139 RCL 03	190 FS?C 17	241 RTN	292 RCL 03
140 SEEKPT	191 GTO 28	242*LBL 01	293 RCL 10
141 ALENG	192 GTO 40	243 7	294 ENTER↑
142 STO 04	193*LBL 01	244 SKPCOL	295 ENTER↑
143 DELCHR	194 SF 26	245 SF 12	296 FS?C 06
144*LBL 29	195 CLA	246 *-A	297 CLX
145 ATOX	196 SF 25	247 RTN	298 +
146 STO 05	197 ACA	248*LBL 02	299 +
147 RCL 10	198 FS?C 25	249 10	300 +
148 MOD	199 GTO 00	250 ACCHR	301 XTOA
149 STO 07	200 BEEP	251 *-LBL	302 CLX
150 2	201 *-DRUCKER-	252 RTN	303 SEEKPT
151 RCL 05	202 PROMPT	253*LBL 03	304 APPCHR
152 RCL 10	203*LBL 00	254 *-GTO	305 RTN
153 /	204 SF 25	255 RTN	306*LBL 26
154 INT	205 CLX	256*LBL 04	307 RCL 04
155 STO 06	206 STO 00	257 *-END-	308 SEEKPT
156 X<Y?	207*LBL 18	258 PRA	309 DELREC
157 GTO 00	208 RCL 00	259 XEQ *-Z-	310 INSREC
158 RCL 06	209 .1	260 RTN	311 ISG 00
159 3	210 %	261*LBL 21	312 CLX
160 X>Y?	211 SEEKPT	262 *-b-	313 ISG 00
161 GTO 03	212 FC? 25	263 1	314 CLX
162 RCL 01	213 GTO 04	264 SEEKPT	315 RTN
163 RCL 07	214 GETREC	265 APPCHR	316*LBL 27
164 X=Y?	215 ATOX	266 CLA	317 XTOA
165 RCL 02	216 STO 02	267 RCL 07	318 RCL 06
166 STO 07	217 RCL 10	268 XTOA	319 XTOA
167*LBL 03	218 /	269 CLX	320 RCL 04
168 RCL 01	219 STO 01	270 SEEKPT	321 SEEKPT
169 RCL 07	220 RCL 02	271 APPCHR	322 3
170 X>Y?	221 LASTX	272 RTN	323 DELCHR
171 DSE 07	222 MOD	273*LBL 22	324 INSCR
172 RCL 06	223 STO 02	274 SF 05	325 RTN
173*LBL 00	224 XEQ IND 01	275*LBL 23	326*LBL 31

327 DSE 06	369 RCL 01	411 GTO 20	453 DSE 06
328 ISG 00	370 STO 02	412*LBL 01	454 ISG 00
329 CLX	371 ISG 01	413 RCL 07	455 CLX
330 ISG 00	372 CLX	414 STO 03	456 ISG 00
331 CLX	373 RCL 01	415 XEQ 23	457 CLX
332 RCL 07	374 STO 03	416 GTO 20	458 RCL 07
333 STO IND 00	375 ISG 01	417*LBL 34	459 STO IND 00
334 RCL 06	376 CLX	418 GTO IND 06	460 RCL 01
335 X=0?	377 XEQ 21	419*LBL 02	461 STO 02
336 GTO 20	378 XEQ 24	420 RCL 01	462 ISG 01
337 CLA	379 CLA	421 STO 02	463 CLX
338 1	380 3	422 ISG 01	464 XEQ 22
339 XEQ 27	381 XTOA	423 CLX	465 CLA
340 GTO 20	382 2	424 RCL 01	466 6
341*LBL 32	383 XTOA	425 STO 03	467 XEQ 27
342 GTO IND 06	384 RCL 03	426 ISG 01	468 CLA
343*LBL 02	385 XTOA	427 CLX	469 RCL 02
344 RCL 01	386 RCL 02	428 XEQ 22	470 XTOA
345 STO 02	387 XTOA	429 XEQ 21	471 APPCHR
346 ISG 01	388 RCL 09	430 XEQ 25	472 GTO 20
347 CLX	389 XTOA	431 CLA	473*LBL 36
348 XEQ 21	390 XEQ 26	432 4	474 DSE 06
349 XEQ 24	391 RCL 08	433 XTOA	475 GTO 01
350 CLA	392 STO IND 00	434 1	476 XEQ 21
351 2	393 GTO 20	435 XTOA	477 RCL 08
352 XTOA	394*LBL 02	436 RCL 02	478 STO 02
353 1	395 RCL 07	437 XTOA	479 XEQ 24
354 XTOA	396 STO 03	438 RCL 07	480 GTO 20
355 RCL 02	397 RCL 08	439 XTOA	481*LBL 01
356 XTOA	398 STO 02	440 XEQ 26	482 ISG 00
357 XEQ 26	399 XEQ 25	441 RCL 08	483 CLX
358 RCL 08	400 XEQ 22	442 STO IND 00	484 ISG 00
359 STO IND 00	401 CLA	443 GTO 20	485 CLX
360 GTO 20	402 3	444*LBL 01	486 RCL 07
361*LBL 01	403 XTOA	445 RCL 07	487 STO IND 00
362 RCL 07	404 1	446 STO 02	488 CLA
363 STO 02	405 XTOA	447 RCL 08	489 6
364 XEQ 22	406 RCL 07	448 STO 03	490 XEQ 27
365 GTO 20	407 XTOA	449 XEQ 24	491 GTO 20
366*LBL 33	408 XEQ 26	450 XEQ 23	492 END
367 GTO IND 06	409 RCL 09	451 GTO 20	
368*LBL 03	410 STO IND 00	452*LBL 35	

Literatur

- [1] W i r t h, N.: Algorithmen und Datenstrukturen.
Stuttgart: Teubner 1975, 1979

Modell einer Datenverarbeitung

von Karl Hackenberg

In den unterschiedlichsten Bereichen unserer zivilisierten Umwelt ergeben sich laufend eine Fülle alpha-numerischer Daten, deren Auswertung und sinngemäße Verknüpfung miteinander zu wesentlichen Erkenntnissen führen können. Somit erscheint es angebracht, einmal ein pragmatisches Datenverarbeitungsprogramm zu kreieren, das - stellvertretend für viele ähnlich geartete Fälle - die Vielfalt des Rechners HP-41C/CV erkennen läßt. Derartige Programme sind hinsichtlich diffiziler Rechenoperationen weniger aufwendig, erfordern jedoch zur Aufnahme von Informationen viel Speicherplatz. Hier erweist sich das erweiterte Funktions/Speicher-Modul (HP 82180 A) und das erweiterte Speicher-Modul (HP 82181 A) als nützliche Zusatzeinrichtung.

1 PROBLEMSTELLUNG

Eine der häufigsten und wohl ältesten Anwendungen der Datenverarbeitung stammt aus dem kaufmännischen Rechnungswesen. Es liegt somit nahe, aus diesem Bereich ein Beispiel auszuwählen. Im einzelnen handelt es sich um einen speziellen Teil der Betriebskostenabrechnung eines größeren Wohnhauses. Er enthält eine Anzahl sogenannter Nebenkosten unterschiedlicher Kostenarten, die jährlich - in Abhängigkeit von der allgemeinen wirtschaftlichen Lage - veränderlich anfallen und dem Mieter je nach Größe der von ihm genutzten Wohnfläche ausschließlich eines Grundbetrages angelastet werden. Offensichtlich ist es für die Vertragspartner (Mieter, Vermieter) vorteilhaft, sich in zeitlichen Abständen einen Auszug absoluter und relativer Werte derartiger Kosten in übersichtlicher Form darstellen zu lassen.

Das Beispiel umfaßt einen Abschnitt von 8 Jahren. Beginnend mit 1975, entstehen pro Jahr die Nebenkosten von jeweils 16 Kostenarten. Z. Z. existieren also bereits 128 Einzelbeträge in DM. Zur eingehenden Information über die ständig zunehmende Datensammlung wären z.B. folgende, wahlweise abzurufende Begriffe von Interesse:

1. Betrag einer Kostenart in einem bestimmten Jahr.
2. Vergleich einer Kostenart zwischen 2 beliebigen Jahren in DM und %.
3. Aufstellung einer Kostenart für n aufeinanderfolgende Jahre in DM.
4. Wie vor, jedoch in prozentualen Änderungen.
5. Summe der Kostenarten eines Jahres und ihre prozentuale Änderung gegenüber der des Vorjahres.
6. Prozentualer Anteil sämtlicher Kostenarten an der Jahressumme.
7. Vergleich unterschiedlicher Kostenarten eines Jahres in DM und %.

2 PROGRAMMBESCHREIBUNG

Um auch nach längerer Anwendungspause eines umfangreichen Programms keine Unklarheiten aufkommen zu lassen, sollte seine Verständlichkeit nicht dem Mangel an Platz für ausreichende Benennungen u.a. geopfert werden. Dieser berechtigten Forderung wird das erweiterte Funktions/Speicher-Modul gerecht. Seine vielfachen Eigenschaften gestatten vor allem, alpha-numerische Texte mit 24 Zeichen pro Zeile ohne besondere Maßnahmen in einen sogenannten Record aufzunehmen. Diesem Vorteil zufolge erscheint es zweckmäßig, die einzelnen Abschnitte des vorliegenden Programms BETRKO (Betriebskosten) in nachstehender Anordnung den verschiedenen Speichern zuzuordnen.

1	Hauptprogramm	BETRKO : Progr.Speicher im Rechner	
1	Textblock (Progr.Funkt.)	BTRF-AS: ASCII-File	} im erweiterten Funktions/ Speicher-Modul
1	Textblock (Kostenarten)	BTRK-AS: ASCII-File	
8	Datenblöcke (DM-Beträge)	BTRKXX : Daten-File	
1	Hilfsprogramm	RECIN : Progr.File	
1	Hilfsprogramm	DREGX : Progr.File	

Der Inhalt der beiden Textblöcke ist aus T a b e l l e 1 und T a b e l l e 2, derjenige sämtlicher Datenblöcke aus T a b e l l e 3 ersichtlich. Die Anzahl der vorzusehenden Register beträgt für jeden Record 55 (CRFLAS) und Datenblock 17 (CRFLD).

Tabelle 1 Record BTRF-AS

BTRF-AS	RUN
*PROGRAMM FUNKTIONEN :	
A KO-NR UND BENENNUNGEN	
B JAEHRL. KO EINER ART	
C VERGL. VON 2 JAEHRL.	
KO EINER ART (DM & %)	
D KO-AUFSTLLG. EINER ART	
FUER n AUF EINANDER	
FOLGENDE JAHRE	
E WIE VOR, MIT AENDE-	
RUNGEN IN %	
F SUMME DER KO INNERHALB	
EIN. JAHR., %-AENDERG.	
GEGENUEBER d. VORJAHR	
G PROZT. ANTEIL DER KO-	
ARTEN AN d. JAHR. SUMME	
H VERGLEICH UNTERSCHDL.	
KO-ARTEN (DM & %)	
END OF FL	

Tabelle 2 Record BTRK-AS

BTRK-AS	RUN
*KOSTEN PRO JAHR IN DM :	
131 HEIZUNG	
332 STROM-PUMPEN	
733 WARTUNG-HEIZUNG	
336 STROM-LUEFTUNG	
438 WASSER-ZULAUF	
439 WASSER-ABLAUF	
240 WARM-WASSER	
742 WARTUNG-LUEFTUNG	
343 STROM-AUFZUG	
744 WARTUNG-AUFZUG	
346 STROM-BELEUCHTUNG	
947 REINIGUNG-HAUS	
948 REINIGUNGS-MATERIAL	
949 REINIGUNG-FENSTER	
954 REINIGUNG-HEGE	
757 WARTUNG-ANTENNE	
END OF FL	

Tabelle 3 Datenblöcke BTRKXX

R00= "BTRK75"	R00= "BTRK76"	R00= "BTRK77"	R00= "BTRK78"
R01= 774,39	R01= 791,16	R01= 794,59	R01= 690,53
R02= 52,68	R02= 20,37	R02= 39,06	R02= 43,25
R03= 0,00	R03= 0,00	R03= 2,44	R03= 7,42
R04= 0,00	R04= 0,00	R04= 36,89	R04= 42,47
R05= 0,00	R05= 15,83	R05= 99,98	R05= 115,60
R06= 0,00	R06= 0,00	R06= 0,00	R06= 0,00
R07= 0,00	R07= 49,39	R07= 3,16	R07= 137,31
R08= 0,00	R08= 0,00	R08= 2,45	R08= 7,45
R09= 26,86	R09= 34,94	R09= 32,58	R09= 35,82
R10= 72,94	R10= 72,63	R10= 76,77	R10= 87,38
R11= 163,15	R11= 197,76	R11= 104,11	R11= 110,99
R12= 100,96	R12= 105,44	R12= 121,23	R12= 139,79
R13= 0,00	R13= 0,00	R13= 0,00	R13= 0,00
R14= 3,68	R14= 4,12	R14= 4,39	R14= 3,71
R15= 0,00	R15= 0,00	R15= 0,00	R15= 0,00
R16= 16,69	R16= 19,03	R16= 18,58	R16= 20,68

R00= "BTRK79"	R00= "BTRK80"	R00= "BTRK81"	R00= "BTRK82"
R01= 781,49	R01= 774,18	R01= 1045,69	R01= 1207,94
R02= 38,33	R02= 44,70	R02= 37,37	R02= 36,10
R03= 7,64	R03= 8,75	R03= 10,38	R03= 10,83
R04= 56,83	R04= 56,04	R04= 50,18	R04= 46,30
R05= 169,10	R05= 167,36	R05= 174,03	R05= 211,02
R06= 84,99	R06= 89,61	R06= 109,44	R06= 125,39
R07= 137,03	R07= 129,35	R07= 147,47	R07= 158,22
R08= 10,24	R08= 12,21	R08= 12,37	R08= 12,91
R09= 35,85	R09= 34,92	R09= 29,71	R09= 34,91
R10= 90,90	R10= 100,48	R10= 98,25	R10= 104,19
R11= 110,84	R11= 110,76	R11= 103,89	R11= 72,94
R12= 122,16	R12= 135,84	R12= 139,84	R12= 143,77
R13= 0,81	R13= 1,51	R13= 1,03	R13= 0,32
R14= 4,23	R14= 5,44	R14= 5,95	R14= 6,10
R15= 27,65	R15= 30,61	R15= 32,48	R15= 33,98
R16= 21,19	R16= 23,09	R16= 23,66	R16= 23,66

Jeder Kostenart sind bestimmte Jahresbeträge zugehörig, deren Übereinstimmung durch den gemeinsamen Record- bzw. Register-Pointer gegeben ist.

Eingaben in das Funktions/Speicher-Modul, ggf. auch hierin auszuführende Korrekturen oder Löschungen, erfordern bestimmte, häufig zu wiederholende Ausführungsbefehle. Eine bewährte Tastenzuordnung dieser USER KEYS wird in T a b e l l e 4 aufgeführt. Im Kassettenlaufwerk (HP 82161 A) gespeichert, können diese Zuordnungen bei Bedarf auf das Tastenfeld des Rechners übertragen werden. Um die Records auf eingeschlichene Fehler zu überprüfen, wird das Hilfsprogramm RECI angewendet, das den Record-Inhalt kontinuierlich ausdruckt. Auch für die jeweils vor der Übergabe in das Funktions/Speicher-Modul im Hauptspeicherregister aufzustellenden Datenblöcke ist ein kleines Hilfsprogramm DREGX vorgesehen. Bei erforderlichen

Tabelle 4 Tastenzuordnungen USER KEYS

USER KEYS:	21 CRFLAS	42 SAVEP
11 CRFLD	22 APPREC	43 GETSUB
12 SAVERX	23 GETREC	51 RCLPT
13 GETRX	24 DELREC	61 ENDIR
14 GETX	25 INSREC	71 PURFL
15 SEEKPT	41 SEEKPTA	81 CLKEYS

Korrekturen löscht es durch eine zuvor in die Anzeige getastete Ziffer der Art aaa.bbb den hierdurch gekennzeichneten Registerabschnitt. Abschließend können beide Hilfsprogramme in das Funktions/Speicher-Modul übertragen (SAVEP) und im Hauptspeicher gelöscht werden. Die Tastenzuordnungen sind nach Betätigung der Taste 81 (CLKEYS) wirkungslos.

Ein Teil-Test des vorliegenden Programms mit wesentlich verringertem Eingabeaufwand ist bereits mit 2 Kostenarten und Datenblöcken benachbarter Jahreszahlen nach folgendem Beispiel durchführbar:

Textblock	Pointer	Datenblöcke
BTRK-AS	0	BTRK81 BTRK82
438 Wasser-Zulauf	1	174,03 211,02
240 Warm-Wasser	2	147,47 158,22

Für die Funktionen unter F und G nur möglich, wenn die in F programmierte Schleifensteuerungszahl 1.01601 vorübergehend durch 1.00201 ersetzt wird.

3 PROGRAMMDURCHFÜHRUNG

Nach Eingabe sämtlicher Programme in den Hauptspeicher, der Records und Datenblöcke in das erweiterte Funktions/Speicher-Modul, erfolgt der Start durch Ausführung von BETRK0. Zunächst wird eine Aufstellung der Programmfunktionen ausgedruckt. Abschließend sind diese auf B, C, D, F und H abrufbereit. Auf einige Besonderheiten dieser Funktionen sei nachträglich hingewiesen:

- A Jedem Text "Kostenart" ist eine dreistellige Ziffer zugehörig (z.B. gegliedert in Kosten-Gruppe und -Art), einem Schlüssel vergleichbar, mit dessen Hilfe der gewünschte alpha-numerische Text dem Record BTRK-AS entnommen werden kann.
- B Jeweils eine Arten-Zeile und Jahres-Spalte der Datenblock-Matrix bestimmen den gesuchten Betrag.

- C Sofern einer von zwei Beträgen gleich Null ist, entfällt die prozentuale Änderung des zweiten Betrages.
- D,E Die Register ab Nr. 11 dienen lediglich zur Speicherung der Jahresbeträge. Hierdurch wird die Berechnung der prozentualen Änderungen wesentlich vereinfacht. Zur besseren Übersicht der untereinander stehenden Dezimalzahlen mit unterschiedlichen Vorkomma-Stellen ist eine Druckformatierung vorgesehen. Sollte der darin logarithmierte Betrag in DM oder % gleich Null sein, wird dieser auf Eins erhöht, um die ERROR-Anzeige zu verhindern. Eine Sperre durch FLAG 02 läßt die Taste E nur nach vorangegangener Aufstellung der DM-Beträge wirksam werden.
- F,G Auch hier wird die vorbeschriebene Sperre für G in Abhängigkeit von F und eine Druckformatierung angewendet. Die Nachkommastellen unter G sind dreistellig, um für die Summe der Prozentsätze möglichst den Wert 100 zu erreichen. Als vorteilhaft erweist sich hier der Befehl ANUM im erweiterten Funktions/Speicher-Modul. Er selektiert den Ziffernanteil einer alpha-numerischen Kette und befördert ihn vom Alpha- in das X-Register.
- H Bei Eingabe der Kosten-Nummern ist nicht bekannt, welche von beiden dem höheren oder geringeren Betrag zugehört. Es muß somit jeweilig geprüft werden, ob die Rechnungsfolge der Angabe des definierten Prozentsatzes entspricht.

4 BEISPIELE

Darstellung von einigen willkürlich ausgewählten Programm-funktionen.

<pre> XEQ B >KO-NR ↑ JAHR19XX ? 343 ENTER↑ 1978 RUN 343 STROM-AUFZUG *BTRK78 DM 35,82 ----- </pre>	<pre> XEQ D >KO-NR ? 240 RUN >ZEITRAUM 19XX↑19YY ? 1975 ENTER↑ 1979 RUN *KOSTEN PRO JAHR IN DM : 240 WARM-WASSER BTRK75 0,00 BTRK76 49,39 BTRK77 3,16 BTRK78 137,31 BTRK79 137,03 ----- </pre>	<pre> XEQ G * %-ANT. d. KO-ART. AN d. SUMME DER BTRK82 : KO-NR %-SATZ 131 54,202 332 1,620 733 0,486 336 2,078 438 9,469 439 5,626 240 7,100 742 0,579 343 1,566 744 4,675 346 3,273 947 6,451 948 0,014 949 0,274 954 1,525 757 1,062 ----- </pre>
<pre> XEQ C >KO-NR ↑ JAHR19XX ? 343 ENTER↑ 1979 RUN >JAHR 19XX ? 1981 RUN 343 STROM-AUFZUG *BTRK79 DM 35,85 *BTRK81 DM 29,71 *ÄNDERUNG IN % : -17,13 ----- </pre>	<pre> XEQ E *ÄNDERUNG IN % : BTRK76 ENTFÄLLT BTRK77 -93,60 BTRK78 4245,25 BTRK79 -0,20 ----- </pre>	<pre> XEQ H >KO-NR X1↑X2 ? 438 ENTER↑ 240 RUN >JAHR 19XX ? 1980 RUN *KO-NR U. BTRG. IN DM : 438 240 167,36 129,35 * %-SATZ DES GERINGEREN BETRAGES : 77,29 ----- </pre>
<pre> XEQ C >KO-NR ↑ JAHR19XX ? 439 ENTER↑ 1978 RUN >JAHR 19XX ? 1981 RUN 439 WASSER-ABLAUF *BTRK78 DM 0,00 *BTRK81 DM 109,44 * %-ANGABE ENTFÄLLT ----- </pre>	<pre> XEQ F >JAHR 19XX ? 1982 RUN *SUMME DER KO IM JAHR : BTRK82 DM 2.228,58 *ÄNDERUNG. GEGENUEB. DEM VORJAHR IN % : 10,23 ----- </pre>	

