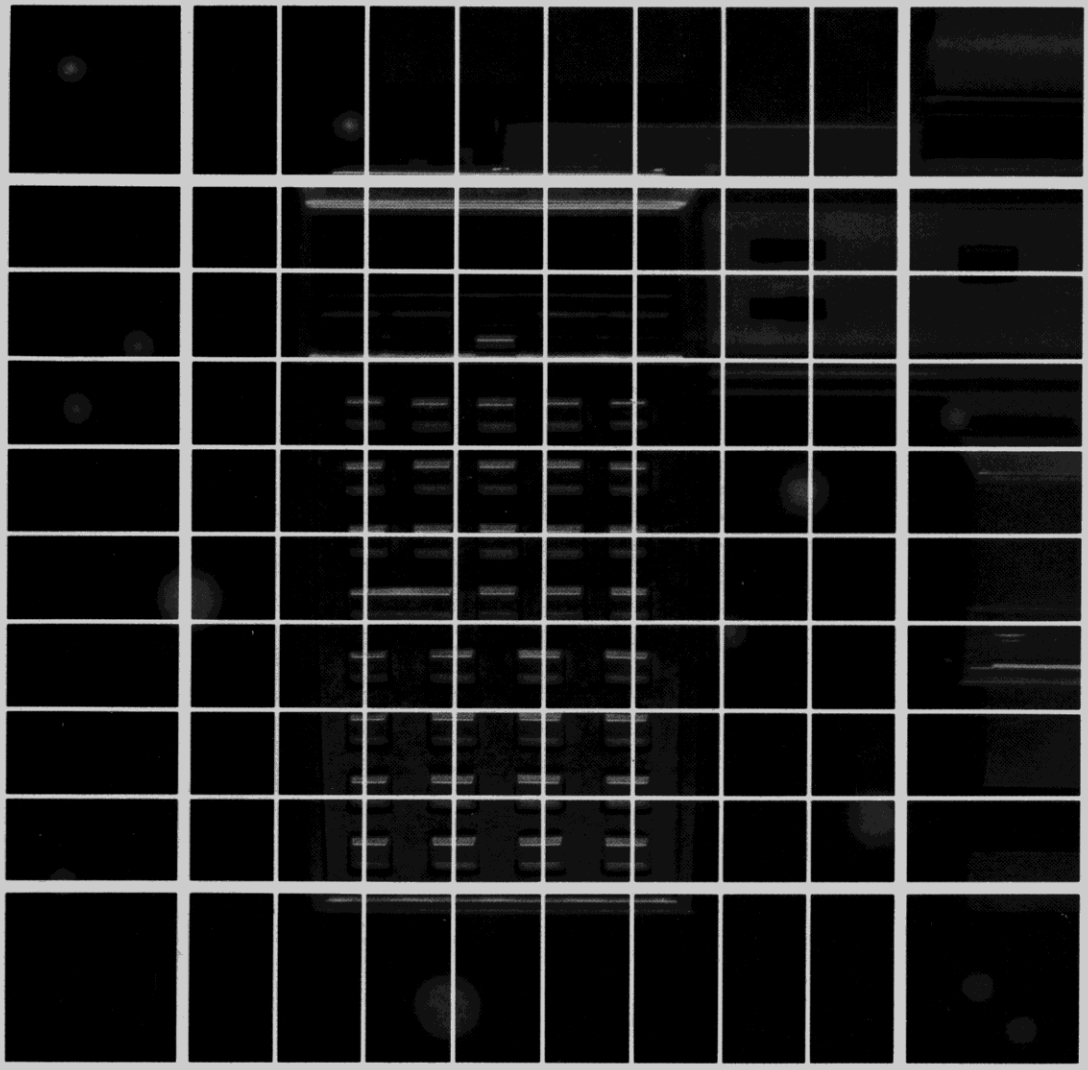
HEWLETT-PACKARD

HP-41CX

BENUTZERHANDBUCH

BAND II



In diesem Handbuch verwendete Konventionen

Darstellung (Symbol)	Bedeutung
[STO]	*Schwarzes Tastensymbol.* Erstfunktionen des Tastenfelds.
[10^x]	*Goldfarbenes Tastensymbol.* Umgeschaltete Zweitfunktionen des Tastenfelds. Drücken Sie zuerst die Umschalttaste (■) und danach die gewünschte Funktionstaste. Dies gilt für das Normal- und Alpha-Tastenfeld.
[DATE]	*Blaues Tastensymbol.* Alpha-Ausführung: Drücken Sie zuerst die Taste [XEQ] und geben Sie dann den Funktionsnamen über das Alpha-Tastenfeld ein. Ausführung über User-Tastenzuordnung: Ordnen Sie die Funktion einer Taste des User-Tastenfelds zu (Seiten 45, 46).
ABC	*Blaue Buchstaben.* Erstzeichen des Alpha-Tastenfelds.
123	*Goldfarbene Ziffern oder Zeichen.* Umgeschaltete Zeichen des Alpha-Tastenfelds.
[T] ■ [T] [·] [T]	*Schwarze Buchstaben in einem Tastenrahmen.* Dies sind in besonderen Fällen wirksame Spezialfunktionen; es handelt sich *nicht* um Alpha-Zeichen. Einer umgeschalteten Funktion ist das ■ Symbol vorangestellt.
Parameter	Der zur Ausführung einer Funktion benötigte *Parameter.*

Eine ausführliche Beschreibung finden Sie auf Seite 16 unter «Darstellung von Tastenfolgen in diesem Handbuch».

