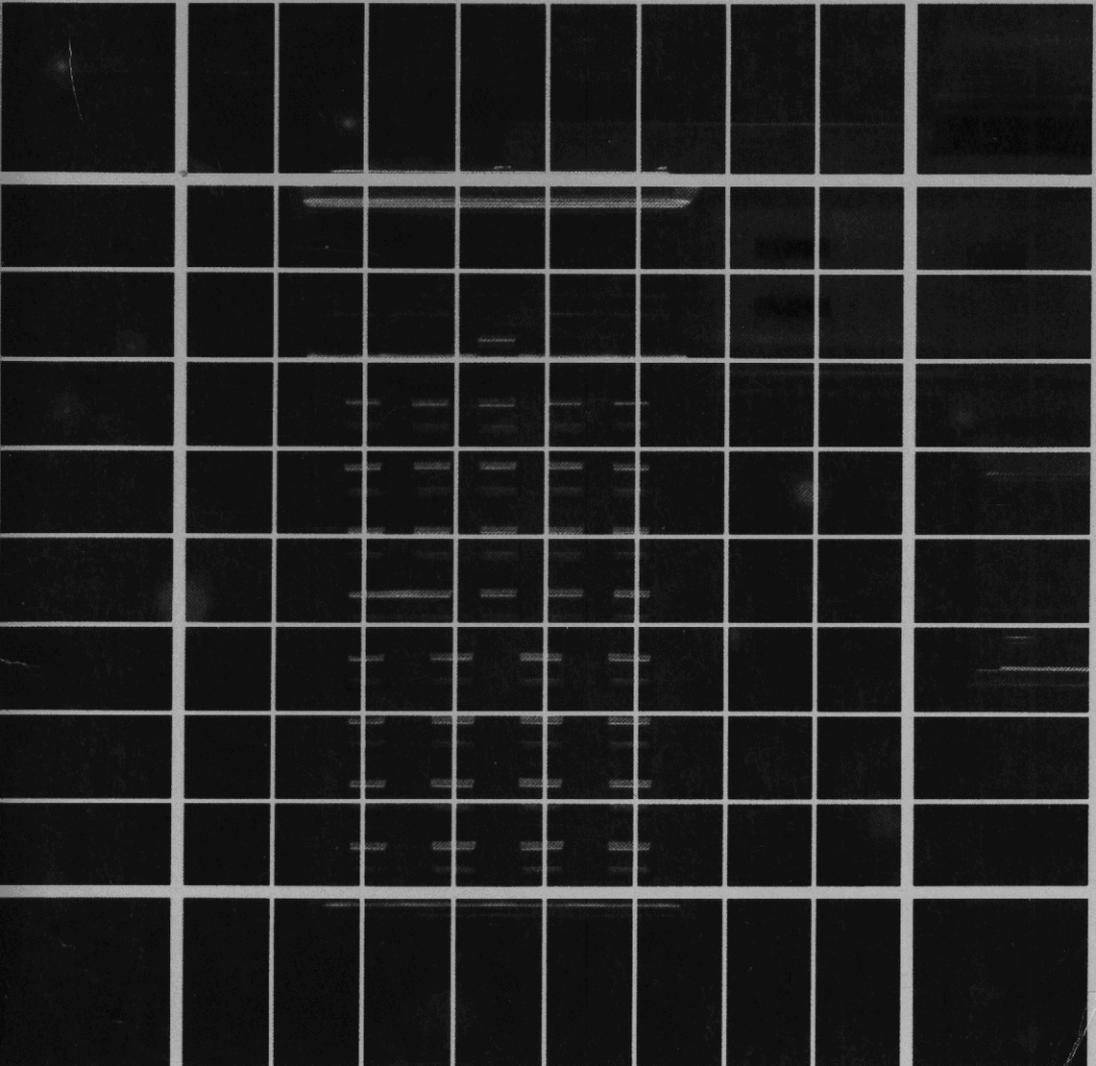


HEWLETT-PACKARD

# HP-41C/CV HANDBUCH

Handbuch für den fortgeschrittenen Anwender







**HP-41C/CV Handbuch**  
Handbuch für den fortgeschrittenen Anwender

September 1980

00041-90330

# Inhaltsverzeichnis

Einleitung .....	5
<b>Teil 1: HP-41C/CV Grundlagen .....</b>	<b>6</b>
Rechner-Modi .....	6
ON .....	6
USER .....	6
PRGM .....	6
ALPHA .....	6
Status-Indikatoren .....	7
Anzeige der Funktion und NULL .....	7
Abfrage nach Angaben (Parameter) .....	7
Angaben durch einzelne Tasten .....	8
Indirekte Parameterangabe .....	8
Verzeichnisse (Kataloge) .....	12
Ausführung von Funktionen und Programmen über die Anzeige .....	13
Anzeige Korrektur und Löschung .....	15
Abspeichern und Zurückrufen von Daten .....	15
Betrachten des Registerinhaltes .....	16
Löschen der Datenspeicherregister .....	16
<b>Teil 2: Speicherplatz .....</b>	<b>17</b>
ALPHA-Register .....	18
Hauptspeicher .....	18
Hauptspeichereinteilung .....	18
Änderung der Hauptspeichereinteilung .....	18
Erweiterung des Hauptspeichers (nur HP-41C) .....	20
Löschen des Hauptspeichers .....	23
Programmspeicher .....	24
<b>Teil 3: Alpha-Modus .....</b>	<b>26</b>
ALPHA-Register .....	26
Anzeige des ALPHA-Registers .....	27
Löschen und Korrigieren von Alpha-Ketten .....	28
Löschen des ALPHA-Registers .....	28
Korrektur von Alpha-Ketten .....	28
Verschiebung von Ketten im ALPHA-Register .....	29
Abspeichern von Alpha-Ketten aus dem ALPHA-Register .....	29
Zurückrufen von Daten ins ALPHA-Register .....	29

<b>Teil 4: User-Modus</b> .....	<b>31</b>
Zuordnen von Funktionen oder Programmen .....	<b>31</b>
Operationen im User-Modus .....	<b>33</b>
Zuordnungen lokaler Marken .....	<b>33</b>
<b>Teil 5: Programmierungsgrundlagen</b> .....	<b>35</b>
Eingeben eines Programmes .....	<b>35</b>
Ausführung eines Programmes .....	<b>37</b>
Programmkomponenten .....	<b>37</b>
Programmzeilen .....	<b>37</b>
Marken .....	<b>38</b>
<b>Teil 6: Programmkorrektur</b> .....	<b>39</b>
Positionieren im Programmspeicher .....	<b>39</b>
Positionieren mit <b>[GTO]</b> <input type="checkbox"/> .....	<b>39</b>
Positionieren mit <b>[RTN]</b> .....	<b>39</b>
Positionierung mit <b>[CATALOG]</b> 1 .....	<b>39</b>
Einzelschritt vor und Einzelschritt zurück .....	<b>40</b>
Im Normal-Modus .....	<b>40</b>
Löschen und Korrektur einzelner Anweisungen .....	<b>41</b>
Löschen von Anweisungen .....	<b>41</b>
Einfügen von Anweisungen .....	<b>42</b>
Löschen von Programmen .....	<b>42</b>
<b>Teil 7: Programmunterbrechungen</b> .....	<b>44</b>
<b>STOP</b> und <b>[R/S]</b> .....	<b>44</b>
<b>STOP</b> .....	<b>44</b>
Stop vom Tastenfeld .....	<b>44</b>
<b>PROMPT</b> .....	<b>44</b>
<b>PSE</b> (Pause) .....	<b>44</b>
<b>OFF</b> .....	<b>45</b>
Stop bei Fehlermeldungen .....	<b>45</b>
<b>Teil 8: Programmverzweigungen</b> .....	<b>46</b>
Die Verwendung von <b>[GTO]</b> in einem Programm .....	<b>46</b>
Die Suche nach Marken .....	<b>46</b>
Suche nach globalen Marken .....	<b>46</b>
Suche nach lokalen Marken .....	<b>47</b>
Vergleichende Funktionen .....	<b>48</b>
Steuerung von Programmschleifen .....	<b>49</b>
<b>Teil 9: Unterprogramme</b> .....	<b>50</b>
Aufruf von Unterprogrammen .....	<b>50</b>
Grenzen bei der Verwendung von Unterprogrammen .....	<b>51</b>

## 4 Inhaltsverzeichnis

<b>Teil 10: Flags</b> .....	<b>52</b>
<b>Teil 11: Peripheriegeräte</b> .....	<b>55</b>
<b>Teil 12: Fehler- und Statusmeldungen</b> .....	<b>57</b>
<b>Funktionsverzeichnis</b> .....	<b>59</b>
<b>Verzeichnis</b> .....	<b>69</b>

# Einleitung

Dieses « HP-41C/CV Handbuch » soll ein Handbuch für den fortgeschrittenen Anwender sein. Es ergänzt das « HP-41C/CV Bedienungs- und Programmierhandbuch », ersetzt es jedoch nicht. Wenn Sie schon Erfahrungen mit anderen programmierbaren Rechnern gemacht haben, wird dieses Handbuch wohl zu Ihrer hauptsächlichen Unterrichtsunterlage werden, um mit den neuen und besonderen Eigenschaften Ihres HP-41C oder HP-41CV vertraut zu werden. Sofern Sie erst das Bedienungs- und Programmierhandbuch durcharbeiten, werden Sie dieses Handbuch als eine gute Zusammenstellung der wichtigsten HP-41C/CV Anwendungsmöglichkeiten erkennen. Zum größten Teil wird hier dasselbe behandelt wie im Bedienungs- und Programmierhandbuch, nur verkürzt und beziehend. Die Beschreibung einiger Grundlagen wie UPN und mathematischen Funktionen wurden in diesem Handbuch nicht wiederholt.

Wie Sie sicher schon gesehen haben, können Funktionen und Programme nicht nur vom Tastenfeld ausgeführt werden, sondern auch « über die Anzeige »\*. Diese Funktionen, die nicht auf oder über einer Taste stehen (Normal-Modus Tastenfeld oder Alpha-Tastenfeld) werden im Handbuch immer dargestellt, wie sie in der Anzeige erscheinen, beispielsweise: **DSE** (im Gegensatz zum Bedienungs- und Programmierhandbuch, wo die Funktionen in « Tasten-Kästchen » dargestellt werden). Diese Funktionen können *nur* über die Anzeige ausgeführt werden (außer, Sie würden sie einer Taste zuordnen).

Beachten Sie bitte, daß der HP-41C und der HP-41CV bis auf ihre Speicherkapazität identisch sind. Wenn Sie 4 Speichererweiterungsmodule HP 82106A oder *ein* HP 82170A in den HP-41C einstecken, erhalten Sie für den HP-41C die gleiche Speicherkapazität, wie sie der HP-41CV besitzt. Da die Speicher beim HP-41CV in das Gerät integriert sind, haben Sie 4 freie I/O Buchsen zur Verfügung, um Software-Module oder Peripheriegeräte in beliebiger Kombination anzuschließen. Da die Unterschiede der beiden Modelle sich auf die anfängliche Speicherkapazität beschränkt, wird in diesem Handbuch für beide Modelle nur noch die Bezeichnung HP-41C benutzt, es sei denn, es würde ausdrücklich etwas anderes ausgeführt.

\* Erläuterung zum Begriff: « über die Anzeige ausführen ».

Funktionen oder Programme, die nicht auf dem Tastenfeld aufgedruckt sind, können folgendermaßen aufgerufen werden :

Beispiel: Funktion **FACT** (= Fakultät).

Ausführung über die Anzeige: **[XEQ] [ALPHA] FACT [ALPHA]**.

# HP-41C/CV Grundlagen

## Rechner-Modi

Die vier Rechner-Modi (Normal, Alpha, User und Programm) werden durch die Modi-Tasten direkt unter der Anzeige gewählt. Der Rechner ist im Normal-Modus, wenn er nicht im Alpha-, User- oder Programm-Modus ist.

### ON

Die **ON**-Taste schaltet den Rechner ein und aus. Nach ca. 10 Minuten Inaktivität schaltet sich der Rechner automatisch aus, um die Batterie zu schonen. (Das Continuous Memory bewirkt jedoch, daß auch in ausgeschaltetem Zustand der Rechner keine Programme, Daten oder Status « vergißt »).

### USER

Die **USER**-Taste schaltet den User-Modus ein bzw. aus. Im User-Modus werden die vom Anwender durchgeführten Tastenfeldzuordnungen aktiv. Der **USER**-Indikator in der Anzeige gibt an, ob der Rechner im User-Modus ist oder nicht. Im Alpha-Modus ist der User-Modus nicht wirksam.

### PRGM

Die **PRGM**-Taste schaltet den Programm-Modus ein oder aus. Im Programm-Modus werden Tastendrucke nicht ausgeführt, sondern als Programmschritte für spätere Ausführung gespeichert. In der Anzeige ist der **PRGM**-Indikator zu sehen. Befindet sich der Rechner im Alpha-Modus, wird durch Einschalten des **PRGM**-Modus der Alpha-Modus ausgeschaltet, auch der **ALPHA**-Indikator erlischt.

### ALPHA

Die **ALPHA**-Taste schaltet den Rechner in den Alpha-Modus oder zurück. Im Alpha-Modus wird die Alpha-Tastatur aktiv, mit den alphanumerischen, numerischen und Sonderzeichen, die auf der Vorderseite der Tasten in blau angegeben sind, bzw. auch die umgeschalteten (Drücken der -Taste vor dem Drücken der Funktionstaste) Zeichen. Der gesamte Zeichensatz ist auf der Unterseite des Rechners zu sehen. Im Alpha-Modus erscheint der **ALPHA**-Indikator in der Anzeige.

## Status-Indikatoren

Die Status-Indikatoren erscheinen in der Anzeige unten. Zusätzlich zum **USER**, **PRGM** und **ALPHA**-Indikator können folgende Indikatoren erscheinen :

**BAT** zeigt ein Nachlassen der Batteriespannung an. Nach dem ersten Anzeigen von **BAT** bleiben Ihnen mit Alkali-Batterien noch ca. 5 bis 15 Tage Rechenzeit. Weitere Informationen über die Lebensdauer der Batterien entnehmen Sie bitte dem Bedienungshandbuch, Seite 240.

**GRAD** oder **RAD** zeigt an, daß der Rechner im Neugrad- oder Bogenmaß-Modus für trigonometrische und Rechteck/Polarfunktionen ist. Erscheint weder **GRAD** noch **RAD** befindet sich der Rechner im Altgrad-Modus.

**SHIFT** zeigt an, daß die goldene Taste (■) gedrückt wurde.

**0 1 2 3 4** gibt an, daß das entsprechende Flag – 0, 1, 2, 3 oder 4 – gesetzt ist.

## Anzeige der Funktion und NULL

Wenn eine Funktionstaste des HP-41C gedrückt wird, erscheint kurz der Funktionsname in der Anzeige, wird die Taste losgelassen, wird diese Funktion ausgeführt. Diese Anzeige der Funktion ist besonders im User-Modus bei Tastenzuordnungen sehr hilfreich, wenn die Funktion der Taste dann nicht auf dem Tastenfeld angegeben ist.

Wird eine Funktionstaste für länger als etwa eine halbe Sekunde gedrückt, wird die Funktion unwirksam, d.h. die Funktion wird nicht ausgeführt und es erscheint **NULL** in der Anzeige. Die ■-Taste kann nur durch erneutes Drücken unwirksam werden, der **SHIFT**-Indikator erlischt dann.

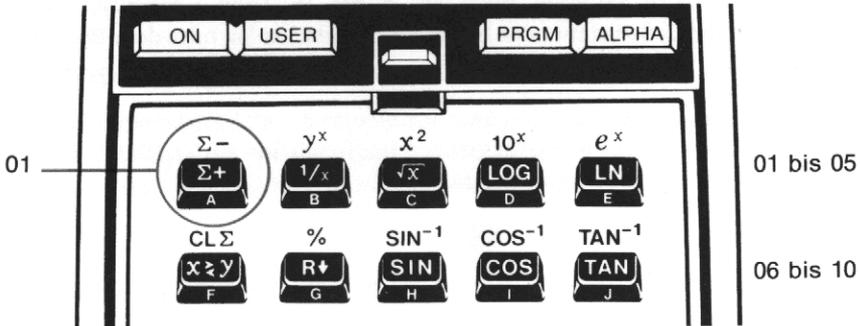
## Abfrage nach Angaben (Parameter)

Erscheinen in der Anzeige ein oder mehr Unterstreichungszeichen ( \_ ) erwartet der Rechner eine Eingabe. Ist die Eingabe einer Zahl oder einer Alpha-Kette nicht abgeschlossen, erscheint ein Unterstreichungszeichen. Das bedeutet, daß weitere Ziffern oder Zeichen, die eingegeben werden, an die bestehende Zahl, Funktion oder Alpha-Kette angehängt werden.

Wurden bestimmte Funktionen – wie z.B. **FIX**, **STO** oder **GTO** ■ – aufgerufen, erscheint in der Anzeige dieser Funktionsname gefolgt von ein, zwei oder drei Unterstreichungszeichen. Dies gibt an, daß ein ein-, zwei- oder dreistelliger Parameter – etwa eine Marke oder eine Adresse – eingegeben werden muß.

## Angaben durch einzelne Tasten

Sofern der Rechner nicht im Alpha-Modus ist, kann ein ein- oder zweistelliger Parameter zwischen 0 und 10 durch Drücken der entsprechenden Taste der oberen zwei Reihen eingegeben werden:



$\Sigma+$  bis  $\text{LN}$  entsprechen 01 bis 05,  $x \leftrightarrow y$  bis  $\text{TAN}$  entsprechen 06 bis 10. Wird ein einstelliger Parameter erwartet, wird nur die rechte Ziffer verwendet (1 bis 5 oder 6 bis 0).

## Indirekte Parameterangabe

Die Parameter für die meisten Funktionen des HP-41C können sowohl direkt als auch indirekt eingegeben werden. Bei indirekter Parameterangabe geben Sie anstelle des Parameters, nach dem der Rechner fragt, die Adresse eines Speicherregisters (das indirekte Speicherregister) ein, das den Parameter enthält. Diese Möglichkeit ist äußerst nützlich, wenn in einem Programm der Parameterwert von den vorhergehenden Berechnungen des Programms abhängig ist. Der Zugang zu den erweiterten Speicherregistern  $R_{100}$  bis  $R_{318}$  kann nur direkt geschehen.

Um einen Parameter indirekt anzugeben:

- Speichern Sie den Wert des Parameters in das indirekte Speicherregister. Das kann sowohl eines der Speicherregister  $R_{00}$  bis  $R_{99}$  sein, wie auch jedes Stack-Register oder das LAST X-Register. Der Rechner verwendet als Parameterwert den absoluten, ganzzahligen Wert des Registerinhaltes.
- Drücken Sie nach der Eingabe der Funktion  $\blacksquare$ . In der Anzeige erscheint dann **IND** \_\_ nach dem Funktionsnamen.
- Geben Sie das indirekte Register an:
  - entweder  $R_{00}$  bis  $R_{99}$ , als zweistellige Angabe (d.h. die Nummer des Speicherregisters)
  - oder ein Stack-Register bzw. das LAST X-Register, mit Eingabe von  $\square$  gefolgt von X, Y, Z, T oder L. Vor der Eingabe dieser Buchstaben muß **ALPHA** nicht gedrückt werden.

Um beispielsweise **FIX** 8 indirekt auszuführen, mit  $R_{10}$  als indirektem Register :

**Tastenfolge**

8 **STO** 10

**■** **FIX** **■**

10

**Anzeige**

8.0000

FIX IND \_\_\_

8.00000000

Speichert den Parameter 8 in Register  $R_{10}$  (Anzeige sei im Moment noch mit **FIX** 4 dargestellt).

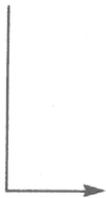
Das Drücken der **■**-Taste gibt an, daß die zweistellige Zahl, die nun eingegeben wird, die Adresse eines Registers ist, das den Parameter enthält, und nicht selbst Parameter ist.

Gibt das indirekte Register an. **FIX** 8 wird durchgeführt.

Das untenstehende Diagramm soll den obigen Vorgang veranschaulichen :

Die Funktion

**FIX** **■** 10



$R_{10}$

Das indirekte Register

8.0000

Entspricht **FIX** 8

Für die meisten Funktionen, deren Parameter indirekt angegeben werden können, ist der Parameter eine Registeradresse. Um 2.54 mittels Y-Register in Register  $R_{10}$  abzuspeichern :

**Tastenfolge**

10 **ENTER**+

2.54 **STO** **■** **•** Y

**Anzeige**

10.0000

2.5400

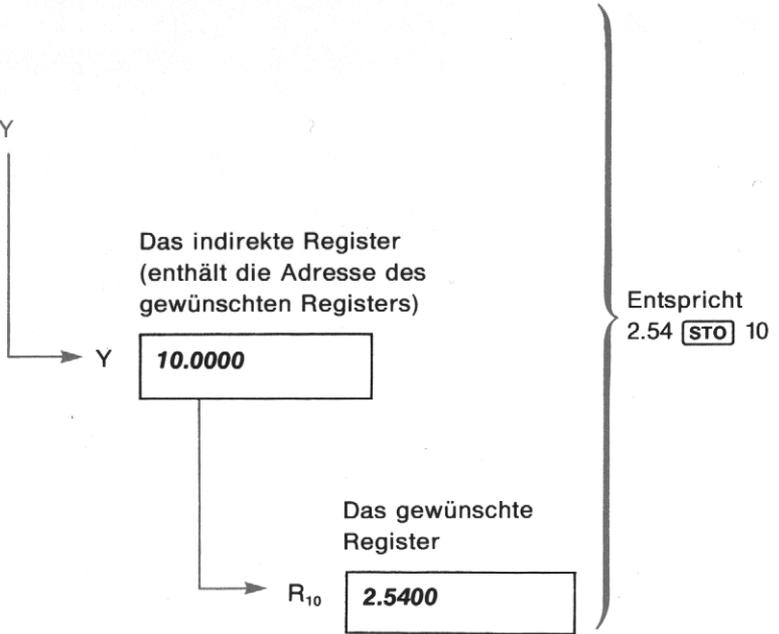
Speichert den Parameter 10 ins Y-Register (Anzeige sei auf **FIX** 4 eingestellt).

2.54 wird in  $R_{10}$  abgespeichert. (Der Buchstabe Y wird durch Drücken der **×**-Multiplikationstaste eingegeben auf deren Vorderseite Y steht.)