5 ANWEISUNGSLISTE UND FLUSSDIAGRAMM BETRKO

ANWEISUNGSLISTE

```

CAT 1
LBL*BETRKO
END      1188 BYTES
LBL*RECIN
END      51 BYTES
LBL*DREGX
.END.    42 BYTES

EMDIR
BTRF-AS A055
BTRK-AS A050
BTRK75 D017
BTRK76 D017
BTRK77 D017
BTRK78 D017
BTRK79 D017
BTRK80 D017
BTRK81 D017
BTRK82 D017
RECIN P006
DREGX P006
      85.00 ***
      PRP "BETRKO"
      01*LBL "BETRKO"
      19 PSIZE CLRG 1
      STO 01 4 STO 04 6
      STO 06 9 STO 09 CF 01
      CF 02 CF 20 CF 29
      SF 00 FIX 2 CLA
      "BTRF-AS" GTO a
      22*LBL a
      CF 00 CLA "BTRK-AS"
      26*LBL a
      FS? 00 0 FC?C 00 1
      SEEKPTA
      32*LBL b
      GETREC AVIEW GTO b
      36*LBL B
      SF 00 CF 02 XEQ IND 04
      XEQ IND 01 XEQ d RCL Z
      XEQ c XEQ IND 06
      XEQ 05 XEQ 02
      XEQ IND 09 RTN
      49*LBL c
      FIX 0 CF 29 CLA
      ARCL X POSFL STO 10
      FIX 2 RTN
      58*LBL d
      CLA "BTRK-AS" 0
      SEEKPTA RTN
      64*LBL C
      SF 00 CF 02 XEQ IND 04
      STO 07 XEQ IND 04
      STO 08 XEQ d RCL T
      XEQ c XEQ IND 06 SF 01
      RCL 07
      77*LBL 00
      XEQ IND 01 RCL 10
      XEQ 05 FS? 01 STO 02
      FC? 01 STO 03 XEQ 02
      RCL 08 FS?C 01 GTO 00
      RCL 02 RCL 03 * X=0?
      GTO 10 RCL 02 RCL 03
      %CH "AENDERUNG"
      "FIN % : " ARCL X
      AVIEW XEQ IND 09 RTN
      103*LBL 01
      CLA "BTRK" 1900 -
      ARCL X ASTO 00 RTN
      111*LBL 02
      SF 29 FIX 2
      "F DM " ARCL X
      AVIEW CF 29 RTN
      119*LBL 04
      FIX 0 FS? 00
      ">KO-NR ↑"
      "FJAHR19XX ?" FC?C 00
      ">JAHR 19XX ?" PROMPT
      RTN
      128*LBL 05
      CLA ARCL 00 RCL 10
      SEEKPTA GETX "*"
      ARCL 00 RTN
      137*LBL 06
      CLA SEEKPT GETREC
      AVIEW RTN
      143*LBL 10
      "*" %-ANGABE "
      "FENTFUELLT" AVIEW
      XEQ IND 09 RTN
      149*LBL D
      SF 02 CF 03 CF 04 11
      STO 03 FIX 0
      ">KO-NR ?" PROMPT
      STO 02 ">ZEITRAUM 19"
      "FXX19YY ?" PROMPT
      1900 ST- Z ST- Y RDN
      1 E-3 * 1 E-5 + +
      STO 05 STO 07 XEQ d
      XEQ IND 06 RCL 02
      XEQ c XEQ IND 06
      178*LBL 03
      CLA FIX 0 "BTRK"
      RCL 05 INT ARCL X
      FS? 03 RTN RCL 10
      SEEKPTA GETX STO 08
      STO IND 03
      192*LBL 11
      FIX 2 8 RCL 08 INT
      X=0? ISG X ABS LOG
      INT - FS? 04 RTN ACA
      SKPCHR RCL 08 ACX
      PRBUF 1 ST+ 03 ISG 05
      GTO 03 XEQ IND 09 RTN
      216*LBL 09
      "-----"
      "F-----" AVIEW
      CF 29 RTN
      222*LBL E
      FC?C 02 RTN 11 STO 03
      RCL 07 STO 05 1
      ST+ 05 "AENDERUNG"

```

```
"FIN % : " AVIEW SF 03
SF 04
```

```
236*LBL 07
XEQ 03 ACA RCL 03
RCL IND 03 X=0? GTO 12
1 ST+ 2 RCL IND 2
X<>Y RDN %CH STO 08
XEQ 11 SKPCHR RCL 08
ACX
```

```
254*LBL 08
PRBUF 1 ST+ 03 ISG 05
GTO 07 CF 03 CF 04
XEQ IND 09 STOP
```

```
264*LBL 12
"ENTFAELT" 6 SKPCHR
ACA GTO 08
```

```
270*LBL F
CF 00 SF 02 0 STO 02
1.01601 STO 03 STO 07
CLA XEQ IND 04
XEQ IND 01 1 SEEKPTA
```

```
283*LBL 13
GETX ST+ 02 ISG 03
GTO 13 FS? 03 GTO 16
"*SUMME DER K"
"FO IM JAHR : " AVIEW
RCL 02 STO 05 " "
ARCL 00 XEQ 02 SF 03
FIX 0 0 STO 02 RCL 07
STO 03 CLA ARCL 00
ANUM 1 - "BTRK"
ARCL X 1 SEEKPTA
GTO 13
```

```
314*LBL 16
CF 03 RCL 02 RCL 05
%CH FIX 2
"*AENDERNG. G"
```

```
"FEGENUEB. DEM" AVIEW
" VORJAHR IN % : " "F "
ARCL X AVIEW
XEQ IND 09 RTN
```

```
329*LBL G
FC?C 02 RTN SF 04
"* %-ANT. d. "
"FO-KO-ART. AN" AVIEW
" d. SUMME DER "
ARCL 00 "F : " AVIEW
" KO-NR "
"FO%-SATZ" AVIEW ADV
```

```
344*LBL 14
FIX 0 CLA ARCL 00
RCL 07 INT SEEKPTA
GETX RCL 05 / 1 E2 *
STO 08 XEQ d RCL 07
INT SEEKPTA GETREC
ANUM CLA " "
ARCL X ACA XEQ 11
SKPCHR RCL 08 FIX 3
ACX PRBUF ISG 07
GTO 14 CF 02 CF 04
FIX 2 XEQ IND 09 RTN
```

```
380*LBL H
CF 00 CF 01 CF 02
SF 03 SF 04 FIX 0
">KO-NR X1+X2 ?" PROMPT
STO 08 X<>Y STO 07
XEQ IND 04 XEQ IND 01
```

```
394*LBL 15
XEQ d CLA FS? 03
ARCL 07 FC?C 03
ARCL 08 POSFL STO 10
```

```
XEQ 05 FS? 04 STO 03
FC? 04 STO 02 FS?C 04
GTO 15 RCL 02 RCL 03
X<=Y? XEQ 17 RCL Y
STO 02 RDN STO 03 CLA
FIX 0 "*KO-NR U. B"
"FRG. IN DM : " AVIEW
CLA " " ARCL 07
" " " ARCL 08
AVIEW CLA FIX 2 SF 29
" " FS? 01 ARCL 02
FC? 01 ARCL 03
" " " "F "
FS? 01 ARCL 03 FC?C 01
ARCL 02 AVIEW RCL 02
RCL 03 X=Y? GTO 10 *
X=0? GTO 10 RCL 02
RCL 03 / 1 E2 * CLA
"* %-SATZ DES"
"GERINGEREN" AVIEW
" BETRAGES : " "F "
ARCL X AVIEW CF 29
XEQ IND 09 RTN
```

```
467*LBL 17
X<>Y SF 01 RTN END
```

```
PRP "RECIN"
```

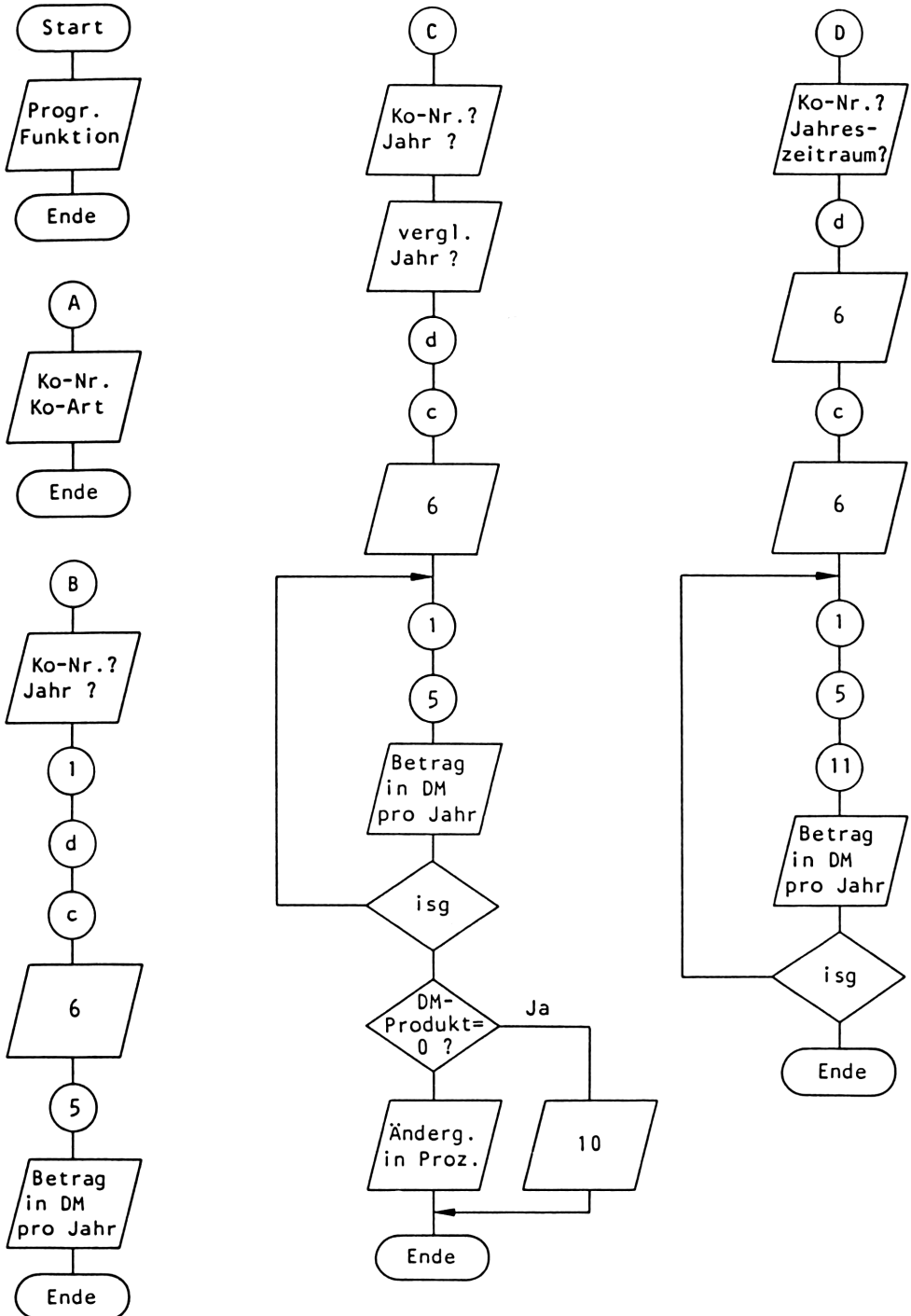
```
01*LBL "RECIN"
AON ">FILE NAME?"
PROMPT AOFF 0 SEEKPTA
```

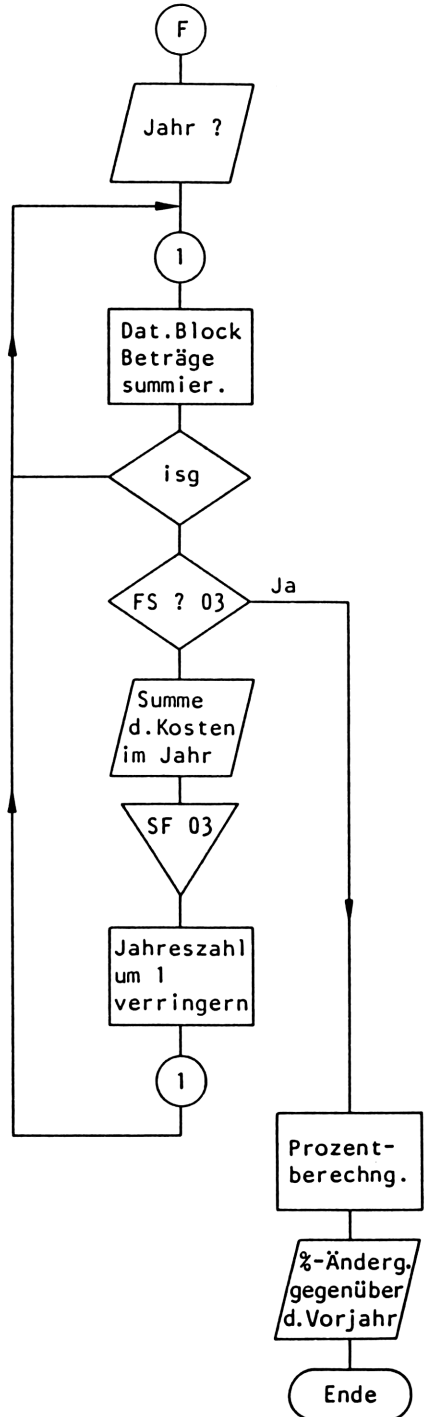
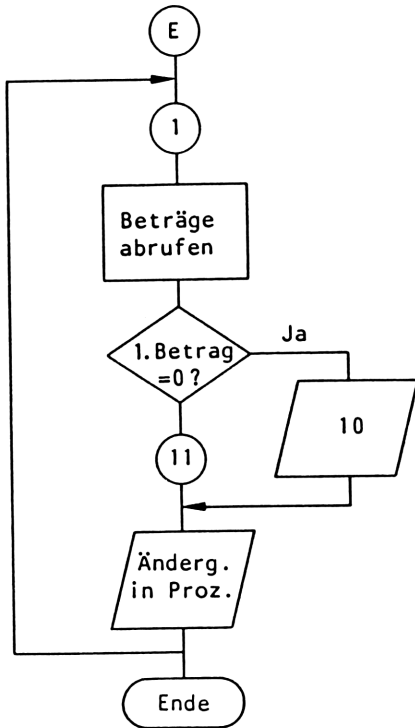
```
08*LBL 01
GETREC AVIEW GTO 01
END
```

```
PRP "DREGX"
```

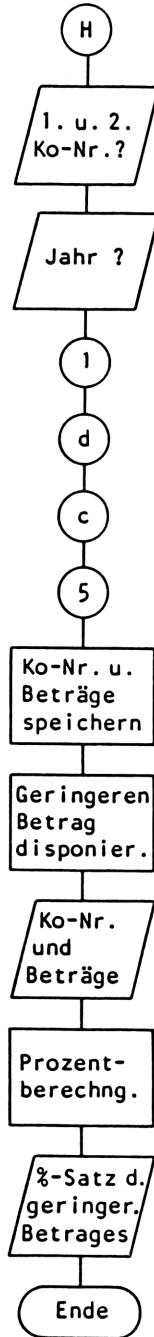
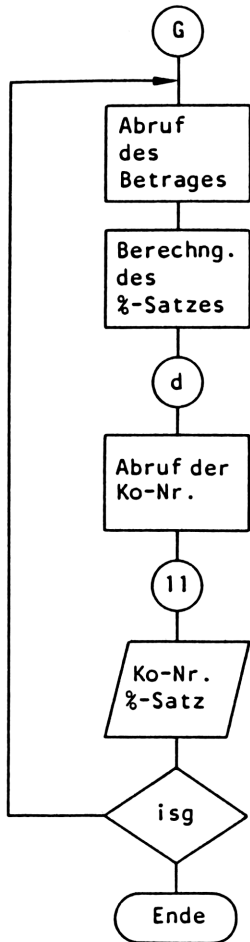
```
01*LBL "DREGX"
">aaa.bbb?" PROMPT 0
```

```
05*LBL 00
STO IND Y ISG Y GTO 00
.END.
```





- Ko = Kosten
- BTRK = Betr. Kosten
- Rec = Record
- Fl = File
- isg = Schleifensteuerung
- (c) = Record Pointer bestimmt Ko-Nr.
- (d) = Record Pointer auf 0 gesetzt
- (1) = File Name für Datenblockbild.
- (5) = Record Pointer bestimmt Betrag
- 6 = Text aus Record BTRK-AS
- 10 = Prozent-Angabe entfällt
- 11 = Druckformatierg.



Schaltalgebra und Logiknetzwerke – mit dem HP-41 getestet

von Herbert Hoffmann

Teil I: Schaltalgebra

Neben verschiedenen graphischen Verfahren bietet die Schaltalgebra mit ihren Rechenregeln eine ausgezeichnete Methode, Vereinfachungen von Schaltkreisen mit bistabilen Schaltelementen aufzuzeigen. Das hier vorgestellte Programm will helfen, die Schaltalgebra anzuwenden. Es testet:

1. die Eindeutigkeit algebraisch dargestellter Schaltungen
2. die Rechenregeln (Axiome) der Schaltalgebra
3. das Dualitätsprinzip
4. das De Morgansche Theorem
5. die konjunktive und disjunktive Normalform
6. die Zulässigkeit berechneter Vereinfachungen
7. das algebraisch dargestellte Netzwerk eines Volladdierers.

1 VORBEMERKUNGEN

Achtung: Vor dem Arbeiten mit dem Programm ist im X-FUNCTIONS-Modul ein ASCII-File mit dem File-Namen "BA" und 8 Register nach Anweisung des Benutzer-Handbuches einzurichten.

Die globale Marke 'BA' (für Boolesche Algebra) am Anfang des Programms ist der Taste 'BEEP' zugeordnet. Nach Betätigung dieser Taste stellt das Programm die erforderlichen Anfangsbedingungen her, z.B. Löschen der Flags 0 - 7. Bei Unklarheiten im Programmablauf sollte immer diese Taste betätigt werden. Am Ende der Vorbereitungen erscheint in der Anzeige "BOOLE: 1815-64" zu Ehren von George Boole, dem Begründer der nach ihm benannten Algebra.

Das Programm ist weitgehend gegen Eingabe- und Bedienungsfehler gesichert. Der öfter vorhandene Befehl 'SF 66' (Setzen der nicht vorhandenen Flag 66) soll genau das bezwecken, was er auslöst: ein "NONEXISTENT" in die Anzeige bringen und das Programm stoppen.

Es sind 5 Eingangs-Variablen 'a', 'b', 'c', 'd' und 'e' vorgesehen. Zur Negation der Eingangs-Variablen sowie ganzer Ausdrücke wird das Minuszeichen verwendet.

Maximal können 48 Symbole eingegeben werden. Folgende Symbole sind zulässig:

a, b, c, d, e, -, +, x (steht für UND-Verknüpfung und kann wahlweise als Multiplikationszeichen '*' oder als Buchstabe 'X' eingegeben werden), =, \neq , ?, \uparrow , /, L, M, \emptyset (für Signalwert Null) und 1 (für Signalwert Eins).

Bei mehr als 24 Symbolen werden zunächst die ersten 24 Symbole eingegeben und mit R/S! bestätigt. Der Rest des algebraischen Ausdrucks wird nach der Frage "FERTIG?" ins Alpha-Register übertragen und mit R/S! weiterverarbeitet. Bei genau 24 Symbolen wird nach der Meldung "FERTIG?" ohne weitere Eingabe R/S! betätigt. Bei weniger als 24 Symbolen unterbleibt die Frage "FERTIG?". Der Rechner meldet nach der Übernahme des algebraischen Ausdrucks die Anzahl der Symbole.

Die lokalen Marken

- A: Ausgabe der gesamten Formel mit jeweils 12 Symbolen. Nach R/S! kommen die nächsten 12 Symbole in die Anzeige. Dieses wird bis zur Meldung "FORMEL KOMPL." wiederholt
- a: Ausgabe der letzten 8 Symbole bei aktueller Länge der Formel. Die vor den Symbolen stehende Ziffer ist die Symbol-Nummer des ersten gezeigten Symbols
- B: Eingabe der gewünschten Ausgangswerte. Der Rechner fragt mit "EING.-VAR.?" nach der Anzahl der Eingangs-Variablen. Nach deren Eingabe verlangt dann der Rechner mit "AW \emptyset /1?" die Eingabe \emptyset oder 1. Bei 5 Variablen erscheint in der Anzeige "I. AW \emptyset /1?" Hier sind die ersten 16 Ausgangswerte einzugeben. Die zweite Hälfte wird nach "II. AW \emptyset /1?" eingetastet.

Die Werte brauchen in beiden Hälften jeweils nur bis zur letzten '1' in den Rechner übertragen zu werden. Die Eingabe wird abgeschlossen mit "AW KOMPL."

b: Verändern der Anzahl der Eingangs-Variablen

C: Konjunktive Normalform bei vorhandenen Ausgangswerten

c: Konjunktive Normalform bei vorhandener Formel.

Bei gesetzter Flag 08 wird zuerst die Kombinations-Nummer angezeigt und nach R/S! die konjunktive Normalform z.B. "(+ : ab \bar{c} d \bar{e})x". In normaler Schreibweise: $(a + \bar{b} + c + \bar{d} + e)$. Das 'x' hinter der Klammer soll zeigen, daß die Maxterme UND-verknüpft sind

D: Disjunktive Normalform bei vorhandenen Ausgangswerten

d: Disjunktive Normalform bei vorhandener Formel.

Hier kann z.B. in der Anzeige erscheinen "(x : a \bar{b} cd \bar{e})+". In normaler Schreibweise bedeutet dies: $(\bar{a}xbxcxd\bar{e})$. Das '+' hinter der Klammer soll zeigen, daß die Minterme ODER-verknüpft sind

E: Ausgangswert eines Ausdrucks für eine einzelne Kombination

e: Test eines Ausdrucks bis zu einem gewünschten Symbol. Diese Taste gestattet das Aufschlüsseln einer Formel und zeigt dadurch leicht die Wirkungen der einzelnen Symbole und die Zusammenhänge

F: Eingabe des algebraischen Ausdrucks, der vom Rechner als "FORMEL?" erfragt wird. Die Symbole sind nach den UPN-Regeln einzugeben, z.B. $a + bxc$ "bcxa+"

G: Prüfen der Gleichheit der Ausdrücke auf beiden Seiten des '='-Zeichens

H: Berechnung und Ausgabe der Ausgangswerte einer Formel. Es werden jeweils 8 Ausgangswerte in die Anzeige gebracht. Eine Betätigung von R/S! zeigt weitere 8 Ausgangswerte. Den Abschluß bildet die Meldung "AW KOMPL."

I: Prüfen der Ungleichheit der Ausdrücke auf beiden Seiten des '≠'-Zeichens

J: Vergleich einer 'Formel' mit vorher eingegebenen Ausgangswerten. Stimmen die Ausgangswerte aller Kombinationen über-

ein, wird dieses am Schluß der Berechnung mit "=" gemeldet. Die Meldung " ≠ KOMB. 7" z.B. zeigt an, daß bei der Kombination 7 erstmals keine Gleichheit besteht

Das Alpha-Register (Eingabe und Anzeige) wird durch Anführungszeichen (") gekennzeichnet.

Um Platz zu sparen, werden bei der Beschreibung nach Möglichkeit 'Kurzformen' verwendet. So steht z.B. für 'Nach der Betätigung der Taste 'H' erscheint in der Anzeige als Ergebnis "0001"' die Kurzform 'H!:"0001"'.

2 DIE EINDEUTIGKEIT ALGEBRAISCH DARGESTELLTER SCHALTUNGEN

Dieses gilt für Schaltungen mit bistabilen Schaltelementen, die nur zwei Zustände einnehmen können. Ein Lichtschalter z.B. hat die beiden Stellungen 'AUS' und 'EIN', auch als '0' und '1' bezeichnet. In dieser Beschreibung sind dafür die Symbole 'Ø' und '1' gewählt.