HEWLETT PACKARD

HP-41CX

Benutzerhandbuch

Band II

Bedienung im Detail

Oktober 1983

00041-90494

Printed in Singapore

© Hewlett-Packard Company 1983

Einführung in Band 2

Dies ist der zweite Band des *HP-41CX Benutzerhandbuchs*. In Band I, *«Grundlagen der Bedienung»*, wird eine breite Einführung in die meisten Aspekte der Bedienung des HP-41CX gegeben. Der zweite Band, *«Die Bedienung im Detail»*, enthält eine fortgeschrittene, detaillierte Behandlung aller Konzepte des HP-41CX. Die beiden Bände bilden ein Handbuch und die Seiten beider Teile sind durchgehend numeriert.

Der Absatz «Verwenden dieses Handbuchs» auf Seite 9 (Band 1) erläutert den Aufbau dieses Benutzerhandbuchs und gibt Hinweise, an welcher Stelle mit dem Lesen begonnen werden sollte. Sie finden in dem genannten Absatz eine Einführung in beide Bände. Der zweite Band ist im Hinblick auf die *Vollständigkeit* der gegebenen Informationen und als *Nachschlagewerk* konzipiert. Im zweiten Band finden Sie unter anderem: alle Anhänge, einen vollständigen Funktionsindex (Innenseite Rückumschlag) und eine ausführliche Zusammenfassung aller Funktionen im Funktionsverzeichnis (die Seiten mit dem blauen Rand vor dem Sachindex). Wenn Sie bereits mit dem HP-41 vertraut sind, sollten Sie nicht versäumen, Anhang I («Vergleich HP-41CX und HP-41C/CV») zu lesen.

Band 2: Die Bedienung im Detail

Inhalt

Teil II: Grundlagen im Detail

Abschnitt 9: Das Tastenfeld und die Anzeige **154**

- Die Modustasten • Die Tastenfelder
- Eingeben von Zahlen und Zeichen • Status-Indikatoren
- Numerische Anzeigeformate • Standardanzeigen und Meldungen
- Verschieben der Anzeige • Spezifizieren von Parametern
- Belegen der Tasten des User-Tastenfelds
- Funktionsanzeige und Null • Die Kataloge • Fehlermeldungen

Abschnitt 10: Der automatische Speicherstack **174**

- Einführung • Berechnungen mit UPN • Das LAST X Register
- Weitere Stackoperationen

Abschnitt 11: Numerische Funktionen **184**

- Einführung • Funktionen einer Variabler
- Funktionen zweier Variablen • Statistische Funktionen

Teil III: Der Speicher im Detail

Abschnitt 12: Der Hauptspeicher **194**

- Aufteilung des Hauptspeichers • Der Programmspeicher
- Der Alarmspeicher • Der Speicher des User-Tastenfelds
- Der Datenspeicher • Operationen mit Datenspeicher-Registern

Abschnitt 13: Erweiterter Speicher **204**

- Einführung • Files im erweiterten Speicher
- Operationen mit Programmfiles • Erzeugen von Daten- und Textfiles
- Zeiger in Daten- und Textfiles • Operationen mit Datenfiles
- Operationen mit Textfiles

Abschnitt 14: Der Texteditor **228**

- Einführung • Die Texteditor-Anzeige • Texteditor-Operationen (ED)
- Verwenden der Funktion ED in Programmen

Teil IV: Zeitfunktionen im Detail

Abschnitt 15: Uhr- und Datumsfunktionen **236**
- Einstellen und Korrigieren der Uhrzeit • Anzeige der Uhr
- Operationen mit Zeitwerten • Datumsfunktionen
- Berechnungen mit Datumswerten • Gültigkeitsbereiche und Fehlermeldungen

Abschnitt 16: Alarmfunktionen **246**
- Alarmarten: Meldealarme, Steueralarme, bedingte Alarme
- Einstellen von Alarmen • Aktivieren und Bestätigen von Meldealarmen
- Der Alarmkatalog • Löschen von Alarmen aus dem Speicher
- Überfällige Alarme • Anwendungsprogramme zur Alarmeinstellung

Abschnitt 17: Stoppuhrbetrieb **266**
- Das Stoppuhr-Tastenfeld • Allgemeiner Stoppuhrbetrieb mit Zeitnahme
- Programmierbare Stoppuhrfunktionen
- Die Stoppuhr als Countdown Timer • Ausdrucken gespeicherter Zeiten
- Beispiel für ein Stoppuhrprogramm

Teil V: Programmierung im Detail

Abschnitt 18: Einfache Programmierung **280**
- Laden eines Programms • Ausführen eines Programms • Programmzeilen
- Nicht programmierbare Operationen • Positionieren des Programmzeigers
- Modifizieren eins Programms • Löschen von Programmen

Abschnitt 19: Flags **288**
- Einführung • Flagtypen • Zusammenfassung der Flagzustände
- Flags und das X-Register

Abschnitt 20: Programmverzweigungen **298**
- Einführung • Verzweigen zu einem Label
- Aufrufen eines Unterprogramms • Bedingte Funktionen • Programmschleifen

Abschnitt 21: Alpha- und interaktive Funktionen **308**
- Einführung • Das Alpha-Register und das X-Register
- Manipulieren von Alpha-Strings • Eingabeaufforderungen
- Erzeugen von Meldungen und akustischen Signalen

Abschnitt 22: Programme zur Zeiterfassung **320**
- Einführung • Programmbeispiele • Benutzen der Programme
- Files zur Zeiterfassung • Erläuterung von ZE • Erläuterung von Σ

Anhänge

Anhang A: Fehler- und Statusmeldungen 354
Anhang B: Mehr über überfällige Alarme 360
Anhang C: Nullzeichen 366
Anhang D: Druckeroperationen 368
Anhang E: Erweiterte Speicher-Module 370
Anhang F: Zeitspezifikationen 374
Anhang G: Batterien, Gewährleistung und Serviceinformation 380
Anhang H: Peripheriegeräte, Erweiterungen und HP-IL 392
Anhang I: Vergleich HP-41CX und HP-41C/CV 398
Anhang J: Strichcode-Listings der Beispielprogramme 404
Funktionsverzeichnis 414
Sachindex 440
Funktionsindex 452