Das folgende Diagramm veranschaulicht den Ablauf der indirekten Speicherung von 2.54 in  $R_{10}$ .

Die Funktion

2.54 **STO** **▀** **◻** Y



Für **GTO** und **XEQ** kann der anzugebende Parameter entweder eine Alpha-Marke oder eine numerische Marke sein. Um beispielsweise das Programm SOLVE mittels dem Register  $R_{02}$  (als indirektes Register) aufzurufen:

**Tastenfolge:**

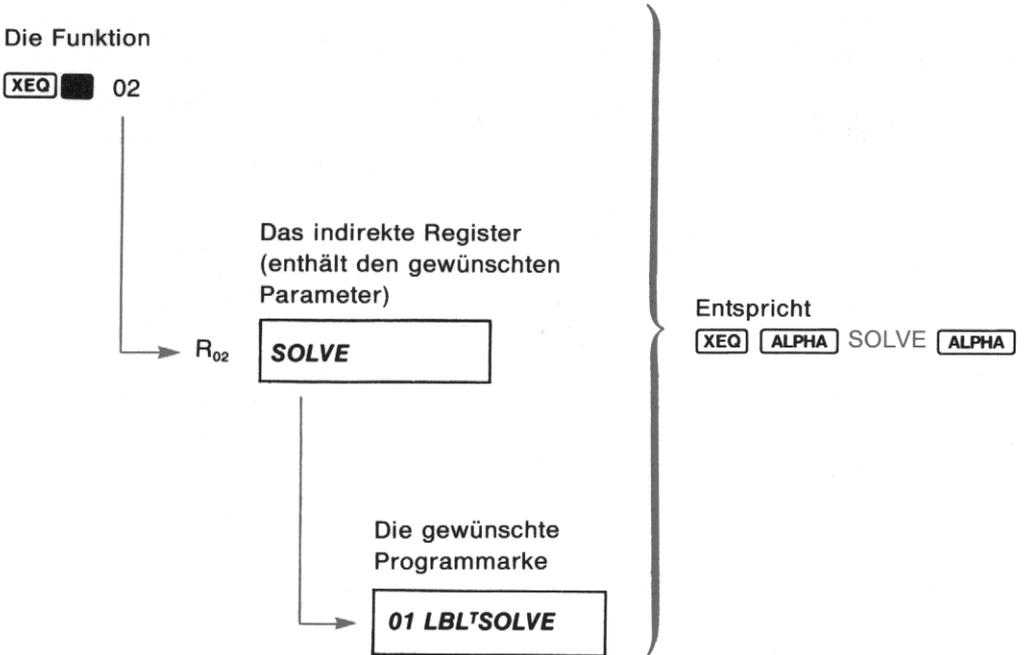
**ALPHA** SOLVE **ASTO** 02 **ALPHA**

**XEQ** **▀** 02

Speichert den Parameter SOLVE in Register  $R_{02}$ .

Startet das Programm, das mit der Marke SOLVE beginnt.

Das folgende Diagramm veranschaulicht den Ablauf, wenn SOLVE indirekt aufgerufen wird:



Eine Funktion mit indirekter Parameterangabe wird ausgeführt, sobald die zweite Ziffer der Adresse des indirekten Registers eingetastet wird. Besteht dieses indirekte Register im Datenspeicher nicht, erscheint in der Anzeige **NONEXISTENT**.

Die Parameter für die folgenden HP-41C Funktionen können indirekt angegeben werden :

<b>FIX</b>	Festkomma-Anzeigeformat	
<b>SCI</b>	Wissenschaftliches Anzeigeformat	
<b>ENG</b>	Technisches Anzeigeformat	
<b>DSE</b>	Dekrement-kontrollierte Schleife	
<b>ISG</b>	Inkrement-kontrollierte Schleife	
<b>STONE</b>	Tonhöhe	
<b>ΣREG</b>	Lage der Summationsregister	
<b>SF</b>	Flag setzen	
<b>CF</b>	Flag löschen	
<b>FS?</b>	Flag prüfen, ob es gesetzt ist	
<b>FC?</b>	Flag prüfen, ob es gelöscht ist	
<b>FS?C</b>	Flag prüfen, ob es gesetzt ist und anschließend löschen	
<b>FC?C</b>	Flag prüfen, ob es gelöscht ist und anschließend löschen	
<b>X &lt;&gt;</b>	Austausch des X-Registers mit einem anderen Register	
<b>CATALOG</b>	Verzeichnis	
<b>STO</b>	Abspeichern	
<b>STO +</b>	Summe abspeichern	jeweils ange-
<b>STO -</b>	Differenz abspeichern	gebenes Register
<b>STO x</b>	Produkt abspeichern	mit X-Register
<b>STO ÷</b>	Quotient abspeichern	
<b>ASTO</b>	Alpha-Kette abspeichern	
<b>RCL</b>	Zurückrufen	
<b>ARCL</b>	Alpha-Kette zurückrufen	
<b>VIEW</b>	Registerinhalt anzeigen	
<b>GTO</b>	Gehe nach ...	
<b>XEQ</b>	Ausführen	

## Verzeichnisse (Kataloge)

Drei Funktions- und Programmverzeichnisse kann der HP-41C ausgeben :

**Catalog 1:** Katalog der vom Anwender geschriebenen Programme. Eine Liste aller globalen Marken und **END**-Anweisungen der vom Anwender geschriebenen Programme. Die Programme werden in der gleichen Ordnung angezeigt, wie sie sich im Programmspeicher befinden. Von jedem Programm werden die globalen Marken und **END**-Anweisungen in der Reihenfolge der Programmzeilennummer angezeigt.

**Catalog 2:** Eine Liste aller Funktionen und Programme, die mit (an den Rechner angeschlossenen) Erweiterungen und Peripheriegeräten verfügbar werden (z.B. Kartenleser, Drucker, Software-Module). Die Liste dieser Funktionen ist nach den jeweiligen Erweiterungen gruppiert.

**Catalog 3:** Eine Liste aller Standardfunktionen, sowohl die über das Normal-Modus Tastenfeld ansprechbaren wie auch aller « versteckten » Funktionen. Diese Liste ist alphabetisch geordnet.

Wenn Sie  **CATALOG** drücken und anschließend 1, 2 oder 3, läuft das betreffende Verzeichnis – in der Anzeige ersichtlich – ab. Während das Verzeichnis aufgelistet wird, können Sie den Ablauf durch Drücken einer beliebigen Taste außer **ON** und **R/S** verlangsamen. **R/S** hält die Auflistung an.

Wenn Sie die Anzeige des Katalogs anhalten:

- wird nach **SST** die nächste Angabe des Verzeichnisses angezeigt;
- wird nach  **BST** die vorhergehende Angabe des Verzeichnisses angezeigt.
- können Sie mit **R/S** den Ablauf fortsetzen;
- wird auf Tastendruck die Ausgabe des Verzeichnisses beendet und die eingegebene Funktion ausgeführt.

Wird Catalog 1 angehalten, können Sie den Rechner einfach an die Programmzeile mit der globalen Marke oder der **END**-Anweisung stellen, die gerade angezeigt wird: Drücken Sie einfach **PRGM**.

## Ausführung von Funktionen oder Programmen über die Anzeige

Jede der Funktionen oder Programme, die in einem der Verzeichnisse aufgelistet werden, kann durch Drücken von **XEQ** und der Eingabe des Funktions- oder Programmnamens ausgeführt werden.

1. Drücken Sie **XEQ**. Die Anzeige zeigt **XEQ \_\_\_**.
2. Drücken Sie **ALPHA** um in den Alpha-Modus zu gehen.
3. Schreiben Sie mit den Alpha-Zeichen den gewünschten Programm- oder Funktionsnamen in die Anzeige. Die Tastatur mit allen Zeichen, auch den mit  Umgeschalteten, sehen Sie auf der vorletzten Seite dieses Handbuchs und auch auf der Rückseite des Rechners.
4. Drücken Sie **ALPHA**, um den Alpha-Modus wieder auszuschalten. Wird eine Funktion aufgerufen, die keinen Parameter benötigt, oder wird ein Programm aufgerufen, dann wird die Funktion oder das Programm ausgeführt, sobald der Rechner den Alpha-Modus verläßt.
5. Benötigt die Funktion einen Parameter – wie z.B. **TONE**, **FC?** oder **SIZE** –, verschwindet nach dem Ausschalten des Alpha-Modus **XEQ** aus der Anzeige und die Funktion wird angezeigt, gefolgt von der erforderlichen Anzahl von Unterstreichungszeichen. Die Funktion wird ausgeführt, sobald der Parameter eingegeben ist.

Beispiel: Um  $e^x - 1$  mit  $x = 0,45$  zu errechnen, geben Sie 0,45 in das X-Register ein und führen die Funktion wie folgt durch:

**Tastensequenz**

.45

**XEQ****ALPHA**

E

↑ (N)

X

- (Q)

1 (Z)

**ALPHA****Anzeige**

.45 \_

XEQ \_ \_

XEQ \_

XEQ E \_

XEQ E ↑ \_

XEQ E ↑ X \_

XEQ E ↑ X - \_

XEQ E ↑ X - 1 \_

0.5683

0,45 (x) eingeben.

**XEQ** drücken. Der Rechner fragt mit Unterstreichungszeichen nach dem Parameter.

Alpha-Modus einschalten.

E auf dem Alpha-Tastenfeld drücken.

↑ ist das umgeschaltete Zeichen von N auf dem Alpha-Tastenfeld.

X auf dem Alpha-Tastenfeld drücken. Nicht verwechseln mit der **X** (Multiplikations-)Taste im Normal-Modus.

- ist das umgeschaltete Zeichen der Q-Taste im Alpha-Modus.

Der Buchstabe 1 ist ein umgeschaltetes Zeichen der Z-Taste im Alpha-Modus, dies entspricht nicht der Ziffer 1 im Normal-Modus.

Alpha-Modus ausschalten; Funktion wird sofort ausgeführt.

Beispiel 2: Führen Sie die Funktion **FS?C** mit Flag 9 aus:**Tastensequenz****XEQ****ALPHA**

FS?C

**ALPHA**

09

**Anzeige**

XEQ \_ \_

XEQ \_

XEQ FS?C \_

FS?C \_ \_

NO

Funktionsname in die Anzeige schreiben.

Alpha-Modus ausschalten. Der Rechner fragt nach einem zweistelligen Parameter: hier nach der Nummer eines Flags.

Die Funktion wird ausgeführt, sobald die zwei Ziffern des Parameters eingetastet wurden. (Antwort des Rechners: Flag 9 war nicht gesetzt.)

Steht der aufgerufene Funktions- oder Programmname nicht in einem der drei Verzeichnisse, wird **NONEXISTENT** angezeigt.

Viele Funktionen, die Ihnen im HP-41C zur Verfügung stehen, können nur mit **[XEQ]** und der Eingabe des Funktionsnamen über die Alpha-Tastatur ausgeführt werden, da es keine entsprechende Taste für sie im Normal-Modus gibt. (Allerdings können diese Funktionen einer Taste zugeordnet und im User-Modus mit einem einzigen Tastendruck ausgeführt werden.) Bei vielen Funktionen, die sowohl über das Tastenfeld wie auch durch Eingabe über die Anzeige ausgeführt werden können, beispielsweise **SIN**, ist der Name für die Eingabe über die Anzeige identisch mit dem Namen, der auf der Taste steht. Für andere – beispielsweise **SQRT** (Quadratwurzel) – ist der Name, der zur Eingabe über die Anzeige einzugeben ist, nicht identisch mit der Bezeichnung auf (über) der Taste. Für alle in Katalog 3 aufgeführten Funktionen ist der Name zur Eingabe über die Anzeige und die entsprechende Tastenangabe im Funktionsverzeichnis am Ende dieses Handbuchs aufgeführt.

## Anzeigekorrektur und Löschung

Wenn das Unterstreichungszeichen in der Anzeige erscheint (bei der Eingabe von Zahlen oder Alphazeichen, um anzuzeigen, daß die Eingabe noch nicht beendet ist), bewirkt **[←]** (Korrektur) die Löschung der am weitesten rechts stehenden Ziffer bzw. des am weitesten rechts stehenden Zeichens.

Im Normal-Modus löscht **[CLx]** den gesamten Inhalt des X-Registers und ersetzt ihn durch Null. Im Alpha-Modus löscht **[CLA]** das gesamte Alpha-Register, auch den Teil, der nicht in der Anzeige zu sehen ist.

Zeigt die Anzeige kein Unterstreichungszeichen (wenn die Ziffern- oder Zeicheneingabe abgeschlossen ist), hat **[←]** dieselbe Wirkung wie **[CLx]** oder **[CLA]**. Ist der Rechner nicht im Alpha-Modus und ist in der Anzeige kein Unterstreichungszeichen, wirkt **[←]** wie **[CLx]**. Ist der Rechner im Alpha-Modus und ist kein Unterstreichungszeichen in der Anzeige, hat **[←]** dieselbe Wirkung wie **[CLA]**.

Zeigt die Anzeige den Inhalt eines Registers nach einem **VIEW**- oder **AVIEW**-Befehl, bringt **[←]** den Inhalt des X-Registers oder (wenn der Rechner im Alpha-Modus ist) des ALPHA-Registers wieder in die Anzeige. In einem Programm kann dies (da **[←]** nicht programmierbar ist) durch die **CLD**-Anweisung (Anzeige löschen) erreicht werden.

**[←]** entfernt auch Fehlermeldungen aus der Anzeige.

## Abspeichern und Zurückrufen von Daten

Um Daten, die sich im X-Register befinden in irgendein Speicherregister, ein Stackregister oder in das LAST X-Register zu speichern :

1. Drücken Sie **[STO]**. Der Rechner verlangt eine zweistellige Adresse mit **STO \_\_**.
2. Drücken Sie zwei Zifferntasten, um die gewünschte Adresse (zwischen 00 und 99) einzugeben ; oder drücken Sie **[◻]** und anschließend X, Y, Z, T oder L (Sie müssen die **[ALPHA]**-Taste zuvor nicht drücken) um das gewünschte Stackregister bzw. LAST X-Register anzugeben.

Werden Daten, gleich ob Zahlen oder Alphazeichen, in ein Register gespeichert, wird ein eventuell dort bestehender Inhalt überschrieben. Jedes Register kann entweder eine Zahl (10 Stellen Mantisse plus 2 Stellen Exponent) oder eine Alpha-Kette (bis sechs Zeichen) fassen.

Um Daten (Zahlen oder Alpha-Ketten) aus einem Register in das X-Register zurückzurufen, drücken Sie die **RCL**-Taste und gehen bei der Adresseneingabe genauso wie bei der Abspeicherung vor. Zurückrufen von Daten ist « nicht-zerstörend », d.h. Daten werden nur in das X-Register kopiert und bleiben im Register. Beim Zurückruf ins X-Register wird ein Stack-Lift durchgeführt.

## Betrachten des Registerinhaltes

Um den Inhalt eines beliebigen Speicherregisters, Stackregister oder des LAST X-Registers zu betrachten, *ohne* ihn ins X-Register zu schreiben :

1. Drücken Sie **VIEW**.
2. Geben Sie – direkt oder indirekt – das gewünschte Register an, genauso wie mit **STO** oder **RCL**.

Im Gegensatz zu **RCL** kopiert **VIEW** den Registerinhalt nicht in das X-Register, **VIEW** zeigt ihn nur an. Weder das angegebene Register, noch das X-Register wird verändert.

**←** beendet das Anzeigen und bringt den Inhalt des X-Registers wieder in die Anzeige. Jede andere Taste beendet das Anzeigen ebenso und führt die Funktion dieser Taste mit dem Inhalt des X-Registers aus – so, wie wenn die Anzeige das X-Register angezeigt hätte, bevor diese Taste gedrückt wurde.

In einem Programm kann das Betrachten mit **CLD** beendet werden. Dies bringt den **→** (Marken ausführenden) Indikator wieder in die Anzeige.

## Löschen der Datenspeicherregister

Um ein einzelnes Speicherregister zu löschen, speichern Sie Null ab.

Um alle Speicherregister zu löschen, führen Sie die **CLRG**-Funktion (clear registers) aus.

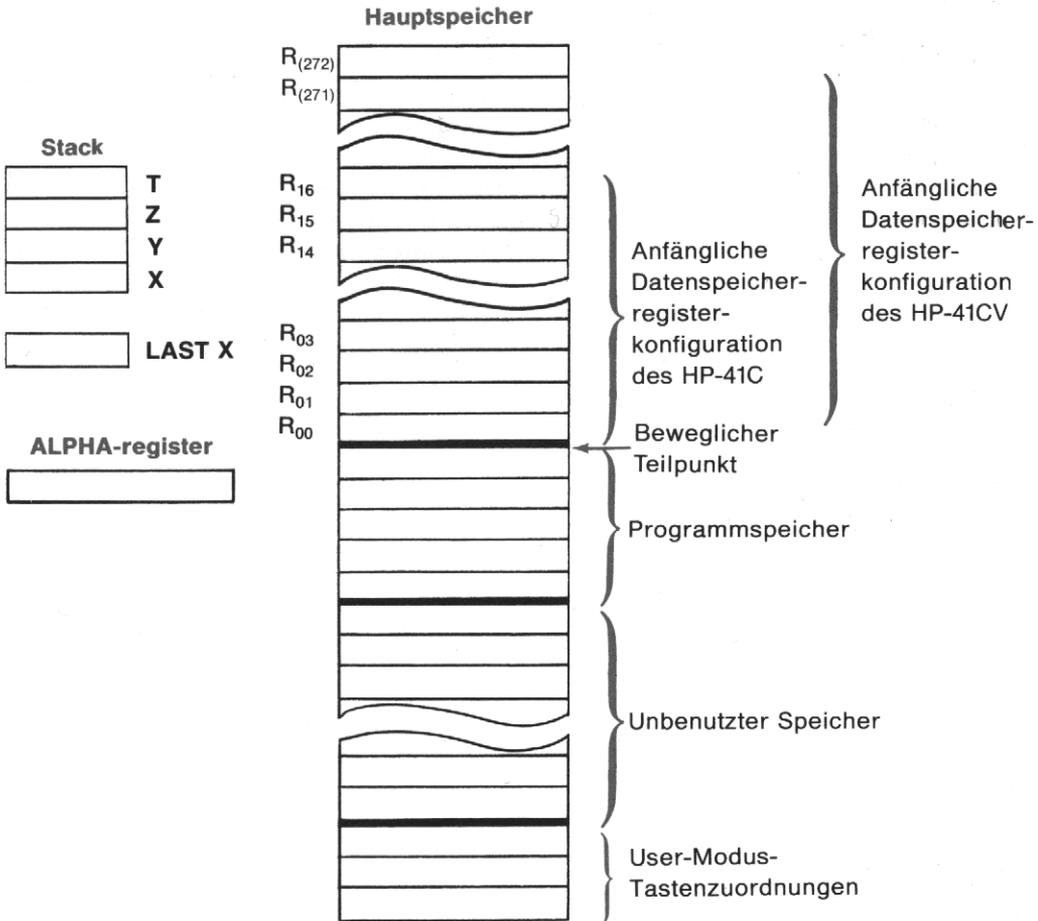
Um alle Speicherregister, alle Programme sowie alle Informationen zu löschen, die das Continuous Memory aufrecht erhält, führen Sie folgende Operationen aus :

1. Schalten Sie den Rechner aus.
2. Halten Sie **←** gedrückt und drücken Sie **ON**.
3. Lassen Sie **←** los.

Sie müssen ein Speicherregister vor dem Abspeichern von Daten nicht löschen, die Speicheroperation löscht das Speicherregister automatisch, bevor Daten abgespeichert werden.

# Speicherplatz

Das Continuous Memory des HP-41C erhält sämtlichen Inhalt, eingeschlossen den der eingesteckten Speichererweiterungsmodule\*, auch, wenn der Rechner ausgeschaltet ist. Einige Flags werden ebenso aufrechterhalten (siehe Teil 10).



\* Der HP-41CV benötigt keine Speichererweiterungsmodule.

## ALPHA-Register

Das ALPHA-Register ist ein separates Register, das Alpha-Zeichen beinhalten kann, die auf dem Alpha-Tastenfeld angegeben sind. Eine Alpha-Kette (d.h. eine Reihe von Alpha-Zeichen) im ALPHA-Register kann maximal 24 Zeichen lang sein, wobei auch Komma, Punkt, Semikolon und Leerzeichen als ein Zeichen zählt.

Ist der Rechner im Alpha-Modus, zeigt die Anzeige normalerweise einen Teil des ALPHA-Registerinhalts. (Ausnahmen siehe Seite 26.) Es können jeweils nur 12 Zeichen gleichzeitig angezeigt werden – hierbei zählen jetzt allerdings Komma, Punkt und Semikolon nicht zu diesen Zwölf, da sie zwischen den Buchstaben stehen.

## Hauptspeicher

Der Hauptspeicher des HP-41C wird zur Datenspeicherung (Zahlen, Ergebnisse, Konstanten, Alpha-Ketten usw.), Programmspeicherung und der Speicherung der User-Modus-Tastenzuordnungen verwendet. Der Anwender kann die Aufteilung in Programmspeicher und Datenspeicher selbst bestimmen.

63 Register bilden den Hauptspeicher des HP-41C\*. Jedes zusätzliche HP 82106 Speichererweiterungsmodul, das eingesteckt wird, vergrößert diesen um weitere 64 Register. Wird anstelle der HP 82106 Speichererweiterungsmodule das HP 82170 Quad-Speichererweiterungsmodul eingesteckt, wird der Hauptspeicher um 256 Register vergrößert. Sind insgesamt vier HP 82106 Speichererweiterungsmodule *oder* ein HP 82170 Speichererweiterungsmodul eingesteckt, hat der Hauptspeicher eine Größe von 319 Registern. Der Hauptspeicher des HP-41CV, für den Sie keine Speichererweiterungsmodule verwenden können, umfaßt immer 319 Register. Der gesamte Hauptspeicher kann beliebig in Programmspeicher und Datenspeicher eingeteilt werden.

## Hauptspeichereinteilung

Wird der Rechner zum ersten Mal, oder nachdem eine Gesamtlöschung (master clear) durchgeführt wurde, eingeschaltet, sind zunächst:

- 17 Register im HP-41C oder 273 Register im HP-41CV dem Datenspeicher zugeordnet;
- 46 unbenutzte Register für Programmspeicherung oder Tastenzuordnungen verfügbar.