Zuerst testen wir die 'Signale' der Variablen 'e', 'd', 'c', 'b' und 'a'. Zum Test legen wir folgendes Schema an:

```

F "e d c b a"
K S: 1 2 3 4 5

```

Es bedeuten: 'F' = Formel, 'K' = Kombination, 'S' = Symbol-Nummer

F! "EING.-VAR.?" 5 R/S! "FORMEL?" "edcba" R/S! "(5)" (fünf Symbole wurden übernommen). e! ": SYMBOL NR.?" (bis zu welchem Symbol soll getestet werden?). Wir nehmen einmal als Beispiel Symbol 'c' mit der Nummer 3. Nach R/S! rechnet das Programm die Kombinationen 0 - 31 durch und gibt aus: "00001111" R/S!: "00001111" R/S!: "00001111" R/S!: "AW KOMPL.". Das Ergebnis wird von links nach rechts gelesen und von oben nach unten in unser Schema eingetragen. Die Ausgangswerte der fünf Variablen zusammen ergibt die Funktionstabelle.

	F	e	d	c	b	a
K	S:	1	2	3	4	5
0		0	0	0	0	0
1		0	0	0	0	1
2		0	0	0	1	0
3		0	0	0	1	1
4		0	0	1	0	0
5		0	0	1	0	1
6		0	0	1	1	0
7		0	0	1	1	1
8		0	1	0	0	0
9		0	1	0	0	1
10		0	1	0	1	0
11		0	1	0	1	1
12		0	1	1	0	0
13		0	1	1	0	1
14		0	1	1	1	0
15		0	1	1	1	1
16		1	0	0	0	0
17		1	0	0	0	1
18		1	0	0	1	0
19		1	0	0	1	1
20		1	0	1	0	0
21		1	0	1	0	1
22		1	0	1	1	0
23		1	0	1	1	1
24		1	1	0	0	0
25		1	1	0	0	1
26		1	1	0	1	0
27		1	1	0	1	1
28		1	1	1	0	0
29		1	1	1	0	1
30		1	1	1	1	0
31		1	1	1	1	1

Die Variablen können verschieden miteinander 'verknüpft' werden. Man unterscheidet 3 Grundverknüpfungen: UND-, ODER- und Negationsglieder. Für ein UND-Gatter gilt: Am Ausgang ist nur dann ein '1'-Signal, wenn alle Eingänge ein '1'-Signal aufweisen. Zur Darstellung eignet sich die 'Multiplikation':

$0 \times 0 = 0$, $0 \times 1 = 0$, $1 \times 0 = 0$, $1 \times 1 = 1$. Für ein ODER-Gatter gilt: Schon ein '1'-Signal an einem Eingang genügt, um am Ausgang ein '1'-Signal zu erzeugen. Mit einer gewissen Einschränkung eignet sich für die Darstellung die 'Addition':

$0 + 0 = 0$, $0 + 1 = 1$, $1 + 0 = 1$, $1 + 1 = 1$ (und nicht wie in der allgemeinen Algebra $1 + 1 = 2$!). Wir können diese Zusammenhänge testen. Nach F! und "EING.-VAR.?" geben wir mit R/S! jeweils 0 ein und dann weiter als Formel z.B. "1 1 +". H!: "1". Oder als weiteres Beispiel "0 1 x" H!: "0".

Nun testen wir einige Verknüpfungen von 2 Variablen. Nach F! und der Frage "EING.-VAR.?" ist jeweils 2 mit R/S! einzugeben. Die folgende Zusammenstellung nennt die Bezeichnung der Verknüpfungen, die Eingabe der Symbole in UPN-Schreibweise und die erzeugten Ausgangswerte.

1. UND-Gatter (Boolesches Produkt, Konjunktion)	"abx"	"0001"
2. ODER-Gatter (Boolesche Summe, Disjunktion)	"ab+"	"0111"
3. NAND-Gatter (NICHT UND)	"abx-"	"1110"
4. NOR-Gatter (NICHT ODER)	"ab+ -"	"1000"
5. Exklusiv-Oder	"ab-xa-bx+"	"0110"
6. Äquivalenz	"a-b-xabx+"	"1001"
7. Eine Variation zu 5	"a-b-xabx+ -"	"0110"
8. Eine Variation zu 6	"ab-xa-bx+ -"	"1001"

Nach dem Gesamttest mit H! sollte ein Test der einzelnen Symbole mit e! angeschlossen werden. Ein Einzeltest zeigt leicht die Zusammenhänge.

3 ÜBERPRÜFUNG DER RECHENREGELN

Die folgende Tabelle zeigt die in der Literatur genannten 'Axiome', ihre Eingabe in den Rechner und das Ergebnis.

Nr.	Axiom	Formel	Taste	Anzeige	Taste	Ergebnis
3.1	$a + \emptyset = a$	"a \emptyset + = a"	R/S!	"(5)"	G!	" = "
3.2	$a x \emptyset = \emptyset$	"a \emptyset x = \emptyset "	R/S!	"(5)"	G!	" = "
3.3	$a + 1 = 1$	"a1+ = 1"	R/S!	"(5)"	G!	" = "
3.4	$a x 1 = a$	"a1x = a"	R/S!	"(5)"	G!	" = "
3.5	$a + a = a$	"aa+ = a"	R/S!	"(5)"	G!	" = "
3.6	$a x a = a$	"aax = a"	R/S!	"(5)"	G!	" = "
3.7	$a + \bar{a} = 1$	"aa-+ = 1"	R/S!	"(6)"	G!	" = "
3.8	$a x \bar{a} = \emptyset$	"aa-x = \emptyset "	R/S!	"(6)"	G!	" = "

Natürlich gelten alle Rechenregeln auch für die negierten Variablen, wie ein Test zeigt.

4 ÜBERPRÜFUNG DES DUALITÄTSPRINZIPS

Die duale Aussage eines Ausdrucks entsteht dadurch, daß man die Operationen (+) und (x) und gleichzeitig die 'Einselemente' (\emptyset) und (1) vertauscht. Als Beispiel: $a + b \neq \bar{a} x \bar{b}$. Der Ausdruck wird wie folgt eingegeben "ab+ \neq a-b-x" und mit I! getestet. Der Rechner meldet mit " \neq ", wenn die 'Umkehrung' gelungen ist.

5 ÜBERPRÜFUNG DES DE MORGANSCHEN THEOREMS

Wird die duale Aussage eines algebraischen Ausdrucks als Ganzes negiert, dann bleiben die Ausgangswerte der Funktion erhalten. Diesen Sachverhalt beschreiben die De Morganschen Gesetze, die nach folgender Tabelle getestet werden können.

Nr.	Gesetz	Eingabe	Taste	Anzeige	Taste	Ergebnis
5.1	$axb = \overline{a+b}$	"abx=a-b-+-"	R/S!	"(10)"	G!	"="
5.2	$a+b = \overline{axb}$	"ab+=a-b-x-"	R/S!	"(10)"	G!	"="
5.3	$\overline{axb} = \overline{a+b}$	"abx-=a-b-+-"	R/S!	"(10)"	G!	"="
5.4	$\overline{a+b} = \overline{axb}$	"ab+-=a-b-x"	R/S!	"(10)"	G!	"="

6 DIE NORMALFORMEN

Manchmal ist es wertvoll, die Normalform eines Ausdrucks zu kennen. Zur Übung wollen wir vom Rechner die disjunktive Normalform des Ausdrucks 'ax(a+b)' nennen lassen. Nach Eingabe der Formel "ab+ax", R/S! (5) d!:"(x:ab⁻)+ " R/S!:"(x:ab)+ " R/S!:"d/D KOMPL.". Die Normalform (disjunktiv) lautet also: ax⁻b+axb. Wir testen: "ab+ax=ab-xabx+" R/S! G! "=".

7 ÜBERPRÜFUNG BERECHNETER VEREINFACHUNGEN

Bei allen Vereinfachungen müssen die geforderten Ausgangswerte (Schließbeigenschaften) erhalten bleiben. Das Ziel ist, die Anzahl der Symbole zu verringern. Einige Beispiele werden durchgerechnet.

Läßt sich der Ausdruck 'ax(a+b)' vereinfachen? Die disjunktive Normalform lautet: f=ax⁻b+axb. Durch Ausklammern von 'a' wird die Formel zu f=ax(⁻b+b). Der Ausdruck in der Klammer kann nach 3.7 durch eine '1' ersetzt werden. Damit f=ax1. Dieser Ausdruck wird nach 3.4 zu 'a'. Also: ax(a+b)=a. Test: "ab+ax=a" G!:"=".

Vereinfachung zulässig.

Der Ausdruck 'a+axb' hat die gleiche disjunktive Normalform wie 'ax(a+b)' und kann ebenfalls durch 'a' ersetzt werden. Bitte testen!

Gegeben sei der Ausdruck $'axb+axc'$. Wir klammern hier 'a' aus und erhalten $f=ax(b+c)$. Test: $"abxacx+=bc+ax"$ G!: $"="$. Vereinfachung zulässig.

Der Ausdruck $'axb+ax\bar{b}+\bar{a}xb'$ ist zu vereinfachen. Wir geben die Formel ein und lassen uns durch c! die konjunktive Normalform nennen. Der Rechner gibt aus $"(+:ab)x; R/S!:"c/C KOMPL."$ Die einfache Lösung lautet also $f=a+b$. Ein Test zeigt Übereinstimmung.

Der Ausgang eines Netzwerkes soll folgende Werte bringen: $"0001000100011111"$. Gesucht wird eine einfache Schaltung.

Wir geben mit B! die gewünschten Ausgangswerte ein und lassen uns mit D! die disjunktive Normalform für 4 Variablen nennen:

$axb\bar{c}\bar{x}\bar{d}+axbxc\bar{d}+axb\bar{c}xd+\bar{a}x\bar{b}xcxd+ax\bar{b}xcxd+\bar{a}xbxcxd+axbxcxd$.

Diese Formel übersteigt mit 63 Symbolen die Aufnahmekapazität unseres Programms, aber wir wollen ja nach Möglichkeit ein einfacheres Netzwerk finden. Aus obigen 7 Mintermen klammern wir axb und $cx\bar{d}$ aus und erhalten:

$(axb)x(\bar{c}\bar{x}\bar{d}+cx\bar{d}+\bar{c}xd+cx\bar{d}) + (cx\bar{d})x(\bar{a}\bar{b}+\bar{a}xb+ax\bar{b}+axb)$

Der Minterm $axbxcxd$ wird dabei als zweimal vorhanden angesehen. Nun geben wir einmal $(\bar{c}\bar{x}\bar{d}+cx\bar{d}+\bar{c}xd+cx\bar{d})$ als Formel ein und testen mit H! die Ausgangswerte. Bei allen 16 Kombinationen erhalten wir den Wert '1'. Der Wert '1' stellt einen dauernd geschlossenen Stromkreis dar. Ebenso verhält es sich mit dem Ausdruck $(\bar{a}\bar{b}+\bar{a}xb+ax\bar{b}+axb)$. Der Gesamtausdruck reduziert sich damit auf die einfache Formel $f=axb+cx\bar{d}$. Diese Formel geben wir ein $"abxc\bar{d}x+"$ und erhalten mit H! die Ausgangswerte: $"0001000100011111"$, die den gewünschten Werten entsprechen. (Siehe auch Fig. 9 im Mikrocomputer-Jahrbuch '84, Seite 189)

8 DAS ALGEBRAISCH DARGESTELLTE NETZWERK EINES VOLLADDIERERS

Das Netzwerk für einen Volladdierer besitzt 3 Eingänge und die beiden Ausgänge Summe 's' und Übertrag 'ü', für die gilt:

$s=01101001$ und $\bar{u}=00010111$. Wir stellen uns die Aufgabe, die Ausgangswerte beider Ausgänge mit möglichst wenigen Gattern zu erzeugen.

Beginnen wir mit dem Ausgang 's'! Die Ausgangswerte 0110 und 1001 könnten von 'Exklusiv-Oder' und 'Äquivalenz' erzeugt sein.

Das Netzwerk 'Exklusiv-Oder' einmal in der Form ' $\bar{a}\bar{x}\bar{b}+axb$ ' eingegeben: " $a-b-xabx+-$ " hat bei 3 Variablen die Ausgangswerte 01100110 . Eine UND-Verknüpfung mit ' \bar{c} ' läßt die letzten beiden '1'-Signale verschwinden: " $a-b-xabx+-c-x$ " H!: " 01100000 ". Das Netzwerk 'Äquivalenz' hat bei 3 Variablen die Ausgangswerte 10011001 . Eine UND-Verknüpfung mit ' c ' läßt die beiden ersten '1'-Signale verschwinden: " $a-b-xabx+cx$ " H!: " 00001001 ". Die Gesamtformel könnte lauten:

" $a-b-xabx+-c-xa-b-xabx+cx+$ ". (Diese Formel besitzt 25 Symbole, das Symbol '+' wird nach der Frage "FERTIG?" mit R/S! eingegeben) H!: " 01101001 ". Damit hätten wir für den Ausgang 's' eine Lösung gefunden. Die Formel enthält jedoch 5 negierte Eingangs-Variablen, die durch andere Gatter ersetzt werden sollten. Wir erinnern uns an die De Morganschen Gesetze und finden unter '5.4': ' $\bar{a}\bar{x}\bar{b}=\overline{a+b}$ '. Mit diesen Möglichkeiten erhalten wir für 's' die Gesamtformel:

$\overline{a+b}+axb+c + (\overline{a+b}+axb)xc$, die wir in UPN-Schreibweise in unser Schema übertragen:

F	"	a	b	+	-	a	b	x	+	c	+	-	a	b	+	-	a	b	x	+	c	x	+	"
K	S:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	

Ein Einzeltest mit e! zeigt gut die Zusammenhänge.

Die Kombinationen mit den Symbolen 1 - 8 und 12 - 19 sind gleich. Nun gestattet das Programm, drei verschiedene Ausgangswerte mit den Symbolen '↑', 'L' und '=' an beliebiger Stelle der Formel festzuhalten und mit den Symbolen '/', 'M' und '?' an beliebiger Stelle der Formel zurückzugeben. Mit dieser Vereinfachung lautet die Formel:

F " $ab+-abx+\uparrow c+/-cx+$ ". Bitte testen!

Jetzt folgt der Ausgang 'ü'. Die disjunktive Normalform lautet:

$\bar{u} = a\bar{x}b\bar{c} + a\bar{x}bx\bar{c} + a\bar{x}bxc + axb\bar{c}$. Durch Ausklammern erhalten wir:

$$\bar{u} = a\bar{x}b(\bar{c} + c) + a\bar{x}c(\bar{b} + b) + bxcx(\bar{a} + a).$$

Die Verknüpfungen innerhalb der Klammern ergeben nach '3.7' jeweils '1' und können entfallen. Damit $\bar{u} = a\bar{x}b + a\bar{x}c + bxc$. Eine weitere Vereinfachung: $\bar{u} = a\bar{x}b + (a + b)xc$.

"abxab+cx+" H!: "00010111". Damit haben wir die Formeln für die Ausgänge 's' und 'ü' in getrennter Form. Die beiden Formeln haben wenig 'Gemeinsamkeiten'. Wir versuchen es einmal mit der konjunktiven Normalform. Für den Ausgang 's' lautet sie:

$$s = (a + b + c)x(\bar{a} + \bar{b} + c)x(\bar{a} + b + \bar{c})x(a + \bar{b} + \bar{c}).$$

Vereinfacht und ohne negierte Variablen:

$s = \overline{a+b} + a\bar{x}b + c + \overline{a+b} + a\bar{x}bxc$. Auch diese Formel wollen wir testen.
In UPN-Schreibweise:

F "a b + - a b x + - c + - a b + - a b x + - c x + -
K S: 1 2 3 4 5 6 7 8 9 10 11 12 13 14 15 16 17 18 19 20 21 22 23 24 25

Die Kombinationen mit den Symbolen 1 - 9 und 13 - 21 sind gleich. Auch hier vereinfachen wir zu $F"ab+abx+-\uparrow c+/-cx+-"$. Bitte testen!

Die konjunktive Normalform für 'ü' lautet:

$$\bar{u} = (a + b + c)x(\bar{a} + \bar{b} + c)x(a + \bar{b} + c)x(a + b + \bar{c})$$

Vereinfacht und ohne negierte Variablen:

$$\bar{u} = \overline{a+b} + a\bar{x}bxc + a\bar{x}b$$

In UPN-Schreibweise: "ab+-abx+-cxabx+".

Es folgt die Gesamtformel in UPN-Schreibweise mit Angabe der Signale bei den einzelnen Symbolen.

	F "a b + - a b x L + - ↑ c + - / c x ↑ + - M / +																						
K S:	1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	17	18	19	20	21	22	23
0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0	1	0	0	0	0
1	1	0	1	0	1	0	0	0	0	1	1	0	1	0	1	0	0	0	0	1	0	0	0
2	0	1	1	0	0	1	0	0	0	1	1	0	1	0	1	0	0	0	0	1	0	0	0
3	1	1	1	0	1	1	1	1	1	0	0	0	0	1	0	0	0	0	1	0	1	0	1
4	0	0	0	1	0	0	0	0	1	0	0	1	1	0	0	1	0	0	0	1	0	0	0
5	1	0	1	0	1	0	0	0	0	1	1	1	1	0	1	1	1	1	1	0	0	1	1
6	0	1	1	0	0	1	0	0	0	1	1	1	1	0	1	1	1	1	1	0	0	1	1
7	1	1	1	0	1	1	1	1	1	0	0	1	1	0	0	1	0	0	0	1	1	0	1

Das Symbol 20 bringt die Signale für den Ausgang 's';
das Symbol 23 liefert die Signale für den Ausgang 'ü'.

Der algebraische Ausdruck kann realisiert werden mit folgenden
Gattern (für die Signal-Nummern):

(1-4) = NOR-, (5-7) = UND-, (9+10) = NOR-, (12-14) = NOR-,
(16+17) = UND-, (19+20) = NOR-, (21-23) = ODER-Gatter.
(Siehe auch Fig. 12 auf Seite 190 im Mikrocomputer-Jahrbuch '84)

Bei umfangreichen schaltalgebraischen Ausdrücken ist es wünschenswert, weitere Ausgangswerte zu speichern. Neben den bereits beschriebenen Symbolen gestattet das Programm mit den Symbolen '2', '5' und '8' drei Ausgangswerte an beliebiger Stelle der Formel festzuhalten und mit den Symbolen '3', '6' und '9' an beliebiger Stelle der Formel zurückzugeben.

Teil II: Logiknetzwerke

Dieser Programmteil wurde entworfen, um die mit Hilfe der Schaltalgebra gefundenen Netzwerke testen zu können. Zum 'Aufbau' von beliebigen logischen Netzwerken stellt das Programm 24 ICs mit je 9 Eingängen und 4 Ausgängen zur Verfügung. Die nicht benutzten 'Anschlüsse' werden vom Programm ignoriert. Jedem IC können UND (U)-, ODER (O)-, NAND (N)- oder NOR (R)-Eigenschaft zugewiesen werden. Zur Negation von einem Signal wird ein 'N'- oder ein 'R'-IC mit einem Eingang verwendet. Die 24 ICs sind mit den Großbuchstaben A - X gekennzeichnet. Ihre Ausgangs-Signale werden in der Reihenfolge A bis X verarbeitet. Dieses ist beim 'Aufbau' der logischen Netzwerke zu beachten. Die globale Marke 'L' ist der Taste FIX zugeordnet. Nach Betätigung von FIX! fragt der Rechner mit 'EING.-VAR.?' nach der Anzahl der vorhandenen Eingangs-Variablen, die mit R/S! einzugeben ist. Dann fragt der Rechner, mit welchen ICs die bezeichnete Eingangs-Variable 'verbunden' ist; die entsprechenden ICs (bis max. 6) werden gemeinsam eingegeben. Anschließend wird nach der Anzahl der Eingänge, der Type und den Verbindungen des Ausgangs gefragt. Z.B. wird mit "B E. TYP A.↑?" nach den Merkmalen des ICs mit der Kennzeichnung 'B' gefragt. Alle 'Daten' werden gemeinsam eingegeben und mit R/S! bestätigt. Ohne die Eingabe einer 'Ausgangs-Verbindung' wird vom Rechner angenommen, daß der IC am Ende des logischen Netzwerkes angeordnet und damit der Aufbau beendet ist.

In diesem Programmteil werden die nachfolgend erläuterten lokalen Marken verwendet.

- B: Eingabe der gewünschten Ausgangswerte (siehe Teil I)
- b: Verändern der Anzahl der Eingangs-Variablen
- C: Kontrolle der einzelnen ICs. Der Rechner erwartet nach "A-X?" die Eingabe einer IC-Kennzeichnung. Die Rückmeldung z.B. "A: 3RECHM" bedeutet: der IC mit der Kennzeichnung A hat 3 Eingänge, wurde als NOR-Glied festgelegt, und sein Ausgang ist verbunden mit den ICs E, C, H und M
- c: Wird betätigt, wenn die 'Daten' eines ICs verändert werden sollen. Im obigen Beispiel lautet die Rückmeldung "A: 3RECHM?". Der Rechner erwartet jetzt eine neue Eingabe mit R/S!
- D: Kontrolle der einzelnen Eingangs-Variablen. Der Rechner erwartet nach "a-e?" die Eingabe einer Variablen. Die Rückmeldung zeigt, mit welchen ICs die Variable verbunden ist
- d: Wird betätigt, wenn die Signale einer Eingangs-Variablen auf andere ICs gelangen sollen
- E: Ausgangswert des Netzwerkes für eine einzelne Kombination
- e: Test eines Netzwerkes bis zu einem gewünschten IC. Die Ausgangswerte des gewählten ICs werden angezeigt. Dadurch kann der Signalfluß überprüft werden
- H: Berechnung und Ausgabe der Ausgangswerte eines Netzwerkes (siehe auch Teil I)
- J: Vergleich der Ausgangswerte eines Netzwerkes mit vorher eingegebenen oder errechneten Ausgangswerten. Stimmen die Ausgangswerte aller Kombinationen überein, wird dieses am Schluß mit "=" gemeldet.

Das Indikationsfeld für die Anzeige des Winkelmodus wird zur Kennzeichnung der beiden 'Programmteile' verwendet. Es gilt: bei Schaltalgebra RAD; bei logischen Netzwerken GRAD.

Nach diesen Vorbemerkungen testen wir einmal eine Verknüpfung mit Exklusiv-ODER-Ausgang in verschiedenen Ausführungsmöglichkeiten. Eine Formel lautet:

$a\bar{b} + \bar{a}b$, in UPN-Schreibweise "ab-xa-bx+". Das entsprechende Netzwerk zeigt B i l d 1. Zur Übung geben wir die Schaltung in den Rechner ein:

Frage	Eingabe	
-	FIX!	
'EING.-VAR.?'	2	R/S!
"a ↑ ?"	"CB"	R/S!
"b ↑ ?"	"AD"	R/S!
"A E. TYP A.↑?"	"1NC"	R/S!
"B E. TYP A.↑?"	"1ND"	R/S!
"C E. TYP A.↑?"	"2UE"	R/S!
"D E. TYP A.↑?"	"2UE"	R/S!
"E E. TYP A.↑?"	"20"	R/S!
'AUFB. KOMPL.'		

Nach Betätigung von H! werden die einzelnen Kombinationen durchlaufen und anschließend die entsprechenden Ausgangssignale genannt: 0110.