Abbildungen und Tabellen

Abbildungen

Das Alpha-Tastenfeld 157
Spezialtasten zur Spezifikation von Parametern 165
Das User-Tastenfeld 167
Aufteilung des Hauptspeichers 195
Das Texteditor-Tastenfeld 231
Flußdiagramm zur Alarmeinstellung 249
Das Alarmkatalog-Tastenfeld 256
Das Stoppuhr-Tastenfeld 269
Tastenpositionen und Tastencodes 316
Zulässige Belegungen der Einschubbuchsen
bei Verwendung von zwei Erweiterten Speicher-Modulen HP 82181A 370
Aufbau des erweiterten Speichers 373
Die Funktion CORRECT 376

Tabellen

Die Statistikregister 191
Zeiteinstellung 237
Anzeigeformate der Uhr 239
Zeitwerte 240
Anhängen eines Zeitwerts an den Inhalt des Alpha-Registers 241
Datumsformate 242
Anhängen eines Datums an den Inhalt des Alpha-Registers 243
Ergebnis der Alarmbestätigung 254
Das Alarmkatalog-Tastenfeld 257
Das Stoppuhr-Tastenfeld 268
Zusammenfassung der Flagzustände 293
Dezimalwerte und Flags 00 bis 07 294
Dezimalcodes der Anzeigezeichen 310
Verhalten überfälliger Alarme 362
Der Kompensationsfaktor 375
Anzeige einer Programmanweisung 396
Speicherkonfiguration 399
Äquivalente Begriffe 403

Inhalt von Band 1: Grundlagen der Bedienung

Verwenden dieses Handbuchs

Teil I: Grundlagen der Bedienung des HP-41

Abschnitt 1: Benutzen des Tastenfelds
Abschnitt 2: Die Anzeige
Abschnitt 3: Speichern und Zurückrufen von Zahlen
Abschnitt 4: Ausführen der Funktionen des HP-41
Abschnitt 5: Standardfunktionen des HP-41
Abschnitt 6: Zeitfunktionen
Abschnitt 7: Grundlagen der Programmierung
Abschnitt 8: Speichern von Text, Daten und Programmen in Files
Fehlerverzeichnis
Sachindex
Funktionsindex

Teil II
Grundlagen im Detail

Abschnitt 9

Das Tastenfeld und die Anzeige

Inhalt

Die Modustasten 155
Die Tastenfelder 155
  Das Normal-Tastenfeld 156
  Das User-Tastenfeld 156
  Das Alpha-Tastenfeld 156
  Spezial-Tastenfelder 158
Eingeben von Zahlen und Zeichen 158
  Eingeben von Zahlen 159
  Eingeben von Zeichen 159
Status-Indikatoren 160
Numerische Anzeigeformate 160
  Wahl des Anzeigeformats 160
  Ziffern- und Dezimaltrennzeichen 161
Standardanzeigen und Meldungen 161
Verschieben der Anzeige 162
Spezifizieren von Parametern 162
  Indirekte Spezifikation 162
  Sondertasten 164
Belegen der Tasten des User-Tastenfelds 166
  Der Katalog der Tastenfeld-Zuordnungen 168
  Aufheben von Tastenzuordnungen 168
  Die beiden obenliegenden Tastenreihen 169
Funktionsanzeige und Null 169
Die Kataloge 170
  Grundlagen der Katalogoperation 170
  Die einzelnen Kataloge 171
Fehlermeldungen 172

Die Modustasten

Die Modustasten sind unterhalb der Anzeige angeordnet: [ON], [USER], [PRGM], [ALPHA]. Diese Tasten legen fest, wie der Computer die anderen Tasten interpretiert. Wenn Sie eine Modustaste drücken, schalten Sie dadurch ein Tastenfeld (d.h. einen Operationsmodus) ein, das im allgemeinen aktiviert bleibt, bis Sie durch nochmaliges Drücken der gleichen Modustaste den vorhergehenden Zustand wiederherstellen.

Die [ON] Taste. Diese Modustaste schaltet den Computer ein und aus. Wenn für ca. 10 Minuten keine Eingabe erfolgt, schaltet sich der HP-41 automatisch aus, um die Batterien zu schonen.* Während der Computer ausgeschaltet ist, bleibt der Inhalt des Hauptspeichers und des erweiterten Speichers sowie der Zustand von bestimmten Flags durch den Permanentspeicher erhalten. Um den Einschaltzustand wiederherzustellen (d.h. den Hauptspeicher und den erweiterten Speicher zu löschen und den voreingestellten Status aller Flags zu wählen), können Sie wie folgt vorgehen:

1. Schalten Sie den Computer aus.
2. Drücken Sie [←] und halten Sie diese Taste niedergedrückt.
3. Drücken Sie die [ON] Taste.
4. Lassen Sie die [←] Taste los.

In der Anzeige erscheint die Meldung **MEMORY LOST**.

Die [USER] Taste. Diese Modustaste schaltet das User-Tastenfeld (die von Ihnen definierte Belegung des Tastenfelds) ein und aus. Wenn das User-Tastenfeld aktiviert ist, erscheint der **USER**-Indikator in der Anzeige und Flag 27 ist gesetzt.

Die [PRGM] Taste. Diese Modustaste schaltet den Computer zwischen dem Programm-Modus und dem Ausführungs-Modus hin und her. Wenn Sie den HP-41 einschalten, ist der Ausführungs-Modus aktiviert – Sie können Funktionen und Programme ausführen. Im Programm-Modus können Sie Programme eingeben oder modifizieren; Funktionen werden als Programmschritte aufgezeichnet und dann erst ausgeführt, wenn Sie das Programm im Ausführungs-Modus starten. Der **PRGM**-Indikator zeigt an, daß sich der Computer im Programm-Modus befindet oder daß im Ausführungs-Modus ein Programm abläuft.