Um die Anzahl der unbenutzten Programmspeicherregister zu sehen, drücken Sie  **GTO**   oder  **CATALOG** 1 und schalten den Rechner dann in den Programm-Modus.

00 REG 46 bedeutet, daß 46 Register unbenutzt sind und der Speicherung von Programmschritten oder Tastenbelegungen oder auch für eine Neueinteilung der Datenregister zur Verfügung stehen.

## Änderung der Hauptspeichereinteilung

Die Einteilung in Daten- und Programmspeicher kann mit **SIZE** geändert werden. Die Zahl, die dem Rechner nach der Eingabe der Funktion noch anzugeben ist, ist die Anzahl

\* Eigentlich besteht ein Register mehr, jedoch wird ein Teil dieses Registers für ein ständiges **.END.** am Ende des Programmspeichers benötigt.

der Register, die dem Datenspeicher zugeordnet werden. Die übrigen Register werden automatisch dem Programmspeicher zugeordnet.

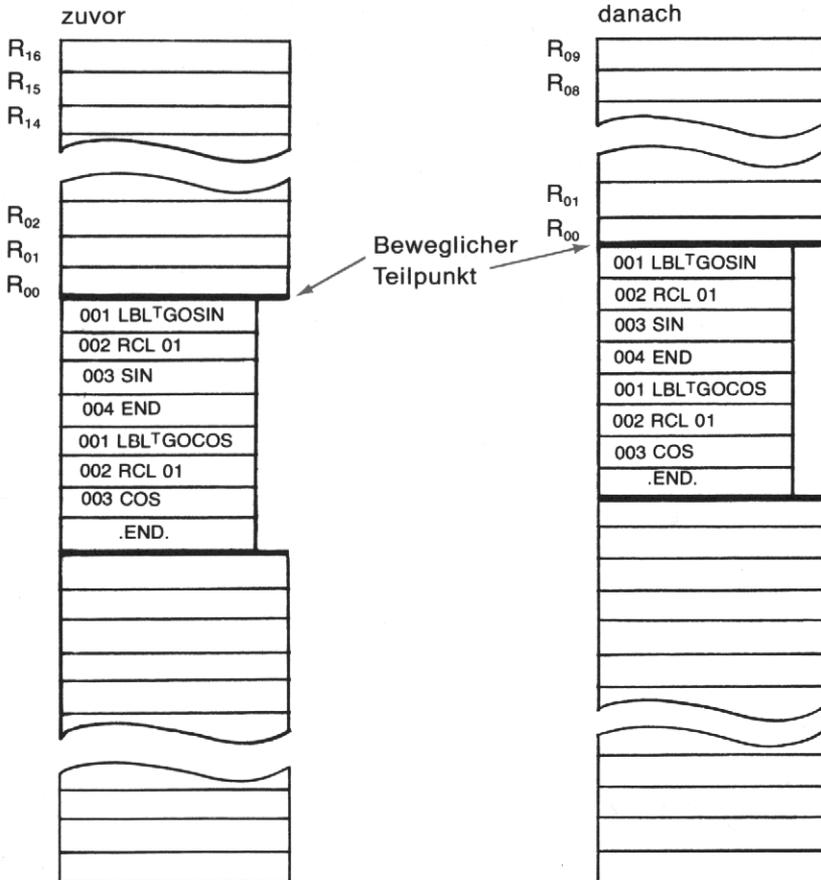
Für eine Änderung der Speicherplatzeinteilung :

1. Führen Sie **SIZE** aus. (**SIZE** ist nicht programmierbar.) In der Anzeige erscheint **SIZE \_\_\_\_**.
2. Geben Sie die Anzahl der Register ein, die dem Datenspeicher zugeordnet werden sollen: eine dreistellige Zahl zwischen 000 und 319 (oder die maximale Anzahl der Register im Hauptspeicher des Rechners).

Nach der Ausführung von **SIZE** werden sowohl gespeicherte Daten wie Programme mit der Speicherteilung im Hauptspeicher nach oben oder nach unten geschoben.

Wird nach **SIZE** eine kleinere Anzahl von Datenspeicher angegeben, als zuvor bestand, wird der Inhalt und die Teilung nach oben geschoben. Die Daten in den am höchsten nummerierten Registern gehen verloren.

Beispiel : Eine Speicherplatzeinteilung wie links würde durch die Ausführung von **010** wie folgt verändert :



Wird nach **SIZE** eine größere Anzahl von Datenspeichern angegeben, als zuvor bestand, wird der Inhalt und die bewegliche Teilung nach unten geschoben. Zeigt der Rechner **PACKING** und danach **TRY AGAIN** an, kann es sein, daß zu wenig unbenutzter Speicherplatz besteht, um die zusätzlichen Datenspeicher unterzubringen. Wird nach einer wiederholten Ausführung der **SIZE**-Funktion (mit derselben Angabe von Datenregister) dasselbe angezeigt, ist diese Speicherplatzeinteilung nicht möglich, sofern Sie nicht Programme oder Tastenzuordnungen löschen oder ein zusätzliches Speichererweiterungsmodul einstecken\*.

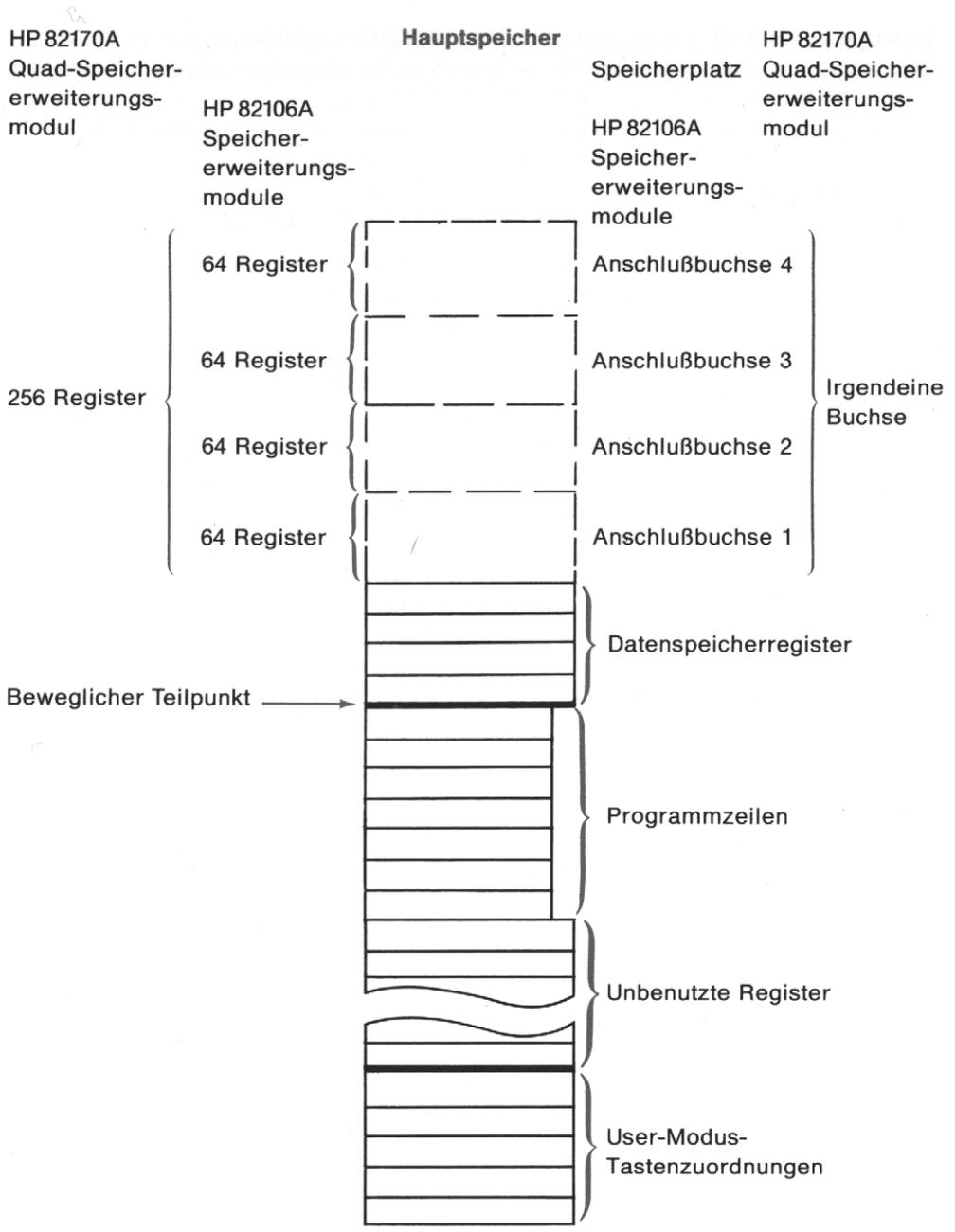
Um festzustellen, wieviele unbenutzte Register noch für Programmspeicherung nach einer **SIZE**-Ausführung bestehen, drücken Sie  **GTO**   und schalten Sie kurz den Programm-Modus ein. In der Anzeige erscheint dann **00 REG nn**, wobei **nn** die Anzahl der unbenutzten, für Programmspeicherung zur Verfügung stehenden, Register ist. Wird **SIZE** während dem Anhalten in einem Unterprogramm durchgeführt, « vergißt » der Rechner alle noch ausstehenden **RTN**- und **END**-(Rücksprung-)Befehle.

### Erweiterung des Hauptspeichers (nur HP-41C)

Mit dem Einstecken von bis zu vier HP 82106A Speichererweiterungsmodulen oder einem HP 82170A Quad-Speichererweiterungsmodul kann der Hauptspeicher maximal 319 Register erreichen.

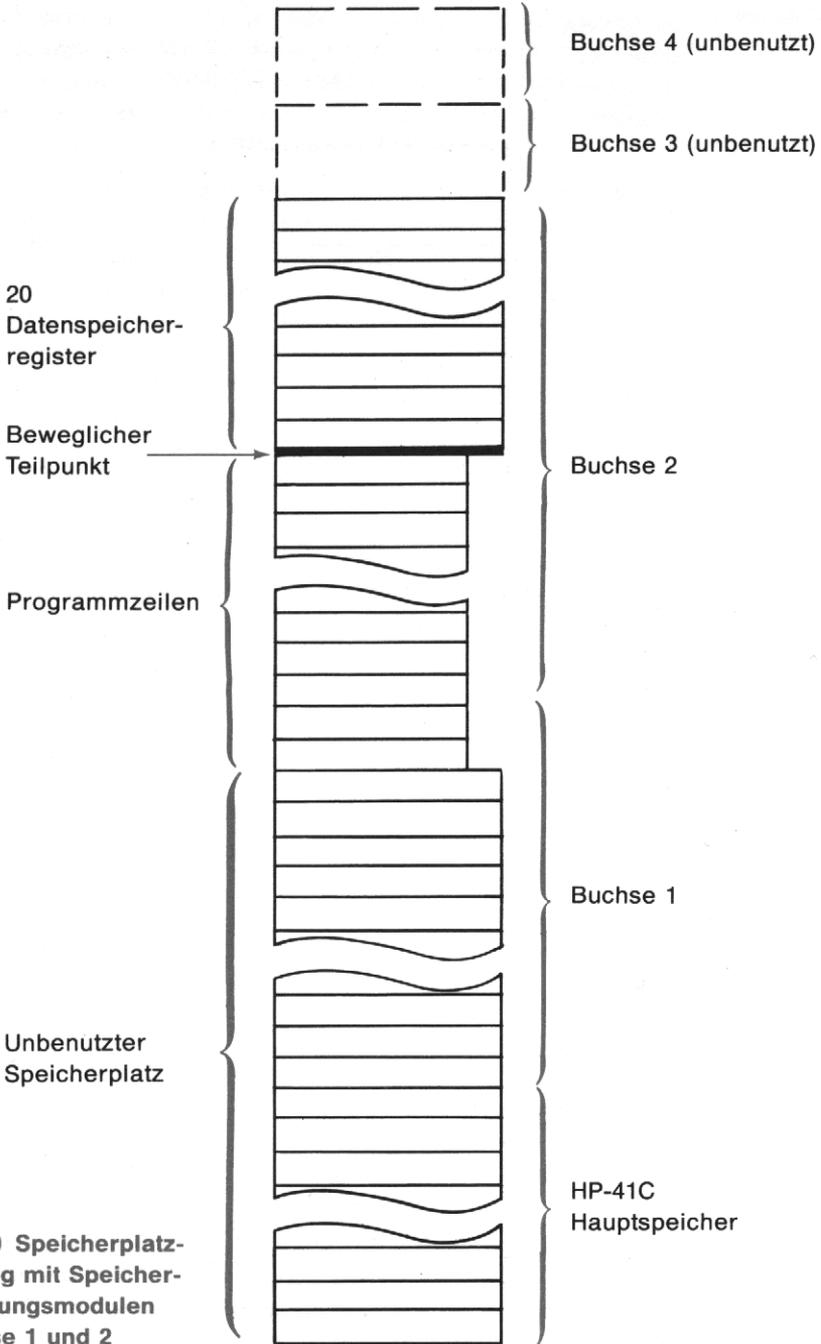
Es ist wichtig, daß beim Anschließen von Peripheriegeräten die Speichererweiterungsmodule HP 82106A zuerst in Buchse 1, dann in Buchse 2 usw. eingesteckt werden, und daß keine Lücken entstehen. Andere Peripheriegeräte sollten in die höher nummerierten Buchsen gesteckt werden.

\* Einstecken von Speichererweiterungsmodulen betrifft nur HP-41C.



Jedes weitere HP 82106A Speichererweiterungsmodul vergrößert den Datenspeicher um 64 Register. Das HP 82170 Quad-Speichererweiterungsmodul vergrößert den Datenspeicher um 256 Register. Mit **SIZE** kann dann eine neue Einteilung von Daten- und Programmspeicher erfolgen.

So würde der HP-41C mit zwei eingesteckten HP 82160A Speichererweiterungsmodulen nach der Ausführung von **SIZE 020** die untenstehende Registereinteilung haben :



**SIZE 020 Speicherplatzeinteilung mit Speichererweiterungsmodulen in Buchse 1 und 2**

Die folgende Tabelle gibt die anfängliche Einteilung und die höchste Anzahl von Registern an, die dem Datenspeicher oder Programmspeicher zugeordnet werden können.

HP-41C plus	Höchste Anzahl Datenspeicherregister	Anfängliche Einteilung		Höchste Anzahl Programm-speicherregister
		Daten-speicher	Programm-speicher	
Kein Modul	63	17	46	63 (445 Bytes)
1 Modul HP 82106A	127	81	46	127 (893 Bytes)
2 Module HP 82106A	191	145	46	191 (1341 Bytes)
3 Module HP 82106A	255	209	46	255 (1789 Bytes)
4 Module HP 82106A oder	319	273	46	319 (2237 Bytes)
1 Quad-Modul HP 82170A				

Das Entfernen von HP 82106A Speichererweiterungsmodulen sollte in umgekehrter Reihenfolge geschehen (d.h. mit dem in der höchsten numerischen Buchse beginnen), wenn der Rechner ausgeschaltet ist. Wird ein Speichererweiterungsmodul entfernt, geht die in ihm gespeicherte Information verloren.

Weiterhin ist zu beachten, beim Entfernen eines Speichererweiterungsmoduls, in dem sich der bewegliche Teilpunkt befindet, der gesamte Inhalt des Continuous Memory gelöscht wird. Dies wird durch **MEMORY LOST** angezeigt. Die folgende Tabelle gibt die Mindestanzahl von Datenspeichern an, die eingeteilt werden sollten, bevor ein oder mehrere Speichererweiterungsmodule entfernt werden. Ist die Anzahl der Datenspeicherregister kleiner als die angegebene Anzahl – oder falls Sie die Einteilung nicht genau wissen –, sollten Sie **SIZE** ausführen und eine Anzahl größer oder gleich der in der Tabelle angegebenen eingeben.

Zu entfernende Speichererweiterungsmodule	Mindestanzahl der Datenspeicherregister
1 HP 82106A Modul	064
2 HP 82106A Module	128
3 HP 82106A Module	192
4 HP 82106A Module oder	256
1 HP 82170A Quad-Modul	

## Löschen des Hauptspeichers

Immer wenn das Continuous Memory, das den Hauptspeicher, Flags, Rechnerstatus usw. enthält, gelöscht wird, beispielsweise wenn die Stromzufuhr längere Zeit unterbrochen

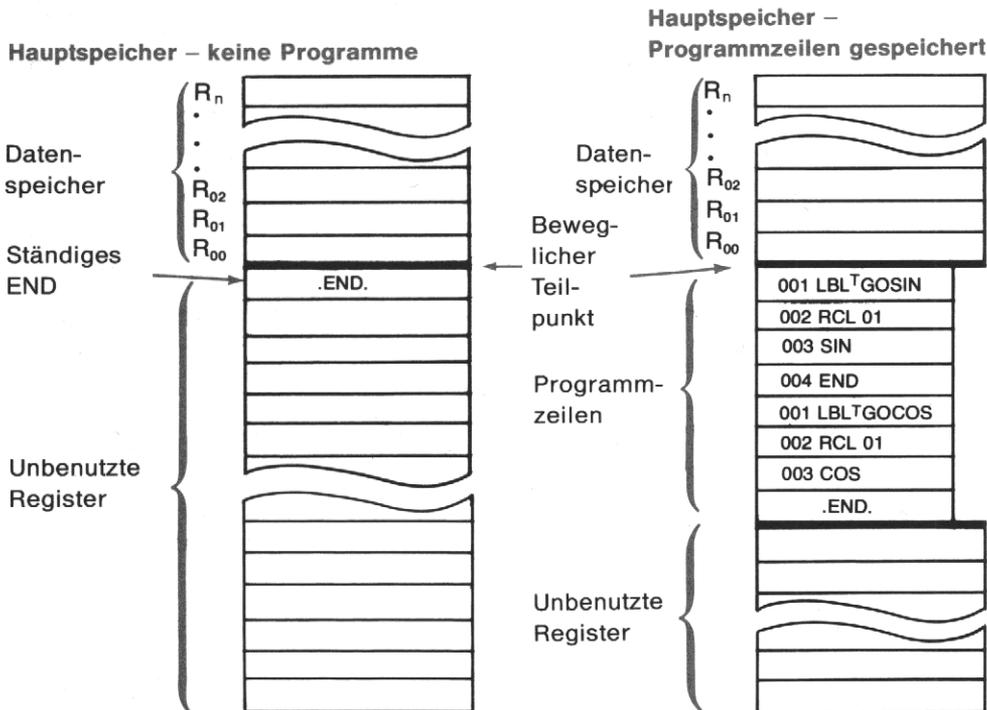
wurde, zeigt der Rechner **MEMORY LOST** an. Um das Continuous Memory absichtlich zu löschen :

1. Schalten Sie den Rechner aus.
2. Halten Sie die  $\leftarrow$ -Taste gedrückt und schalten Sie den Rechner ein.
3. Lassen Sie die  $\leftarrow$ -Taste los.

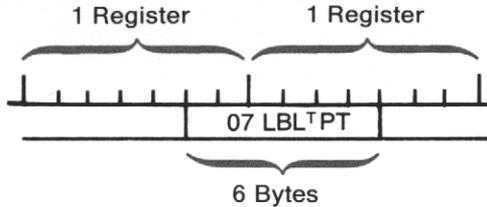
## Programmspeicher

In ein Programm geschriebene Anweisungen werden in unbenutzten Speicherregistern abgespeichert. Jedes Register enthält sieben Bytes. Funktionen belegen ein, zwei, drei oder vier Bytes – abhängig von der jeweiligen Funktion. Die Anzahl Bytes, die jede Funktion, belegt ist im Funktionsverzeichnis am Ende dieses Handbuchs angegeben. Zahlen benötigen ein Byte pro Ziffer, zusätzlich ein Byte für jedes  $\square$ , **[CHS]** und **[EEX]**, das mit dieser Zahl eingegeben wird. Alpha-Daten benötigen ein Byte für jedes Zeichen plus ein Byte für die gesamte Kette.

Jede Funktion, Zahl oder Alpha-Kette, die in ein Programm eingegeben wird, gilt als eine Programmzeile. Diese Zeilen werden fortlaufend nummeriert, wobei jedes Programm mit Zeile 1 beginnt. Die Anzahl der Programmzeilen hängt von der Anzahl der Funktionen, Zahlen und der Alpha-Ketten ab, wogegen die Anzahl der Register, die mit diesen Programmzeilen belegt werden, von den einzelnen Funktionen und der Länge der Zahlen und Alpha-Ketten abhängt.



Die Bytes, die durch eine Anweisung in einer Programmzeile belegt werden, müssen nicht unbedingt im selben Register enthalten sein. In der Abbildung zu Beispiel belegt die Anweisung **LBLPT** die letzten zwei Bytes in einem Register und die ersten vier Bytes im nächsten Register. Die restlichen Bytes in diesem Register würden durch die unmittelbar folgenden Anweisungen nach **LBLPT** belegt.



Unter bestimmten Umständen sind einige Bytes eines Registers unbelegt. Dies geschieht, wenn andere Anweisungen, die in diesem Register Bytes belegen, gelöscht werden und wenn eine Zahl in ein Programm eingegeben wird. Dasselbe kann geschehen, wenn eine Funktion, Zahl oder Alpha-Kette in ein Programm oder an das Ende eines Programmes, das nicht das letzte Programm im Programmspeicher ist, eingefügt wird (d.h. immer, wenn eine Anweisung irgendwo eingefügt wird, mit Ausnahme am Ende des Programmspeichers). Sind dort, wo zwischen belegten Bytes und den nachfolgenden Programmzeilen zusätzliche Anweisungen eingefügt werden sollen, keine unbenutzten Bytes verfügbar, werden alle nachfolgenden Anweisungen sieben Bytes nach unten geschoben und die neue Anweisung in die entsprechende Anzahl Bytes geschoben. Die restlichen Bytes dieses Registers bleiben unbenutzt.

Alle im Programmspeicher verteilten unbenutzten Bytes können nach der Ausführung von **PACK** verwendet werden. Wird der Programmspeicher gepackt, werden die Anweisungen aller Programme nach oben in unbenutzte Bytes geschoben, und – je nach Anzahl dieser Bytes – zusätzlich Register zur Programmspeicherung verfügbar. Der Programmspeicher wird automatisch gepackt, wenn **GTO**   ausgeführt wird, wenn ganze Programme gelöscht werden (mit **CLP**) und wenn Anweisungen eingefügt oder angehängt werden, für die augenblicklich nicht genügend unbenutzte Bytes zur Verfügung stehen.