Wir wollen versuchen, die negierten Eingangsglieder \bar{a}, \bar{b} 'verschwinden' zu lassen. Mit BEEP! schalten wir auf 'Schaltalgebra' und lassen uns mit C! die konjunktive Normalform nennen: $(+:ab)x$ und $(+:a-b-)x$. In normaler Schreibweise: $(a+b)x(\bar{a} + \bar{b})$. Nach der De Morganschen Regel (siehe 5.3 in Teil I) erhalten wir als Gesamtformel $(a+b) \times \overline{a\bar{b}}$. B i l d 2 zeigt das entsprechende Netzwerk.

Wir schalten mit FIX! wieder um auf logische Netzwerke, 'bauen' das Netzwerk auf und betätigen J!. Der Rechner meldet mit "=" Übereinstimmung. Mit e! testen wir einmal einzeln und erhalten für IC 'A': 0111, für IC 'B': 1110. Durch eine weitere Umformung wird $(a+b) \times \overline{a\bar{b}}$ zu $\overline{a+b} + a\bar{b}$. Das dazugehörige Netzwerk nach B i l d 3 bringt am Ausgang von IC 'A' die Signale 1000 und am Ausgang von IC 'B' 0001. Diese Version kann beim Aufbau eines Volladdierers verwendet werden.

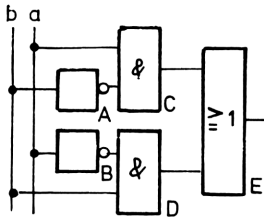


Bild 1 Verknüpfung mit
Exklusiv-ODER-Ausgang

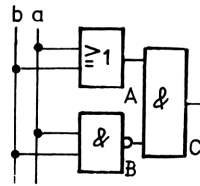


Bild 2 Vereinfachung
von Bild 1

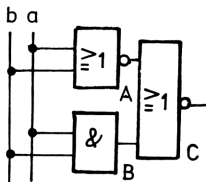


Bild 3 Baustein
zum Aufbau eines
Volladdierers

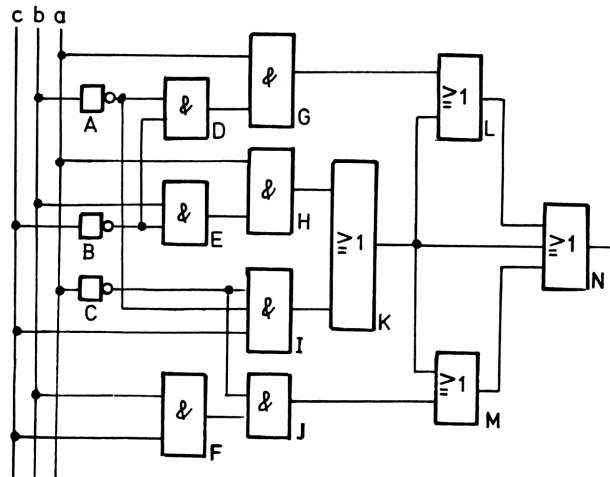


Bild 4 Weiteres Beispiel-Netzwerk

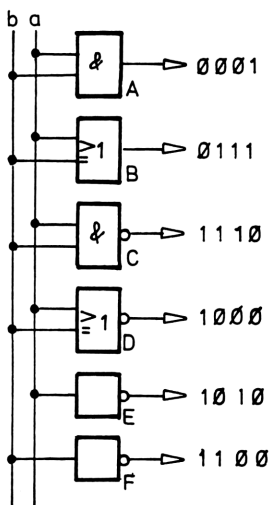


Bild 5 Netzwerk mit 6 Ausgängen

Ist ein Netzwerk optimal entworfen oder kann es noch vereinfacht werden? Diese Frage wollen wir für das Netzwerk nach Bild 4 stellen. Die schaltalgebraische Formel würde uns die Frage beantworten können. Das Aufstellen der Formel jedoch ist nicht gerade einfach. Hier kann das Programm helfen. Nach FIX! bauen wir das Netzwerk auf und erhalten mit H! die Ausgangswerte 01011010. Nach BEEP! und D! und Eingabe von 3 Eingangs-Variablen nennt der Rechner die disjunktive Normalform, die in normaler Schreibweise folgende Form hat:

$a\bar{b}x\bar{c} + a\bar{b}x\bar{c} + \bar{a}x\bar{b}x\bar{c} + \bar{a}x\bar{b}x\bar{c}$.

Durch Ausklammern der Variablen 'b' wird die Formel zu $a\bar{x}\bar{c} + \bar{a}x\bar{c}$. Diese Formel entspricht einem Exklusiv-ODER, welches wir nach Bild 2 oder Bild 3 aufbauen können. Hierbei ist jedoch zu beachten, daß 3 Eingangs-Variablen vorhanden sind und anstelle der Variablen 'b' hier die Variable 'c' tritt. Bei der Frage "b ↑?" wird "↑" eingegeben oder mit R/S! die Frage einfach übersprungen. Nach J! zeigt der Rechner mit "=" Übereinstimmung mit der (künstlich) aufgeblähten Ausgangsschaltung.

Das Netzwerk nach Bild 5 hat 6 Ausgänge. Wenn wir bei IC A nur "2U" eingeben, meldet der Rechner schon hier 'AUFB. KOMPL.'

Damit der Rechner das Netzwerk vollständig übernimmt, wird bei den ICs A-E als Ausgang das Zeichen "↑" eingegeben. Ein Einzeltest mit jeweils e! bringt die eingezeichneten Ausgangssignale.

Anmerkung

Die endgültige Form einer Schaltung liegt natürlich beim Konstrukteur. Es können z.B. Parallelkontakte zur Verringerung der Übergangswiderstände von ihm gewünscht sein, die rein rechnerisch zu einem Einzelkontakt reduziert würden.

ANWEISUNGSLISTE

PRP "BA"	43 RCL 10	87 ASHF	132 2
	44 10	88 ASHF	133 X<Y
01*LBL "BA"	45 +	89 PROMPT	134 Y+X
02 CLX	46 1 E3	90 FS? 17	135 RTN
03 X<Y	47 /	91 GTO 03	136 SF 66
04 CF 29	48 11	92 "FORMEL KOMPL."	
05 FIX 0	49 +	93 PROMPT	137*LBL B
06 103	50 STO 00	94 SF 66	138 65
07 PSIZE	51 FS? 17	95*LBL a	139 PSIZE
08 ΣREG 59	52 11,034	96 RCL 00	140 103
09 "BOOLE:1815-64"	53 STO 00	97 STO 00	141 PSIZE
10 RAD	54*LBL 02	98 FRC	142 XEQ b
11 PROMPT	55 100	99 1 E3	143 16
	56 ATOX	100 *	144 X<Y?
12*LBL F	57 X=Y?	101 10	145 SF 00
13 CF 17	58 95	102 -	146 CLA
14 "BA"	59 X>Y?	103 X>0?	147 FS? 00
15 CLX	60 96	104 ST+ 00	148 "I."
16 SEEKPTA	61 STO IND 00	105 CLA	149*LBL 15
17 CLFL	62 ISG 00	106 RCL 00	150 AON
18 XEQ b	63 GTO 0?	107 10	151 "F AW 0/1?"
19 24	64 FC? 17	108 -	152 PROMPT
20 AON	65 GTO 00	109 ARCL X	153 AOFF
21 "FORMEL?"	66 RCL 00	110 "F : "	154 ALENG
22 PROMPT	67 FRC		155 X>Y?
23 ALENG	68 35	111*LBL 04	156 X<Y
24 STO 10	69 +	112 95	157 STO 00
25 APPREC	70 STO 00	113 RCL IND 00	158 65
26 X*Y?	71 GETREC	114 X=Y?	159 FS?C 01
27 GTO 01	72 GTO 02	115 100	160 01
28 "FERTIG?"		116 96	161 STO 01
29 ASTO Y	73*LBL 00	117 X<Y	
30 PROMPT	74 40	118 X=Y?	162*LBL 05
31 ASTO Y	75 XTOA	119 101	163 1
32 X=Y?	76 ARCL 10	120 XTOA	164 ATOX
33 GTO 00	77 41	121 ISG 00	165 48
34 ALENG	78 XTOA	122 GTO 04	166 -
35 ST+ 10	79 PROMPT	123 PROMPT	167 X<0?
36 APPCHR	80 SF 66	124 SF 66	168 LOG
			169 X>Y?
37*LBL 00	81*LBL A	125*LBL b	170 ACOS
38 CLX	82 CLX	126 5	171 STO IND 01
39 SEEKPT	83 SEEKPT	127 "EING.-VAR.?"	172 ISG 01
40 GETREC		128 PROMPT	173 RCL 00
	84*LBL 03	129 X>Y?	174 DSE 00
41*LBL 01	85 "YYYYYYZZZZ?"	130 ACOS	175 GTO 05
42 AOFF	86 ARCLREC	131 STO 09	176 FS?C 00

177 SF 01	221 CLA	270 GTO 06	315 GTO 00
178 16	222 -31	271 "Fe"	316 AON
179 "II."	223 RCL 07	272 FC? 30	317 " :IC?"
180 FS? 01	224 INT	273 XTOA	318 PROMPT
181 GTO 15	225 ARCL X	274 CF 05	319 AOFF
182 "AW KOMPL."	226 "F :"		320 ATOX
183 PROMPT	227 FS? 08	275*LBL 06	321 64
184 SF 66	228 PROMPT	276 FIX 0	322 -
	229 FS? 06	277 41	323 GTO 01
185*LBL c	230 +	278 XTOA	
186 SF 06	231 ABS	279 FC? 06	324*LBL 08
187 FS? 42	232 OCT	280 "++"	325 " :SYMBOL NR.?"
188 GTO 33	233 LASTX	281 FS? 06	326 PROMPT
	234 8	282 "FX"	327*LBL 01
189*LBL d	235 MOD	283 PROMPT	328 X>Y?
190 SF 04	236 FIX IND X	284*LBL 17	329 SF 66
191 FS? 42	237 CLA	285 ISG 05	330 10
192 GTO 34	238 40	286 ISG 07	331 +
193 XEQ H	239 XTOA	287 GTO 16	332 1 F3
194 GTO 00	240 FC? 06	288 "d/D"	333 /
195*LBL C	241 "FX"	289 FS?C 06	334 11
196 FS? 42	242 FS? 06	290 "c/C"	335 +
197 GTO 33	243 "++"	291 "F KOMPL."	336 STO 00
198 SF 06	244 "F :"	292 PROMPT	
	245 "Fa"	293 SF 66	337*LBL H
199*LBL D	246 CLX		338 SF 02
200 FS? 42	247 FC? 39	294*LBL E	339 XEQ 20
201 GTO 34	248 XTOA	295 31	340 CF 02
202 XEQ b	249 FS?C 01	296 "KOMB.?"	341 RCL 05
203 1	250 GTO 06	297 PROMPT	342 STO 06
204 -	251 "fb"	298 X>Y?	343 FS?C 04
205 1 E3	252 FC? 30	299 ACOS	344 RTH
206 /	253 XTOA	300 STO 07	345 8
207 STO 07	254 FS?C 02	301 SF 07	346 ST/ 04
208 65,1	255 GTO 06	302 XEQ 21	
209 STO 05	256 "fc"	303 "F AW="	347*LBL 18
	257 FC? 37	304 ARCL X	348 " "
210*LBL 00	258 XTOA	305 PROMPT	349 RCL 07
211 RCL 07	259 FS?C 07	306 SF 66	350 8
212 FRC	260 GTO 06		351 X<Y?
213 STO 07	261 "fd"	307*LBL G	352 STO 07
	262 RCL Z	308 SF 01	
214*LBL 16	263 10	309 XEQ 20	353*LBL 07
215 RCL IND 05	264 /	310 CF 01	354 ARCL IND 06
216 FS? 06	265 FIX IND X	311 PROMPT	355 ISG 06
217 XEQ 45	266 CLX		356 DSE 03
218 X=0?	267 FC? 39	312*LBL e	357 GTO 07
219 GTO 17	268 XTOA	313 RCL 10	358 PROMPT
220 SF IND 09	269 FS?C 04	314 FS? 43	

359 STO 03	405 FS? 03	446*LBL 42	485*LBL 51
360 DSE 04	406 STO 62	447*LBL 08	486 RCL 97
361 GTO 18	407 FS? 04	448 *	487 RTN
362 " AW KOMPL. "	408 STO 63	449 RTN	
363 PROMPT	409 RCL 00		488*LBL 53
364 SF 66	410 X<>F	450*LBL 47	489 STO 98
	411 RCL 00	451 +	490 RTN
365*LBL I	412 STO 00	452 SORT	
366 SF 03	413 FS? 42	453 RND	491*LBL 54
367 XEQ 20	414 GTO 26	454 RTN	492 RCL 98
368 CF 03	415*LBL 22	455*LBL 45	493 RTN
369 PROMPT	416 X<> IND 03	456 X=0?	494*LBL 56
370 SF 66	417 STO L	457 GTO 00	495 STO 99
371*LBL J	418 X<> IND 00	458 CLX	496 RTN
372 XEQ 20	419 XEQ IND L	459 RTN	
373 PROMPT	420 ISG 00		497*LBL 57
374 SF 66	421 GTO 22	460*LBL 00	498 RCL 99
	422 FS?C 07	461 SIGN	499 RTN
375*LBL 20	423 RTN	462 RTN	
376 65,1	424 GTO 23		500*LBL 32
377 STO 05		463*LBL 29	501 RTN
378 STO 06	425*LBL 48	464*LBL 61	
379 2	426 0	465 STO 01	502*LBL 23
380 RCL 09	427 RTN	466 RTN	503 FC? 02
381 Y+X			504 GTO 00
382 STO 03	428*LBL 49	467*LBL 63	505 STO IND 06
383 STO 04	429 1	468 RCL 01	506 GTO 24
384 1	430 RTN	469 RTN	
385 -			507*LBL 00
386 1 E3	431*LBL 97	470*LBL 94	508 FC? 03
387 /	432 RCL 59	471 STO 02	509 GTO 00
388 STO 07	433 RTN	472 RTN	510 RCL 01
			511 X*Y?
389*LBL 21	434*LBL 98	473*LBL 47	512 GTO 24
390 "KOMB. "	435 RCL 60	474 RCL 02	513 " ="
391 CLX	436 RTN	475 RTN	514 GTO 00
392 RCL 07			
393 INT	437*LBL 99	476*LBL 76	515*LBL 00
394 ARCL Y	438 RCL 61	477 STO 64	516 FS? 01
395 AVIEW	439 RTN	478 RTN	517 RCL 01
396 X<>F			518 FC? 01
397 STO 00	440*LBL 95	479*LBL 77	519 RCL IND 06
398 1	441 RCL 62	480 RCL 64	520 X=Y?
399 FS? 00	442 RTN	481 RTN	521 GTO 24
400 STO 59			522 " *"
401 FS? 01	443*LBL 96	482*LBL 50	
402 STO 60	444 RCL 63	483 STO 97	523*LBL 00
403 FS? 02	445 RTN	484 RTN	524 "F KOMB. "
404 STO 61			

525 RCL 07	572 65	618 1	664+LBL 30
526 INT	573 STO 04	619 ST+ 04	665 CLA
527 ARCL X	574 CF 23	620 RTN	666 ARCL IND 01
528 RTN			667 ATOX
	575+LBL 19	621+LBL 26	668 48
529+LBL 24	576 CLA	622 ΣREG 75	669 -
530 ISG 06	577 RCL 04	623 CLΣ	670 RCL IND 02
531 ISG 07	578 XTOA	624 ΣREG 41	671 ATOX
532 GTO 21	579 "+ E. TYP A.1?"	625 CLΣ	672 RDN
533 " ="	580 PROMPT	626 ΣREG 47	673 XEQ IND T
534 FS? 03	581 FC?C 23	627 CLΣ	674 X=0?
535 " +"	582 ACOS	628 ΣREG 57	675 GTO 00
536 RTN	583 ALENG	629 CLΣ	676 ALENG
537+LBL "L"	584 2	630 ΣREG 59	677 X=0?
538 CLX	585 X>Y?	631 RCL 09	678 GTO 01
539 X<F	586 ACOS	632 STO 01	679 X<Y
540 FIX 0	587 ASTO IND 00	633 98	680 GTO 00
541 GRAD	588 X=Y?	634 STO 02	681+LBL 01
542 103	589 GTO 00	635 59	682 X<Y
543 PSIZE	590 1	636 STO 97	
544 11.034	591 ST+ 04	637+LBL 27	683+LBL 31
545 STO 00	592 ISG 00	638 CLA	684 ATOX
546 XEQ b	593 GTO 19	639 ARCL IND 02	685 30
547 SF IND 09		640 ALENG	686 -
548 98	594+LBL 00	641 RCL IND 97	687 RDN
549 STO 04	595 AOFF	642 X=0?	688 ST+ IND T
550 AON	596 RCL 00	643 GTO 00	689 DSE Y
551 "a"	597 INT		690 GTO 31
552 XEQ 25	598 10	644+LBL 28	
553 FS?C 01	599 -	645 ATOX	691+LBL 00
554 GTO 00	600 STO 1P	646 30	692 ISG 02
555 "b"	601 LASTX	647 -	693 ISG 01
556 XEQ 25	602 +	648 X<0?	694 GTO 30
557 FS?C 02	603 1 E3	649 GTO 00	695 FS?C 07
558 GTO 00	604 /	650 RDN	696 RTN
559 "c"	605 11	651 ST+ IND T	697 GTO 23
560 XEQ 25	606 +	652 DSE Y	
561 FS?C 07	607 STO 00	653 GTO 28	698+LBL 70
562 GTO 00	608 "AUFB. KOMPL."		699 X=Y?
563 "d"	609 PROMPT	654+LBL 00	700 CLX
564 XEQ 25	610 SF 66	655 1	701 X=Y?
565 FS?C 04		656 ST+ 02	702 1
566 GTO 00	611+LBL 25	657 ST+ 97	
567 "e"	612 CF 23	658 DSE 01	703+LBL 02
568 XEQ 25	613 "+ 1?"	659 GTO 27	704 X=0?
569 CF 05	614 PROMPT	660 RCL 00	705 GTO 00
	615 FC?C 23	661 STO 01	706 CLX
570+LBL 00	616 "A"	662 35.1	707 RTN
571 RDN	617 ASTO IND 04	663 STO 02	

708*LBL 00	723 "A-X?"	739 AON	755 1
709 SIGN	724 PROMPT	740 PROMPT	756 +
710 RTN	725 AOFF	741 FC?C 23	757 ARCL INT X
	726 ATOX	742 ACOS	758 FS?C 04
711*LBL 79	727 XTOA	743 AOFF	759 GTO 00
712 X=0?	728 "F: "	744 ASTO IND X	760 PROMPT
713 1	729 54	745 PROMPT	761 ACOS
714 RTN	730 -		762*LBL 00
	731 ARCL IND X	746*LBL 34	763 CF 23
715*LBL 05	732 FS?C 06	747 AON	764 "F?"
716 X=Y?	733 GTO 00	748 "a-e?"	765 AON
717 CLX	734 PROMPT	749 PROMPT	766 PROMPT
718 X=Y?	735 ACOS	750 AOFF	767 FC?C 23
719 1		751 ATOX	768 ACOS
720 RTN	736*LBL 00	752 XTOA	769 AOFF
721*LBL 33	737 CF 23	753 "F: "	770 ASTO IND X
722 AON	738 "F?"	754 ASTO 00	771 PROMPT
			772 END

Berechnung einer Transistor-Verstärkerstufe (Emitterschaltung)

von Hans-Joachim Soyta

Die Dimensionierung einfacher Transistor-Verstärkerstufen erfordert einen erheblichen Zeitaufwand, zumal der Hobby-Elektro- niker nicht immer die hierfür erforderlichen Formeln auswendig kennt und deshalb erst auf entsprechende Fachliteratur zurück- greifen muß. Dabei zeigt sich, daß gerade die Berechnung der Kondensatoren einen erheblichen Rechenaufwand erfordern.

Das nachstehend beschriebene Programm für den HP-41 ermöglicht die überschlägige Berechnung einfacher Kleinsignal-Transistor- Verstärkerstufen.

1 BESCHREIBUNG

Die Bemessung der Bauteile erfolgt entsprechend der Schaltung nach B i l d 1 (Emitterschaltung). Das hier beschriebene Pro- gramm eignet sich für die Verwendung sowohl mit als auch ohne Drucker. Wer Speicherplatz sparen will und auf Druckerausgabe verzichtet, kann selbstverständlich alle Druckbefehle sowie die Befehle FS? 21 und FC? 21 entfallen lassen. Wem bei der Eingabe kein Drucker zur Verfügung steht, muß alle Druckbefehle mit vorangestelltem "XEQ" eingeben (Ausnahme ADV). Dadurch bleibt die Möglichkeit offen, später einen Drucker anzuschlie- ßen und das Programm in übersichtlicher Form auszudrucken. Das Programm hat in der hier beschriebenen Form einen Umfang von 733 Bytes sowie 20 Speicher. Es benötigt folgende Eingabedaten,

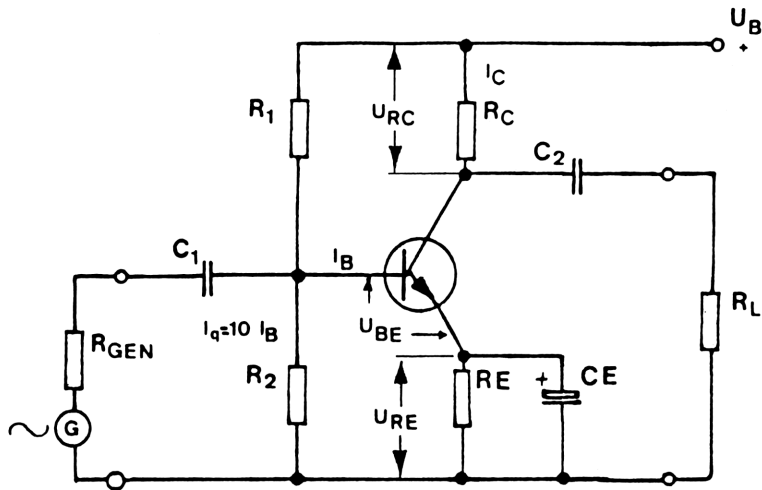


Bild 1 Zugrunde liegende Schaltung

die hier kurz erläutert werden und evtl. als Richtwerte dienen können:

Kollektorgleichstrom I_C

Die Wahl der Größe von I_C hängt vom Verwendungszweck ab und liegt z.B. bei Mikrofonverstärkern zwischen 0,5 mA - 1,5 mA. Das Ansteuern einer etwa nachfolgenden stärkeren Stufe kann bereits einen Strom von 1 - 10 mA erfordern. Relais, mehrere Gatter oder LED-Anzeigen benötigen bereits einen Ausgangsstrom von 10 - 50 mA.