Die [ALPHA] Taste. Diese Modustaste schaltet das Alpha-Tastenfeld ein und aus; der alphanumerische Zeichensatz beinhaltet unter anderem die in blauer Farbe auf den abgeschrägten Tastenvorderseiten stehenden Zeichen. Wenn das Alpha-Tastenfeld aktiviert ist, erscheint der **ALPHA**-Indikator in der Anzeige und Flag 48 ist gesetzt. Das Alpha-Tastenfeld wird durch Drücken von [ON] oder [PRGM] ausgeschaltet.

Die Tastenfelder

Funktionsnamen sind in diesem Handbuch farbig dargestellt; die jeweils gewählte Farbe zeigt an, wie die Funktion auszuführen ist. In der folgenden Übersicht über die Tastenfelder werden diese Farbkonventionen erläutert und Sie erhalten Informationen über die grundlegenden Eigenschaften der verschiedenen Tastenfelder des HP-41.

* Die Abschaltautomatik ist in Abhängigkeit von Flag 44 desaktiviert (Flag 44 gesetzt) oder aktiviert (Flag 44 gelöscht). Die Ausführung der Funktion [ON] bedingt ein Setzen von Flag 44, das Einschalten des Computers ein Löschen von Flag 44.

Das Normal-Tastenfeld

Über das Normal-Tastenfeld haben Sie Zugriff auf die Funktionen, die mit weißen Zeichen auf die Tasten aufgetragen sind und auf die Funktionen, die in goldfarbener Schrift über den Tasten erscheinen. Nach einem Löschen des Permanentspeichers ist das Normal-Tastenfeld aktiviert.

Wenn Sie die Taste ■ drücken, erscheint der **SHIFT**-Indikator um anzuzeigen, daß eine Zweitfunktion ausgeführt wird. Der Indikator erlischt, wenn Sie eine weitere Taste drücken (um die Zweitfunktion auszuführen), oder wenn Sie nochmals ■ drücken (um die Umschaltanweisung aufzuheben).

Erstfunktionen werden in diesem Handbuch durch den Funktionsnamen in schwarzer Schrift in einem schwarzen Tastenrahmen dargestellt; Zweitfunktionen erscheinen in goldfarbener Schrift in einem goldfarbenen Tastenrahmen. [LN] z.B. ist die Erstfunktion der Taste in der rechten oberen Ecke des Tastenfelds und [e^x] ist die umgeschaltete Zweitfunktion. Diese Regel gilt auch für andere Tastenfelder: [AVIEW] ist eine Zweitfunktion des Alpha-Tastenfelds.

Wenn eine Taste eine besondere Bedeutung in Zusammenhang mit dem auf der Tastenvorderseite stehenden Buchstaben hat, wird die Taste durch einen schwarzen Buchstaben in einem schwarzen Tastenrahmen dargestellt. Die Tastenfolge zur Ausführung von [ARCL] X zum Beispiel ist [ARCL] [·] [X], wobei [X] die Taste [6] repräsentiert.

Das User-Tastenfeld

Das User-Tastenfeld ist eine vom Benutzer definierte Version des Normal-Tastenfelds. Sie können jeder Taste (mit Ausnahme der Modustasten und der Umschalttaste ■) eine Funktion oder ein globales Label zuordnen. Die Funktion bzw. das durch das globale Label repräsentierte Programm kann nun durch Drücken der neudefinierten Taste des User-Tastenfelds ausgeführt werden.

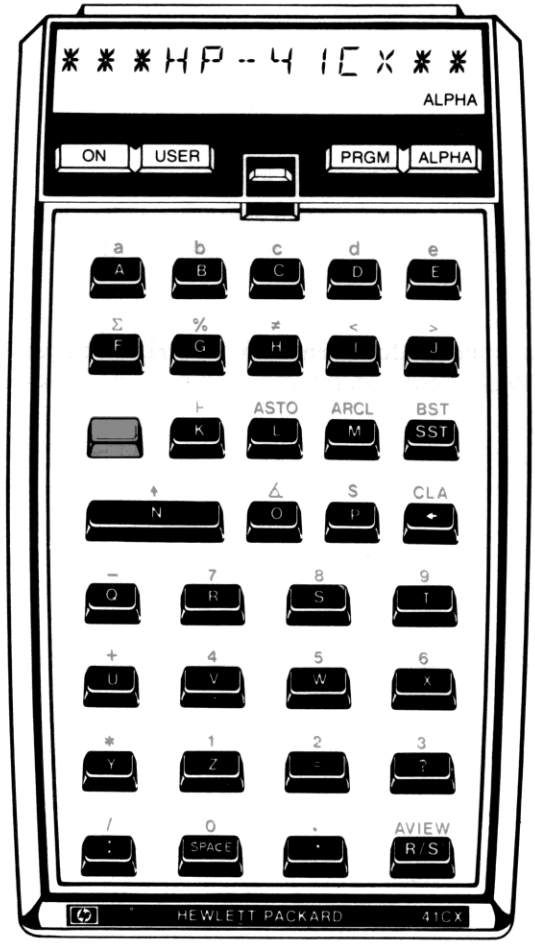
Auch die umgeschalteten Tastenpositionen können neu zugeordnet werden; eine Taste kann daher bis zu vier verschiedene Funktionen ausführen, abhängig davon, ob das User-Tastenfeld eingeschaltet ist und ob vor der Funktionstaste die Umschalttaste ■ gedrückt wird. Die Funktionsweise des User-Tastenfelds wird auf Seite 166 erläutert.

Viele Funktionen sind nicht auf dem Normal-Tastenfeld vorhanden, können aber dem User-Tastenfeld zugeordnet werden. Diese Funktionen werden in diesem Handbuch durch den in blauer Farbe gedruckten Funktionsnamen in einem blauen Tastenrahmen dargestellt.