## Alpha-Modus

Das Drücken der **ALPHA**-Taste schaltet den Rechner in den Alpha-Modus oder zurück. Ebenso kann der Rechner mit **AON** in den Alpha-Modus geschaltet werden und mit **AOFF** zurück.

Wenn der HP-41C im Alpha-Modus ist:

- Ist das Alpha-Tastenfeld aktiv. Dieses Tastenfeld ist auf der hinteren Umschlagseite dieses Handbuches wie auch auf der Unterseite des Rechners dargestellt. Zeichen und Funktionen, die dort über den Tasten stehen, werden ausgeführt, indem zuvor die **ALPHA**-Taste gedrückt wird.
- In der Anzeige wird anstelle des X-Registers das ALPHA-Register angezeigt. Zeichen, die eingegeben werden, werden automatisch in das ALPHA-Register aufgenommen – mit folgenden Ausnahmen:
  - in der Anzeige steht **ASN**, **CLP**, **GTO**, **LBL** oder **XEQ**, gefolgt von einem oder mehreren Unterstreichungszeichen. In diesem Fall können bis zu sieben Zeichen als Argument der Funktion eingegeben werden;
  - der Rechner ist im Programm-Modus.

Ist der Rechner sowohl im Programm-Modus als auch im Alpha-Modus, werden eingegebene Zeichen (höchstens 15) als Daten in eine Programmzeile geschrieben (außer, die Anzeige würde eine der fünf oben genannten Funktionen anzeigen). Die Anzeige der Programmzeile zeigt ein  $\tau$  nach der Zeilennummer um anzuzeigen, daß eine Alpha-Kette eingegeben wurde. Wird diese Zeile im Programm ausgeführt, wird das ALPHA-Register gelöscht und diese Daten in das ALPHA-Register aufgenommen (jedoch nicht automatisch in die Anzeige!).

### ALPHA-Register

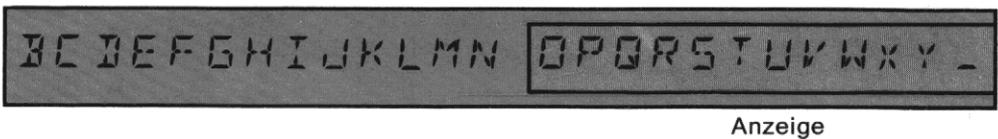
Das ALPHA-Register kann bis zu 24 Zeichen enthalten. Jeder Punkt, Doppelpunkt, Komma und Leerzeichen zählt als ein Zeichen. Die Anzeige kann nur 12 Zeichen des ALPHA-Registers gleichzeitig anzeigen (bzw. auch nur 11 plus das Unterstreichungszeichen), jedoch zählen hierbei Punkt, Doppelpunkt und Komma nicht dazu, da sie zwischen den anderen Zeichen stehen können.

Um Alpha-Zeichen in das angezeigte ALPHA-Register einzugeben, schalten Sie den Rechner in den Alpha-Modus und drücken die entsprechenden Tasten der Alpha-Tastatur. Zeichen werden von links nach rechts in die Anzeige geschrieben. Wenn Sie die

Eingabe fortsetzen, nachdem das elfte Zeichen eingegeben wurde, werden die Zeichen nach links verschoben. Diese aus der Anzeige verschobenen Zeichen bleiben im ALPHA-Register.



Bei der Eingabe des 24. Zeichens ertönt ein Summton als Warnung, daß die Eingabe eines weiteren Zeichens das erste Zeichen aus dem ALPHA-Register hinausschiebt, dieses somit verlorengeht. Wurden schon 24 Zeichen in das ALPHA-Register geschrieben, können Sie die Eingabe von weiteren Zeichen fortsetzen. Allerdings wird bei jedem zusätzlichen Zeichen der Summer ertönen und ein Zeichen auf der linken Seite des ALPHA-Registers verlorengehen.



Das Unterstreichungszeichen ( \_ ) zeigt an, daß Alpha-Zeichen, die eingegeben werden, an die bestehende Alpha-Kette angehängt werden. Dieses Unterstreichungszeichen ist nicht in der Anzeige, wenn die Alpha-Eingabe beendet ist.

Folgende Operationen beenden eine Alpha-Eingabe :

**ALPHA** **ASTO** **AVIEW** **BST** **CLA** **OFF** **ON** **R/S** **SST**

## Anzeige des ALPHA-Registers

Immer wenn der Rechner in den Alpha-Modus geschaltet wird (außer im Programm-Modus oder bei einem **AON** in einem laufenden Programm) werden vom Alpha-Register die ersten 12 Zeichen von links angezeigt, die restlichen Zeichen des ALPHA-Registers (sofern welche bestehen) werden dann durch die Anzeige « geschoben ».

Um den Inhalt des ALPHA-Registers in einem Programm anzuzeigen, ohne in den Alpha-Modus zu schalten, führen Sie **AVIEW** aus. Ist der Rechner bereits in den Alpha-Modus geschaltet worden, schließt **AVIEW** die Alpha-Eingabe ab, bevor das ALPHA-Register angezeigt wird.

Ist der Rechner nicht im Alpha-Modus, beendet **↵** die Anzeige des ALPHA-Registers und bringt den Inhalt des X-Registers wieder in die Anzeige (dasselbe gilt auch für **VIEW**). Ebenso beendet das Drücken einer beliebigen Taste die Anzeige des ALPHA-Registers, zusätzlich wird die Funktion der gedrückten Taste ausgeführt.

Ist der Rechner im Alpha-Modus, löscht  nach einem **AVIEW** natürlich das ganze ALPHA-Register.

In einem laufenden Programm wird das ALPHA-Register angezeigt, wenn **AVIEW** ausgeführt wird. Es wird nicht angezeigt, wenn nur **AON** ausgeführt wird, außer, wenn das Programm unterbrochen wird – etwa mit **PSE** oder **STOP**. Die Anzeige des ALPHA-Registers kann mit der **CLD**-Anweisung gelöscht werden; diese Anweisung bringt den  $\rightarrow$ -Indikator (Marken ausführenden) wieder in die Anzeige.

## Löschen und Korrigieren von Alpha-Ketten

### Löschen des ALPHA-Registers

 **CLA** (clear Alpha) löscht das gesamte ALPHA-Register, auch den Teil, der nicht angezeigt wird.

Ist die Alpha-Eingabe beendet (wenn kein Unterstreichungszeichen in der Anzeige erscheint), wirkt  wie  **CLA**.

Nach beendeter Eingabe wird das ALPHA-Register automatisch gelöscht, wenn ein neues Zeichen vom Tastenfeld aus eingegeben wird oder wenn das Programm eine Anweisung ausführt, die nur aus einer Alpha-Kette besteht.

### Korrektur von Alpha-Ketten

Ist die Alpha-Eingabe noch nicht beendet (wenn also das Unterstreichungszeichen in der Anzeige erscheint):

- Werden Zeichen eingegeben, wenn die entsprechende Taste gedrückt wird.
- Löscht  das am weitesten rechts stehende Zeichen.

Um Zeichen im ALPHA-Register zu löschen oder anzuhängen, nachdem die Eingabe schon beendet ist, reaktivieren Sie zuerst die Alpha-Eingabe durch  **APPEND**\*. Dies bringt das Unterstreichungszeichen wieder in die Anzeige.

Soll ein Programm Zeichen an eine schon im ALPHA-Register stehende Alpha-Kette anhängen, drücken Sie  **APPEND** im Programm-Modus und geben Sie die entsprechenden Zeichen in diese Programmzeile. Die Anzeige dieser Programmzeile zeigt ein  $\vdash$  nach dem  $\tau$ , was bedeutet, daß diese Alpha-Kette an die schon im ALPHA-Register stehende angehängt wird. Erscheint dieses  $\vdash$  vor der Alpha-Kette nicht (d.h. **APPEND** wurde nicht in diese Programmzeile eingegeben), ersetzen diese Alpha-Zeichen die bestehende Kette des ALPHA-Registers und werden nicht angehängt.

\* **APPEND** ist eine umgeschaltete Funktion auf dem Alpha-Tastenfeld. Um diese Funktion auszuführen, schalten Sie den Rechner in den Alpha-Modus und drücken Sie  K.

## Verschiebung von Ketten im ALPHA-Register

**ASHF** (Alpha shift) verschiebt alle Zeichen im ALPHA-Register um sechs Stellen nach links. Die sechs Zeichen, die zuvor am weitesten links standen, gehen dabei verloren.

## Abspeichern von Alpha-Ketten aus dem ALPHA-Register

Alpha-Ketten im ALPHA-Register können in jedes Datenspeicherregister, Stackregister oder ins LAST X-Register mit **ASTO** (Alpha store) abgespeichert werden.

Um die sechs am weitesten links stehenden Zeichen aus dem ALPHA-Register in ein Register abzuspeichern:

1. Führen Sie **ASTO** aus oder drücken Sie  **ASTO** (im Alpha-Modus).
2. Geben Sie – direkt oder indirekt – das gewünschte Register an, wie Sie es mit **STO** auch machen würden.

**ASTO** kopiert die sechs am weitesten links stehenden Zeichen vom ALPHA-Register in das angegebene Register. Diese Zeichen bleiben unverändert auch im ALPHA-Register.

Müssen weitere Zeichen aus dem ALPHA-Register abgespeichert werden, führen Sie **ASHF** aus, um die nächsten sechs Zeichen an das linke Ende des ALPHA-Registers zu bringen. Führen Sie dann **ASTO** aus. Die wiederholte Ausführung von **ASHF** und **ASTO** speichert die nächsten sechs Zeichen aus dem ALPHA-Register und die nochmalige Ausführung speichert die letzten sechs Zeichen. Da jedes Register nur sechs Zeichen aufnehmen kann, muß nach jedem **ASTO** ein anderes Register angegeben werden.

Numerische Zeichen (genau wie Alpha-Zeichen), die vom ALPHA-Register mit **ASTO** abgespeichert wurden, werden auch als alphanumerische Daten und nicht als numerische Daten behandelt. Sie können folglich auch nicht für mathematische Operationen verwendet werden (wie z.B. Speicherarithmetik). Ein derartiger Versuch erzeugt eine **ALPHA DATA** Fehlermeldung. Allerdings können Alpha-Ketten (alphabetische wie numerische) nach der Speicherung ins X- und Y-Register mit **X=Y?** und **X≠Y?** verglichen werden.

## Zurückrufen von Daten ins ALPHA-Register

Um Daten (numerische oder Alpha-) aus einem Register ins ALPHA-Register zurückzurufen:

1. Führen Sie **ARCL** aus oder drücken Sie  **ARCL** (im Alpha-Modus).
2. Geben Sie – direkt oder indirekt – das gewünschte Register an wie Sie es mit **RCL** auch machen würden.

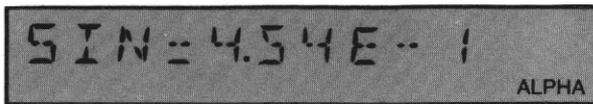
Die zurückgerufenen Daten werden automatisch an die rechte Seite des bestehenden Inhalts des ALPHA-Registers angehängt. Ist die sich ergebende Kette länger als 24

Zeichen, bleiben die 24 am weitesten rechts stehenden Zeichen im ALPHA-Register, die übrigen Zeichen gehen verloren. Daten in den angegebenen Registern werden nicht verändert.

Enthält das nach **ARCL** angegebene Register numerische Daten, wird die Kopie dieser Daten als Alpha-Daten behandelt. Jedoch werden die Zeichen, die im ALPHA-Register angehängt werden durch das Anzeigeformat bestimmt (FIX, SCI oder ENG). Ist das Anzeigeformat SCI oder ENG, wird der Exponentenangabe ein **E** vorangestellt und Leerzeichen vor dem Exponenten weggelassen. Die Zahl  $1,23456789 \times 10^{44}$  wird beispielsweise wie folgt in das ALPHA-Register zurückgerufen :

Anzeigeformat	Angezeigt im X-Register	Entsprechende Daten im ALPHA-Register
SCI 2	1.23 44	1.23E44
SCI 7	1.2345679 44	1.2345679E44
SCI 9	1.2345678 44	1.234567890E44

Numerische Daten – genau wie Alpha-Daten – werden beim Zurückrufen ins ALPHA-Register an die Daten angehängt, die schon im ALPHA-Register bestehen. Beispiel:



Schon bestehender  
Inhalt im  
ALPHA-Register

Mit **ARCL**  
angehängte Daten

Im Unterschied zu **RCL** beendet **ARCL** nicht die Alpha-Dateneingabe.

## User-Modus

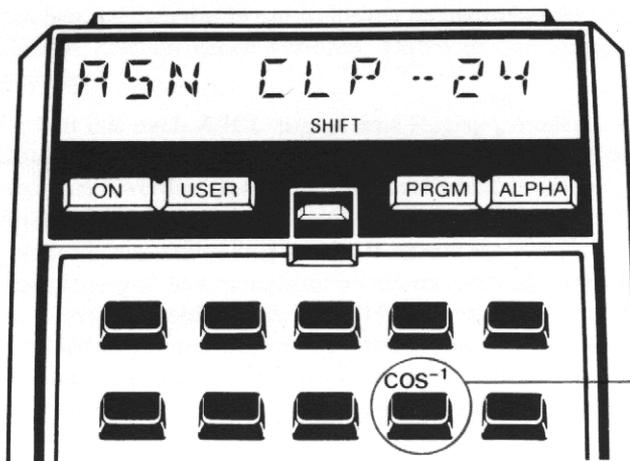
Der User-Modus gibt Ihnen die Möglichkeit, Ihren Rechner persönlich zu gestalten. Programme, Unterprogramme, Rechnerfunktionen oder Funktionen von angeschlossenen Peripheriegeräten können beliebigen Tasten des HP-41C zugeordnet werden. Das Einschalten des User-Modus aktiviert diese Funktionszuordnungen. Um die auf solche Art neu belegten Tasten des Rechners zu kennzeichnen können Tastenfeldschablonen mit entsprechenden Aufklebern verwendet werden.

### Zuordnen von Funktionen oder Programmen

Um eine Funktion oder ein Programm einer Taste zuzuordnen :

1. Drücken Sie  **[ASN]** im Normal oder User-Modus. Der Rechner zeigt dann **ASN\_** an.
2. Drücken Sie **[ALPHA]**, um den Rechner in den Alpha-Modus zu schalten.
3. Tasten Sie die Funktionsbezeichnung oder den Namen der Programmarke ein, die zugeordnet werden soll. Eine Programmarke, die zugeordnet werden soll, muß eine globale Alpha-Marke sein, nicht eine lokale oder numerische Marke. (Die verschiedenen Marken sind auf Seite 38 beschrieben.) Lokale Alpha-Marken sind automatisch einigen Tasten zugeordnet (siehe Seite 33).
4. Drücken Sie **[ALPHA]**, um die Alpha-Eingabe zu beenden.
5. Drücken Sie die Taste (oder  und die Taste), der die Funktion oder der Programmnamen zugeordnet werden soll. Wenn Sie diese Taste gedrückt halten, zeigt die Anzeige die Funktionsbezeichnung oder die Programmarke an, die zugeordnet wurde und den Tasten-Code der neu zugeordneten Taste. Halten Sie die Taste solange gedrückt, bis **NULL** erscheint, wird die Zuordnung nicht ausgeführt.

Der Tasten-Code ist eine Zeilen-/Spaltenangabe der Position der zugeordneten Taste. Zum Beispiel ist der Tasten-Code für die **[COS]**-Taste 24, das ist die Taste in der zweiten Zeile und vierten Spalte. Den Tasten-Codes für umgeschaltete Tasten wird ein Minus-Zeichen vorangestellt.



Ordnet die **CLP**-Funktion der  $\boxed{\text{COS}^{-1}}$ -Tastenposition zu.

Taste in der  
2. Zeile  
4. Spalte

Insgesamt sind 68 Tastenpositionen für eine User-Modus-Tastenzuordnung verfügbar. Die einzigen Tasten, denen keine Funktion oder Programm zugeordnet werden können sind  $\blacksquare$ ,  $\boxed{\text{ON}}$ ,  $\boxed{\text{USER}}$ ,  $\boxed{\text{PRGM}}$  und  $\boxed{\text{ALPHA}}$ .

Jede Funktionsbezeichnung oder Programmarke die in einem der drei Verzeichnisse ( $\blacksquare$   $\boxed{\text{CATALOG}}$ ) erscheint, kann einer Taste zugeordnet werden. Wird versucht, eine Funktion oder ein Programm zuzuordnen, das sich nicht in einem der drei Verzeichnisse befindet, wird **NONEXISTENT** angezeigt.

Um eine Zuordnung rückgängig zu machen (dieser Taste also wieder die ursprüngliche Funktion zuzuordnen), drücken Sie  $\blacksquare$   $\boxed{\text{ASN}}$   $\boxed{\text{ALPHA}}$   $\boxed{\text{ALPHA}}$  und diese Taste. Die Normal-Modus-Funktion dieser Taste wird dann auch wieder im User-Modus ausgeführt.

#### Beispiel:

$\blacksquare$   $\boxed{\text{ASN}}$   $\boxed{\text{ALPHA}}$  STO  $\boxed{\text{ALPHA}}$   $\boxed{1/x}$

Ordnet die **STO**-Funktion der  $\boxed{1/x}$ -Taste zu.

$\blacksquare$   $\boxed{\text{ASN}}$   $\boxed{\text{ALPHA}}$  CIRCLE  $\boxed{\text{ALPHA}}$   $\blacksquare$   $\boxed{e^x}$

Ordnet den Programmnamen **CIRCLE** der  $\boxed{e^x}$ -Taste zu. (Dieses Programm muß sich im Programmspeicher befinden.)

$\blacksquare$   $\boxed{\text{ASN}}$   $\boxed{\text{ALPHA}}$   $\boxed{\text{ALPHA}}$   $\boxed{1/x}$

Ordnet der  $\boxed{1/x}$ -Taste die ursprüngliche Funktion wieder zu. (Auch im User-Modus wird auf Tastendruck nun wieder  $\boxed{1/x}$  ausgeführt.)

Bitte beachten Sie stets, daß zugeordnete Funktionen nur im User-Modus auf den entsprechenden Tasten aktiv sind, im Normal-Modus wird nur die auf der Taste angegebene Funktion ausgeführt.

## Operationen im User-Modus

Mit dem Drücken der **USER**-Taste wird der User-Modus ein- oder ausgeschaltet. Im User-Modus werden Funktionen und Programme auf den zugeordneten Tasten wirksam. So kann eine einzige Taste des Tastenfeldes für vier verschiedene Funktionen verwendet werden. Im Normal- und User-Modus sind mit der -Taste Doppelbelegungen möglich.

Wenn Sie im User-Modus eine Taste drücken, der eine Funktion zugeordnet wurde, entspricht dies der Ausführung der Funktion mit **XEQ** und der Eingabe der Funktionsbezeichnung. Ebenso wird, wenn eine Taste gedrückt wird, der eine Programmarke zugeordnet wurde, das Programm ab der entsprechenden Marke ausgeführt – genauso wie wenn Sie das mit **XEQ** und der Angabe der Programmarke gemacht hätten.

Wenn einer Taste keine Funktion oder Programmarke zugeordnet wurde, wird auf Tastendruck – im User-Modus wie im Normal-Modus – die auf (oder über) der Taste angegebene Funktion ausgeführt (mit Ausnahme der lokalen Marken – siehe unten).

Um zu überprüfen, ob einer Taste eine Funktion oder ein Programm zugeordnet wurde, drücken Sie diese Taste im User-Modus und halten Sie sie gedrückt, bis **NULL** erscheint (damit die Funktion nicht ausgeführt wird). In der Anzeige erscheint dann die Funktionsbezeichnung bzw. der Programmname, der auf dieser Taste aktiv ist. Dem Programmnamen ist ein <sup>T</sup> vorangestellt.

Die Tastenzuordnung jeder Taste bleibt durch das Continuous Memory erhalten, bis der entsprechenden Taste eine andere Funktion oder ein anderes Programm oder wieder die Normal-Modus-Funktion zugeordnet wird.

Zuordnungen von Funktionen oder Programmen, die in den Verzeichnissen 2 oder 3 aufgeführt werden, benötigen ein Register (sieben Bytes) für jede ungerade Anzahl von Zuordnungen, die gemacht wurden. Beispielsweise benötigt die erste Zuordnung ein Register, die zweite Zuordnung keinen zusätzlichen Platz, die dritte Zuordnung wieder ein Register und die vierte Zuordnung keinen zusätzlichen Platz usw. Zuordnungen von Programmen, die im Verzeichnis 1 aufgeführt sind benötigen keinen Platz im Programmspeicher, außer den Platz für das Programm selbst.

## Zuordnungen lokaler Marken

Ist der Rechner im User-Modus, sind den Tasten der oberen zwei Reihen automatisch lokale Marken zugeordnet (sofern nicht eine andere Funktion oder Programmarke vorher dieser Taste zugeordnet wurde). Die zugeordnete Marke entspricht dem Buchstaben, der im Alpha-Modus wirksam ist : A bis E in der oberen Reihe, F bis J in der zweiten Reihe und a bis e in der oberen Reihe nachdem  gedrückt wurde.

Da dieses lokale und keine globalen Marken sind, beginnt der Rechner mit der Ausführung des Programmes an dieser Marke nur, wenn diese Marke in dem Programm,

in dem der Rechner gerade steht, enthalten ist\*. Ist der Rechner im User-Modus und eine der Tasten wird gedrückt, wird eine Funktion nach der folgenden Priorität ausgeführt:

1. Wurde ein Programm oder eine Funktion dieser Taste zugeordnet, so wird diese ausgeführt.
2. Besteht eine lokale Marke dieses Buchstabens im Programm, in dem der Rechner gerade steht, beginnt der Rechner mit der Ausführung des Programms an dieser Stelle.
3. Trifft keine der vorhergehenden Bedingungen zu, wird die Funktion ausgeführt, die auf der Taste (oder darüber) angegeben ist. Die Ausführung dieser Funktion kann im User-Modus bedeutend länger dauern als im Normal-Modus – bedingt durch die Zeit, die für die Suche nach lokalen Marken aufgewendet wird. Um Zeit zu sparen gehen Sie am besten aus dem User-Modus heraus, wenn Sie eine Normal-Modus-Funktion ausführen wollen.