U_{RC}

Der vom Rechner nach Eingabe von U_B (Versorgungsspannung) angezeigte Wert in Volt dient nur als Hinweis zur Ermittlung der Arbeitsgeraden in einem Kennlinienfeld (siehe hierzu B i l d 2). Dabei wird ein Mindestwert der Potentialdifferenz U_{CE} von 0,5 Volt berücksichtigt. Nach dem Betätigen der Taste R/S wird das Programm fortgesetzt.

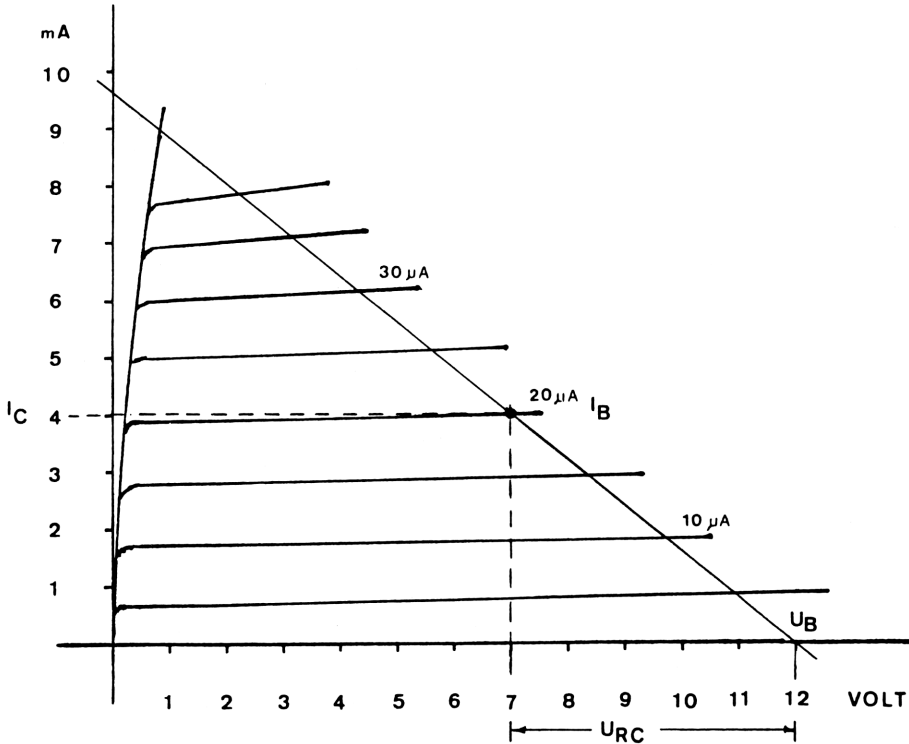


Bild 2 Arbeitsgerade im Kennlinienfeld

Basisstrom I_B

Der Basisstrom steht im Verhältnis

$$I_B = \frac{\text{Kollektorgleichstrom } I_C}{\text{Stromverstärkung } B} \quad \text{z.B.} \quad \frac{4 \text{ mA}}{200} = 0,02 \text{ mA}.$$

Sofern der Stromverstärkungsfaktor B nicht bekannt ist, kann der dazugehörige Basisstrom aus dem Kennlinienfeld des verwendeten Transistors abgelesen werden. (Siehe auch hierzu als Beispiel das entsprechende Kennlinienfeld nach Bild 2.) Wird für I_B kein Wert eingegeben, so erfolgt nach Betätigung der Taste R/S die Aufforderung zur Eingabe des Stromverstärkungs-Faktors: $B = ?$.

Stromverstärkung B

Transistoren werden von den Herstellern nach der Stromverstärkung gruppiert und mit A, B, C gekennzeichnet. Der entsprechende Stromverstärkungsfaktor ist den jeweiligen Kenndaten des verwendeten Transistors zu entnehmen. Der Transistor BC 107 z.B. weist bei einem Kollektorstrom von $I_C = 2 \text{ mA}$ folgende mittlere B-Werte auf: Typ A = 170, Typ B = 290, Typ C = 500. Diese Werte stellen nur Durchschnittswerte dar; Transistoren weisen nämlich erhebliche Exemplarstreuungen auf. Von seiten der Hersteller werden deshalb die Kennwerte durch Angabe des Streubereichs ergänzt. Sofern man die Mühe nicht scheut, kann auch das Verhältnis I_B zu I_C des betreffenden Transistors experimentell selbst ermittelt werden.

Querstrom I_q

Der Querstrom I_q soll etwa 5 - 10mal so groß wie der Basisstrom I_B sein. Je höher dieser Wert gewählt wird, je stabiler wird das Schaltungsverhalten, aber ungünstiger und damit kleiner der Eingangswiderstand. Wird während des Programmablaufs hierfür kein Wert eingegeben, so rechnet das Programm mit einem I_q -Faktor von 10 I_B .

Basis-Emitter-Spannung U_{BE}

Sofern der erforderliche Wert nicht aus dem betreffenden Kennlinienfeld ermittelt werden kann, rechnet das Programm nach R/S automatisch mit 0,60 Volt (für Silizium-Transistoren).

Untere Grenzfrequenz U_{GF}

Für die Ermittlung der Kondensatorwerte ist die untere noch zu verstärkende Frequenz einzugeben. Wird kein Wert hierfür eingegeben, so rechnet das Programm nach Betätigung der Taste R/S mit 50 Hz.

h-Parameter

Die Werte h_{11e} (Eingangswiderstand in Emitterschaltung) und h_{21e} (Stromverstärkung) sind ebenfalls zur Berechnung der Kon-

densatorwerte erforderlich und stehen im Zusammenhang mit dem Kollektorgleichstrom I_C bzw. U_{CE} . Diese Kennwerte aus den Datenblättern gelten nur für einen bestimmten Arbeitspunkt (in der Regel bei U_{CE} 5 Volt und I_C 2 mA). Soll der Transistor jedoch in einem anderen Arbeitspunkt betrieben werden, so sind die entsprechenden Werte umzurechnen.

Formel: $h_{neu} = H_e \times h_{gegeben}$.

Beispiel: Der Transistor BC 107 A weist laut Datenbuch bei $I_C = 2$ mA und $U_{CE} = 5$ Volt folgende dynamische Kennwerte auf:

$$h_{11e} = 2,7 \text{ k}\Omega, h_{21e} = 222.$$

Bei einem Wert von z.B. $I_C = 4$ mA wäre der neue Wert

$$h_{11e} = 0,50 \times 2,7 \text{ k}\Omega = 1,35 \text{ k}\Omega \text{ und}$$

$$h_{21e} = 1,10 \times 222 = 244.$$

Generator-Innenwiderstand R_{GEN}

Für die Berechnung der Kondensatoren ist der Innenwiderstand der Signalquelle einzugeben. Wird kein Wert angegeben, so rechnet das Programm nach R/S intern mit 2,0 k Ω .

Lastwiderstand R_L

Das vorher Gesagte gilt auch hier für den Lastwiderstand R_L . Intern wird mit 10,0 k Ω gerechnet, wenn keine andere Eingabe erfolgt.

Anmerkung:

Bei der Bemessung ist die zulässige maximale Verlustleistung des betreffenden Transistors zu beachten. Eine Überschreitung dieses Grenzwertes verringert die Lebensdauer des Transistors oder/und kann zur Zerstörung führen. Die totale Verlustleistung beträgt:

$$P_{tot} = U_{CE} \times I_C.$$

2 PROGRAMMABLAUF

(Erforderlicher Speicherplatz mindestens SIZE 020)

Wahlschalter des Druckers in Stellung "MAN".

Anzeige	Hinweis	Eingabe/Taste	
1.	Programmstart mit	XEQ	EM
2. UB = ? VOLT	Eingabe der Versorgungsspannung, z.B. 12 Volt	12	R/S
3. URC = 5,00 V	Anzeige des Spannungsabfalls an R_C zur Ermittlung der Arbeitsgeraden, siehe Bild 2		R/S
4. IC = ? MA	Eingabe des gewählten Kollektorstroms in mA, siehe Beispiel Bild 2	4	R/S
5. IB = ? MA	Eingabe des Basisstroms in mA, sofern hier keine Eingabe erfolgt	0,02	R/S R/S
6. B = ?	Sofern keine Eingabe von IB erfolgte, Eingabe des Stromverstärkungsfaktors B	R/S
7. IQ-FAKTOR ?	Eingabe des Querstrom-Faktors I_q , sofern hier keine Eingabe erfolgt (intern wird dann mit 10 gerechnet)	R/S R/S
8. UBE = ? VOLT	Eingabe der gewählten Spannung BE, z.B. sofern keine Eingabe erfolgt (intern wird mit 0,60 V gerechnet)	0,62	R/S R/S
9. UGF = ? HZ	Eingabe der unteren noch zu verstärkenden Grenzfrequenz, z.B. (intern 50 Hz)	40	R/S
10. HE11 ? K-OHM	Eingabe des Eingangswiderstands laut Datenbuch; umgerechnet z.B.	1,35	R/S
11. HE21 ?	Eingabe des Wertes laut Datenbuch; umgerechnet z.B.	244	R/S
12. RGEN = ? K-OHM	Generator-Innenwiderstand der zu verstärkenden Signalquelle, z.B. (intern 2,0 kOhm)	2	R/S
13. RL = ? K-OHM	Eingabe des Lastwiderstands, z.B. (intern 10,0 kOhm)	5	R/S
	Ende der Dateneingabe		

Der Ausdruck erfolgt entsprechend Bild 3. Sofern kein Drucker angeschlossen ist, erfolgt ein STOP nach jedem Ausgabewert.

Die ausgedruckten Werte "K-OHM" und "MF" sind auf die am nächsten liegenden Normgrößen zu runden.

BERECHNUNG EINER EMITTERSCHAL TUNG

UB = 12,00 VOLT	R1 = 44,91 K-OHM
IC = 4,00 MA	R2 = 10,60 K-OHM
IB = 0,020 MA	RC = 1,25 K-OHM
UBE = 0,62 VOLT	RE = 0,38 K-OHM
UGF = 40,00 HZ	
HE11= 1,35 K-OHM	C1 = 2,80 MF
HE21= 244	C2 = 0,69 MF
	CE = 641,45 MF
RGEN= 2,00 K-OHM	
RL = 10,00 K-OHM	

Bild 3 Beispielausdruck

Die zur Berechnung verwendeten Gleichungen

$$R_1 = \frac{U_B - U_{BE} - 1,5}{10^* \cdot I_B + I_B}$$

$$C_2 = \frac{1}{2\pi \cdot 50^* \cdot 0,51 (R_C + R_L)}$$

$$R_2 = \frac{U_{BE} + 1,5}{10^* \cdot I_B}$$

$$C_1 = \frac{1}{2\pi \cdot 50^* \cdot 0,51 (R_{EIN} + R_G)}$$

$$R_C = \frac{U_B - 1,5 - 0,5}{2 \cdot I_C}$$

$$R_E = \frac{U_{RE} (1,5)}{I_C}$$

$$C_E = \frac{H_{21}}{2\pi \cdot 50^* \cdot 0,51 (H_{11} + R_G)}$$

$$R_{EIN} = \frac{1}{\frac{1}{H_{11}} + \frac{1}{R_1} + \frac{1}{R_2}}$$

$$R_G = \frac{1}{\frac{1}{R_{GEN}} + \frac{1}{R_1} + \frac{1}{R_2}}$$

Anmerkung: Die mit * versehenen Werte werden automatisch vom Programm verwendet, sofern hierfür keine Eingaben erfolgen.

3 ANWEISUNGSLISTE

01*LBL "EM"	48 AVIEW	95 STO 03	142 XEQ 09
02*LBL 00	49 STOP	96 CLX	143 STO 07
03 CLRG	50 CLX	97*LBL 04	144 CLA
04 " VOLT"	51 FS? 55	98 CF 21	145 ARCL 16
05 ASTO 15	52 SF 21	99 "IQ-FAKTOR ?"	146 XEQ 10
06 " K-OHM"	53 "IC = "	100 PROMPT	147*LBL 06
07 ASTO 16	54 XEQ 08	101 X=0?	148 CF 29
08 " MF"	55 "I?"	102 10	149 FIX 0
09 ASTO 17	56 ARCL 18	103 STO 19	150 "HE21 = ?"
10 " MA"	57 PROMPT	104 CLA	151 PROMPT
11 ASTO 18	58 CLA	105 CLX	152 X=0?
12 CLA	59 X=0?	106 FS? 55	153 GTO 06
13 FC? 21	60 GTO 01	107 SF 21	154 "HE21 = "
14 GTO 01	61 STO 02	108 FIX 2	155 XEQ 08
15 "BERECHNUNG EINE"	62 XEQ 09	109 "UBE = "	156 STO 08
16 "IR"	63 ARCL 18	110 XEQ 08	157 XEQ 11
17 PRA	64 XEQ 10	111 "I?"	158 SF 29
18 SF 12	65*LBL 02	112 ARCL 15	159 FIX 2
19 "EMITTERSCHAL"	66 FIX 3	113 PROMPT	160 ADV
20 "I-TUNG"	67 "IB = ? MA"	114 CLA	161 "RGEN=?"
21 PRA	68 PROMPT	115 X=0?	162 ARCL 16
22 CF 12	69 X=0?	116 .6	163 PROMPT
23 ADV	70 GTO 03	117 STO 05	164 X=0?
24*LBL 01	71 "IB = "	118 XEQ 09	165 2
25 FIX 2	72 XEQ 08	119 ARCL 15	166 STO 04
26 "UB = "	73 RCL X	120 XEQ 10	167 "RGEN= "
27 XEQ 08	74 XEQ 09	121 "UGF = "	168 XEQ 08
28 "I?"	75 STO 03	122 XEQ 08	169 RCL X
29 ARCL 15	76 CLA	123 "I? HZ"	170 XEQ 09
30 PROMPT	77 ARCL 18	124 PROMPT	171 CLA
31 CLA	78 XEQ 10	125 X=0?	172 ARCL 16
32 X=0?	79 GTO 04	126 50	173 XEQ 10
33 GTO 01	80*LBL 03	127 XEQ 09	174 CLA
34 STO 00	81 CF 29	128 3.2	175 " RL = "
35 XEQ 09	82 FIX 0	129 *	176 XEQ 08
36 ARCL 15	83 "B = ?"	130 STO 06	177 "I?"
37 XEQ 10	84 PROMPT	131 " HZ"	178 ARCL 16
38 CLX	85 X=0?	132 XEQ 10	179 PROMPT
39 RCL 00	86 GTO 02	133*LBL 05	180 CLA
40 2	87 "B = "	134 "HE11=?"	181 X=0?
41 -	88 XEQ 08	135 ARCL 16	182 10
42 2	89 STO 03	136 PROMPT	183 STO 13
43 /	90 XEQ 11	137 X=0?	184 XEQ 09
44 CF 21	91 RCL 03	138 GTO 05	185 ARCL 16
45 "URC = "	92 1/X	139 "HE11= "	186 XEQ 10
46 ARCL X	93 RCL 02	140 XEQ 08	187 CLA
47 "I V"	94 *	141 RCL X	188 ADV

189 ADV	227 *RC = "	265 RCL 11	303 RTN
190 FC? 21	228 XEQ 07	266 RCL 13	304+LBL 08
191 " K."	229 1.5	267 +	305 FS? 21
192 FC? 21	230 RCL 02	268 RCL 06	306 ACA
193 ASTO 16	231 /	269 *	307 RTN
194 RCL 00	232 *RE = "	270 1/X	308+LBL 09
195 1.5	233 XEQ 07	271 1 E3	309 FS? 21
196 -	234 ADV	272 *	310 ACX
197 RCL 05	235 RCL 07	273 *C2 = "	311 RTN
198 -	236 1/X	274 XEQ 12	312+LBL 10
199 ENTER↑	237 RCL 09	275 RCL 07	313 FC? 21
200 RCL 19	238 1/X	276 RCL 14	314 CLX
201 RCL 03	239 +	277 +	315 FC? 21
202 *	240 RCL 10	278 RCL 06	316 RTN
203 RCL 03	241 1/X	279 *	317 ACA
204 +	242 +	280 1/X	318 PRBUF
205 /	243 1/X	281 RCL 08	319 CLX
206 STO 09	244 STO 12	282 *	320 RTN
207 *R1 = "	245 CLX	283 1 E3	321+LBL 11
208 XEQ 07	246 RCL 04	284 *	322 FC? 21
209 1.5	247 1/X	285 *CE = "	323 CLX
210 RCL 05	248 RCL 09	286 XEQ 12	324 FC? 21
211 +	249 1/X	287 ADV	325 RTN
212 RCL 19	250 +	288 ADV	326 ACX
213 /	251 RCL 10	289 ADV	327 PRBUF
214 RCL 03	252 1/X	290 ADV	328 CLX
215 /	253 +	291 ADV	329 RTN
216 STO 10	254 1/X	292 CLA	330+LBL 12
217 *R2 = "	255 STO 14	293 STOP	331 ARCL X
218 XEQ 07	256 RCL 12	294 GT0 00	332 ARCL 17
219 RCL 00	257 +	295+LBL 07	333 AVIEW
220 2	258 RCL 06	296 ARCL X	334 FC? 21
221 -	259 *	297 ARCL 16	335 STOP
222 2	260 1/X	298 AVIEW	336 CLX
223 /	261 1 E3	299 FC? 21	337 CLA
224 RCL 02	262 *	300 STOP	338 .END.
225 /	263 *C1 = "	301 CLA	
226 STO 11	264 XEQ 12	302 CLX	

Die Berechnung der Zahl PI (π)

von Eckard Gehrke

1 DIE ZAHL PI

"Pi, (Ludolphsche Zahl) Bezeichnung für die durch den griechischen Buchstaben π symbolisierte, irrationale, transzendente Zahl, die das konstante Verhältnis des Kreisumfangs ($2\pi r$) zum Durchmesser ($2r$) angibt: $\pi = 3,14159\dots$ " [1].

Keine andere Zahl beschäftigte die Menschheit über Jahrtausende so intensiv wie diese Zahl π .

Im Alten Testament wird die Näherung $\pi = 3$ verwendet: "Und er machte das Meer, gegossen, von einem Rand zum andern zehn Ellen weit rundherum und fünf Ellen hoch und eine Schnur von dreißig Ellen war das Maß ringsherum." (1. Könige 7,23) Die erste bewiesene Approximation für die Zahl π gab Archimedes an. Mit der Näherung des Kreises durch ein- und umbeschriebene Vielecke erhielt er für π : $3\frac{10}{71} < \pi < 3\frac{1}{7}$. In allen Wissenschaften brachte das Mittelalter, das im Englischen treffend mit "dark ages" bezeichnet wird, keine wesentlich neuen Erkenntnisse hervor.

Nachdem die Schriften der großen griechischen Mathematiker wieder entdeckt worden waren, berechnete 1610 Ludolph van Ceulen π mit der Archimedischen Approximation auf 35 Dezimalen genau. Ihm zu Ehren wird π auch als Ludolphsche Zahl bezeichnet. Grundlegende Fortschritte wurden durch die Entdeckung der Infinitesimalrechnung und die Theorie der unendlichen Reihen

ermöglicht. Die Darstellung der zyklometrischen Funktionen durch unendliche Reihen gestattete es, den Wert von π genau zu bestimmen. Nachdem 1767 J.H. Lambert die Irrationalität der Zahl π bewiesen und F. von Lindemann 1882 den Nachweis der Transzendenz von π erbracht hatte, war das Wesen von π endgültig geklärt.

In unserem Jahrhundert berechnete man π mittels elektronischer Digitalrechner. 1958 wurde π in Paris auf 10000 Dezimalen unter Benutzung einer Formel von J. Machin berechnet. Dem Anhang ist eine Kopie des Ausdrucks beigelegt.

2 ZIELSETZUNG DER ARBEIT

Man wird sich fragen, was man mit einem programmierbaren Taschenrechner noch an der Zahl π berechnen will. Sie ist doch schon auf eine Million Dezimalen berechnet. In der Tat sollen mit dem Taschenrechner nur 800 Dezimale von π bestimmt werden.

Der Schwerpunkt der Arbeit liegt auf der Entwicklung des Algorithmus zur Berechnung von π . Es soll ein Algorithmus entworfen werden, der präzise die Berechnung beschreibt und für jeden Rechner als Grundlage dienen kann. In einem zweiten Schritt wird der Algorithmus in ein Programm für den programmierbaren Taschenrechner HP-41CV übersetzt.

Es gibt verschiedene Möglichkeiten der Problemreduktion. In dieser Arbeit wird die Modularisierung verwendet, das Problem wird in mehrere Teilprobleme aufgeteilt, die einzeln bearbeitet werden. Dieses Prinzip gehört zum Konzept der strukturierten Programmierung, das man gewöhnlich nur bei höheren Programmiersprachen verwendet. Anhand dieser Arbeit soll gezeigt werden, daß dieses Konzept auch bei Taschenrechnern anwendbar und äußerst nützlich ist. Die Modularisierung wird nach der "Top-Down"-Methode durchgeführt. Hierzu entwickelt man zunächst eine grobe Gesamtstruktur, die dann soweit verfeinert wird, bis man Moduln erhält, die einfach zu programmieren sind.

3 PROGRAMMBESCHREIBUNG

3.1 Mathematische Grundlagen

Für die Berechnung wird die bereits erwähnte Formel von John Machin benutzt [3], die π erklärt durch

$$\pi = 16 \arctg \frac{1}{5} - 4 \arctg \frac{1}{239} = \sum_{n=0}^{\infty} \frac{(-1)^n}{2n+1} a_n + \sum_{n=0}^{\infty} \frac{(-1)^{n+1}}{2n+1} b_n$$

mit

$$a_n = a_{n-1}/25, a_0 = 16/5, b_n = b_{n-1}/57121, b_0 = 4/239 \text{ und } n \in \mathbb{N}$$

Es sei $\bar{\pi}$ die Näherung für π mit $|\pi - \bar{\pi}| < \epsilon$ ($\epsilon = 10^{-800}$). Aus der Konvergenz der Reihe folgt:

$$\bar{\pi} = \sum_{n=0}^m \frac{(-1)^n}{2n+1} a_n + \sum_{n=0}^l \frac{(-1)^{n+1}}{2n+1} b_n \text{ mit } a_m < \epsilon \text{ und } b_l < \epsilon, m, l \in \mathbb{N}.$$

3.2 Beschreibung des Algorithmus

Es wird mit 3 Registerblöcken gearbeitet (R_1, R_2, R_3). In R_1 werden die einzelnen Folgeglieder der beiden Reihen aufsummiert, in R_2 erfolgt zunächst die Berechnung von a_n und in R_3 von $a_n/(2n+1)$. Ist $a_n < \epsilon$, wird in R_2 b_n berechnet und in R_3 $b_n/(2n+1)$.

Jeder Block besteht aus einer wohlbestimmten Anzahl von Registern, die von der gewünschten Genauigkeit und den Stellen p , die je Register geführt werden, abhängig ist. Bei 10 Stellen je Register und einer Genauigkeit von 800 Dezimalen werden je Block 80 Register benötigt. Möchte man auch die letzte Dezimale sicher angeben, muß man mindestens 2 Dezimale mehr berechnen, je Block ist ein zusätzliches Register erforderlich.