Das Alpha-Tastenfeld

Über das Alpha-Tastenfeld haben Sie Zugriff auf Buchstaben, Funktionen, Symbole und Ziffern (die Ziffern werden vom Computer als Zeichen und nicht als Zahlen interpretiert). Die in blauer Farbe auf den Tastenvorderseiten erscheinenden Buchstaben und Symbole sind die Erstzeichen des Alpha-Tastenfelds. Die Ziffern 0 bis 9 und die arithmetischen Operatoren sind umgeschaltete Zeichen auf dem Alpha-Tastenfeld, d.h. Sie müssen zuerst die Umschalttaste drücken und dann die Taste, die das gewünschte Zeichen auf der Tastenoberseite zeigt. Das untenstehende Diagramm zeigt das gesamte Alpha-Tastenfeld, einschließlich aller Funktionen und umgeschalteter Zeichen.

Das Alpha-Tastenfeld



Das Alpha-Tastenfeld findet Verwendung bei den folgenden beiden Operationen:

- Eintasten eines Funktionsnamens oder eines globalen Labels als Parameter der Funktionen [ASN], [CLP], [COPY], [GTO], [LBL] und [XEQ]. In diesen Fällen sind die Alpha-Zeichen Teil der Anweisung.
- Eintasten von Zeichen in das Alpha-Register. Die Zeichen bleiben gespeichert, bis sie überschrieben werden oder das Alpha-Register gelöscht wird. Das Alpha-Register dient zur Anzeige von Meldungen, zur Spezifikation von Filenamen und globaler Labels bei der Ausführung von bestimmten Funktionen und zur Manipulation von Datenbytes.

In diesem Handbuch erscheinen Erstzeichen des Alpha-Tastenfelds in blauer Farbe; umgeschaltete Zeichen sind goldfarben gedruckt. Beachten Sie, daß eine goldfarben gedruckte Ziffer ein Alpha-Zeichen darstellt und eine in schwarzer Farbe gedruckte Ziffer eine Zahl des Normal-Tastenfelds repräsentiert.

Spezial-Tastenfelder

Außer den Tastenfeldern, die durch die Modustasten [USER] und [ALPHA] eingeschaltet werden, verfügt der HP-41 noch über Spezial-Tastenfelder. Diese werden durch das Ausführen von bestimmten Funktionen aktiviert. Die wichtigsten Beispiele hierzu sind:

- Das Texteditor-Tastenfeld, aktiviert durch [ED]. Dieses Tastenfeld basiert auf dem Alpha-Tastenfeld, enthält aber zusätzliche Funktionen zur Manipulation von Textfiles im erweiterten Speicher. Die Funktion [ED] wird in Abschnitt 14 beschrieben.
- Das Alarmkatalog-Tastenfeld, aktiviert durch [ALMCAT] und [CAT] 5. Die Funktionen dieses Tastenfelds erlauben Ihnen, die gespeicherten Alarme aufzulisten und zu modifizieren. Alarme werden in Abschnitt 16 beschrieben.
- Das Stoppuhr-Tastenfeld, aktiviert durch [SW] und [SWPT]. Stoppuhr-Operationen werden in Abschnitt 17 beschrieben.

Eingeben von Zahlen und Zeichen

Es besteht kein wesentlicher Unterschied, ob nun Zahlen über das Normal- oder User-Tastenfeld in das X-Register (die gewöhnlich verwendete numerische Anzeige) eingegeben werden oder Zeichen über das Alpha-Tastenfeld in das Alpha-Register eingetastet werden. In beiden Fällen gelten die folgenden Regeln:

- Wenn Sie die erste Ziffer bzw. das erste Zeichen eingeben, erscheint hinter der Ziffer bzw. dem Zeichen das *Eingabezeichen* (_) in der Anzeige.
- Das Eingabezeichen bedeutet, daß die nächste Eingabe an die angezeigte Ziffern- bzw. Zeichenkette angehängt wird.
- Bei angezeigtem Eingabezeichen kann die Eingabe durch Drücken der Taste [←] korrigiert werden. Dabei wird das rechts außen stehende Zeichen bzw. die rechts außen stehende Ziffer gelöscht und das Eingabezeichen rückt um eine Stelle nach links.*
- Wenn die Eingabe beendet wird, verschwindet das Eingabezeichen aus der Anzeige. Ihre nächste Eingabe wird dann vom Computer als Anfang einer neuen Ziffern- oder Zeichenkette betrachtet.*

* Wenn Sie zehn Ziffern oder einen zweistelligen Exponenten in das X-Register eingeben, verschwindet das Eingabezeichen, da die Anzeige nicht über mehr Stellen verfügt. Die Eingabe ist jedoch noch nicht abgeschlossen: Ihre nächste Eingabe beginnt *keine* neue Zahl und Drücken der Taste [←] *löscht* die rechts außen stehende Ziffer.

Eingeben von Zahlen

Sie können bis zu 10 Ziffern in das X-Register eingeben – weitere Ziffern werden vom Computer ignoriert. Die einzigen zur Zifferneingabe verwendeten Tasten sind die Zifferntasten [0] bis [9] sowie die Tasten [.], [CHS] (*change sign*), [EEX] (*enter exponent*) und [←]. Die Zahleneingabe wird beendet, wenn Sie eine andere Taste als die Zifferneingabe-Tasten, [■] oder [USER] drücken – jede weitere Ziffer wird vom HP-41 als Teil einer neuen Zahl betrachtet.

Das Drücken der Taste [CLx] ersetzt den Inhalt des X-Registers durch Null; diese Null wird von der als nächstes eingegebenen Ziffer überschrieben. Die Taste [←] hat den gleichen Effekt wie die Taste [CLx], wenn die Anzeige nur eine einzige Ziffer enthält oder wenn die Zahleneingabe abgeschlossen ist.