Besteht im Programm, in dem der Rechner steht, eine lokale Marke dieses betreffenden Buchstabens, zeigt die Funktionsanzeige ein **XEQ** gefolgt von der Marke an. Andernfalls wird die Normal-Modus-Funktion (bzw. die zugeordnete Funktion/Programmarke) angezeigt.

\*Die Suche nach lokalen Marken wird nur in einem Programm ausgeführt, in dem der Rechner gerade steht (d.h. zwischen **END**-Anweisungen). (Weitere Information auf Seite 47.)

## Programmiergrundlagen

### Eingeben eines Programmes

Um ein Programm in den Rechner einzugeben :

1. Schalten Sie den Rechner in den Programm-Modus (drücken Sie **PRGM**).
2. Drücken Sie **■** **GTO** **□** **□** um den Rechner an eine unbenutzte Stelle des Programmspeicherplatzes zu stellen. Die Anzeige zeigt die Anzahl der unbenutzten Programmspeicherregister an, die für neue Programmanweisungen verfügbar sind.
3. Drücken Sie **■** **LBL** und geben Sie eine globale Alpha-Marke ein.
4. Geben Sie Funktionen, Zahlen und Alpha-Ketten genau wie im Normal-Modus ein. Im Programm-Modus werden diese Anweisungen allerdings nicht ausgeführt sondern in der Reihenfolge der Eingabe gespeichert.

Sie werden nun zwischen Normal-, Alpha- und User-Modus hin- und herschalten, so wie Sie es auch bei der Lösung von Aufgaben tun. Alle Operationen des HP-41C, auch die Funktionen, die nicht auf dem Tastenfeld ersichtlich sind, die Alpha-Modus-Funktionen und die zugeordneten Funktionen im User-Modus, sind programmierbar mit den folgenden Ausnahmen :

**CLP** (Programm löschen)

**←** (Korrektur)

**BST** (Einzelschritt zurück)

**SST** (Einzelschritt vorwärts)

**DEL** (Löschen von Programmzeilen)

**ASN** (Zuordnen)

**USER** (User-Modus-Taste)

**PACK** (Packen des Programmspeichers)

**SIZE** (Anzahl der Datenregister)

**PRGM** (Programm-Modus-Taste)

**GTO** **□** (Schritt zu einer Zeilennummer)

**CATALOG** (Verzeichnisauflistung)

**ON** (keine automatische Ausschaltung)

**ON** (Einschalttaste)

**COPY** (Kopieren von Programmen)

**GTO** **□** **□** (Springen ans Ende des Programmspeichers)

**■** **GTO** **□** **□** bewirkt folgendes :

- Ist die letzte Anweisung im Programmspeicher keine **END**-Anweisung, fügt der Rechner eine **END**-Anweisung ans Ende des letzten Programmes an. Dies trennt das letzte Programm im Speicher von den Programmen, die danach eingegeben werden.
- Der Rechner wird an das Ende des Programmspeichers positioniert (d.h. nach dem letzten Programm im Programmspeicher, am Anfang des unbenutzten Speichers).
- Der Programmspeicher wird gepackt. Das bedeutet, daß Programmanweisung nach oben in unbenutzte Bytes geschoben werden, die sich im Programmspeicher befanden (z.B. aus Einfügungen oder Löschen von Zeilen).

- Ist der Rechner im Programm-Modus, wird **00 REG** angezeigt, gefolgt von einer zwei- oder dreistelligen Zahl, die die Anzahl der unbenutzten Register im Programmspeicher angibt.

Nach dem Anfügen von Anweisungen an das Ende des Programmspeichers können Sie feststellen, wieviele Register unbenutzt sind, indem Sie **[SST]** drücken. Der Rechner zeigt dann **.END.REG** an, gefolgt von der Anzahl der unbenutzten Register. Erneutes Drücken von **[SST]** stellt den Rechner in Zeile 01 des gegenwärtigen Programmes und ermöglicht Ihnen so, das gesamte Programm mit **[SST]** nochmals anzusehen. **[BST]** stellt den Rechner zurück an die letzte Zeile des gegenwärtigen Programmes, dies ermöglicht Ihnen, Anweisungen anzuhängen.

Die Anzahl der unbenutzten Register, die für Programmanweisungen zur Verfügung stehen, können auf verschiedene Weisen erhöht werden :

- Führen Sie **PACK** aus, um den Programmspeicher zu packen. Dies macht vor allem dann neue Register verfügbar, wenn seit der letzten PACK-Ausführung Anweisungen eingefügt oder gelöscht wurden oder wenn Tastenbelegungen (von Funktionen aus Verzeichnis 2 oder 3) aufgehoben wurden.
- Führen Sie **SIZE** aus und geben Sie eine kleinere Anzahl von Datenregistern an, als augenblicklich eingeteilt sind.
- Löschen Sie Programme (mit **CLP**).
- Löschen Sie einzelne oder mehrere Programmzeilen und führen Sie anschließend **PACK** durch.
- Heben Sie Tastenzuordnungen von Funktionen aus Verzeichnis 2 oder 3\* auf und führen Sie danach **PACK** aus.
- Stecken Sie ein zusätzliches Speichererweiterungsmodul ein und führen danach **SIZE** mit der entsprechenden Anzahl von Datenregistern aus (nur HP-41C, siehe Seite 20).

Bestehen nicht genügend Bytes um eine Programmanweisung, die eingegeben wird, in den Programmspeicher aufzunehmen, packt der Rechner den Programmspeicher und zeigt **TRY AGAIN** an. Zeigt der Rechner **TRY AGAIN** an, nachdem Sie die Anweisung erneut eingegeben haben, können keine neuen Anweisungen eingegeben werden bevor nicht weitere Register in der oben beschriebenen Weise verfügbar gemacht werden.

Ein Programm, das am Ende des Programmspeichers eingegeben wird (d.h. nachdem **[GTO]** ausgeführt wurde) muß nicht mit einer **END-** oder **RTN-**Anweisung abgeschlossen werden. Eine spezielle **END-**Anweisung besteht immer am Ende des Programmspeichers und belegt drei Bytes des letzten Registers vor unbenutzten Registern\*\*. Diese Anweisung kann nicht gelöscht werden und es können keine Anweisungen nach diesem **END** eingegeben werden. Diese **END-**Anweisung wird als **.END.** angezeigt.

\* Die Aufhebung einer Tastenzuordnung macht nur dann zusätzliche Register verfügbar, wenn eine ungerade Anzahl von Tastenzuordnungen bestand (siehe Seite 33).

\*\* Deshalb können noch bis zu sechs unbelegte Bytes zwischen der letzten Anweisung des Programmspeichers und der ständigen **END-**Anweisung stehen.

## Ausführung eines Programmes

Um ein Programm zu starten wird der Rechner zuerst aus dem Programm-Modus zurückgeschaltet, und dann initialisiert (d.h. Daten in die entsprechenden Daten- oder Stackregister eingegeben). Ein Programm kann dann auf verschiedene Arten ausgeführt werden :

1. Durch Drücken von **[XEQ]** und der Angabe des Programmnamens.
2. Durch Zuordnen des Programmnames mit **[ASN]** zu einer Taste. Drücken dieser Taste im User-Modus.
3. Indem der Rechner an den Anfang des Programmes gestellt und **[R/S]** oder **[SST]** gedrückt wird. **[R/S]** startet die Durchführung des Programmes von der augenblicklichen Zeile ab. **[SST]** führt nur die augenblickliche Programmzeile aus und geht einen Schritt weiter.

Ist ein Programm gestartet, werden die Anweisungen im Programmspeicher ausgeführt bis der Rechner auf eine **END-** (oder **RTN-**)Anweisung trifft oder das Programm angehalten wird. Während der Ausführung erscheint in der Anzeige der **PRGM-**Indikator und der **→** (markenausführende)-Indikator. Jedes Mal, wenn das Programm an eine Marke kommt, rückt **→** um eine Position nach rechts. Nachdem dieses Zeichen die letzte Position auf der rechten Seite erreicht hat springt es wieder zur linken Seite und läuft erneut durch die Anzeige. **→** verschwindet nach einer **VIEW-** oder **AVIEW-**Anweisung. Der Inhalt des Registers, das mit dieser Anweisung betrachtet wird, bleibt in der Anzeige stehen bis eine **CLD-**Anweisung (clear display) oder eine nächste **VIEW-** oder **AVIEW-**Anweisung ausgeführt wird. **CLD** bringt wieder **→** in die Anzeige.

## Programmkomponenten

### Programmzeilen

Im Programm-Modus wird immer eine Programmzeile angezeigt. Jede Zeile enthält eine abgeschlossene Anweisung, bestehend aus: 1) einer Funktion; 2) einer Alpha-Kette bis zu 15 Zeichen oder 3) einer Zahl (bis zu 10 Stellen plus einem zweistelligen Exponenten)\*.

Die Zeilen werden mit dem Eingeben der Anweisungen automatisch geschrieben und in den Programmspeicher gespeichert. Jeder Zeile ist eine Zeilennummer zugeordnet. Jedes einzelne Programm im HP-41C beginnt mit Zeilennummer 01.

Muß eine Funktion mit ihrer Funktionsbezeichnung über die Anzeige in eine Programmzeile geschrieben werden (also nicht durch Drücken einer Taste), muß **[XEQ]** gedrückt werden, bevor die Funktionsbezeichnung in die Anzeige geschrieben wird. Wird **[XEQ]** nicht vorher gedrückt, kann die Bezeichnung nicht als eine Funktion erkannt werden,

```

01 BLTAREA
02 X↑2
03 PI
04 *
05 END
01 LBL'ZOT
02 TANS =
03 ARCL 00
04 END
    
```

\* Wie im Normal-Modus muß für eine Zahl, die aus 9 oder 10 Ziffern plus einem Exponenten bestehen soll, vor der neunten Stelle ein Komma eingegeben werden.

sondern wird als Eingabe einer Alpha-Kette behandelt und ins ALPHA-Register geschrieben, wenn die Zeile ausgeführt wird.

In der Anzeige einer Programmzeile im Programm-Modus weist ein  $\tau$  daraufhin, daß die folgenden Zeichen eine Alpha-Kette darstellen. Dieses  $\tau$  erscheint ebenfalls nach einer **XEQ**- oder **GTO**-Sprunganweisung zu einer Alpha-Marke und nach dem **LBL** einer Alpha-Marke (mit Ausnahme der lokalen Marken).

## Marken

Marken werden durch Drücken von  **LBL** in den Programmspeicher eingegeben, gefolgt von der gewünschten Markenbezeichnung. Marken verwendet man um komplette Programme zu benennen oder um Schleifen oder Unterprogramme abzugrenzen. HP-41C Marken können entweder aus (zweistelligen) Zahlen (numerische Marken) oder aus Alpha-Zeichen (Alpha-Marken) bestehen. Sie werden weiterhin eingeteilt in lokale Marken und globale Marken in Übereinstimmung mit der unterschiedlichen Art, in der nach diesen Marken gesucht wird (siehe auch Seite 46).

**Numerische Marken.** Numerische Marken bestehen aus zwei Ziffern. Man kann zwei Arten unterscheiden :

- Marken LBL 00 bis 14 benötigen nur ein Byte Programmspeicherplatz ;
- Marken LBL 15 bis 99 benötigen zwei Bytes im Programmspeicherplatz.

**Lokale Alpha-Marken.** In einem Programm gelten die Marken mit den einzelnen Buchstaben A bis J und a bis e als lokale Marken. Sie sollten nicht als Programm-Marken (zur Bezeichnung eines Programmes) verwendet werden sondern nur innerhalb eines Programmes. Lokale Alpha-Marken benötigen zwei Bytes Programmspeicherplatz.

**Globale Alpha-Marken.** Globale Alpha-Marken können aus jeder Kombination von bis zu sieben Alpha-Zeichen bestehen (inklusive Ziffern) mit Ausnahme von Komma (,), Punkt (.) und Doppelpunkt (;). Die einzelnen Buchstaben A bis J und a bis e gelten als lokale und nicht als globale Marken. Bei der Anzeige einer Programmzeile im Programm-Modus steht zwischen **LBL** und der globalen Marke ein  $\tau$ . Globale Marken benötigen vier Bytes Programmspeicherplatz für die **LBL**-Anweisung plus ein weiteres Byte für jeden Buchstaben der Marke.

Der Rechner behandelt die globalen Marken auf verschiedene Weisen anders als lokale Marken :

- Eine globale Marke kann von jedem Punkt des Programmspeichers aus angesteuert werden.
- Globale Marken erscheinen im Katalog 1.
- Nur ein Programm mit einer globalen Alpha-Marke kann einer Taste (für die User-Modus-Ausführung) zugeordnet werden.

Aus diesen Gründen ist es empfehlenswert, ein Programm mit einer globalen Alpha-Marke zu beginnen.

## Programmkorrektur

### Positionieren im Programmspeicher

#### Positionieren mit **GTO** **•**

Unabhängig davon, ob der HP-41C im Programm-Modus ist oder nicht :

- setzt **■** **GTO** **•** und danach eine globale Alpha-Marke den Rechner an diese Stelle im Programmspeicher. Die Suche nach dieser Marke beginnt am Ende des Programmspeichers in Richtung Anfang, und der Rechner hält an, sobald er eine Marke dieser Bezeichnung trifft. Findet der Rechner diese Marke nicht, zeigt er **NONEXISTENT** an und ist wieder am vorhergehenden Punkt im Programmspeicher ;
- setzt **■** **GTO** **•** und danach eine dreistellige Zahl den Rechner an diese Zeilennummer in dem Programm, in dem er sich gerade befindet.

Zeilennummern größer als 1000 sprechen Sie mit **■** **GTO** **•** dann **EEX** an und geben danach die letzten drei Ziffern der Zeilenzahl ein.

Geben Sie nach **■** **GTO** **•** eine Zahl an, die größer ist als die höchste Zeilennummer, wird der Rechner nur an die letzte Zeilennummer des betreffenden Programms gesetzt.

#### Positionieren mit **RTN**

Ist der Rechner nicht im Programm-Modus können Sie ihn mit **■** **RTN** an die Zeile 00 des gegenwärtigen Programms stellen.

#### Positionieren mit **CATALOG** 1

Um den Rechner an eine Programmzeile mit einer globalen Alpha-Marke oder einer **END**-Anweisung zu positionieren :

1. Drücken Sie **■** **CATALOG** 1 um die globalen Marken und **END**-Anweisungen aufzulisten, die sich im Programmspeicher befinden.
2. Um die Auflistung zu verlangsamen können Sie irgendeine Taste außer **ON** und **R/S** drücken.
3. Drücken Sie **R/S** wenn die gewünschte lokale Marke oder die **END**-Anweisung erscheint, um die Auflistung zu stoppen.

4. Drücken Sie **SST** bzw. **BSST**, wenn Sie die nachfolgende oder vorhergehende Angabe des Katalogs ansehen wollen.

Der Rechner ist jeweils an der Stelle im Programmspeicher, deren globale Marke oder **END**-Anweisung er anzeigt.

5. Mit **←** können Sie die Anzeige des Katalogs löschen.

Die oben beschriebene Art der Positionierung des Rechners mit **CATALOG** 1 ist die einzige Möglichkeit, in ein Programm zu kommen, das ohne globale Marke geschrieben wurde. Diesen Programmen geht bei der Auflistung von **CATALOG** 1 der angezeigten **END**-Anweisung keine globale Marke voran.

## Einzelschritt vor und Einzelschritt zurück

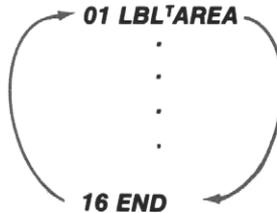
**SST** (single step) und **BSST** (back step) ermöglicht die Positionierung des Rechners in einem Programm und zeigt dem Anwender Schritt für Schritt immer eine Programmzeile.

Im Programm-Modus:

- wird nach **SST** die nächste Zeile des Programms angezeigt;
- wird nach **BSST** die vorhergehende Zeile des Programms angezeigt.

**SST** und **BSST** wirken nur im gegenwärtigen Programm. Wird in der letzten Programmzeile **SST** gedrückt, wird der Rechner wieder an den Anfang des Programms gestellt. Wird **BSST** am Anfang eines Programmes gedrückt, wird der Rechner an das Ende dieses Programmes gestellt.

Wird hier **SST** gedrückt, wird der Rechner an den Anfang des Programmes gestellt.



Wird hier **BSST** gedrückt, wird der Rechner ans Ende dieses Programmes gestellt.

## Im Normal-Modus

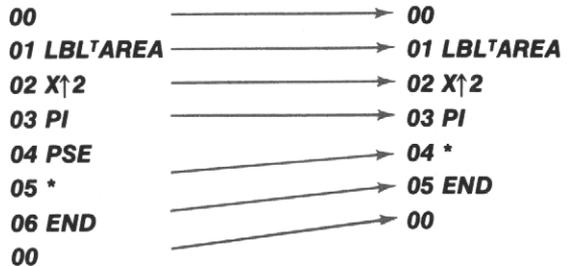
Im Normal-Modus kann mit **SST** ein Programm zeilenweise ausgeführt werden. Wird die Taste **SST** gedrückt, wird die folgende Programmzeile angezeigt, und sobald die Taste losgelassen wird, wird diese Zeile ausgeführt. Drücken Sie **BSST**, wird die vorhergehende Zeile angezeigt, es wird jedoch keine Funktion ausgeführt.

# Löschen und Korrektur einzelner Anweisungen

## Löschen von Anweisungen

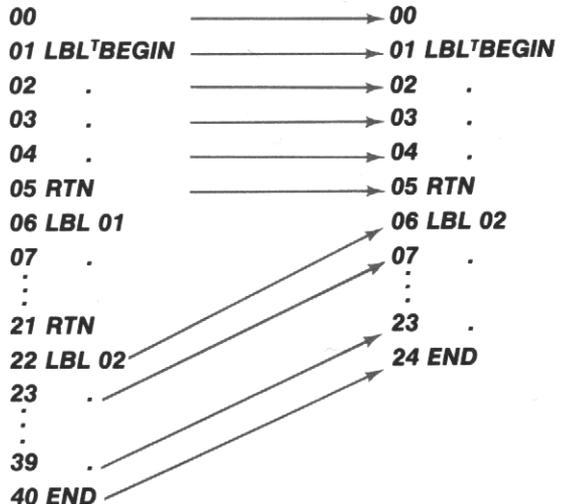
**Einzelne Zeilen.** Um eine einzelne Zeile zu löschen, schalten Sie den Rechner in den Programm-Modus, positionieren Sie den Rechner an die Programmzeile, die Sie löschen wollen, und drücken Sie dann  $\boxed{\leftarrow}$ . Wenn diese Zeile gelöscht wird, springt der Rechner eine Zeile zurück und zeigt diese auch an. Die Zeilennummern der folgenden Programmzeilen werden um eins vermindert.

Wird der Rechner auf Zeile 04 gestellt und mit  $\boxed{\leftarrow}$  eine Zeile gelöscht, rücken die nachfolgenden Anweisungen um eine Zeile nach oben.



**Zeilenblöcke.** Führen Sie im Programm-Modus **DEL** (delete lines) gefolgt von einer dreiziffrigen Zahl aus, löscht der Rechner ab der angezeigten Zeile so viele Zeilen wie angegeben, höchstens jedoch bis zur nächsten **END**-Anweisung (diese bleibt bestehen). Der Rechner geht dann eine Zeile zurück und zeigt diese an. Alle folgenden Zeilennummern werden um die Anzahl der gelöschten Zeilen reduziert.

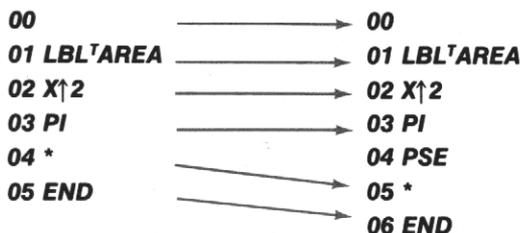
Steht der Rechner auf Zeile 06 und wird **DEL 016** ausgeführt, werden 16 Programmzeilen, beginnend mit Zeile 06 gelöscht und die nachfolgenden Anweisungen 16 Zeilen nach oben geschoben.



## Einfügen von Anweisungen

Um eine Anweisung in ein Programm einzufügen, stellen Sie den Rechner auf die Programmzeile, nach der die Anweisung eingefügt werden soll und geben dann die Anweisung ein. Der Rechner zeigt die neue Zeile an, und alle nachfolgenden Zeilennummern werden um eins erhöht.

Steht der Rechner auf Zeile 03 und wird eine **PSE**-Anweisung eingetastet, wird diese Anweisung als Zeile 04 eingefügt und die nachfolgenden Anweisungen um eine Zeile nach unten geschoben.



Stehen nicht mehr genügend Bytes im Programmspeicher zur Verfügung, packt der Rechner den Programmspeicher und zeigt **PACKING** an und danach **TRY AGAIN**. Zeigt der Rechner nach einem neuen Eingabeversuch wieder **TRY AGAIN** an, können keine Anweisungen mehr eingegeben werden, solange nicht Register verfügbar gemacht werden wie auf Seite 36 beschrieben.

Wenn Sie nach einer Einfügung sehen wollen, wieviele Register unbenutzt sind, drücken Sie  **GTO**  000. Der Rechner zeigt daraufhin **00 REG** und die Anzahl der unbenutzten Register an.

## Löschen von Programmen

**CLP** (clear program) gefolgt von einem Programmnamen (d.h. einer globalen Marke) löscht das bezeichnete Programm. Wird **CLP** ausgeführt:

1. Sucht der Rechner von unten her den Programmspeicher nach der angegebenen Marke ab.
2. Löscht der Rechner diese Marke und alle folgenden Programmzeilen bis (einschließlich) zu einer **END**-Anweisung. Es werden auch Programmzeilen vor der globalen Marke bis zu einer **END**-Anweisung gelöscht.
3. Wird die Zuordnung des Programmes zu einer Taste aufgehoben – falls eine bestand.
4. Packt der Rechner den Programmspeicher um die unbenutzten Bytes, die das gelöschte Programm zurückließ, wieder zu verwenden.