Bei der Durchführung der Grundrechenarten ist zu berücksichtigen, daß anstatt der Basis 10 (Dezimalsystem) die Basis 10^p verwendet wird. Additionen, Subtraktionen und Divisionen (andere Rechenarten werden nicht benötigt) können bei Berücksichtigung des Basiswechsels problemlos durchgeführt werden. Das Grundproblem ist auf 5 Teilprobleme reduziert worden:

Das Hauptprogramm und 4 Unterprogramme (Berechnung eines neuen Folgegliedes NFG $a_n/(2n+1)$ bzw. $b_n/(2n+1)$, sowie ADDition, SUBtraktion und DIVision).

3.2.1 Definition der Variablen

R_{kj} bezeichnet das j -te Register des k -ten Blocks ($k \in \{1,2,3\}$, $j \in \{1,2,\dots,81\}$). Mit F0, F1 und F2 sind 3 Flags definiert. Eine "Flagge" ist entweder gesetzt ($F=1$) oder gelöscht ($F=0$). In Abhängigkeit des Zustands der Flag kann bei der Abfrage eine Entscheidung getroffen werden. DR ist der Divisor, mit dem die Division in dem entsprechenden Unterprogramm durchgeführt wird. Die Variable RE speichert den ganzzahligen Rest, der bei einer Division entsteht. Für $2n+1$ wird die Variable D benutzt. Die Variablen M, N und O dienen als Speicher von Zwischenergebnissen.

Anmerkung:

Im folgenden werden die 5 Algorithmen beschrieben. Aus Platzgründen ist das jeweils dazugehörige Flußdiagramm im Anhang angegeben. Zu jedem Algorithmus sollte man das entsprechende Diagramm vor Augen haben.

3.2.2 Das Hauptprogramm

Es wird zunächst die Reihe $\sum_{n=0}^m (-1)^n a_n/D$ berechnet, Flag 0 wird dazu gesetzt. In einer Iteration wird das im Unterprogramm NFG berechnete Folgeglied a_n/D zur Näherung von π addiert ($n \in \{0,2,4,\dots\}$). Das nächste Folgeglied a_n/D muß aufgrund des alternierenden Faktors $(-1)^n$ subtrahiert werden ($n \in \{1,3,5,\dots\}$). a_n entsteht durch die Division von a_{n-1} mit 25, $R_2(n) = R_2(n-1)/25$. Die Division wird durchgeführt, indem jedes R_{2j} , $j \in \{1,2,\dots,81\}$, $i=1$, durch 25 geteilt wird. Entstehende Reste werden in das nächste Register übernommen. Nach einigen Divisionen $R_2/25$ enthält R_{21} den Wert 0, R_{21} ändert seinen Wert nicht mehr. Bei den folgenden Divisionen $R_{2j}/25$ ist daher $j \in \{2,3,\dots,81\}$, $i=2$. Das nächste auf den Wert 0 zu prüfende Register ist R_{22} .

Wenn $i=81$ und $R_{21}=0$ ist, ist das Abbruchkriterium erfüllt, $a_n < 10^{-810}$.

Die Dezimaldarstellung des ersten Folgegliedes a_n/D lautet: $a_0/1=16/5=3,2$. Aus dem angegebenen Algorithmus ist ersichtlich, daß bereits das erste Glied durch 25 geteilt wird:

$R_2(0)=R_2(-1)/25$. $R_2(-1)$ sei der Wert von R_2 zu Beginn der Berechnungen. Für R_{21} ergibt sich: $R_{21}=16/5 \cdot 25=80$. Mit der Basis 10^{10} erhält man für $R_2(0)$: $R_{21}=3$ und $R_{22}=2 \cdot 10^9$. In R_{21} führt man nur eine Ziffer. Setzt man hingegen $R_{21}=80 \cdot 10^9$, ermöglicht dies, auch in R_{21} 10 Ziffern zu führen.

Die Berechnung der zweiten Reihe erfolgt analog. Zu Beginn ist Flag 0 zu löschen; $D=1$, $R_{21}=9,56 \cdot 10^{11}$, $DR=57121$. Bei der Iteration muß beachtet werden, daß für $n \in \{0,2,4,\dots\}$ die Glieder zu subtrahieren und für $n \in \{1,3,5,\dots\}$ zu addieren sind.

Der letzte Teil des Hauptprogramms ist die Ausgabe. Bevor man die Zahl π , die in den Registern von R_1 gespeichert ist, anzeigt oder ausdruckt, muß die letzte anzugebende Dezimale gerundet werden.

3.2.3 Ein neues Folgeglied

Nach dem Rücksprung (RTN) aus diesem Unterprogramm enthält R_3 ein neues Folgeglied a_n/D (b_n/D). Zunächst wird a_{n-1} (b_{n-1}) durch 25 (57121) dividiert und sowohl in R_2 als auch in R_3 gespeichert. Das Ergebnis der Division a_n/D (b_n/D) wird für die anschließende Addition oder Subtraktion in R_3 festgehalten. Flag 1 wird vor der Berechnung von a_n (b_n) gesetzt und anschließend gelöscht.

3.2.4 Die Addition

Nach der Abarbeitung dieses Unterprogramms ist das zuvor berechnete neue Folgeglied a_n/D (b_n/D) zur letzten Näherung von π , die in R_1 gespeichert war, addiert worden.

Führt man eine Addition zweier Zahlen "zu Fuß" durch, werden die beiden Zahlen entsprechend untereinander geschrieben, zuerst addiert man die beiden Ziffern in der äußersten rechten Spalte und notiert den Einerteil der Summe. Ist die Summe der beiden Ziffern nicht kleiner als 10, merkt man sich den Über-

trag 1 und addiert ihn zur nächsten Spalte. Anderenfalls ist der Übertrag 0. Ein größerer Übertrag ist bei der Addition zweier Ziffern nicht möglich.

Die Addition auf der Basis 10^{10} erfolgt genauso. Zu Beginn der Addition ist kein Übertrag vorhanden, Flag 2 ist zu löschen. Die Addition beginnt mit den jeweiligen äußersten rechten Registern von R_3 und R_1 : R_{3j} , R_{1j} , $j=81$. Zu R_{3j} wird der Übertrag addiert. M speichert den Wert von R_{1j} für die weitere Verwendung.

Ist die Summe $R_{1j} := R_{3j} + R_{1j}$ größer oder gleich 10^{10} , muß ein Übertrag vorgenommen werden, $F2=1$. Führt der Rechner nur p Stellen je Zahl, so wird eine aus p+1 Ziffern bestehende Zahl auf p Ziffern gerundet. Wird nun $R_{1j} := R_{1j} - 10^{10}$ gebildet, kann es zu Rundungsfehlern kommen.

Die Funktionen $\text{frc}(a)$ und $\text{rnd}(a)$ ($a \geq 0$) seien folgendermaßen definiert: $\text{frc}(a)$ schneidet den ganzzahligen Teil von a ab, es verbleibt nur der Dezimalteil. $\text{rnd}(a)$ rundet auf die nächste ganze Zahl. Ist der Dezimalteil von $a < 0,5$ wird ab-, sonst aufgerundet. Die zu behebende Rundung wird von der Einerziffer der Summe der beiden Einerziffern verursacht. Diese Ziffer ist durch $10 \cdot N = 10 \cdot \text{frc}(\text{frc}(R_{1j}/10) + \text{frc}(M/10))$ bestimmt. Setzt man $R_{1j} := R_{1j} + 10 \cdot N - 10 \cdot \text{rnd}(N)$, so wird ein eventueller Rundungsfehler behoben.

Damit R_{3j} bei Aufruf des Unterprogramms NFG ein neues Folgeglied aufnehmen kann, wird $R_{3j}=0$ gesetzt. Der Index j wird um 1 vermindert. Falls $j \geq i-1$ ist, wird die Iteration erneut durchlaufen, sonst erfolgt der Rücksprung. Für $i=1$ erfolgt der Rücksprung bereits bei $j=1$.

3.2.5 Die Subtraktion

Dieses Unterprogramm subtrahiert ein zuvor berechnetes Folgeglied in R_3 von R_1 (R_1).

Wie bei der Addition beginnt auch die Subtraktion mit dem äußersten rechten Register $R_{1j} := R_{1j} - R_{3j} - F2$, $j=81$. F2 speichert

einen eventuellen Übertrag, der zu Beginn der Iteration den Wert 0 hat. Falls bei der Subtraktion R_{1j} negativ wird, wird zu R_{1j} 10^{10} addiert und der Übertrag 1 gesetzt.

Für j gelten die Bedingungen, die bei der Addition formuliert wurden.

3.2.6 Die Division

Mit Hilfe des zweimaligen Aufrufs dieses Unterprogramms wird ein neues Folgeglied a_n/D (b_n/D) berechnet. Ist $DR=25$ ($DR=57121$) wird a_n (b_n) ausgerechnet, beim zweiten Aufruf wird durch D dividiert. Wie bei der Addition und der Subtraktion stützt sich das Verfahren auf das Ausrechnen "zu Fuß". Die Division beginnt mit dem ersten linken Register mit $R_{kj} \neq 0$. In 3.2.2 wurde bereits festgestellt, daß dieses Register R_{ki} ist. Ein eventueller alter Rest muß gelöscht werden. Die Division a_n/D (b_n/D) findet in R_3 statt, daher $k=3$. Vor dem Aufruf der Division $a_{n-1}/25$ ($b_{n-1}/57121$) ist Flag 1 gesetzt. Diese Division wird in R_2 durchgeführt, denn dort wird a_n (b_n) gespeichert, also $k=2$. Flag 1 gestattet es, den Wert von k zu bestimmen ($k=3$; $F1=1 \Rightarrow k=2$).

Bevor die Iteration beschrieben wird, werden zwei Funktionen definiert. $\text{int}(a)$ schneidet den Dezimalteil von a ab. Mit $n \bmod m$ sei eine Funktion erklärt, die den Rest der Division n/m bestimmt. Es sei angemerkt, daß die vier definierten Funktionen für positive Zahlen alle durch die Funktion $\text{int}(a)$ ausgedrückt werden können. Da die Darstellung der Funktionen recht kompliziert wird, wurde zugunsten einer besseren Lesbarkeit der Diagramme darauf verzichtet.

Zunächst wird R_{kj} durch DR dividiert und der ganzzahlige Anteil der Variablen M übergeben. Da der Rest RE von R_{kj-1} stammt, muß er, bevor dividiert wird, mit 10^{10} multipliziert werden. $N = \text{int}(RE \cdot 10^{10}/DR)$. Bevor die Summe aus N und M den neuen Wert von R_{kj} ergibt, muß der Divisionsrest $RE_1 = DR \bmod R_{kj}$ bestimmt werden.

Hier tritt wieder das Problem der Rundung auf. Die Aufrundung wird mit folgendem Kunstgriff behoben: Wenn $RE_1/DR + M - M = 1$ ist, wurde gerundet. Da Reste immer kleiner als Divisoren sind, gilt $RE_1/DR < 1$. Von R_{kj} muß der Wert 1 subtrahiert werden. Die Abrundung stellt kein Problem dar, da der Dezimalteil als ganzzahliger Rest in RE_1 erhalten bleibt.

Analog wird bei $RE_2 = (RE \cdot 10^{10}) \bmod DR$ und N verfahren. Die beiden Reste addieren sich zu einem Rest bezüglich der Division $(RE \cdot 10^{10} + R_{kj})/DR$, $RE = RE_1 + RE_2$. Es gilt $RE_1 < DR$ und $RE_2 < DR$. Die Summe RE kann größer als DR werden. In diesem Fall muß der ganzzahlige Anteil von RE/DR zu R_{kj} addiert werden. Der tatsächliche Rest ergibt sich dann aus $RE := RE \bmod DR$. Wird ein Glied a_n/D (b_n/D) berechnet, so ist die Iteration beendet. Der Index j wird um 1 erhöht, und die Iteration wird erneut durchlaufen. Wenn $j > 81$ ist, wird abgebrochen.

Es wurde schon festgestellt, daß nach der Berechnung von a_n (b_n) in R_2 a_n (b_n) auch in R_3 gespeichert sein muß. Da gerade bei Taschenrechnern auf Rechenzeiten zu achten ist, ist in dem Algorithmus eine Feinheit eingebaut. Ohne Beweis sei behauptet, daß die Division $a_n/25$ abbrechend ist, sie besitzt eine endliche Dezimaldarstellung. Ist $R_{kj} = 0$ und $R_{kj+1} = 0$, kann der Rücksprung aus dem Unterprogramm erfolgen. In Abhängigkeit des Zustands von Flag 0 wird die Entscheidung getroffen, ob a_n oder b_n berechnet wird.

3.3 Das Taschenrechnerprogramm

Für die Berechnung wird der programmierbare Taschenrechner HP-41CV verwendet. Der HP-41 arbeitet mit dem UPN-System, dem eine klammerfreie Darstellung aller Operationen zugrunde liegt. Der HP-41 kann keine Variablen definieren. Er besitzt numerierte Speicher. Ein Aufruf wird mit RCL nm vorgenommen, mit a STO nm wird die Zahl a in Register nm gespeichert. Für die verwendeten Variablen ergibt sich: R 00 nimmt i auf, R 01 und R 02 werden für den Schleifenzähler j benötigt. D wird in R 03 gespeichert und DR in R 04. Die folgenden 81 Speicher R 05 - R 85 bilden R_1 . In diesen Registern werden die Folgeglieder zu 1

aufsummiert, die Register R 86 - R 166 (R_2) nehmen a_n (b_n) auf, die Division mit D wird in den Registern R 167 - R 247 abgewickelt. Die Adressierung dieser Register erfolgt indirekt mit R 01 und R 02. Bei den Unterprogrammen Addition und Subtraktion werden die Register von R_1 mit RCL IND 01 aufgerufen, die von R_3 mit RCL IND 02. Die Berechnung von R 01 und R 02 erfolgt in dem Unterprogramm Schleifenzähler. Für die Division wird nur R 01 benötigt. Das Register R 0 (]) nimmt den Rest RE auf. R M ([) und R N (\) sind für M und N bestimmt.

Der HP-41 kann nur zu Marken ("Labels") springen. Diese sind in den Diagrammen mit Kreisen angegeben. Für die Marken NFG, ADD, DIV und SUB werden die Marken 02, 03, 06 und A benutzt. Unterprogramme werden mit XEQ ausgeführt. Bei einem RTN kehrt der Rechner in die Zeile zurück, die dem Unterprogrammaufruf folgt. Einfache Sprünge erfolgen mit GT0.

Bei Fragen, die mit "nein" beantwortet werden, überspringt der Rechner eine Zeile. Die Schleifensteuerung erfolgt mit ISG und DSE. Für i ergibt sich: R 00 hat den Startwert 1,081 (a,b). Kommt der Rechner zu einer ISG-Anweisung, wird a um 1 erhöht: 2,081. Ist $a > b$, überspringt der Rechner eine Zeile. Bei einer DSE-Anweisung wird a um 1 vermindert. Falls $a \leq b$ ist, wird eine Zeile übersprungen.

Mit Hilfe der Zeilen 02 - 05 zeigt der Rechner während der Berechnung "PI=?" im Display an. Der Rest des Programms läßt sich mit den Diagrammen und den Bemerkungen anhand des kommentierten Programmausdrucks nachvollziehen. Um Speicherplatz während der Berechnung zu sparen, wurde das Ausgabeprogramm ausgegliedert. Zunächst wird die letzte Ziffer gerundet, mit LBL 00 beginnt die Ausgabe. Ein Diagramm wird dafür nicht angegeben.

3.3.1 Die Bedienung des Programms

Mit SIZE 248 werden die Speicherregister reserviert. Das Anzeigeformat muß auf FIX 0 gesetzt werden. Mit XEQ^TPI wird das Programm gestartet. In den folgenden 33,4 Stunden berechnet der HP-41 800 Dezimale von π . Dazu werden 580 Folgeglieder a_n/D und 180 Folgeglieder b_n/D berechnet. Der Rechner schaltet sich an-

schließend aus. Nach dem Einschalten wird mit SIZE 087 Speicherplatz für das Ausgabeprogramm geschaffen. Nach dem Einlesen wird es mit XEQ^TAG gestartet. Bei eingeschaltetem Drucker erhält man folgendes Ergebnis:

¶ auf 800 Dezimale genau

$\pi =$		
3,141592653	0454326648	7190702179
5897932384	2133936072	8609437027
6264338327	6024914127	7053921717
9502884197	3724587006	6293176752
1693993751	6063155881	3846748184
0582097494	7488152092	6766940513
4592307816	0962829254	2000568127
4062862089	0917153643	1452635608
9862803482	6789259036	2778577134
5342117067	0011330530	2757789609
9821480865	5488204665	1736371787
1328230664	2138414695	2146844090
7093844609	1941511609	1224953430
5505822317	4330572703	1465495853
2535940812	6575959195	7105079227
8481117450	3092186117	9689258923
2841027019	3819326117	5420199561
3852110555	9310511854	1212902196
9644622948	8074462379	0864034418
9549303819	9627495673	1598136297
6442881097	5188575272	7477130996
5665933446	4891227938	0518707211
1284756482	1830119491	3499999983
3378678316	2983367336	7297804995
5271201909	2440656643	1059731732
1456485669	0860213949	8160963186
2346034861	4639522473	

4 DISKUSSION DER ERGEBNISSE

Ein Vergleich dieser 800 Dezimalen mit denen, die in [3] angegeben sind, bestätigt die Richtigkeit der ausgedruckten Dezimalen. Die 800. Dezimale wurde korrekt auf 6 gerundet. Der entwickelte Algorithmus gestattet es, ¶ entsprechend dem vorhandenen Speicherplatz beliebig genau zu berechnen. Er ist prinzipiell auch für größere Rechneranlagen geeignet. Würde man den modularen Aufbau des Algorithmus auch im Rechner verwirklichen, könnte man mit dem HP-41 ca. 1300 Dezimale berechnen. Mit der Ausgliederung des Ausgabeprogramms wurde dies angedeutet. Die-

sen Prozeß weiterzuführen, hat zur Folge, daß während der Berechnung der Rechner dauernd überwacht werden muß (Einlesen von Magnetkarten). Deshalb wurde darauf verzichtet. Würde man alle Möglichkeiten des HP-41-Systems nutzen, wäre es meiner Meinung nach möglich, π mit dem angegebenen Programm auf ca. 3000 Dezimalen genau zu berechnen. Hierzu müßte man das Programm in einem Erweiterungsmodul zur Verfügung haben.

Zum Schluß sei noch etwas über die Vorteile der strukturierten Programmierung gesagt. Alle Funktionswerte von Funktionen, die man nur mit Additionen, Subtraktionen und Divisionen definieren kann, lassen sich beliebig genau ausrechnen. Lediglich die benutzten Registeradressen müssen den neuen Verhältnissen angepaßt werden.

Mit Hilfe des Unterprogramms "Division" lassen sich alle Brüche p/q $p, q \in \mathbb{N}$ in beliebig genauer Dezimaldarstellung angeben. Eine zweite transzendente Zahl läßt sich nun sehr leicht berechnen. Die Zahl e , die Basis des natürlichen Logarithmus, auch Eulersche Zahl genannt, ist mit folgender Definition leicht zu berechnen:

$$e = \sum_{n=0}^{\infty} 1/n! \quad \text{mit } n! = 1 \cdot 2 \cdot 3 \cdot \dots \cdot (n-1) \cdot n \text{ und } 0! = 1.$$

In anderer Form ergibt sich:

$$e = 1 + \sum_{n=1}^{\infty} a_n \quad \text{mit } a_n = a_{n-1}/n \text{ und } a_0 = 1.$$

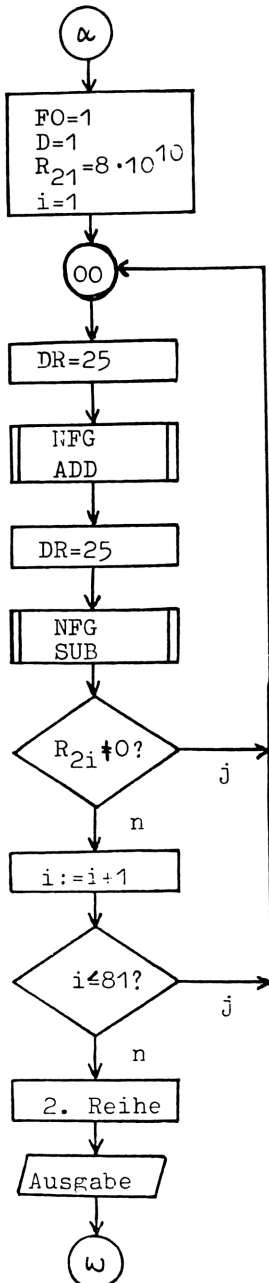
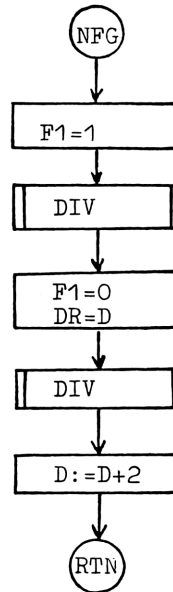
Zur Berechnung werden nur zwei Registerblöcke benötigt: R_1 und R_2 . In R_1 wird die Näherung von e gespeichert. Die Folgeglieder a_n werden in R_2 berechnet. Es ist ersichtlich, daß nur die Unterprogramme "Addition" und "Division" gebraucht werden. Im Anhang ist ein Programm angegeben, das e mit dem HP-41 auf 3000 Dezimalen genau berechnet.