Eingabe von Exponenten. Um eine Zahl der Form $a \times 10^b$ einzugeben, tasten Sie zuerst die Ziffern und das Dezimaltrennzeichen von a ein; drücken Sie die Taste [CHS], wenn a negativ ist. Um mehr als acht Ziffern für a einzugeben, müssen Sie vor der neunten Ziffer an irgendeiner Stelle ein Dezimaltrennzeichen einfügen.

Drücken Sie die Taste [EEX]. Alle rechts von der achten Ziffer stehenden Ziffern verschwinden aus der Anzeige, bleiben aber intern erhalten. Geben Sie nun den ein- oder zweistelligen Exponenten b ein und drücken Sie [CHS], wenn b negativ ist. Wenn Sie [EEX] drücken, ohne vorher einen Wert für a eingegeben zu haben, verwendet der Computer automatisch für a den Wert 1.

Eingabe von π. Das Drücken der Taste [π] hat den gleichen Effekt wie das Eintasten der Zahl 3.141592654 und anschließendes Beenden der Zahleneingabe.

Eingeben von Zeichen

Im Ausführungs-Modus. Die eingetasteten Zeichen werden im Alpha-Register gespeichert, wenn das Alpha-Tastenfeld eingeschaltet ist und durch die Eingabe kein Parameter spezifiziert wird. Wenn Sie innerhalb eines Programms eine Eingabe in das Alpha-Register über das Tastenfeld vornehmen wollen, fügen Sie eine [AON] Anweisung vor der Anweisung ein, die das Programm unterbricht. Vor der Fortsetzung der Programmausführung ist dann das Alpha-Tastenfeld durch eine [AOFF] Anweisung wieder auszuschalten.

Im Alpha-Register können bis zu 24 Zeichen abgelegt werden. Bei der Eingabe des 24. Zeichens erzeugt der Computer ein Tonsignal, um Sie zu warnen, daß die Speicherkapazität des Alpha-Registers erschöpft ist. Wenn das Alpha-Register voll ist und Sie ein weiteres Zeichen eingeben, wird das links außen stehende Zeichen aus dem Alpha-Register geschoben und geht damit verloren.

Die Zeicheneingabe wird durch [ASTO], [BST], [SST], [AVIEW], [R/S] oder durch Ausschalten des Alpha-Tastenfelds beendet. Die Zeicheneingabe kann nach Drücken von [⊢] (*append*) oder [ARCL] fortgesetzt werden.

Durch Drücken der Taste [CLA] werden alle Zeichen aus dem Alpha-Register gelöscht. Wenn die Zeicheneingabe abgeschlossen ist, hat die Taste [←] den gleichen Effekt wie die Taste [CLA].

Im Programm-Modus. In einer Programmzeile können bis zu 15 Zeichen gespeichert werden; in der Anzeige steht vor einem Alpha-String ein «hochgestelltes T» (ᵀ). Die in einer Programmzeile gespeicherten Alpha-Zeichen werden in das Alpha-Register übertragen, wenn das Programm ausgeführt wird. Setzen Sie ein [⊢] an den Anfang eines Zeichenstrings, wenn Sie den String in das Alpha-Register eingeben wollen, ohne daß der vorhergehende Inhalt gelöscht wird. Sie können zum Beispiel mehr als 15 Zeichen im Alpha-Register ablegen, indem Sie zwei Programmzeilen benutzen und an den Anfang der zweiten Zeile eine [⊢] Anweisung setzen. (Das Zeichen ⊢ erscheint nur, wenn die Programmzeile angezeigt wird. Die «Anhängoperation» wird erst ausgeführt, wenn das Programm abläuft.)

Hinweis: Bei Ausgabe auf einen Drucker oder einen Bildschirm erscheinen Alpha-Strings zwischen Anführungszeichen. Programmzeilen sind nur dann Alpha-Strings, wenn sie mit einem Anführungszeichen beginnen und enden; wenn eine Programmzeile *nicht* in Anführungszeichen gesetzt ist, handelt es sich um eine Funktion. Verwechseln Sie nicht einen Ihnen unbekannten Funktionsnamen mit einem Alpha-String – drücken Sie auf jeden Fall die Taste [XEQ], bevor Sie einen Funktionsnamen eintasten.

Status-Indikatoren

Die Status-Indikatoren erscheinen am unteren Rand der Anzeige. Außer den oben erwähnten Indikatoren (**USER**, **PRGM**, **ALPHA** und **SHIFT**) verfügt der HP-41 noch über die folgenden Status-Indikatoren:

- Bei abfallender Batteriespannung erscheint der **BAT**-Indikator in der Anzeige. Bei Verwendung von Alkalibatterien kann der Computer nach dem ersten Aufleuchten des **BAT**-Indikators noch ungefähr 5 bis 15 Tage betrieben werden; bei Verwendung des wiederaufladbaren Batteriensatzes HP 82120A verbleiben noch ungefähr 2 bis 50 Minuten Betriebszeit. Diese Zeitspannen reduzieren sich, wenn Sie den Magnetkartenleser HP 82104A oder den optischen Lesestift HP 82153A verwenden. Nähere Informationen zu den Batterien finden Sie in Anhang G.
- Die Indikatoren **GRAD** und **RAD** zeigen an, in welchem trigonometrischen Modus (Neugrad oder Bogenmaß) sich der Computer befindet. (Wichtig bei Ausführung der trigonometrischen Funktionen und bei Umwandlung zwischen rechtwinkligen und Polarkoordinaten.) Wenn weder **GRAD** noch **RAD** angezeigt wird, befindet sich der HP-41 im Altgrad-Modus.
- **0 1 2 3 4** bedeutet, daß der entsprechende Flag (00, 01, 02, 03 oder 04) gesetzt ist.