Der Programmspeicher würde beispielsweise durch **CLP TEST 1** oder durch **CLP TEST 2** wie folgt geändert :

vorher	nachher
<b>00</b>	<b>00</b>
<b>01 LBL'ZOT</b>	<b>01 LBL'ZOT</b>
<b>02 'ANS =</b>	<b>02 'ANS =</b>
<b>03 ARCL 00</b>	<b>03 ARCL 00</b>
<b>04 END</b>	<b>04 END</b>
<b>00</b>	<b>00</b>
<b>01 LBL'TEST1</b>	<b>01 LBL'HEAT</b>
<b>02 STO 01</b>	<b>02 30</b>
<b>03 RTN</b>	
<b>04 LBL'TEST 2</b>	
<b>05 STO 02</b>	
<b>06 END</b>	
<b>00</b>	
<b>01 LBL'HEAT</b>	
<b>02 30</b>	

Die Ausführung von **CLP** ALPHA ALPHA (d.h. ohne Angabe eines Programmnamens) löscht das Programm, in dem der Rechner augenblicklich steht.

Um alle Programme des Programmspeichers gleichzeitig zu löschen (sowie alle anderen Informationen auch, die das Continuous Memory aufrecht erhält):

1. Schalten Sie den Rechner aus.
2. Halten Sie die ←-Taste gedrückt und schalten Sie den Rechner ein.
3. Lassen Sie die ←-Taste los.

## Programmunterbrechungen

### **STOP** und **R/S**

#### **STOP**

Im Programm-Modus kann eine **STOP**-Anweisung durch Drücken der **R/S**-Taste (run/stop) oder durch Eingeben von **XEQ** **ALPHA** **STOP** **ALPHA** eingegeben werden. Wenn ein laufendes Programm dann auf diese Anweisung stößt wird ein **STOP** ausgeführt, und das Programm hält an, bereit, mit der nächsten Zeile des Programms fortzufahren.

#### **Stop von Tastenfeld**

Läuft ein Programm, kann es mit **R/S** angehalten werden. Obwohl die **STOP**-Funktion anderen Tasten zugeordnet werden kann, hält ein Druck auf diese Tasten im User-Modus das Programm nicht an. Nur die **R/S**-Taste selbst (und natürlich die **ON**-Taste) kann ein laufendes Programm anhalten. Weiterhin stoppt **R/S** ein laufendes Programm im Normal- wie im User-Modus, selbst wenn auf diese Taste eine andere Funktion zugeordnet wäre. Läuft das Programm nicht, startet **R/S** das Programm beginnend mit der Zeile, in der der Rechner gerade steht.

### **PROMPT**

Eine **PROMPT**-Anweisung im Programm hält ein laufendes Programm an und zeigt den Inhalt des ALPHA-Registers. Ebenso wie **AVIEW** zeigt **PROMPT** nur den Inhalt des ALPHA-Registers an, schaltet jedoch den Rechner nicht in den Alpha-Modus.

### **PSE (Pause)**

Wenn ein laufendes Programm eine **PSE**-Anweisung (Pause) durchführt, hält das Programm für einen Moment an. Während dieser Pause – die ungefähr eine Sekunde dauert – zeigt der Rechner entweder das X-Register an oder (wenn der Rechner im Alpha-Modus ist) das ALPHA-Register. Zeigt jedoch die Anzeige den Inhalt eines Registers aufgrund einer **VIEW**- oder **AVIEW**-Anweisung, verlängert das **PSE** diese Anzeige nur. Bei jeder **PSE**-Anweisung, die ausgeführt wird, blinkt der **PRGM**-Indikator einmal.

Während der Pause (oder einer Reihe von Pausen) ist das gesamte Tastenfeld aktiv, und es können numerische oder Alphadaten in den Rechner eingegeben werden. Die Pause wird wiederholt, wenn Sie eine der folgenden Tasten drücken: 0 bis 9, **□**, **EEX**, **CHS**, **↔**, **ALPHA**, **USER**, **■** und alle Alpha-Zeichen. Wird eine andere Taste gedrückt, wird

sowohl die Pause als auch das Programm beendet und die gedrückte Funktion wird ausgeführt.

## **OFF**

Führt das Programm eine **OFF**-Anweisung aus, wird der Rechner ausgeschaltet.

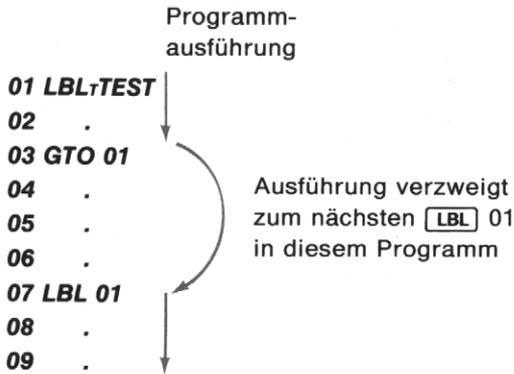
## **Stop bei Fehlermeldungen**

Versucht der HP-41C in einem laufenden Programm eine Operation auszuführen die eine Fehlermeldung hervorruft, zeigt der Rechner diese Fehlermeldung und hält an. Um die Zeile anzusehen, die die Fehlermeldung verursacht hat, schalten Sie den Rechner kurz in den Programm-Modus. Dies löscht die Fehlermeldung – wie es auch  macht. Flag 24 oder Flag 25 können gesetzt werden, um Stops wegen Fehlermeldungen zu ignorieren. Fehlermeldungen und deren Bedingungen sind im Teil 12 dieses Handbuchs beschrieben.

## Programmverzweigungen

### Die Verwendung von **GTO** in einem Programm

Die Programmausführung kann mit der **GTO**-Anweisung (go to label) zu jeder lokalen oder globalen Marke verzweigen.



Wenn das Programm auf eine **GTO 01**-Anweisung trifft, hält die Programmausführung sofort an und der Rechner sucht nach der Marke **LBL 01**, wo die Programmausführung fortgesetzt wird.

Die Marke, zu der das Programm verzweigen soll, kann sowohl indirekt als auch direkt angegeben werden (sofern es nicht eine lokale Alpha-Marke ist). Ist die im indirekten Register angegebene Marke nicht auffindbar, oder ist die Nummer oder die Alpha-Kette im indirekten Register keine erlaubte Marke (Zahl größer als 99 oder lokale Alpha-Marke), dann wird die Programmausführung in der Zeile der **GTO**-Anweisung angehalten und **NONEXISTENT** wird angezeigt. (Weitere Informationen über indirekte Parameterangabe siehe Seite 8.)

### Die Suche nach Marken

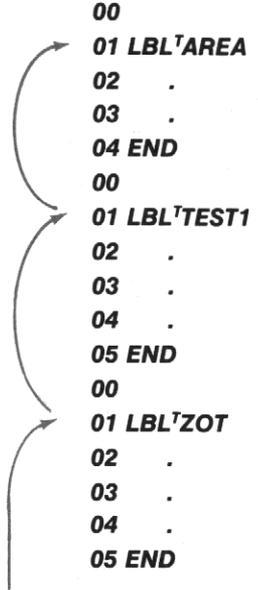
#### Suche nach globalen Marken

Um eine globale Marke zu finden sucht der Rechner erst alle globalen Marken im Programmspeicher ab, dann die der Speichererweiterungsmodule und Peripheriegeräte, die an den Rechner angeschlossen sind, und zuletzt die Standardfunktionen des HP-41C. (Die Reihenfolge der Suche entspricht auch der Reihenfolge der entsprechenden Kataloge: CATALOG 1, CATALOG 2 und dann CATALOG 3.)

Setzt sich eine globale Alpha-Marke aus denselben Buchstaben wie eine der Standardfunktionen zusammen (z.B. **ABS**, **DEG** oder **SIN**), führt der Rechner das so benannte Programm aus, wenn es nach **XEQ** oder **GTO** angegeben wurde. Ist jedoch diese Marke nicht im Programmspeicher oder in einer Erweiterung (also nicht in Catalog 1 oder 2 aufgelistet), wird die Standardfunktion ausgeführt.

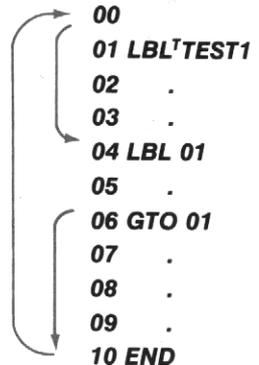
Im Programmspeicher beginnt die Suche mit der letzten globalen Marke. Es wird der Programmspeicher nach oben abgesucht, wobei die Anweisungen zwischen den globalen Marken ausgelassen werden, bis der Rechner den Anfang des Programmspeichers erreicht hat oder die angegebene Marke gefunden hat. Wird die Marke gefunden, wird das Programm ab dort ausgeführt.

Wird eine angegebene Marke nicht gefunden (ist sie also auch nicht in Catalog 1, 2 oder 3 enthalten), wird **NON-EXISTENT** angezeigt und der Rechner steht an der Stelle, an der er vor der Suche stand.



### Suche nach lokalen Marken

Die Suche nach lokalen Marken wird nur im laufenden Programm durchgeführt, d.h. in dem Programm, das der Rechner gerade ausführt oder in dem er steht. Um die lokale Marke zu finden sucht der Rechner die nachfolgenden Programmzeilen ab. Wird die angegebene Marke nicht vor dem Ende des Programmes gefunden (vor einem **END**-Befehl), wird die Suche nach der Marke am Anfang des Programmes fortgesetzt, bis die angegebene Marke gefunden wird. Wird diese Marke nicht gefunden, wird **NON-EXISTENT** angezeigt und der Rechner an die Stelle gestellt, an der er vor der Suche stand.



Auf Grund dieses Suchprozesses können lokale Marken mehrmals im selben Programm verwendet werden.

Die Suche nach einer lokalen Marke kann, abhängig von der Länge des Programmes, erhebliche Zeit dauern. Um die Suchzeit zu verkürzen, merkt sich der Rechner die Position der meisten lokalen Marken bei der ersten Ausführung von **GTO**, so daß für

folgende Ausführungen dieser **[GTO]**-Anweisung die Suche entfällt. Die Ausnahmen sind die Marken 00 bis 14 (die « Kurzform »-Marken), wenn sie weiter als 112 Bytes von der **[GTO]**-Anweisung entfernt sind. Diese Marken müssen nach jedem **[GTO]** gesucht werden. Die Suche ist jedoch auch hier nicht erforderlich, wenn diese Marken weniger als 112 Bytes nach der **[GTO]**-Anweisung kommen.

Diese Kurzform-Marken benötigen nur ein Byte Speicherplatz, die dazugehörige **[GTO]**-Anweisung zwei Bytes. Lokale Alpha-Marken und die anderen numerischen Marken benötigen zwei Bytes und die zugehörigen **[GTO]**-Anweisungen drei Bytes. Um Programmplatz einzusparen sollten also Kurzform-Marken verwendet werden, jedoch können diese – abhängig von ihrer Entfernung von der **[GTO]**-Anweisung – die Rechenzeit verlängern.

## Vergleichende Funktionen

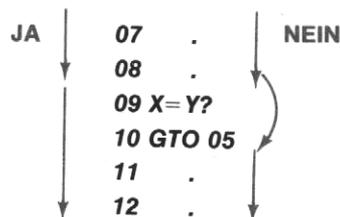
Die auf dem HP-41C verfügbaren vergleichenden Funktionen sind:

**[X=Y?]**   **X≠Y?**   **[X>Y?]**   **X<Y?**   **[X≤Y?]**  
**[X=0?]**   **X≠0?**   **X>0?**   **X<0?**   **X<=0?**

Zwei dieser Vergleiche, **[X=Y?]** und **X≠Y?** können zum Vergleich von numerischen Daten wie auch alphanumerischen Daten verwendet werden. Die anderen Vergleiche arbeiten nur mit numerischen Daten.

Jede Vergleichsoperation stellt eine Frage: **[X=Y?]** fragt zum Beispiel, ob der Inhalt des X-Registers genau gleich dem Inhalt des Y-Registers ist.

«Ausführung falls JA-Regel»: Ist  $x = y$  wird die nächste Anweisung ausgeführt. Ist  $x \neq y$  wird die nächste Anweisung ausgelassen.



Ist die Antwort JA, dann wird die nächste Zeile ausgeführt. Das ist die Regel: « Ausführung falls JA » – wenn die Antwort auf den Vergleich JA lautet, wird die nächste Programmzeile ausgeführt. Ist die Antwort NEIN, wird die nächste Programmzeile übersprungen und mit der übernächsten Anweisung fortgesetzt.

Wird eine dieser Vergleichsoperationen manuell über das Tastenfeld ausgeführt, wird die Antwort in die Anzeige geschrieben. Falls die Bedingung zutrifft, wird **YES (JA)** in die Anzeige geschrieben, trifft die Bedingung nicht zu, **NO (NEIN)**.

## Steuerung von Programmschleifen

Der HP-41C hat zwei Funktionen zur Steuerung von Schleifen. Diese Funktionen sind **ISG** (increment and skip if greater) und **DSE** (decrement and skip if equal). Beide Funktionen enthalten interne Zähler, die eine Schleifensteuerung erlauben. Diese Funktionen interpretieren eine Kontrollzahl in einer besonderen Weise. Diese Kontrollzahl kann in jedes Datenregister, Stackregister oder in das LAST X-Register eingegeben werden. Das Register, das diese Kontrollzahl enthält, muß angegeben werden, wenn der Rechner nach **ISG** oder **DSE** nach einer Eingabe fragt. Das Format der Kontrollzahl ist

**iiii.fffcc**

**iiii** ist der Schleifenzähler. Dieser Wert kann zwischen ein und fünf Ziffern haben. Jedes Mal, wenn **ISG** oder **DSE** ausgeführt wird, wird der Schleifenzählwert um den Wert von **cc** erhöht oder vermindert. Wird **ISG** oder **DSE** verwendet, um eine Schleife auszuführen, dann wird die Anzahl der Durchläufe mit dem anfänglichen Wert von **iiii** begonnen.

**fff** ist der Endwert. Wenn **ISG** oder **DSE** verwendet wird, um eine Schleife auszuführen, wird jedes Mal, wenn diese Anweisung ausgeführt wird, der Wert von **iiii** mit dem Wert von **fff** verglichen. War der Anfangswert von **iiii** Null ist **fff** die Anzahl der Schleifen, die durchlaufen werden sollen. **fff** muß aus drei Ziffern bestehen (z.B. 100, 009 usw.).

**cc** ist der Inkrement- oder Dekrementwert. Dieser Wert muß aus zwei Ziffern bestehen (z.B. 01, 03 usw.). Falls **cc** 00 ist, verwendet der Rechner den Wert 01 stattdessen.

Jedes Mal, wenn **DSE** ausgeführt wird, untersucht der Rechner die Zahl im Kontrollregister und dekrementiert (vermindert) **iiii** um **cc**. Dann wird überprüft, ob **iiii** gleich oder kleiner **fff** ist. Trifft dies zu, überspringt der Rechner die nächste Programmzeile.

Jedes Mal, wenn **ISG** ausgeführt wird, wird erst der Wert **iiii** im Kontrollregister um **cc** inkrementiert (erhöht). Ist dann **iiii** größer als **fff**, wird die nächste Programmzeile übersprungen und das Programm fortgesetzt.

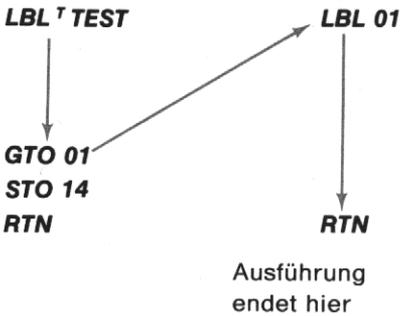
In Programmen stehen **ISG**- oder **DSE**-Anweisungen in Schleifen typischerweise meist vor **GTO**-Anweisungen, um die Ausführung des Programms an einer anderen Stelle weiterzuführen, wenn der Endwert **fff** erreicht wurde. Wird **ISG** oder **DSE** manuell vom Tastenfeld aus ausgeführt, wird das Kontrollregister verändert aber natürlich keine Programmzeilen übersprungen.

## Unterprogramme

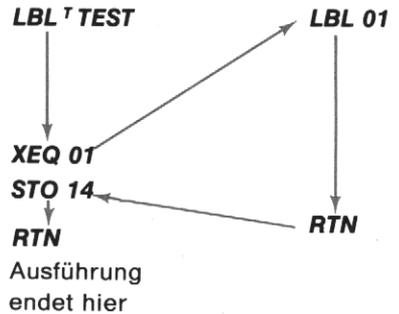
### Aufruf von Unterprogrammen

Eine Programmanweisung, die aus einem **XEQ** und einer Marke besteht, ist ein Unterprogramm-Aufruf\*. Ein Unterprogramm setzt die Durchführung des Programms an der aufgerufenen Marke fort, wie bei einer Verzweigung. Wenn aber nun in einer Zeile eine **RTN**- oder **END**-Anweisung angetroffen wird, springt der Rechner zurück an die Zeile, die der Unterprogramm-Aufruf-Anweisung folgt.

#### Verzweigung



#### Unterprogramm



Unterprogramme (und deren Aufruf) können mit lokalen Marken im laufenden Programm aufgerufen werden oder auch mit globalen Marken im gesamten Programmspeicher.

Die Marke, an der das Programm fortgesetzt werden soll kann sowohl indirekt als auch direkt angegeben werden (außer lokale Alpha-Marken). Kann die Marke, die im indirekten Register angegeben wurde, nicht gefunden werden, oder ist die Zahl oder Alpha-Kette im indirekten Register nicht zulässig (z.B. eine Zahl größer als 99 oder eine lokale Alpha-Marke), wird das Programm an der Zeile, die die **XEQ**-Anweisung enthält, angehalten, und der Rechner zeigt **NONEXISTENT** an. (Weitere Angaben zu indirekter Adressierung siehe Seite 8.)

\* **XEQ** führt Unterprogramme aus wie es **GSE** auf anderen HP Rechnern mit Marken-Adressierung macht (wie beim HP-67/97, HP-19/29C und HP-34C).

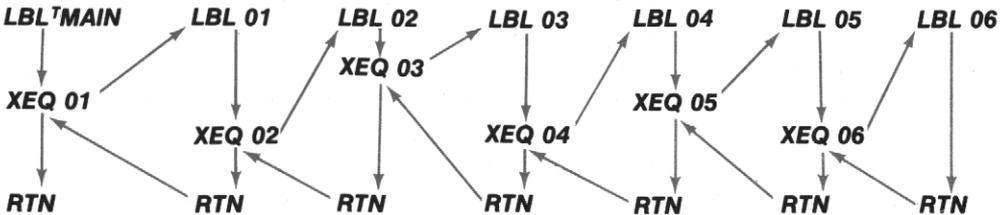
Die Suche nach Marken bei einem Unterprogramm-Aufruf geschieht in der gleichen Weise wie bei den **GTO**-Anweisungen. Wird eine **END**-Anweisung angetroffen, bevor die angegebene Marke gefunden wurde, wird die Suche am Anfang des Programmes fortgesetzt. Wird eine **RTN**-Anweisung angetroffen, wird die Suche nach der Marke nach dieser **RTN**-Anweisung fortgesetzt. Bestehen in einem Programm mehrere Unterprogramme mit lokalen Marken, ist es aus diesem Grunde wichtig, daß jedes Unterprogramm mit einer **RTN**-Anweisung endet, außer dieses Unterprogramm würde am Ende eines Programmes stehen. Unterprogramme mit globalen Marken können entweder mit einer **RTN**- oder einer **END**-Anweisung abgeschlossen werden. Unterprogramme, die am Ende eines Programmes stehen können mit derselben **END**-Anweisung abgeschlossen werden wie das Programm selbst. Es ist nicht nötig, solch ein Unterprogramm mit einem eigenen **END** abzuschließen.

00 LBL DEMO
01 .
02 .
03 LBL 01
04 .
05 .
06 .
07 RTN
08 LBL 02
09 .
10 RTN
11 LBL 03
12 .
13 .
14 END

Wird im Unterprogramm-Aufruf eine lokale Marke angegeben, die im laufenden Programm nicht enthalten ist, oder eine globale Marke, die im Programmspeicher nicht enthalten ist und auch nicht in einer Erweiterungseinheit (also nicht in Catalog 1 oder 2 aufgeführt ist), dann wird **NONEXISTENT** angezeigt, und der Unterprogramm-Aufruf wird nicht ausgeführt. Das Programm wird in der Zeile mit dem Unterprogramm-Aufruf angehalten.

## Grenzen bei der Verwendung von Unterprogrammen

Die Anzahl der Unterprogramme ist nur begrenzt durch die Anzahl der noch ausstehenden **RTN**- oder **END**-Anweisungen. Der Rechner kann sich bis zu sechs noch auszuführende **RTN**-Rücksprungbefehle merken, kann also aus einer sechsten Unterprogrammebene zum Hauptprogramm zurückkehren.



Wird ein Unterprogramm manuell vom Tastenfeld aus aufgerufen, wird **RTN** gedrückt, oder wird **SIZE** durchgeführt, gehen alle gespeicherten, noch auszuführende **RTN**- und **END**-Rücksprunginformationen verloren.

## Flags

Ein Flag ist ein Status-Anzeiger, der entweder im Status « gesetzt » oder « gelöscht » sein kann. In Programmen können Flags verwendet werden, um Entscheidungen zu fällen, wie Sie es von den Vergleichsfunktionen kennen (siehe Teil 8).

Der HP-41C enthält insgesamt 56 Flags. Von diesen sind die Flags 00 bis 29 Anwenderflags; diese können vom Anwender gesetzt, gelöscht oder geprüft werden. Die Flags 30 bis 55 sind Systemflags; diese können vom Anwender getestet, aber nicht verändert werden.

Es gibt sechs Flag-Funktionen. Jede benötigt zur Ausführung eine zweistellige Flagnummer oder eine indirekte Adresse:

**SF** *nn* setzt Flag *nn* (*nn* = 00 bis 29)

**CF** *nn* löscht Flag *nn* (*nn* = 00 bis 29)

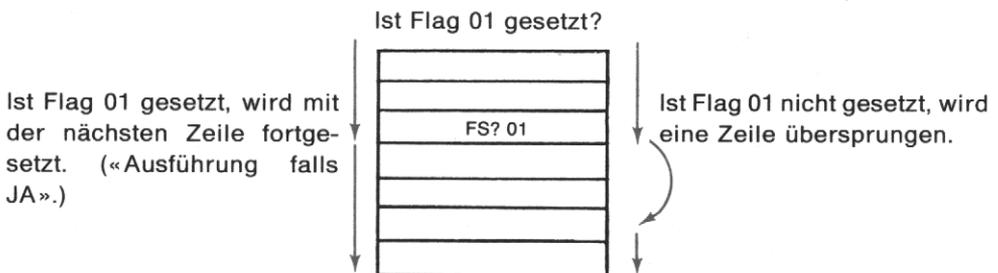
**FS?** *nn* fragt, ob Flag *nn* gesetzt ist (*nn* = 00 bis 55)

**FC?** *nn* fragt, ob Flag *nn* gelöscht ist (*nn* = 00 bis 55)

**FS?C** *nn* fragt, ob Flag *nn* gesetzt ist, und Flag *nn* wird nach dieser Abfrage gelöscht (*nn* = 00 bis 29)

**FC?C** *nn* fragt, ob Flag *nn* gelöscht ist, und Flag *nn* wird nach dieser Abfrage gelöscht (*nn* = 00 bis 29)

Wenn in einem Programm ein Flag abgefragt wird und die Antwort JA ist, wird die nächste Programmzeile ausgeführt. Ist die Antwort auf die Abfrage NEIN, wird die nächste Programmzeile übersprungen.