Die Verbindung der beiden transzendenten Zahlen liefert die Formel:

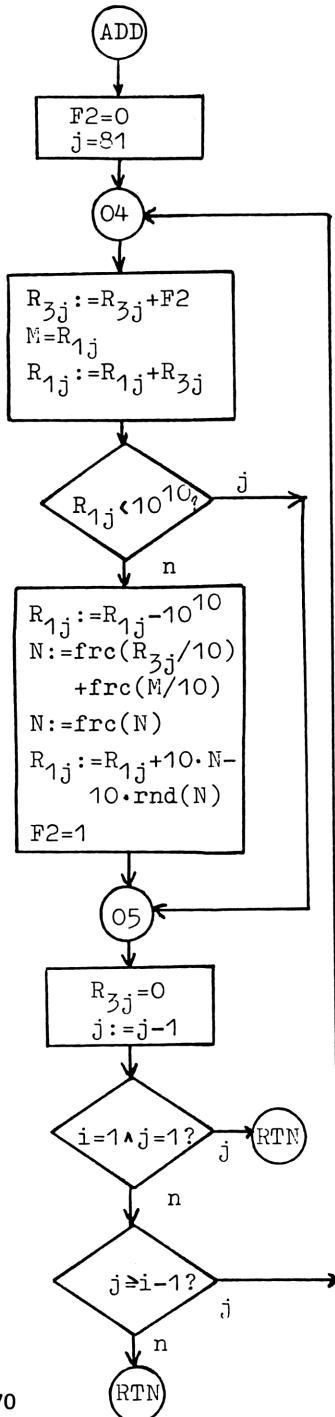
$e^{i\pi} = -1.$

5 ANHANG

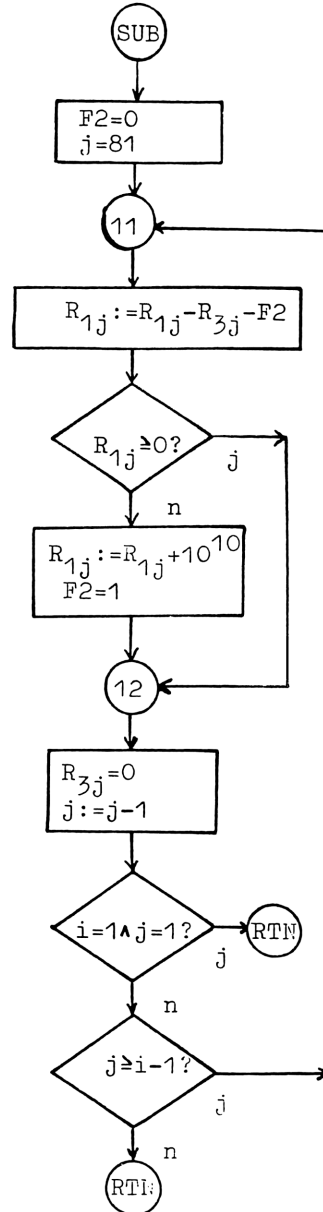
5.1 Flußdiagramme

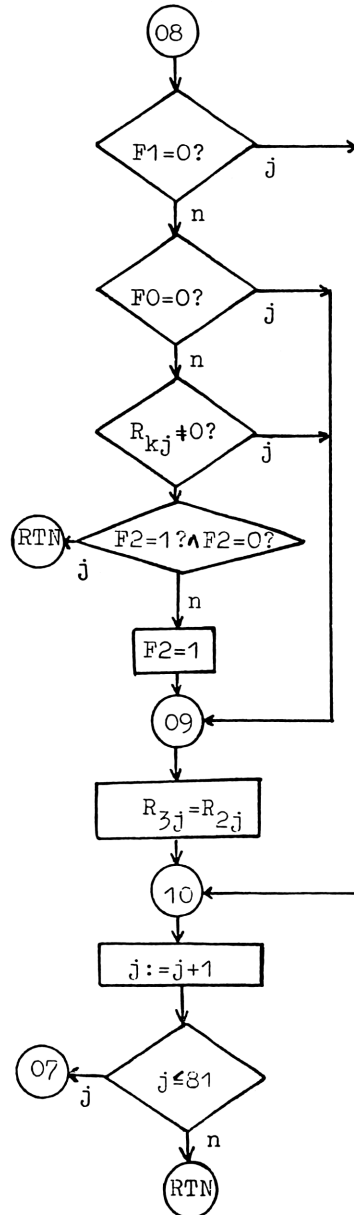
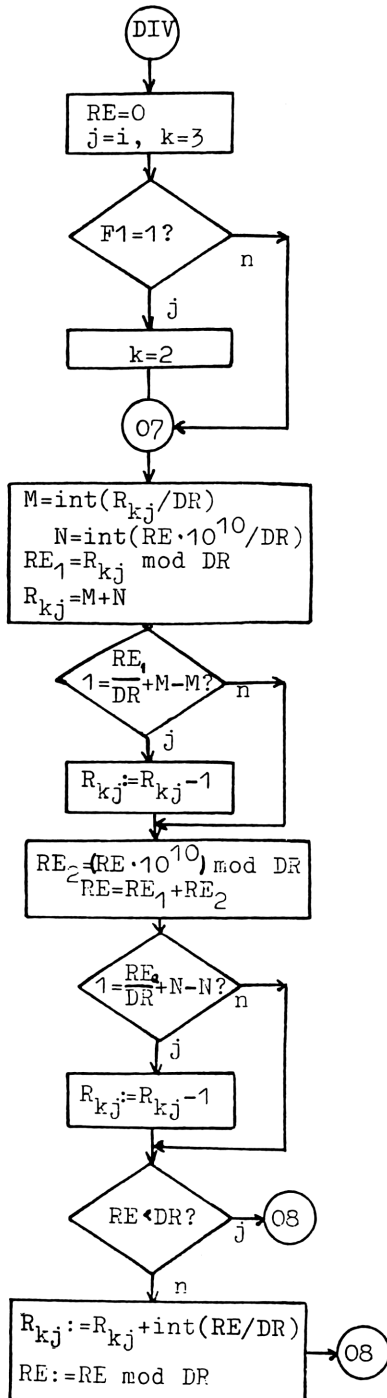
Das HauptprogrammEin neues Folgeglied

Die Addition



Die Subtraktion



Die Division

5.2 Programmausdrucke

Eckard Gehrke Schlesierstr. 19 6704 Mutterstadt	Eingabe: 2. Reihe	Addition	Division
	31 CF 00	66*LBL 03	111*LBL 06
	32 1	67 CF 02	112 CLA
	33 STO 03	68 XEQ 8	113 166.166
Die Zahl π	34 9.56 E11	69*LBL 04	114 RCL 00
	35 STO 06	70 0	115 FS? 01
	36 1.081	71 FS?C 02	116 85.085
Eingabe: 1. Reihe	37 STO 00	72 1	117 +
		73 ST+ IND 02	118 STO 01
		74 RCL IND 02	119*LBL 07
		75 RCL IND 01	120 RCL IND 01
		76 STO I	121 RCL 04
		77 +	122 /
01*LBL "PI"		78 STO IND 01	123 INT
02 "PI=?"	Verarbeitung	79 E10	124 STO I
03 RCL d		80 X>Y?	125 RCL J
04 AVIEW	38*LBL 01	81 GTO 05	126 E10
05 STO d	39 57121	82 ST- IND 01	127 *
06 CLRG	40 STO 04	83 RCL IND 02	128 RCL 04
07 SF 00	41 XEQ 02	84 10	129 /
08 1	42 XEQ A	85 /	130 INT
09 STO 03	43 57121	86 FRC	131 STO \
10 8 E10	44 STO 04	87 RCL I	132 +
11 STO 06	45 XEQ 02	88 10	133 X<> IND 01
12 1.081	46 XEQ 03	89 /	134 RCL 04
13 STO 00	47 RCL 00	90 FRC	135 MOD
	48 85	91 +	136 STO Z
	49 +	92 FRC	137 RCL 04
Verarbeitung	50 RCL IND X	93 ENTER†	138 /
	51 X*0?	94 RND	139 RCL I
	52 GTO 01	95 10	140 +
14*LBL 00	53 ISG 00	96 *	141 LASTX
15 25	54 GTO 01	97 X<>Y	142 -
16 STO 04	55 OFF	98 10	143 1
17 XEQ 02		99 *	144 X=Y?
18 XEQ 03	Ende des Hauptprogramms	100 -	145 ST- IND 01
19 25		101 ST- IND 01	146 RCL T
20 STO 04		102 SF 02	147 X<> J
21 XEQ 02	Unterprogramme	103*LBL 05	148 E10
22 XEQ A		104 1	149 *
23 RCL 00		105 ST- 01	150 RCL 04
24 85	Neues Folgeglied	106 0	151 MOD
25 +		107 STO IND 02	152 ST+ J
26 RCL IND X		108 DSE 02	153 RCL 04
27 X*0?	56*LBL 02	109 GTO 04	154 /
28 GTO 00	57 SF 01	110 RTN	155 RCL \
29 ISG 00	58 XEQ 06		156 +
30 GTO 00	59 CF 01		157 LASTX
	60 RCL 03		158 -
	61 STO 04		159 1
	62 XEQ 06		160 X=Y?
	63 2		
	64 ST+ 03		
	65 RTN		

161 ST- IND 01	196 FC? 02	Ausgabeprogramm	
162 RCL J	197 0		
163 RCL 04	198 FS?C 02		
164 X>Y?	199 1	01*LBL "AG"	
165 GTO 08	200 -	02 RCL 05	
166 MOD	201 RCL IND 02	03 1 E9	
167 STO J	202 -	04 /	29 ARCL IND T
168 1	203 STO IND 01	05 INT	30 RCL I
169 ST+ IND 01	204 0	06 4	31 0
170*LBL 08	205 X<=Y?	07 X>Y?	32 STO I
171 FC? 01	206 GTO 12	08 GTO 00	33 "I++++"
172 GTO 10	207 E10	09 1	34 STO I
173 FC? 00	208 ST+ IND 01	10 ST+ 84	35 "I+++"
174 GTO 09	209 SF 02	11*LBL 00	36 RCL I
175 RCL IND 01	210*LBL 12	12 CF 28	37 CLA
176 X=0?	211 1	13 FIX 9	38 STO I
177 GTO 09	212 ST- 01	14 CLA	39 "I++++++"
178 FS?C 02	213 0	15 RCL 05	40 X<> Z
179 RTN	214 STO IND 02	16 1 E9	41 STO I
180 SF 02	215 DSE 02	17 /	42 AVIEW
181*LBL 09	216 GTO 11	18 ARCL X	43 ISG T
182 RCL 01	217 RTN	19 AVIEW	44 GTO 01
183 81		20 FIX 0	45 CLST
184 +	Schleifenzähler	21 CF 29	46 END
185 RCL IND 01		22 6.084	
186 STO IND Y		23 STO T	
187*LBL 10	218*LBL B	24*LBL 01	
188 ISG 01	219 RCL 00	25 RCL T	
189 GTO 07	220 INT	26 STO T	
190 RTN	221 E3	27 CLA	
	222 /	28 "0000"	
Subtraktion	223 85.003		
	224 +		
	225 STO 01		
191*LBL A	226 162.162		
192 CF 02	227 +		
193 XEQ B	228 STO 02		
194*LBL 11	229 END		
195 RCL IND 01			

5.3 10000 Dezimale von π

Nach Francois Genuys, IBM Frankreich, Paris [3]

```

= 3,14159 26535 89793 23846 26433 83279 50288 41971 69399 37510
   58209 74944 59230 78164 06286 20899 86280 34825 34211 70679
   82148 08651 32823 06647 09384 46095 50582 23172 53594 08128
   48111 74502 84102 70193 85211 05559 64462 29489 54930 38196
   44289 10975 66593 34461 28475 64823 37867 83165 27120 19091

   45648 56692 34603 48610 45432 66482 13393 60726 02491 41273
   72458 70066 06315 58817 48815 20920 96282 92540 91715 36436
   78925 90360 01133 05305 48820 46652 13841 46951 94151 16094
   33057 27036 57595 91953 09218 61173 81932 61179 31051 18548
   07446 23799 62749 56735 18857 52724 89122 79381 83011 94912

   98336 73362 44065 66430 86021 39494 63952 24737 19070 21798
   60943 70277 05392 17176 29317 67523 84674 81846 76694 05132
   00056 81271 45263 56082 77857 71342 75778 96091 73637 17872
   14684 40901 22495 34301 46549 58537 10507 92279 68925 89235
   42019 95611 21290 21960 86403 44181 59813 62977 47713 09960

   51870 72113 49999 99837 29780 49951 05973 17328 16096 31859
   50244 59455 34690 83026 42522 30825 33446 85035 26193 11881
   71010 00313 78387 52886 58753 32083 81420 61717 76691 47303
   59825 34904 28755 46873 11595 62863 88235 37875 93751 95778
   18577 80532 17122 68066 13001 92787 66111 95909 21642 01989

   38095 25720 10654 85863 27886 59361 53381 82796 82303 01952
   03530 18529 68995 77362 25994 13891 24972 17752 83479 13151
   55748 57242 45415 06959 50829 53311 68617 27855 88907 50983
   81754 63746 49393 19255 06040 09277 01671 13900 98488 24012
   85836 16035 63707 66010 47101 81942 95559 61989 46767 83744

   94482 55379 77472 68471 04047 53464 62080 46684 25906 94912
   93313 67702 89891 52104 75216 20569 66024 05803 81501 93511
   25338 24300 35587 64024 74964 73263 91419 92726 04269 92279
   67823 54781 63600 93417 21641 21992 45863 15030 28618 29745
   55706 74983 85054 94588 58692 69956 90927 21079 75093 02955

   32116 53449 87202 75596 02364 80665 49911 98818 34797 75356
   63698 07426 54252 78625 51818 41757 46728 90977 77279 38000
   81647 06001 61452 49192 17321 72147 72350 14144 19735 68548
   16136 11573 52552 13347 57418 49468 43852 33239 07394 14333
   45477 62416 86251 89835 69485 56209 92192 22184 27255 02542

   56887 67179 04946 01653 46680 49886 27232 79178 60857 84383
   82796 79766 81454 10095 38837 86360 95068 00642 25125 20511
   73929 84896 08412 84886 26945 60424 19652 85022 21066 11863
   06744 27862 20391 94945 04712 37137 86960 95636 43719 17287
   46776 46575 73962 41389 08658 32645 99581 33904 78027 59009

```

94657	64078	95126	94683	98352	59570	98258	22620	52248	94077
26719	47826	84826	01476	99090	26401	36394	43745	53050	68203
49625	24517	49399	65143	14298	09190	65925	09372	21696	46151
57098	58387	41059	78659	59772	97549	89301	61753	92846	81382
68683	86894	27741	55991	85592	52459	53959	43104	99725	24680
84598	72736	44695	84865	38367	36222	62609	91246	08051	24388
43904	51244	13654	97627	80797	71569	14359	97700	12961	60894
41694	86855	58484	06353	42207	22258	28488	64815	84560	28506
01684	27394	52267	46767	88952	52138	52254	99546	66727	82398
64565	96116	35488	62305	77456	49803	55936	34568	17432	41125
15076	06947	94510	96596	09402	52288	79710	89314	56691	36867
22874	89405	60101	50330	86179	28680	92087	47609	17824	93858
90097	14909	67598	52613	65549	78189	31297	84821	68299	89487
22658	80485	75640	14270	47755	51323	79641	45152	37462	34364
54285	84447	95265	86782	10511	41354	73573	95231	13427	16610
21359	69536	23144	29524	84937	18711	01457	65403	59027	99344
03742	00731	05785	39062	19838	74478	08478	48968	33214	45713
86875	19435	06430	21845	31910	48481	00537	06146	80674	91927
81911	97939	95206	14196	63428	75444	06437	45123	71919	21799
98391	01591	95618	14675	14269	12397	48940	90718	64942	31961
56794	52080	95146	55022	52316	03881	93014	20937	62137	85595
66389	37787	08303	90697	92077	34672	21825	62599	66150	14215
03068	03844	77345	49202	60541	46659	25201	49744	28507	32518
66600	21324	34088	19071	04863	31734	64965	14539	05796	26856
10055	08106	65879	69981	63574	73638	40525	71459	10289	70641
40110	97120	62804	39039	75951	56771	57700	42033	78699	36007
23055	87631	76359	42187	31251	47120	53292	81918	26186	12586
73215	79198	41484	88291	64470	60957	52706	95722	09175	67116
72291	09816	90915	28017	35067	12748	58322	28718	35209	35396
57251	21083	57915	13698	82091	44421	00675	10334	67110	31412
67111	36990	86585	16398	31501	97016	51511	68517	14376	57618
35155	65088	49099	89859	98238	73455	28331	63550	76479	18535
89322	61854	89632	13293	30898	57064	20467	52590	70915	48141
65498	59461	63718	02709	81994	30992	44889	57571	28289	05923
23326	09729	97120	84433	57326	54893	82391	19325	97463	66730
58360	41428	13883	03203	82490	37589	85243	74417	02913	27656
18093	77344	40307	07469	21120	19130	20330	38019	76211	01100
44929	32151	60842	44485	96376	69838	95228	68478	31235	52658
21314	49576	85726	24334	41893	03968	64262	43410	77322	69780
28073	18915	44110	10446	82325	27162	01052	65227	21116	60396
66557	30925	47110	55785	37634	66820	65310	98965	26918	62056
47693	12570	58635	66201	85581	00729	36065	98764	86117	91045
33488	50346	11365	76867	53249	44166	80396	26579	78771	85560
84552	96541	26654	08530	61434	44318	58676	97514	56614	06800
70023	78776	59134	40171	27494	70420	56223	05389	94561	31407

11270	00407	85473	32699	39081	45466	46458	80797	27082	66830
63432	85878	56983	05235	80893	30657	57406	79545	71637	75254
20211	49557	61581	40025	01262	28594	13021	64715	50979	25923
09907	96547	37612	55176	56751	35751	78296	66454	77917	45011
29961	48903	04639	94713	29621	07340	43751	89573	59614	58901
93897	13111	79042	97828	56475	03203	19869	15140	28708	08599
04801	09412	14722	13179	47647	77262	24142	54854	54033	21571
85306	14228	81375	85043	06332	17518	29798	66223	71721	59160
77166	92547	48738	98665	49494	50114	65406	28433	66393	79003
97692	65672	14638	53067	36096	57120	91807	63832	71664	16274
88880	07869	25602	90228	47210	40317	21186	08204	19000	42296
61711	96377	92133	75751	14959	50156	60496	31862	94726	54736
42523	08177	03675	15906	73502	35072	83540	56704	03867	43513
62222	47715	89150	49530	98444	89333	09634	08780	76932	59939
78054	19341	44737	74418	42631	29860	80998	88687	41326	04721
56951	62396	58645	73021	63159	81931	95167	35381	29741	67729
47867	24229	24654	36680	09806	76928	23828	06899	64004	82435
40370	14163	14965	89794	09243	23789	69070	69779	42236	25082
21688	95738	37986	23001	59377	64716	51228	93578	60158	81617
55782	97352	33446	04281	51262	72037	34314	65319	77774	16031
99066	55418	76397	92933	44195	21541	34189	94854	44734	56738
31624	99341	91318	14809	27777	10386	38773	43177	20754	56545
32207	77092	12019	05166	09628	04909	26360	19759	88281	61332
31666	36528	61932	66863	36062	73567	63035	44776	28035	04507
77235	54710	58595	48702	79081	43562	40145	17180	62464	36267
94561	27531	81340	78330	33625	42327	83944	97538	24372	05835
31147	71199	26063	81334	67768	79695	97030	98339	13077	10987
04085	91337	46414	42822	77263	46594	70474	58784	77872	01927
71528	07317	67907	70715	72134	44730	60570	07334	92436	93113
83504	93163	72840	42512	19256	51798	06941	13528	01314	70130
47816	43798	51852	90928	54520	11658	39341	96562	13491	43415
95625	36586	55705	52690	49652	09858	03385	07224	26482	93972
85847	83163	05777	75606	88876	44624	82468	57926	03953	52773
48030	48029	00587	60758	25104	74709	16439	61362	67604	49256
27420	42083	20956	61190	62545	43372	13153	59584	50687	72460
29016	18766	79524	06163	42522	57719	54291	62991	93064	55377
99140	37340	43287	52628	88963	99587	94757	29174	64263	57455
25407	90914	51357	11136	94109	11939	32519	10760	20825	20261
87985	31887	70584	29725	91677	81314	96990	09019	21169	71737
27847	68472	68608	49003	37702	42429	16513	00500	51683	23364
35038	95170	29893	92233	45172	20138	12806	96501	17844	08745
19601	21228	59937	16231	30171	14448	46409	03890	64495	44400
61986	90754	85160	26327	50529	83491	87407	86680	88183	38510
22833	45085	04860	82503	93021	33219	71551	84306	35455	00766
82829	49304	13776	55279	39751	75461	39539	84683	39363	83047

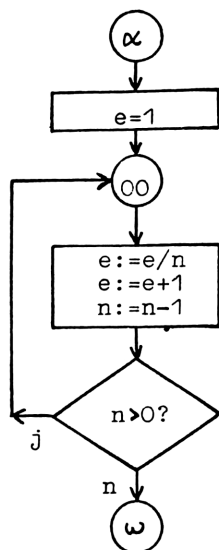
46119	96653	85815	36420	56853	38621	86725	23340	28308	71123
28278	92125	07712	62946	32295	63989	89893	58211	67456	27010
21835	64622	01349	67151	88190	97303	81198	00497	34072	39610
36854	06643	19395	09790	19069	96395	52453	00545	05806	85501
95673	02292	19139	33918	56803	44903	98205	95510	02263	53536
19204	19947	45538	59381	02343	95544	95977	83779	02374	21617
27111	72364	34354	39478	22181	85286	24085	14006	66044	33258
88569	86705	43154	70696	57474	58550	33232	33421	07301	54594
05165	53790	68662	73337	99585	11562	57843	22988	27372	31989
87571	41595	78111	96358	33005	94087	30681	21602	87649	62867
44604	77464	91599	50549	73742	56269	01049	03778	19868	35938
14657	41268	04925	64879	85561	45372	34786	73303	90468	83834
36346	55379	49864	19270	56387	29317	48723	32083	76011	23029
91136	79386	27089	43879	93620	16295	15413	37142	48928	30722
01269	01475	46684	76535	76164	77379	46752	00490	75715	55278
19653	62132	39264	06160	13635	81559	07422	02020	31872	77605
27721	90055	61484	25551	87925	30343	51398	44253	22341	57623
36106	42506	39049	75008	65627	10953	59194	65897	51413	10348
22769	30624	74353	63256	91607	81547	81811	52843	66795	70611
08615	33150	44521	27473	92454	49454	23682	88606	13408	41486
37767	00961	20715	12491	40430	27253	86076	48236	34143	34623
51897	57664	52164	13767	96903	14950	19108	57598	44239	19862
91642	19399	49072	36234	64684	41173	94032	65918	40443	78051
33389	45257	42399	50829	65912	28508	55582	15725	03107	12570
12668	30240	29295	25220	11872	67675	62204	15420	51618	41634
84756	51699	98116	14101	00299	60783	86909	29160	30288	40026
91041	40792	88621	50784	24516	70908	70006	99282	12066	04183
71806	53556	72525	32567	53286	12910	42487	76182	58297	65157
95984	70356	22262	93486	00341	58722	98053	49896	50226	29174
87882	02734	20922	22453	39856	26476	69149	05562	84250	39127
57710	28402	79980	66365	82548	89264	89025	45661	01729	67026
64076	55904	29099	45681	50652	65305	37182	94127	03369	31378
51786	09040	70866	71149	65583	43434	76933	85781	71138	64558
73678	12301	45876	87126	60349	91390	95620	09939	36103	10291
61615	28813	84379	09904	23174	73363	94804	57593	14931	40529
76347	57481	19356	70911	01377	51721	00803	15590	24853	09066
92037	67192	20332	29094	33467	68514	22144	77379	39375	17034
43661	99104	03375	11173	54719	18550	46449	02636	55128	16228
82446	25759	16333	03910	72253	83742	18214	08835	08657	39177
15096	82887	47826	56995	99574	49066	17583	44137	52239	70968
34080	05355	98491	75417	38188	39994	46974	86762	65516	58276
58483	58845	31427	75687	90029	09517	02835	29716	34456	21296
40435	23117	60066	51012	41200	65975	58512	76178	58382	92041
97484	42360	80071	93045	76189	32349	22927	96501	98751	87212
72675	07981	25547	09589	04556	35792	12210	33346	69749	92356

30254	94780	24901	14195	21238	28153	09114	07907	38602	51522
74299	58180	72471	62591	66854	51333	12394	80494	70791	19153
26734	30282	44186	04142	63639	54800	04480	02670	49624	82017
92896	47669	75831	83271	31425	17029	69234	88962	76684	40323
26092	75249	60357	99646	92565	04936	81836	09003	23809	29345
95889	70695	36534	94060	34021	66544	37558	90045	63288	22505
45255	64056	44824	65151	87547	11962	18443	96582	53375	43885
69094	11303	15095	26179	37800	29741	20766	51479	39425	90298
96959	46995	56576	12186	56196	73378	62362	56125	21632	08628
69222	10327	48892	18654	36480	22967	80705	76561	51446	32046
92790	68212	07388	37781	42335	62823	60896	32080	68222	46801
22482	61177	18589	63814	09183	90367	36722	20888	32151	37556
00372	79839	40041	52970	02878	30766	70944	47456	01345	56417
25437	09069	79396	12257	14298	94671	54357	84687	88614	44581
23145	93571	98492	25284	71605	04922	12424	70141	21478	05734
55105	00801	90869	96033	02763	47870	81081	75450	11930	71412
23390	86639	38339	52942	57869	05076	43100	63835	19834	38934
15961	31854	34754	64955	69781	03829	30971	64651	43840	70070
73604	11237	35998	43452	25161	05070	27056	23526	60127	64848
30840	76118	30130	52793	20542	74628	65403	60367	45328	65105
70658	74882	25698	15793	67897	66974	22057	50596	83440	86973
50201	41020	67235	85020	07245	22563	26513	41055	92401	90274
21624	84391	40359	98953	53945	90944	07046	91209	14093	87001
26456	00162	37428	80210	92764	57931	06579	22955	24988	72758
46101	26483	69998	92256	95968	81592	05600	10165	52563	75678

5.4 Die Zahl e

Es sei $e_n = \sum_{k=0}^n 1/k!$. Dann gilt $|e - e_n| < \epsilon$ mit
 $\epsilon = (n+2)/(n+1)/(n+1)!$. Für $\epsilon = 10^{-3002}$ erhält man $n = 1143$.