Einige Status-Indikatoren haben eine besondere Bedeutung, wenn das Texteditor-Tastenfeld ([ED]) eingeschaltet ist. Wenn Sie den Texteditor-Modus verlassen, wird wieder der ursprüngliche Zustand hergestellt.

Numerische Anzeigeformate

Intern stellt der Computer jede Zahl in der Form $a \times 10^b$ dar. (Dabei ist a eine Zahl mit neun Dezimalstellen, $1 \leqslant |a| < 10$ und b eine zweistellige ganze Zahl, $0 \leqslant |b| < 100$. Sie können das Anzeigeformat für Zahlen festlegen, ohne dabei die interne Zahlendarstellung zu verändern. (Wenn Sie die interne Zahlendarstellung mit der Anzeige in Übereinstimmung bringen wollen, schlagen Sie bitte auf Seite 000 die Rundungsfunktion [RND] nach.) Das von Ihnen eingestellte Anzeigeformat und die gewählten Trennzeichen werden vom Permanentspeicher erhalten.

Wahl des Anzeigeformats

Der HP-41 verfügt über drei Anzeigeformate für Zahlen, die durch die Funktionen [FIX], [SCI] und [ENG] eingestellt werden.

[FIX] *n*. In diesem Format werden Zahlen mit bis zu n Dezimalstellen angezeigt ($0 \leqslant n \leqslant 9$). Wenn der ganzzahlige Anteil einer Zahl mehr als $(10-n)$ Stellen hat, werden entsprechend weniger als n Dezimalstellen angezeigt. Das voreingestellte Format zum Beispiel ist [FIX] **4**, d.h. Zahlen werden mit vier Dezimalstellen angezeigt; wenn eine Zahl acht Stellen vor dem Dezimaltrennzeichen hat, werden nur zwei Dezimalstellen angezeigt.

Die letzte angezeigte Ziffer ist aufgerundet, wenn die erste nicht mehr angezeigte Ziffer größer oder gleich 5 ist. Wenn der Dezimalanteil einer Zahl weniger als n Stellen hat, werden nachlaufende Nullen angehängt. Wenn der Betrag einer Zahl zu groß oder zu klein für das gewählte Format ist, schaltet der Computer vorübergehend auf das Anzeigeformat [SCI] *n*.

[SCI] *n*. In diesem Format wird eine Zahl durch eine Ziffer vor dem Dezimaltrennzeichen, *n* Dezimalstellen ($0 \leqslant n \leqslant 9$) und einen zweistelligen Zehnerexponenten dargestellt. Wenn $n \leqslant 7$ ist, wird die Zahl auf *n* Dezimalstellen gerundet. Es können maximal 7 Dezimalstellen angezeigt werden, d.h. bei Verwendung von [SCI] **8** oder [SCI] **9** rundet der Computer eine nicht angezeigte Dezimalstelle. (Diese Formate erweisen sich bei Verwendung eines Druckers als nützlich.)

[ENG] *n*. Dieses Anzeigeformat unterscheidet sich von dem Format [SCI] *n* nur dadurch, daß der Exponent immer ein Vielfaches von drei ist. Das Dezimaltrennzeichen wird nach rechts verschoben, um Änderungen des Exponenten auszugleichen.

Ziffern- und Dezimaltrennzeichen

Die Flags 28 und 29 bestimmen, wie Punkte und Kommata bei der Anzeige von Zahlen verwendet werden. In Mitteleuropa ist normalerweise das Komma das Dezimaltrennzeichen (Trennung von ganzzahligem und gebrochenem Anteil) und der Punkt das Zifferntrennzeichen (Bildung von Zifferngruppen bei großen Zahlen). In den angelsächsischen Ländern dient der Punkt als Dezimaltrennzeichen und das Komma als Zifferntrennzeichen.

Flag 28 legt die Rolle von Punkt und Komma fest. In der Voreinstellung ist Flag 28 *gesetzt* und der Punkt ist das Dezimaltrennzeichen und das Komma das Zifferntrennzeichen. Bei gelöschtem Flag 28 wird das Komma als Dezimaltrennzeichen und der Punkt als Zifferntrennzeichen verwendet.

Flag 29 legt fest, ob ein Zifferntrennzeichen angezeigt wird oder nicht; dies geschieht unabhängig davon, welches Symbol als Zifferntrennzeichen verwendet wird. In der Voreinstellung ist Flag 29 *gesetzt* und das Zifferntrennzeichen wird angezeigt. Bei gelöschtem Flag 29 wird kein Zifferntrennzeichen angezeigt und, wenn das Anzeigeformat [FIX] **0** eingestellt ist, das Dezimaltrennzeichen unterdrückt.

Die Flags 28 und 29 ermöglichen es Ihnen, die Darstellung zu wählen, mit der Sie am besten vertraut sind.

Standardanzeigen und Meldungen

Die Anzeige des Computers enthält entweder eine Standardanzeige oder eine Meldung. Der Inhalt des X-Registers ist die Standardanzeige, außer in den folgenden Fällen:

- Das Alpha-Tastenfeld ist eingeschaltet und es wird kein Parameter spezifiziert. In diesem Fall ist der Inhalt des Alpha-Registers die Standardanzeige.
- Wenn sich der Computer im Programm-Modus befindet, ist die momentane Programmzeile die Standardanzeige.
- Wenn ein Programm ausgeführt wird, ist der Indikator für ein ablaufendes Programm (↻) die Standardanzeige.

Jede andere Anzeige ist eine Meldung. Dies gilt unter anderem für den Texteditor (Abschnitt 14), die Uhr (Abschnitt 15) und die Stoppuhr (Abschnitt 17) sowie für in Programmen verwendete Meldungen (Abschnitt 21). Die Beispiele in diesem Abschnitt behandeln unter anderem die Anzeige bei der Spezifikation von Parametern, die Funktionsanzeige, die Kataloge sowie Fehlermeldungen. Wenn die Anzeige eine Meldung enthält, ist Flag 50 gesetzt.