Wird eine Flag-Abfrage manuell vom Tastenfeld ausgeführt, erscheint **YES** (JA) oder **NO** (NEIN) in der Anzeige.

Flag-Nummer	Flag-Name	Beschreibung	Zustand beim Einschalten des Rechners*
00 bis 10	Allgemeine Anwendung	Status der Flags 00 bis 04 erscheint in der Anzeige als Indikator	M
11 bis 20	Spezielle Anwendung	Status wird manchmal vom Rechner geändert.	C
11	Autostart	Wenn Flag vor dem Ausschalten gesetzt wird, beginnt die Ausführung des Programms automatisch beim Einschalten des Rechners	C
12	doppelt große Buchstaben (Drucker)	Wenn das Flag gesetzt ist werden alle Buchstaben doppelt so groß wie normal gedruckt	C
13	Kleinbuchstaben (Drucker)	Wenn das Flag gesetzt ist, werden Kleinbuchstaben gedruckt	C
14	Magnetkarten-überschrieb	Wenn Flag gesetzt ist, können geschützte Magnetkarten überschrieben werden.	C
21	Druckersteuerung	Wenn Flag gesetzt ist, wird angenommen, daß ein Drucker angeschlossen ist.	**
22	numerische Eingabe	Wird gesetzt, wenn numerische Daten vom Tastenfeld oder optischen Lesestift eingegeben wird.	C
23	Alpha-Eingabe	Wird gesetzt, wenn Alpha-Zeichen vom Tastenfeld oder optischen Lesestift eingegeben werden.	C
24	Bereichsfehlerignorierflag	Wenn Flag gesetzt ist, werden Überlaufmeldungen (wenn eine Zahl größer $9,999999999 \times 10^{99}$ erzeugt wird) unterdrückt.	C
25	Fehlerignorierflag	Wenn Flag gesetzt ist, werden Fehlermeldungen ignoriert. (Flag wird anschließend gelöscht.)	C
26	Akustisches Signal	Wenn Flag gelöscht ist, wird akustisches Signal unterdrückt	S
27	User-Modus	Wenn Flag gesetzt ist, ist der Rechner im User-Modus.	C
28	Dezimalpunkt	Wenn Flag gelöscht ist, erscheint Dezimalkomma (Beispiel: <i>123.456.789,0</i> ).	M
29	Zifferngruppierung	Wenn Flag gelöscht ist, wird die Zifferneinteilung unterdrückt (Beispiel: <i>123456789.0</i> ).	M

30	Katalog	Wird für die Katalogfunktion gebraucht.	n.a.
31-35	Peripherieflags	Intern für bestimmte Peripheriegeräte gebraucht.	n.a.
36-39	Stellenzahlflags	Zusammen verwendet zur Bestimmung der angezeigten Dezimalstellen in FIX, SCI oder ENG.	M
40-41	Anzeigeformat	Zusammen verwendet für Anzeigeformat.	M
42	Grad-Modus	Wenn das Flag gesetzt ist, ist der Rechner im Grad-Winkelmodus.	M
43	RAD-Modus	Wenn das Flag gesetzt ist, ist der Rechner im Bogenmaß-Winkelmodus.	M
44	Einschaltungsdauer-Flag	Wenn das Flag gelöscht ist, wird der Rechner automatisch nach 10 Minuten Inaktivität abgeschaltet.	n.a.
45	Dateneingabe-Flag	Intern für die Dateneingabe verwendet.	n.a.
46	Tasten-Teilfolge-Flag	Intern bei der Ausführung von Funktionen verwendet.	n.a.
47	SHIFT-Flag	Intern für umgeschaltete Operationen verwendet.	n.a.
48	Alpha-Modus-Flag	Ist gesetzt, wenn der Rechner im Alpha-Modus ist.	C
49	Batteriekontrolle	Ist gesetzt, wenn Batteriespannung zu niedrig ist.	n.a.
50	Meldungsflag	Ist gesetzt, wenn eine Fehlermeldung, eine Status-Meldung oder eine <b>VIEW/AVIEW</b> -Anzeige angezeigt wird.	n.a.
51	SST	Intern für die Einzelschritt-Programmausführung verwendet.	n.a.
52	PRGM-Modus	Zur Kontrolle des Programm-Modus verwendet.	n.a.
53	Ein-/Ausgabe	Zur Bestimmung, ob angeschlossene Peripheriegeräte zur Ein-/Ausgabe bereit sind.	n.a.
54	Pause	Ist gesetzt, wenn Peripherie bereit ist.	n.a.
55	Drucker-Anwesenheit	Ist gesetzt, wenn ein Drucker an den HP-41C angeschlossen ist. Arbeitet mit Flag 21 in Übereinstimmung.	**

\* S = gesetzt (set); C = gelöscht (clear); M = vom Continuous Memory aufrechterhalten; n.a. = nicht anwendbar für den Benutzer.

\*\* Status von Flag 21 und 55 wird immer überprüft, wenn der Rechner angeschaltet wird. Beim Einschalten werden beide Flags gesetzt, wenn ein Drucker ans System angeschlossen ist.

## Peripheriegeräte

Namen von Programmen und Funktionen, die mit Software-Modulen und Peripheriegeräten ausgeführt werden können, werden aufgelistet, wenn **CATALOG** 2 gedrückt wird und die Peripherieeinheit angeschlossen ist. Vor den Programmnamen steht in der Anzeige ein  $\tau$ . Solche Programme können in den Programmspeicher des Rechners kopiert und danach verändert werden. Vor den Funktionsnamen steht in der Anzeige kein  $\tau$ , diese können auch nicht in den Rechner kopiert oder verändert werden.

Funktionen und Programme können mit der **XEQ**-Taste ausgeführt werden, wenn das entsprechende Peripheriegerät angeschlossen ist. Soll diese **XEQ**-Anweisung in einem Programm enthalten sein, ist es nicht notwendig, daß die Peripherieeinheit beim Schreiben des Programmes angeschlossen ist. Jedoch hängt die Form, in der diese Anweisung danach angezeigt wird, davon ab, ob das Peripheriegerät angeschlossen ist oder nicht:

- Ist das Peripheriegerät nicht angeschlossen, wenn das Programm geschrieben wird, werden dessen Funktionen oder Programme immer mit **XEQ** gefolgt vom Programm- oder Funktionsnamen angezeigt.
- War das Peripheriegerät angeschlossen, als das Programm geschrieben wurde:
  - werden Anweisungen, die ein Programm ausführen als **XROM** gefolgt von einem Programmnamen angezeigt, wenn das Peripheriegerät angeschlossen ist;
  - werden Anweisungen, die eine Funktion ausführen, als einfacher Funktionsname angezeigt, wenn das Peripheriegerät angeschlossen ist;
  - werden Anweisungen, die entweder eine Funktion oder ein Programm ausführen als **XROM** gefolgt von einer zweistelligen Gerätenummer und einer zweistelligen Programm- oder Funktionsnummer angezeigt, wenn das Peripheriegerät ausgesteckt ist.

Ist beispielsweise das Mathematik-Modul und der Thermodrucker angeschlossen, wenn ein Programm geschrieben wird:

Anzeige der Anweisung wenn Einheit angeschlossen ist	Anzeige der Anweisung wenn die Einheit ausgesteckt wurde	
<b>XROM</b> TANH	<b>XROM 01,35</b>	Führt das Programm <b>TANH</b> aus, das die Nummer 35 im Peripheriegerät 01 (das Mathematik-Modul) hat.
<b>PRX</b>	<b>XROM 29,20</b>	Führt die Funktion <b>PRX</b> aus, die Nummer 20 im Katalog der Peripherieeinheit 29 (Drucker) ist.

Anweisungen, die in den Programmspeicher eingegeben werden, während ein Peripheriegerät angeschlossen ist, benötigen zwei Bytes Programmspeicher. Anweisungen, die in den Programmspeicher eingegeben werden, während die Peripherieeinheit nicht angeschlossen ist, benötigen zwei Bytes plus ein zusätzliches Byte für jeden Buchstaben des Funktions- oder Programmnamen.

Programme aus Software-Modulen oder Peripheriegeräten können nicht nur, wie die Programme im Programmspeicher, ausgeführt werden, sie können auch mit  **GTO**  und danach der Programmname angesprochen werden, und sie können mit **SST** und **BST** angesehen werden. Jedoch können diese Programme nicht geändert werden (d.h. es können keine Programmzeilen eingefügt oder gelöscht werden), sofern sie nicht in den Programmspeicher des Rechners kopiert wurden. Diese Kopie kann auf zwei Arten gemacht werden:

- Führen Sie **COPY** aus und geben den gewünschten Programmnamen an.
- Ist der Rechner schon in dem zu kopierenden Programm (etwa mit  **GTO**  und Programmname) führen Sie **COPY** aus und drücken **ALPHA** **ALPHA**.

Nachdem dieses gemacht wurde:

1. Sucht der Rechner nach dem Programmnamen erst in Katalog 1 und dann in Katalog 2. Wird der Name nicht gefunden oder wurde ein Funktionsname statt eines Programmes angegeben, wird **NONEXISTENT** angezeigt. Existiert das angegebene Programm bereits im Programmspeicher, wird **RAM** angezeigt.
2. Bestimmt der Rechner die Länge des angegebenen Programms und das Ausmaß unbenutzten Speicherplatzes.
3. Bestehen genügend unbenutzte Register im Programmspeicher, um das Programm unterzubringen, wird es kopiert. Im andern Falle packt der Rechner den Programmspeicher, zeigt **PACKING** und danach **TRY AGAIN** an. Wird erneut **TRY AGAIN** nach einem weiteren **COPY**-Versuch angezeigt, muß erst unbenutzter Speicherplatz geschaffen werden, bevor das Programm kopiert werden kann (auf Seite 36 beschrieben).

Der Versuch, ein Programm, das nicht mit **COPY** in den Programmspeicher des Rechners kopiert wurde, zu verändern, wird eine **ROM**-Fehlermeldung in der Anzeige ergeben.

## Fehler- und Statusmeldungen

Wurde versucht, auf dem HP-41C eine unerlaubte Operation durchzuführen, wird diese Operation nicht ausgeführt und es erscheint eine Fehler/Statusmeldung in der Anzeige. Um diese zu löschen, drücken Sie  $\square$ . Wurde der Fehler während eines laufenden Programmes verursacht, können sie sich durch Schalten des Rechners in den Programm-Modus die fehlererzeugende Programmzeile anzeigen lassen.

Anzeige	Bedeutung
<b>ALPHA DATA</b>	Der HP-41C versuchte eine numerische Operation wie Addition oder Subtraktion mit nicht-numerischen Daten wie z.B. Alpha-Ketten durchzuführen.
<b>DATA ERROR</b>	Der HP-41C versuchte eine unerlaubte Operation auszuführen. Diese Fehler sind :
$\square$	bei $x = 0$
$y^x$	bei $y = 0$ und $x \leq 0$ bei $y < 0$ und $x$ nicht ganzzahlig
$\sqrt{x}$	bei $x < 0$
$1/x$	bei $x = 0$
LOG	bei $x \leq 0$
LN	bei $x \leq 0$
<b>LN1 + X</b>	bei $x \leq -1$
$\cos^{-1}$	bei $ x  > 1$
$\sin^{-1}$	bei $ x  > 1$
<b>TONE</b>	bei $ x  \geq 10$ oder $x < 0$
<b>MEAN</b>	bei $n = 0$
<b>OCT</b>	bei $ x  > 1073741823$ (dezimal) oder $x$ nicht ganzzahlig
<b>DEC</b>	wenn $x$ eine Alpha-Kette, 8 oder 9 enthält oder $x$ nicht ganzzahlig ist
<b>%CH</b>	bei $y = 0$
$\square$ , $\square$	bei absolutem Wert der Ziffern $\geq 10$ oder nicht ganzzahlig
$\square$	
<b>FACT</b>	bei $x < 0$ oder nicht ganzzahlig
<b>MEMORY LOST</b>	Das Continuous Memory (Langzeitspeicher) des Rechners wurde gelöscht.
<b>NONEXISTENT</b>	Der HP-41C versuchte ein Register zu verwenden, das nicht existiert oder augenblicklich nicht dem Datenspeicher zugeordnet ist.

- Es wurde versucht, eine nicht existierende Funktion mit **XEQ** oder **ASN** anzusprechen.
- Es wurde versucht, mit **ASN**, **GTO** oder **XEQ** eine Alpha-Marke oder eine numerische Marke anzusprechen, die im System nicht existiert.
- Es wurde versucht, eine Druckfunktion anzusprechen, wobei kein Drucker angeschlossen ist.
- NULL** Das Drücken der Taste wurde unwirksam, nachdem die Taste länger als etwa eine halbe Sekunde gedrückt wurde.
- PRIVATE** Siehe Anleitung zum HP 82104A Magnetkartenleser. Es wurde versucht, ein geschütztes (private) Programm anzusehen.
- OUT OF RANGE** Eine Zahl hat den darstellbaren Zahlenbereich im Speicher oder Rechenwerk des HP-41C überschritten.  
Bereichsüberschreitung =  $\pm 9,999999999 99$ .  
**SDEV**, wo die Standardabweichung von  $x$  ( $S_x = \sqrt{M/[n(n-1)]}$ ) oder  $y$  ( $S_y = \sqrt{N/[n(n-1)]}$ ) eine Division durch Null ergibt oder die Wurzel aus einer negativen Zahl. ( $M = n\sum x^2 - (\sum x)^2$ ;  $N = n\sum y^2 - (\sum y)^2$ .)  
**FACT** bei  $x > 69$ .
- PACKING** Programmspeicher wird gepackt.
- TRY AGAIN** Nach einer Packoperation muß die letzte Eingabe erneut eingegeben werden. Dies kann ein **XEQ**, **ASN**, **GTO**   sein oder der Versuch, einen Befehl in ein Programm einzufügen.
- YES** Antwort, wenn die Prüfung eines Flags das Ergebnis JA ergibt. Ebenso Antwort auf eine Vergleichsoperation zwischen  $x$  und 0 oder  $y$  falls JA.
- NO** Antwort, wenn die Prüfung eines Flags das Ergebnis NEIN ergibt. Ebenso Antwort auf eine Vergleichsoperation zwischen  $x$  und 0 oder  $y$  falls NEIN.
- RAM** Es wurde versucht, mit **COPY** ein Programm in ein **RAM** zu kopieren, das dort bereits besteht.
- ROM** Es wurde versucht mit **DEL**, **CLP**,  ein Programm zu verändern oder etwas einzufügen, das sich in einem ROM befindet.

Viele Peripheriegeräte, die in den HP-41C eingesteckt werden enthalten ihre eigenen Fehler- und Statusmeldungen. Diese Meldungen erscheinen ebenfalls in der Anzeige des HP-41C. Mehr darüber lesen Sie bitte in den entsprechenden Bedienungsanleitungen.

# Funktionsverzeichnis

Funktion		Beschreibung	Funktion ver- langt folgende Eingabe	Stack- lift <sup>1</sup>	Indirekt und LAST X <sup>2</sup>	Bytes <sup>3</sup>	Betrof- ene Flags
über Anzeige	Tastensfeld						
		Umschalttaste		N			47
+		Addition		V	L	1	
-		Subtraktion		V	L	1	
*		Multiplikation		V	L	1	
/		Division		V	L	1	
1/x		Reziprokwert		V	L	1	
10↑X		10 <sup>x</sup>		V	L	1	
ABS		Absolutwert		V	L	1	
ACOS		Arkus-Kosinus		V	L	1	
ADV		Papiervorschub <sup>4</sup>		V		1	21, 55
AOFF		Alpha-Modus aus		V		1	23, 48
AON		Alpha-Modus ein		N		1	
		Alpha-Zeichen anhängen	Alpha-Zeichen	V		1	
ARCL		Alpha-Zurückruf. Ruft Daten ins ALPHA-Register zurück	Register- adresse	V	I	2	
ASHF		Alpha-Shift (links)		V		1	

Funktion		Beschreibung	Funktion ver- langt folgende Eingabe	Stack- lift <sup>1</sup>	Indirekt und LAST X <sup>2</sup>	Bytes <sup>3</sup>	Betrof- fene Flags
über Anzeige	<b>ASIN</b>	Arkus-Sinus		V	L	1	
	<b>ASN</b>	Tastenzuordnung	Funktion, Taste	V		1 <sup>1</sup>	
	<b>ASTO</b>	Alpha-Data in Register abspeichern	Register- adresse	V	I	2	
	<b>ATAN</b>	Arkus-Tangens		V	L	1	
	<b>AVIEW</b>	ALPHA-Register betrachten		V		1	21, 50, 55
	<b>BEEP</b>	Akustisches Signal		V		1	26
	<b>BST</b>	Einzel-schritt zurück		V			
	<b>CAT</b>	Verzeichnis	Katalog-Nr. (1, 2 oder 3)	V	I		30
	<b>CF</b>	Flag löschen	nn Flag-Nr.	V	I	2	00-29
	<b>CHS</b>	Vorzeichenwechsel		N		1	
	<b>CLA</b>	ALPHA-Register löschen		V		1	
	<b>CLD</b>	Anzeige löschen		V		1	
	<b>CLP</b>	Programm löschen	Programm-Marke	V			
	<b>CLRG</b>	Register löschen		V		1	
	<b>CLΣ</b>	Statistik-Register löschen		V		1	
	<b>CLST</b>	Stack-Register löschen		V		1	

Funktion über Anzeige   Tastenfeld	Beschreibung	Funktion ver- langt folgende Eingabe	Stack- lift¹	Indirekt und LAST X²	Bytes³	Betrof- fene Flags
CLX COPY	X-Register löschen Kopiert Programm aus einem Software-Modul oder Peripheriegerät in den Programmspeicher		U	V	1	
COS D-R DEC DEG DEL	Korrekturtaste Kosinus Grad in Bogenmaß Umwandlung Oktal in Dezimal Umwandlung Altgrad-Modus Löscht Programmzeilen		N V V V V V	L L L	1 1 1 1 1	42, 43
DSE  END ENG	Dekrement, Sprung wenn gleich  Exponent Ende des Programms Technisches Anzeigeformat	nnn Zeilen, die gelöscht werden sollen Registeradresse nn nn Exponent von 10 Nach- kommastellen	V N V V	I   I	2  1 3 2	36-41

Funktion über Anzeige Tastenfeld	Beschreibung	Funktion ver- langt folgende Eingabe	Stack- lift <sup>1</sup>	Indirekt und LAST X <sup>2</sup>	Bytes <sup>3</sup>	Betrof- fene Flags
<b>ENTER</b> ↑ 	Schiebt Zahl vom X- ins Y-Register		U		1	
<b>E↑X</b> 	ex		V	L	1	
<b>E↑X-1</b>	ex für Zahlen nahe Null		V	L	1	
<b>FACT</b>	Fakultät		V	L	1	
<b>FC?</b>	Flag gelöscht?	nn Flag-Nr.	V	I	2	00-55
<b>FC?C</b>	Flag gelöscht? und löschen	nn Flag-Nr.	V	I	2	00-29
<b>FIX</b> 	Festkommaformat	n Nachkommastellen	V	I	2	36-41
<b>FRC</b>	Dezimalanteil einer Zahl		V	L	1	
<b>FS?</b> 	Flag ersetzt?	nn Flag-Nr.	V	I	2	00-55
<b>FS?C</b>	Flag gesetzt? und löschen	nn Flag-Nr.	V	I	2	00-55
<b>GRAD</b>	Neugrad-Modus		V		1	42
<b>GTO</b> 	Sprungbefehl zu Marke	Marke nn oder Alpha-Marke	V	I	6	
	Sprungbefehl zu Zeilennummer oder Marke (nicht programmierbar)	Zeilennummer oder Marke	V	I		
	Sprungbefehl ans Ende des Programmspeichers		V			

Funktion		Beschreibung	Funktion ver- langt folgende Eingabe	Stack- lift'	Indirekt und LAST X <sup>2</sup>	Bytes <sup>3</sup>	Betrof- fene Flags
über Anzeige	Tastefeld						
<b>HMS</b>		Dezimale Stunden in Stunden, Minuten, Sekunden umwandeln		V	L	1	
<b>HMS +</b>		Addition von Stunden, Minuten und Sekunden		V	L	1	
<b>HMS -</b>		Subtraktion von Stunden, Minuten und Sekunden		V	L	1	
<b>HR</b>		Umwandlung Stunden, Minuten, Sekunden in dezimale Stunden		V	L	1	
<b>INT</b>		Ganzzahliger Anteil		V	L	1	
<b>ISG</b>	<b>ISG</b>	Inkrement, Sprung wenn größer	Register- adresse <b>nn</b>	V	I	2	
<b>LASTX</b>	<b>LAST X</b>	Ruft Inhalt des LAST X- Registers ins X-Register		V		1	
<b>LBL</b>	<b>LBL</b>	Programm-Marke	Alpha-Marke oder <b>nn</b> Marke	V		7	
<b>LN</b>	<b>LN</b>	Natürlicher Logarithmus		V	L	1	
<b>LN1 + X</b>		Natürlicher Logarithmus für Argumente nahe 1		V	L	1	
<b>LOG</b>	<b>LOG</b>	Logarithmus Basis 10		V	L	1	
<b>MEAN</b>		Mittelwert		V	L	1	