Modifiziert man das angegebene Verfahren, so kann man mit folgendem Algorithmus erreichen, daß nur das Unterprogramm Division und ein Registerblock erforderlich ist.



Die Registerbelegung wurde folgendermaßen vorgenommen: R 00 - R 301 (R₁) enthalten e. Der Index j wird in R M ([]) gespeichert, der Divisor DR = n in R N (\). R 0 ([]) nimmt den Rest RE auf. Die Register R P (↑) und R a dienen als Zwischenspeicher.

Nach einem SIZE 302 kann das Programm mit XEQ ZE gestartet werden. Nach 6d 8h 24min ist die Berechnung beendet. Als Ausgabeprogramm dient das Programm AG. Es kann erst in den Rechner geladen werden, wenn das Programm ZE gelöscht ist. Die Adressen müssen den geänderten Registerbelegungen angepaßt werden. Es ergibt sich für e:

e = 2,718281828... .

Die Zahl e

01*LBL "ZE"	20 X<>Y	39 X=Y?	58 RCL \
02 CLRG	21 RCL]	40 ST- IND [59 X>Y?
03 1143	22 *	41 RCL T	60 GTO 02
04 STO \	23 RCL \	42 X(>)]	61 MOD
05 1	24 /	43 E10	62 STO]
06 STO 00	25 INT	44 *	63 1
07*LBL 00	26 STO a	45 RCL \	64 ST+ IND [
08 ,301	27 +	46 MOD	65*LBL 02
09 STO [28 X(>) IND [47 ST+]	66 ISG [
10 0	29 RCL \	48 RCL \	67 GTO 01
11 STO]	30 MOD	49 /	68 1
12*LBL 01	31 STO Z	50 RCL a	69 ST+ 00
13 RCL IND [32 RCL \	51 +	70 ST- \
14 RCL \	33 /	52 LASTX	71 RCL \
15 /	34 RCL ↑	53 -	72 X>0?
16 INT	35 +	54 1	73 GTO 00
17 E10	36 LASTX	55 X=Y?	74 OFF
18 X<>Y	37 -	56 ST- IND [75 END
19 STO ↑	38 1	57 RCL]	

6 LITERATURVERZEICHNIS

- [1] Meyers Enzyklopädisches Lexikon, Bd. 18, 1980,
Bibliographisches Institut, Mannheim.
- [2] Die Bibel, Revidierter Text 1975,
1978, Deutsche Bibelstiftung, Stuttgart
- [3] G e n u y s, Francois: Dix mille decimales de π , Chiffres,
Revue de l' Association Francaise de Calcul, 1958, Nr. 1,
Paris.
- [4] B l a t t e r, Christian: Analysis, Bd. 1 & 2, 2. Auflage,
1979, Springer, Berlin, Heidelberg, New York

Bei der Firma Lampert in 6800 Mannheim möchte ich mich für die
Bereitstellung eines Druckers herzlich bedanken.

Überführung einer gegebenen Permutation aus n Zahlenelementen durch Transposition in eine geordnete Folge steigender Wertigkeit

von Karl Hackenberg

1 AUFGABENSTELLUNG

Statistische Untersuchungen im kaufmännischen und technischen Bereich sowie innerhalb der Naturwissenschaften führen häufig zur Zusammenstellung von Tabellen, für die hinsichtlich klarer und schneller Beurteilung eine übersichtliche Form unerlässlich ist. Im allgemeinen wird zumindest eine Spalte einer solchen Tabelle Zahlenwerte steigender oder fallender Tendenz aufweisen, während alle übrigen Spalten abhängige Größen in mehr oder weniger geordneter Form ausfüllen. Hierbei könnte es sich als notwendig erweisen, die eine oder andere Spaltenart in geordneten Wertigkeiten darzustellen. Eine größere Anzahl von Zahlenelementen, ggf. mit unterschiedlich langen Dezimalstellen, negativen Vorzeichen und Wiederholungen in eine Folge mit steigender Wertigkeit manuell zu ordnen, ist zeitraubend und fehlerverdächtig. Hier bietet sich der Rechner mit nachstehend beschriebenem Programm an, das nach einer der Datenmenge entsprechenden Bearbeitungsdauer die geordnete Folge einschließlich spezifisch mathematischer Größen abgibt.

2 LÖSUNGSWEG

Die Elemente einer gegebenen Zahlenfolge (Permutation) sind wunschgemäß so zu vertauschen, daß die neue Zahlenfolge eine steigende Wertigkeit aufweist, z.B.:

Gegebene Permutation	9	6	0,2	-5	3
Gewünschte Permutation	-5	0,2	3	6	9

Diese Maßnahme erfordert sequentielle Vertauschung von jeweils 2 Elementen (Transposition), sofern ihre Reihenfolge der gewünschten entgegengesetzt verläuft (Inversion). Um im vorstehenden Beispiel die angegebene Endpermutation zu erhalten, sind 8 Inversionen erforderlich. Würden Anfangs- und Endfolge wertemäßig entgegengesetzt verlaufen, so ergäben sich 10 maximal mögliche Inversionen.

Zur Aufstellung einer Routine für den Rechner möge ein einfaches Schema (1) dienen. Der Übersichtlichkeit halber sollen nur 4 Elemente der Zahlen 1 bis 4 ohne Wiederholung bei größtmöglicher Anzahl von Inversionen verwendet werden.

a	b	c	d	e	f	g	h
4	3	3	3	2	2	2	1
3	4	2	2	3	1	1	2
2	2	4	1	1	3	3	3
1	1	1	4	4	4	4	4

(1)

Die Spalte a, von oben nach unten gelesen, enthält die gegebenen Elemente in größtmöglicher Abweichung vom gewünschten Ergebnis der Spalte h. Links vom ausgezogenen Strich steht jeweils das zu vertauschende Elementenpaar, rechts davon die vollzogene Inversion. Die gepunktete Linie zwischen Spalte f und g besagt, daß an dieser Stelle eine Inversion entfällt.

Zwischen Beginn und Ende des Ordnungsvorganges können Elementenfolgen auftreten, die eine übereinstimmende Anzahl von Inversionen aufweisen. Z.B. resultieren schon bei 4 unterschiedlichen Elementen mit $4! = 24$ Permutationen nachstehende Häu-

figkeiten mit zugehöriger Anzahl der Inversionen:

Häufigkeiten	1	3	5	6	5	3	1
Inversionen	6	5	4	3	2	1	0

Durch schrittweise Überprüfung benachbarter Zahlenelemente einer vorgegebenen Permutation nach Schema (1) bleiben Inversionshäufigkeiten größer als 1 unberücksichtigt.

3 PROGRAMMAUFSTELLUNG

Aus dem Schema (1) ist der Algorithmus bereits ersichtlich. Nach Speicherung der Elemente in gegebener Reihenfolge auf LBL E werden zunächst die beiden ersten der Spalte a durch LBL 04 auf steigende Wertigkeit überprüft und ggf. miteinander vertauscht. Anderenfalls, d.h. wenn steigende Wertigkeit bereits vorliegt, wird dieser Zustand von LBL 02/R07 lediglich registriert. Anschließend wiederholt sich der Vorgang in Spalte b oder c für eine neue Permutation, jedoch um einen Schritt nach unten versetzt. Vor Beginn der Wiederholung muß aber noch durch LBL 00 festgestellt werden, ob mit dem letztlich überprüften Datenpaar das Ende der eingegebenen Datenfolge erreicht worden ist. Im bejahenden Fall wird durch LBL 01 für den Neubeginn auf das erste Datenpaar zurückgeschaltet.

Ist die gewünschte Permutation steigender Wertigkeit bei einem bestimmten Abfragestand eingetreten, so müssen noch (n-1) folgende Datenpaare eine negative Abfrage, also keine Inversion ergeben, bevor der Übergang zur Ergebnisanzeige erfolgen kann. Negative Abfragen, im Register 07 addiert, entscheiden somit darüber, wann der Rechenvorgang beendet ist. Sollte einer negativen Abfrage nochmals eine Inversion folgen, muß der Inhalt von Register 07 jeweils auf Null reduziert werden.

4 PROGRAMMDURCHFÜHRUNG

Vor dem Start auf A ist ab Register 11 eine dem Datenumfang entsprechende Anzahl von Speicherplätzen bereitzustellen. Alle weiteren Anweisungen erfolgen durch die Textanzeigen.

Unter D (Details) lassen sich einige der jeweiligen Rechnungsdurchführung eigentümliche Ergebnisse abrufen. Außer der Anzahl der Elemente wird die der stattgefundenen Inversionen angezeigt. Anschließend folgt die Summe der maximal möglichen Inversionen. Diese Anzeige bezieht sich somit auf die einer vorgegebenen Permutation zugehörige Folge größter Unordnung. Der Quotient aus der Anzahl der erforderlichen zu der Anzahl der maximal möglichen Inversionen ist ein Maß für den Grad der Unordnung einer gegebenen gegenüber der geordneten Folge. Dieser Quotient in Prozent wird als Inversionskoeffizient bezeichnet.

Hierzu sei abschließend vermerkt, daß die Anzahl der maximal möglichen Inversionen analytisch aus der Anzahl der Elemente und der ihrer Wiederholungen nach

$$I_{\max} = 0,5 [n(n-1) - w_1(w_1+1) - w_2(w_2+1) - \dots]$$

bestimmt werden kann. Für den Rechner ist es jedoch sehr umständlich, zwischen beliebig vielen und vor allem unterschiedlichen Arten von Wiederholungen zu unterscheiden. Daher wird im vorliegenden Programm zur Bestimmung von I_{\max} die im ersten Durchgang geordnete Zahlenfolge nochmals, aber nunmehr rückläufig invertiert.

5 TESTBEISPIEL

Die Auswertung einer Versuchsreihe ergibt durch mathematisch bedingte Modulierungen eine stark streuende Zahlenfolge, die aus anschaulichen Gründen in steigender Wertigkeit darzustellen ist.

6 ANWEISUNGSLISTE

PRF "TRAPOS"

01*LBL "TRAPOS"
-START >A" PROMPT

04*LBL A
CLRG FIX 2 SF 00
CF 01 CF 28 11 STO 01
STO 03 6 STO 10
"ZAHLENELEM. >E" AVIEW
"NACH EINGABE" "t" >R/S"
AVIEW ADV RTN

22*LBL E
FC? 00 GTO A STO 00
STO IND 01 1 ST+ 01
ST+ 02 RCL 02 1 -
STO 04 RCL 00 STOP

36*LBL 04
RCL IND 03 ISG 03 0
GTO 08

41*LBL 10
RCL IND 03 DSE 03

44*LBL 08
RCL IND 03 X>Y? GTO 02
X=Y? GTO 02 FS? 01
GTO 12 DSE 03
STO IND 03 ISG 03 0
X<>Y STO IND 03 0
STO 07 1 ST+ 08
GTO 11

63*LBL 12
ISG 03 0 STO IND 03
DSE 03 X<>Y STO IND 03
0 STO 07 1 ST+ 09
GTO 11

75*LBL 02
1 ST+ 07 RCL 07
RCL 04 X=Y? GTO 03

82*LBL 11
XEQ 00 FS? 01 GTO 10
GTO 04
87*LBL 00

11 FC? 01 RCL 04
FS? 01 0 + RCL 03
X=Y? XEQ 01 RTN

98*LBL 01
FS? 01 RCL 04 FC? 01
0 ENTER+ 11 + STO 03
RTN GTO 02

109*LBL 03
BEEP FS? 01 GTO 09
"END-PERMUT." AVIEW
"VISUELL >C" AVIEW
"DETAILS >D" AVIEW
XEQ 05 PRREGX RTN

122*LBL C
XEQ 05 STO 05

125*LBL 07
RCL 05 INT STO 03
RCL IND 03 PSE ISG 05
GTO 07 RTN

134*LBL 05
11 ENTER+ ENTER+
RCL 04 + 1 E-3 * +
FIX 2 RTN

145*LBL D
FIX 0 RCL 02
"Σ ELEMENTE" "
XEQ IND 10 RCL 08
"Σ INVERS." "
XEQ IND 10

153*LBL 13
SF 01 RCL 01 1 -
STO 03 GTO 10

160*LBL 09
FIX 0 RCL 09
"Σ INV. MAX." "
XEQ IND 10 RCL 08 X<>Y
/ 1 E2 *
"INV.KOFFZ. %" "
XEQ IND 10 CF 00 CF 01
"ENDE" AVIEW RTN

177*LBL 06
"t= " ARCL X AVIEW
END

XEQ "TRAPOS"

START >A

XEQ A

ZAHLENELEM. >E
NACH EINGABE >R/S

105,91 XEQ E
89,30 XEQ E
0,07 XEQ E
67,35 XEQ E
89,30 XEQ E
14,76 XEQ E
-2,93 XEQ E
19,40 XEQ E
89,30 XEQ E
44,18 XEQ E
67,35 XEQ E
19,40 XEQ E
-0,72 XEQ E
2,93 XEQ E
19,40 XEQ E
RUN

*END-PERMUT.
VISUELL >C
*DETAILS >D

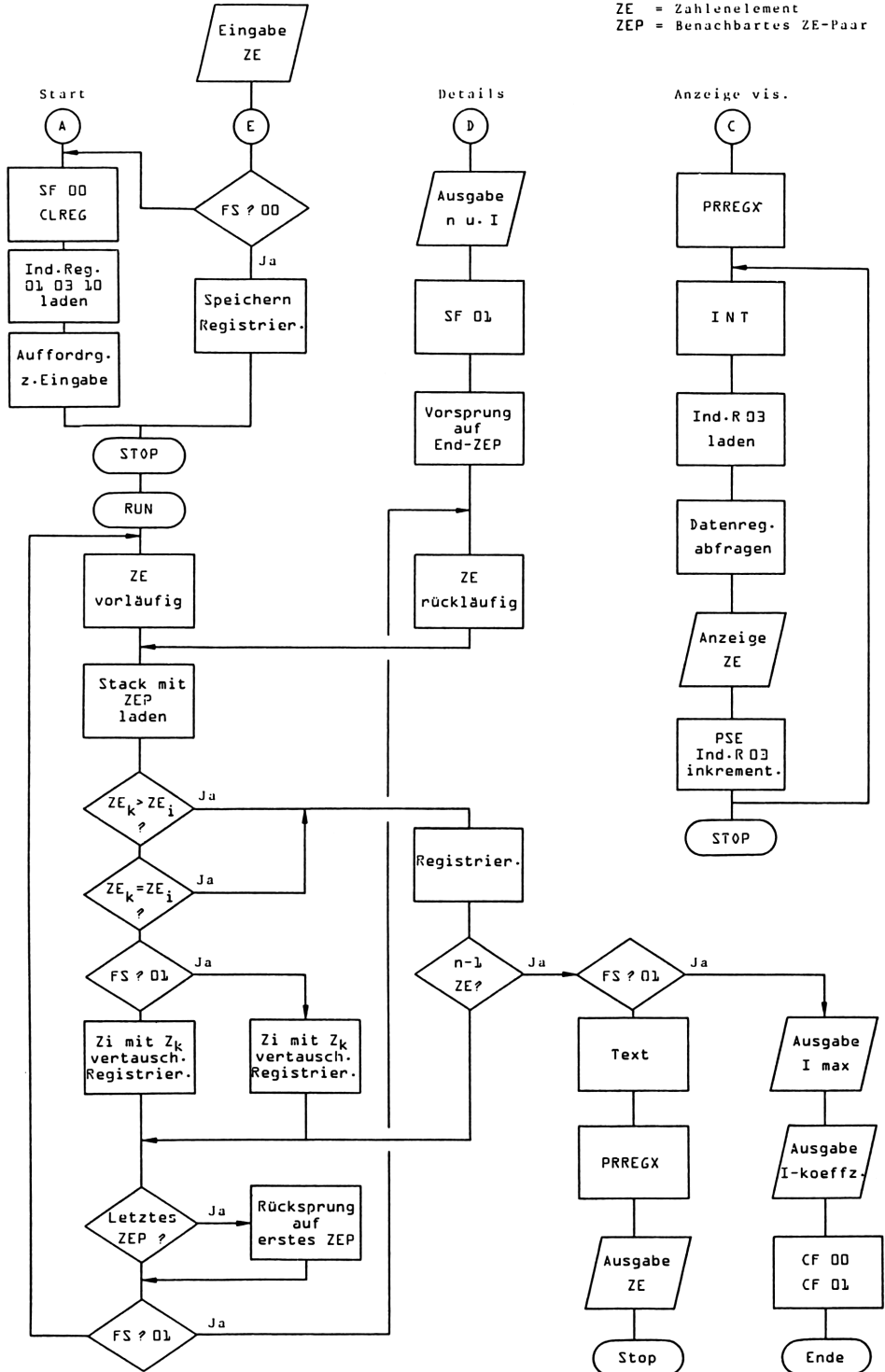
R11= -2,93
R12= -0,72
R13= 0,07
R14= 2,93
R15= 14,76
R16= 19,40
R17= 19,40
R18= 19,40
R19= 44,18
R20= 67,35
R21= 67,35
R22= 89,30
R23= 89,30
R24= 89,30
R25= 105,91

XEQ D

*Σ ELEMENTE = 15
*Σ INVERS. = 65
*Σ INV. MAX. = 98
*INV.KOFFZ. % = 66
ENDE

7 FLUSSDIAGRAMM

n = Anzahl d. ZE
I = Anzahl d. Invers.
ZE = Zahlenelement
ZEP = Benachbartes ZE-Paar



INFO-KARTE

Bitte informieren Sie mich über die neue Vieweg-Infoware

Ich habe einen Mikrocomputer:

Typ: _____ Hersteller: _____

Ich benutze den Mikrocomputer

hauptsächlich privat

☐

hauptsächlich beruflich

☐

Vorwiegend wende ich den Mikrocomputer in folgenden Bereichen an:

_____ Diese Karte entnahm ich dem Buch: _____

Meine Buchhandlung: _____

Gleichzeitig bestelle ich: _____ Ex. _____

_____ Ex. _____

Anschrift:

_____ Name

_____ Vorname

_____ Beruf/Funktion

_____ Straße

_____ PLZ

_____ Ort

VIEWEG-INFOWARE

Lieber Leser,

Sie haben ein Mikrocomputer-Buch aus dem Vieweg-Infoware-Angebot gekauft. Wir wünschen Ihnen viel Freude beim Lesen und Erfolg beim Durcharbeiten.

Gern informieren wir Sie in Zukunft über Neuerscheinungen. Wenn Sie daran interessiert sind, schicken Sie uns bitte diese Karte ausgefüllt zurück.

Wenn Sie an der Weiterentwicklung der Vieweg-Infoware mitarbeiten wollen, z. B. durch Veröffentlichung ausgetesteter Programme zu bestimmten Anwendungsgebieten, dann schreiben Sie uns.

Mit freundlichen Grüßen
Verlag Vieweg

Bitte
mit
60 Pf.
freimachen

Antwortkarte

Friedr. Vieweg & Sohn
Verlagsgesellschaft mbH

Postfach 5829

D - 6200 Wiesbaden 1

VIEWEG PROGRAMMBIBLIOTHEK

Mikrocomputer

Die Bände der Programmbibliothek enthalten ausgetestete Programme zu jeweils einem ausgewählten Themenschwerpunkt oder für einen aktuellen Mikrocomputer. Dabei wird der jeweilige Entwicklungsstand der Rechnertechnik berücksichtigt.

Die Programmbibliothek ist mithin als Ergänzung der Programmsammlung im Mikrocomputer Jahrbuch anzusehen.

Die Programme sind, ausgehend von einer konkreten Aufgabenstellung, in der Regel in ihrem Ablauf beschrieben und durch ausgeführte Beispiele ergänzt. Wenn es nötig scheint, sind auch theoretische Grundlagen für die Programmierung erläutert.

Durch die graphischen, tabellarischen oder in Textform gegebenen Ablaufbeschreibungen wird die Übertragbarkeit auf andere Rechner-typen erleichtert, so daß die wirtschaftliche Nutzung der einzelnen Bände möglich ist. An Hand gleichartiger Aufgabenstellungen wird fallweise auch die unterschiedliche Arbeitsweise verschiedener Rechnertypen aufgezeigt.

Der Herausgeber bemüht sich ständig um eine sorgfältige Auswahl und Begutachtung der eingesandten Programme. Trotzdem kann keine Gewährleistung für vollständige Fehlerfreiheit übernommen werden. Programme zeigen ja oft erst nach vielen Testläufen mit wechselnden Parametern und Grenzbedingungen logische Fehlreaktionen und Sackgassen.

Für die Fälle, die zu Anregungen oder Kritik führen, sind in jedem Band die Anschriften der einzelnen Autoren angegeben. Wir erhoffen uns dadurch einen regen Gedankenaustausch zwischen Autoren und Benutzern der Programmbibliothek, der sich für beide Seiten als nützlich erweisen dürfte.