Funktion über Anzeige Tastensfeld	Beschreibung	Funktion ver- langt folgende Eingabe	Stack- lift <sup>1</sup>	Indirekt und LAST X <sup>2</sup>	Bytes <sup>3</sup>	Betrof- fene Flags
<b>MOD</b>	Rest		V	L	1	
<b>OCT</b>	Dezimal in Oktal Umwandlung		V	L	1	
<b>OFF</b>	Ausschalten		V		1	
<b>ON</b>	Ein-/Ausschalten		N			11
<b>P-R</b>	Rechner schaltet sich nicht automatisch ab		V	L	1	44
<b>PACK</b>	Umwandlung Polar- in rechtwinklige Koordinaten		V			
<b>%</b>	Programmspeicher packen		V			
<b>%CH</b>	Prozent		V	L	1	
<b>PI</b>	Prozentualer Unterschied		V	L	1	
<b>PROMPT</b>	Pi (3,141592654)		V		1	52
<b>PSE</b>	Programm-Modus Ein-/Ausschalter		V			21, 50, 55
<b>R↑</b>	Textausgabe und Stop		V		1	54
	Pause		V		1	
	Zyklisches Verschieben des Stacks nach oben		V		1	

Funktion über Anzeige	Tastenfeld	Beschreibung	Funktion ver- langt folgende Eingabe	Stack- lift'	Indirekt und LAST X <sup>2</sup>	Bytes <sup>3</sup>	Betrof- fene Flags
R-D		Umwandlung Bogenmaß in Altgrad		V	L	1	
R-P	[R-P]	Umwandlung Rechteckkoordinaten in Polarkoordinaten		V	L	1	
RAD		Bogenmaß-Modus		V		1	43
RCL	[RCL]	Holt Daten aus einem Register ins X-Register	Register- adresse	V	I	8	
RDN	[R↓]	Zyklisches Verschieben des Stacks nach unten		V		1	
RND		Runden		V	L	1	
RTN	[RTN]	Rücksprung		V		1	
SCI	[SCI]	Wissenschaftliches Anzeigeformat	n Nachkomma- stellen	V	I	2	
SDEV		Stichprobenstandard- abweichung		V	L	1	
SF	[SF]	Flag setzen	nn Flag-Nr.	V	I	2	
Σ+	[Σ+]	Summationen in Statistik- registern		U	L	1	
Σ-	[Σ-]	Summationskorrektur		U	L	1	
ΣREG		Angabe des Statistikblocks	nn	V	I	2	

Funktion über Anzeige	Funktion Tastenfeld	Beschreibung	Funktion ver- langt folgende Eingabe	Stack- liff <sup>1</sup>	Indirekt und LAST X <sup>2</sup>	Bytes <sup>3</sup>	Betrof- fene Flags
<b>SIGN</b>		Vorzeichen von x		V	L	1	
<b>SIN</b>	<b>SIN</b>	Sinus		V	L	1	
<b>SIZE</b>		Anzahl der Datenspeicher- register	nnn, Anzahl der Datenspei- cherregister	V			51
<b>SQRT</b>	$\sqrt{x}$	Quadratwurzel		V	L	1	
<b>SST</b>	<b>SST</b>	Einzelschritt vorwärts		V			
<b>ST +</b>	<b>STO</b> <b>+</b>	Speicherarithmetik (+)	Register- adresse	V	I	2	
<b>ST -</b>	<b>STO</b> <b>-</b>	Speicherarithmetik (-)	Register- adresse	V	I	2	
<b>ST *</b>	<b>STO</b> <b>x</b>	Speicherarithmetik (*)	Register- adresse	V	I	2	
<b>ST /</b>	<b>STO</b> <b>+</b>	Speicherarithmetik (/)	Register- adresse	V	I	2	
<b>STO</b>	<b>STO</b>	Abspeicherung numerischer Daten in Register	Register- adresse	V	I	*	
<b>STOP</b>	<b>R/S</b>	Stoppt Programmausführung		V		1	
<b>TAN</b>	<b>TAN</b>	Tangens		V	L	1	
<b>STONE</b>		Ton	n Tonhöhe	V	I	2	26

über Anzeige	Funktion	Beschreibung	Funktion verlangt folgende Eingabe	Stack-lift <sup>1</sup>	Indirekt und LAST X <sup>2</sup>	Bytes <sup>3</sup>	Betroffene Flags
	<input type="checkbox"/> USER <sup>5</sup>	User-Modus-Taste		N			27
<b>VIEW</b>	<input type="checkbox"/> VIEW	Registerinhalt betrachten	Register- adresse	V	I	2	21,50,55
<b>X = 0?</b>	<input type="checkbox"/> X=0?	Vergleichende Funktion		V		1	
<b>X ≠ 0?</b>		Vergleichende Funktion		V		1	
<b>X &lt; 0?</b>		Vergleichende Funktion		V		1	
<b>X &lt; = 0?</b>		Vergleichende Funktion		V		1	
<b>X &gt; 0?</b>		Vergleichende Funktion		V		1	
<b>X = Y?</b>	<input type="checkbox"/> X=Y?	Vergleichende Funktion <sup>10</sup>		V		1	
<b>X ≠ Y?</b>		Vergleichende Funktion		V		1	
<b>X &lt; Y?</b>		Vergleichende Funktion		V		1	
<b>X &lt; = Y?</b>	<input type="checkbox"/> X≤Y?	Vergleichende Funktion		V		1	
<b>X &gt; Y?</b>	<input type="checkbox"/> X>Y?	Vergleichende Funktion		V		1	
<b>X &lt;&gt;</b>		Austausch des X-Registers mit einem beliebigen Register	Register- adresse	V	I	2	
<b>X &lt;&gt; Y</b>	<input type="checkbox"/> X?Y	Austausch des X-Registers mit dem Y-Register		V		1	

Funktion		Beschreibung	Funktion ver- langt folgende Eingabe	Stack- lift <sup>1</sup>	Indirekt und LAST X <sup>2</sup>	Bytes <sup>3</sup>	Betrof- fene Flags
über Anzeige	Tastefeld						
<b>XEQ</b>	$\boxed{\text{XEQ}}$	Aufruf einer Funktion/ eines Programmes	Numerische oder Alpha-Marke oder Funktionsname	V	I	9	
<b>X<math>\uparrow</math>2</b>	$\boxed{x^2}$	Quadratwurzel		V	L	1	
<b>Y<math>\uparrow</math>X</b>	$\boxed{y^x}$	Exponentialfunktion		V	L	1	

<sup>1</sup>V = Stack-Lift vorbereitende Funktion; U = Stack-Lift unterdrückende Funktion; N = Stack-Lift neutrale Funktion.

<sup>2</sup>I = Parameter kann indirekt angegeben werden; L = wird ins LAST X-Register geschrieben.

<sup>3</sup>Anzahl der Bytes, die im Programmspeicher belegt werden. Kein Eintrag bedeutet, daß diese Funktion nicht programmierbar ist. Numerische Daten benötigen ein Byte für jede Ziffer in dieser Zahl, plus ein weiteres Byte für jedes  $\boxed{\cdot}$ ,  $\boxed{\text{CHS}}$  und  $\boxed{\text{EE}}$ , das mit der Zahl eingegeben wird. Alpha-Daten belegen ein Byte für jedes Zeichen der Alpha-Kette plus ein Byte für die gesamte Alpha-Kette.

<sup>4</sup>Gilt nur, wenn der Drucker angeschlossen ist.

<sup>5</sup>Dieser Taste können keine Programme oder Funktionen zugeordnet werden.

<sup>6</sup>Braucht 2 Bytes, wenn **nn** zwischen 00 und 14 ist. Braucht 3 Bytes, wenn **nn** zwischen 15 und 99 ist. Braucht 3 Bytes, wenn eine lokale Alpha-Marke angegeben wurde. Braucht 2 Bytes plus 1 zusätzliches Byte für jeden Buchstaben der Marke, wenn eine globale Alpha-Marke angegeben wurde. Braucht 2 Bytes, wenn der Parameter indirekt angegeben ist.

<sup>7</sup>Braucht 1 Byte, wenn **nn** zwischen 00 und 14 ist. Braucht 2 Bytes, wenn **nn** zwischen 15 und 99 ist. Braucht 2 Bytes, wenn lokale Alpha-Marke angegeben wurde. Braucht 4 Bytes plus 1 zusätzliches Byte für jeden Buchstaben der Marke, wenn globale Alpha-Marke angegeben wurde.

<sup>8</sup>Braucht 1 Byte, wenn **nn** zwischen 00 und 15 ist. Braucht 2 Bytes, wenn **nn** zwischen 16 und 99 ist. Ist das Register indirekt angegeben, oder wird das X-, Y-, Z-, T- oder LAST X-Register angegeben, werden 2 Bytes gebraucht.

<sup>9</sup>Braucht 3 Bytes, wenn numerische oder lokale Alpha-Marke angegeben wurde. Braucht 2 Bytes plus ein zusätzliches Byte für jedes Zeichen der Marke oder des Funktionsnamens. Braucht 2 Bytes, wenn die Marke indirekt angegeben ist.

<sup>10</sup>Kann sowohl numerische wie auch Alpha-Daten vergleichen.

<sup>11</sup>Zuordnungen von Funktionen und Programmen, die im Katalog 2 oder 3 aufgelistet werden, benötigen ein Register (sieben Bytes) für jede ungerade Anzahl von Zuordnungen. Beispielsweise braucht die erste Zuordnung ein Register, die zweite keinen weiteren Speicherplatz, die dritte Zuordnung ein weiteres ganzes Register usw. Zuordnungen von Programmen, die im Programmspeicher gespeichert sind brauchen keinen zusätzlichen Speicherplatz, die Zuordnung ist mit der Programm-Marke gespeichert.

# Stichwortverzeichnis

## A

Abfrage nach Parameter 7  
Abspeichern von Alpha-Ketten 27, 29  
Abspeichern von Daten 15  
Änderung der Hauptspeichereinteilung 18, 19  
**ALPHA** 6, 8, 13, 27, 31, 32, 44  
**ALPHA DATA** 29, 57  
Alpha-Daten 24, 29, 30, 37, 48  
Alpha-Kette 18, 24, 26, 28, 29, 37, 38  
Alpha-Ketten, Abspeicherung 29  
Alpha-Ketten, Korrigieren 28  
Alpha-Ketten, Löschen 28  
Alpha-Ketten, Verschiebung 29  
Alpha-Ketten, Zurückrufen 29, 30  
Alpha-Marken, globale 10, 12, 13, 31, 38, 39, 46–47, 50  
Alpha-Marken, lokale 31, 33, 38, 47–48, 50  
Alpha-Modus 6, 18, 26–30, 31  
ALPHA-Register 15, 18, 26–30, 44  
Anweisungen einfügen 41  
Anweisungen löschen 41  
Anzeigeindikatoren 6–7  
Anzeigekorrektur 15  
**AOFF** 26  
**AON** 26, 27, 28  
**APPEND** 28  
**ARCL** 29–30  
**ASHF** 29  
**ASN** 31, 35, 37  
**ASN** 26, 31  
**ASTO** 27, 29  
Ausführen eines Programms 37  
**AVIEW** 15, 27–28, 37, 44, 54

## B

**BAT**-Indikator 7  
Betrachten des Registerinhaltes 15, 16, 28, 37, 44, 54  
**BST** 13, 27, 35, 36, 39, 40, 56  
Bytes, Programmspeicher 24, 56, 59–68  
Bytes, unbelegte 25, 36

## C

**CATALOG** 13, 18, 35, 39, 55  
CATALOG 1 12, 13, 18, 33, 38, 39–40, 46, 47  
CATALOG 2 12, 33, 36, 46, 47, 55  
CATALOG 3 12, 33, 36, 46, 47

**CF** 52  
**CHS** 24, 44  
**CLA** 15, 27, 28  
**CLD** 15, 16, 28, 37  
**CLP** 25, 26, 35, 36, 42–43  
**CLRG** 16  
**CLx** 15  
Continuous Memory 16, 17, 23, 33, 43  
**COPY** 35, 56

## D

**DATA ERROR** 57  
Daten, Alpha- 24, 29, 30, 37, 48  
Daten, numerische 24, 29, 30, 48  
Datenspeicherregister 15–16, 18–20, 23  
Datenspeicherregister, Löschen 16  
Dekrement 49  
**DEL** 35, 41  
**DSE** 49

## E

**EEX** 24, 39, 44  
Einfügen von Anweisungen 42  
Eingeben eines Programms 35  
Einzelschritt vor 13, 27, 35, 36, 37, 39, 40, 56  
Einzelschritt zurück 13, 27, 35, 36, 39, 40, 56  
**END** 12, 18, 20, 35–37, 39–42, 47, 50, 51  
**.END.** 18, 24, 36  
Erweiterung des Hauptspeichers 20  
Exponent 24, 37, 39, 44

## F

**FC?** 52  
**FC?C** 52  
Fehlermeldungen 11, 15, 45, 56–58  
Flags 52–54, 59–68  
Flag-Indikatoren 7  
**FS?** 52  
**FS?C** 52  
Funktionen, Ausführung über die Anzeige 13  
Funktionen, vergleichende 29, 48  
Funktionsnamen 13, 38, 55  
Funktionsverzeichnis 59–68

## 70 Stichwortverzeichnis

### G

- Gesamtlöschung 16, 18, 43
- Globale Alpha-Marke 10, 12, 13, 31, 38, 39, **46–47**
  - Indikator 7
- GTO** 26, 49, 50
- GTO** 46, 48
- GTO**  35, 39, 56
- GTO**   18, 20, 25, 35–36, 42
- GSB** 50

### H

- Hauptspeicher 18, 21
- Hauptspeichereinteilung 18, 21

### I

- IND** 8
- Indikatoren 6, 7
- Indirekte Adresse 8–11, 52
- Indirekte Parameterangabe **8**, 59–68
- Indirekte Register **8–11**, 50
- Indirekte Speicherung 10
- Indirekter Sprungbefehl 10, 11
- Inkrement 49
- ISG** 49

### K

- Kataloge 12–13, 18, 32, 33, 35
- Korrektur/Korrekturtaste **13**, **15**, 16, 23, **28**, 35, 39, **41**, 44, 57
- Korrektur einzelner Anweisungen 41
- Kurzform-Marken 48

### L

- LBL** 26, 46
- Löschen einzelner Anweisungen 36, 41
- Löschen von Programmen 36, 42–43
- Löschen von Zeilenblöcken 41
- Lokale Marken 31, 33, 38, 50
- Lokale Marken, Suche nach 46, 51

### M

- Marken 13, 31, 33, 38, 46–48, 50
- Marken, globale Alpha- 10, 12, 13, 31, 38, 39, 46–47, 50
- Marken, indirekte Angabe 10, 50
- Marken, Kurzform- 48
- Marken, lokale Alpha- 31, 33, 38, 47–48, 50
- Marken, numerische 10, 31, 38, 50
- Master clear 16, 18, 37
- MEMORY LOST** 23, 57

### N

- NO** 48, 52, 58
- NONEXISTENT** 11, 15, 32, 47, 50, 51, 56, **57–58**
- Normal-Modus 6, 15, 32, 40
- NULL** 7, 31–34, 58
- Numerische Daten **24**, 29, 30, 48
- Numerische Marken 10, 31, 38, 50

### O

- OFF** 27, 45
- ON** 35
- ON**  6, 16, 23, 27, 32, 35, 44
- OUT OF RANGE** 58

### P

- PACK** 25, 35, 36
- Packen 25, 36, 42, 56, 58
- PACKING** 25, 36, 42, 56, 58
- Parameterangabe 7–12
- Parameterangabe, indirekt 8–12
- Pause 44
- Peripheriegeräte 12, 55–56, 58
- Positionieren 35, 39–40
- PRGM**-Indikator 6, 37
- PRGM**  6, 32, 55
- PRIVATE** 58
- Programm, Ausführen eines 37
- Programm, Eingeben eines 35–36
- Programm, Löschen 36, **42–43**
- Programm-Modus 6, 26, 35, 36, 38, 41
- Programmarken 13, 31, 33, 38, 46–48, 50
- Programmnamen 33, 35, 55, 56
- Programmnamen, Zuordnen von 31–34, 42
- Programmschleifen 46–49
- Programmspeicher 18, 19, 23, **24–25**
- Programmunterbrechungen 28, 44–45
- Programmzeilen 24, 36, 37
- PROMPT** 44
- PRX** 56
- PSE** 28, 44, 54

### R

- RAD**-Indikator 7
- RAM** 56, 57, **58**
- RCL**  16
- REG** 18, 20, 36, 42
- Register, ALPHA 15, 18, **26–30**, 44
- Register, indirekte 8, 50
- Register, LAST X 8, 15, 59–68
- Register, unbenutzte 18, 24, 36, 56
- ROM** 56, **58**
- R/S**  **13**, 27, 37, 39, 44
- RTN** 20, 36, 37, 50, 51
- RTN**  39, 51

## S

**SF** 52  
**SHIFT**-Indikator 7  
**SIZE** 18–23, 35, 36  
 Speichererweiterungsmodule 18, 20, 21  
 Speicherplatzeinteilung 18, 21  
 Sprunganweisung 13, 26, 38, 46, 48  
**SST** 13, 27, 35, 36, 37, 39, 40, 56  
 Stack lift 59–68  
 Status-Indikatoren 6, 7  
 Statusmeldungen 57  
 Steuerung von Programmschleifen 49  
**STO** 16–18  
**STOP** 28, 44  
 Stop bei Fehlermeldungen 45

## T

Tasten-Code 32  
**TRY AGAIN** 20, 36, 42, 56, 57

## U

Unbenutzte Bytes 25, 36  
 Unbenutzte Register 18, 24, 36, 56  
 Unterprogramme 50–51  
 Unterprogramme, Grenzen 51  
**USER** 6, 31, 32, 33, 35, 44  
**USER**-Indikator 6  
 User-Modus 6, 31–34, 37

## V

Vergleich zwischen Alpha-Ketten 29  
 Vergleichende Funktionen 29, 48  
 Verschiebung von Alpha-Ketten 29  
 Verzeichnisse 12, 33, 35  
**VIEW** 15, 16, 28, 37, 44, 54

## X

**XEQ\_** 13, 26  
**XEQ** 13, 33, 37, 50, 55  
**XROM** 55  
**x=y?** 29, 48  
**X ≠ Y?** 29, 48  
**x>y?** 48  
**X < Y?** 48  
**x=sy?** 48  
**x=0?** 48  
**X ≠ 0?** 48  
**X > 0?** 48  
**X < 0?** 48  
**X = < 0?** 48

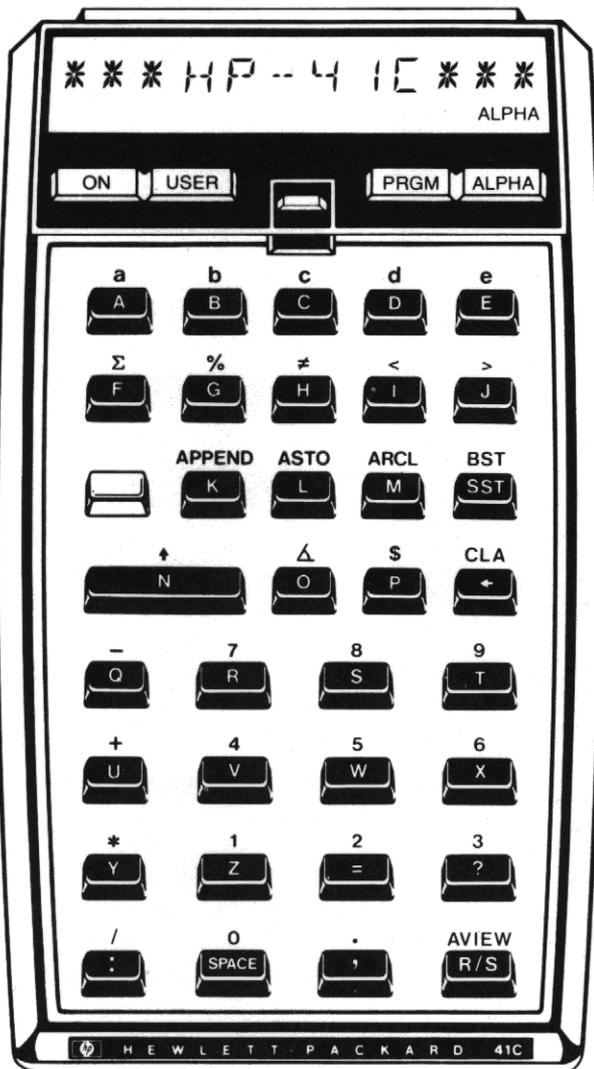
## Y

**YES** 48, 52, 58

## Z

Zeilen, Programm- 24, 36, 37  
 Zeilennummern 24, 37  
 Zifferneingabe 15  
 Zuordnen lokaler Marken 33  
 Zuordnen von Funktionen und Programmen 31–34, 42  
 Zurückrufen von Alpha-Ketten 29, 30  
 Zurückrufen von Daten 15–16, 29–30





ALPHA-Tastenfeld

## **VERKAUFS-NIEDERLASSUNGEN:**

### **Hewlett-Packard GmbH:**

6000 Frankfurt 56, Bernerstraße 117, Postfach 560140, Tel. (0611) 5004-1  
7030 Böblingen, Herrenbergerstraße 110, Tel. (07031) 667-1  
4000 Düsseldorf 11, Emanuel-Leutze-Straße 1 (Seestern), Tel. (0211) 59711  
2000 Hamburg 60, Kapstadtring 5, Tel. (040) 63804-1  
8021 Taufkirchen, Eschenstraße 5, Tel. (089) 6117-1  
3000 Hannover 91, Am Großmarkt 6, Tel. (0511) 466001  
8500 Nürnberg, Neumeyerstraße 90, Tel. (0911) 522083/85  
1000 Berlin 30, Keithstraße 2-4, Tel. (030) 249086

### **Hewlett-Packard (Schweiz) AG:**

Allmend 2, CH-8967 Widen, Tel. (057) 50111

### **Hewlett-Packard Ges.m.b.H., für Österreich/für sozialistische Staaten:**

Wagramerstraße-Liebgasse, A-1220 Wien

### **Hewlett-Packard S.A., Europa-Zentrale:**

7, rue du Bois-du-Lan, Postfach, CH-1217 Meyrin 2-Genf, Schweiz, Tel. (022) 827000

## **SERVICE-NIEDERLASSUNGEN:**

### **Hewlett-Packard GmbH:**

6000 Frankfurt 56, Bernerstraße 117, Postfach 560140, Tel. (0611) 5004-1

### **Hewlett-Packard (Schweiz) AG:**

Allmend 2, CH-8967 Widen, Tel. (057) 50111

### **Hewlett-Packard Ges.m.b.H., für Österreich/für sozialistische Staaten:**

Wagramerstraße-Liebgasse, A-1220 Wien



**HEWLETT  
PACKARD**