Verschieben der Anzeige

Um mehr Zeichen anzuzeigen als die Anzeige auf einmal aufnehmen kann, schiebt der Computer die Zeichenkette durch die Anzeige, bis daß das letzte Zeichen sichtbar wird. Wenn Sie während des Verschiebens irgendeine Taste drücken, wird sofort das rechte Ende der Zeichenkette angezeigt. Die Funktion, die der gedrückten Taste zugeordnet ist, wird nicht ausgeführt.

Spezifizieren von Parametern

Einige Funktionen können erst ausgeführt werden, nachdem ein Parameter eingegeben wurde. Sie müssen einen Parameter spezifizieren, wenn hinter dem Funktionsnamen ein oder mehrere Eingabezeichen (_) in der Anzeige erscheinen.

- Beachten Sie bei Spezifikation eines numerischen Parameters (z.B. Registeradresse, Flagnummer, lokales numerisches Label oder Nummer einer Programmzeile), wieviele Eingabezeichen angezeigt werden und geben Sie dann die gewünschten Ziffern ein. (Sie müssen unter Umständen führende Nullen hinzuzufügen, z.B. 042 zur Spezifikation der Programmzeile 42.)
- Spezifikation eines Alpha-Parameters (z.B. Funktionsname oder globales Label). Drücken Sie [ALPHA], um das Alpha-Tastenfeld einzuschalten, tasten Sie den Namen oder das Label ein und drücken Sie dann wieder [ALPHA], um die Parameterspezifikation abzuschließen.

Indirekte Spezifikation

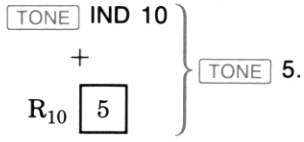
Die Parameter der meisten Funktionen können indirekt spezifiziert werden. Als Reaktion auf das Eingabezeichen geben Sie nicht den Parameter selbst ein, sondern die Adresse eines Registers (das «indirekte Register»), das den Parameter enthält. Die indirekte Spezifikation ist besonders nützlich, wenn der Wert eines Parameters von vorhergehenden Berechnungen innerhalb eines Programms abhängt oder wenn ein Programmteil mehrmals ausgeführt wird, um auf verschiedene Register zuzugreifen. Die Hauptspeicher-Register $R_{(100)}$ bis $R_{(318)}$ können immer nur indirekt spezifiziert werden.

Gehen Sie folgendermaßen vor, um einen Parameter indirekt zu spezifizieren:

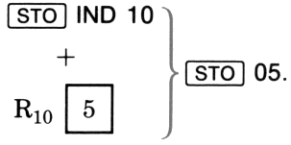
1. Führen Sie die Funktion aus.
2. Drücken Sie ■, wenn das Eingabezeichen angezeigt wird. Daraufhin erscheint hinter dem Funktionsnamen IND__ in der Anzeige.
3. Spezifizieren Sie das indirekte Register.

In den folgenden Beispielen wird die indirekte Spezifikation für drei verschiedene Arten von Parametern gezeigt. In jedem Beispiel ist R_{10} das indirekte Register, das den Parameter 5 enthält. Im ersten Beispiel ist 5 nur eine Zahl, im zweiten Beispiel ist 05 eine Adresse und im dritten Beispiel ist 05 ein Label.

Beispiel: In R_{10} ist die Zahl 5 gespeichert. Wenn Sie [TONE] **IND 10** ausführen, wird die Zahl in R_{10} zum Parameter für die Funktion [TONE]. [TONE] **IND 10** ist gleichbedeutend mit [TONE] **5**, sofern in R_{10} die Zahl 5 abgelegt ist.

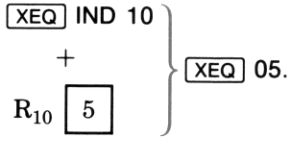


Beispiel: In R_{10} ist die Zahl 5 gespeichert. Wenn Sie [STO] **IND 10** ausführen, wird die Adresse in R_{10} zum Parameter für [STO]. [STO] **IND 10** ist gleichbedeutend mit [STO] **05**, sofern in R_{10} die Zahl 5 abgelegt ist.



Die indirekte Spezifikation eines Parameters wird sehr häufig zur *«indirekten Adressierung»* verwendet, d.h. zur indirekten Angabe einer Registeradresse. Mit Hilfe der indirekten Adressierung kann innerhalb einer Programmschleife auf mehrere Register zugegriffen werden. Wenn eine Schleife zum Beispiel die Anweisungen [RCL] **IND 10**, [1/X], [STO] **IND 10** enthält, wird die Zahl in R_{05} durch ihren Kehrwert ersetzt, sofern R_{10} die Zahl 5 enthält (siehe vorhergehendes Beispiel). Die Schleife kann die Anweisung enthalten, die in R_{10} abgelegte Adresse von 5 auf 6 zu erhöhen und wieder von vorne zu beginnen. Beim nächsten Durchlaufen der Schleife wird dann die Zahl in R_{06} durch ihren Kehrwert ersetzt und die Adresse in R_{10} wird von 6 auf 7 erhöht. Programmschleifen werden in Abschnitt 20 («Programmverzweigungen») erläutert.

Beispiel: In R_{10} ist die Zahl 5 gespeichert. Wenn Sie [XEQ] **IND 10** ausführen, wird das Label in R_{10} zum Parameter für die Funktion [XEQ]. [XEQ] **IND 10** ist gleichbedeutend mit [XEQ] **05**, sofern in R_{10} die Zahl 5 abgelegt ist.



Die Parameter der in Katalog 1 aufgelisteten globalen Label und die Parameter der in Katalog 2 aufgelisteten programmierbaren Funktionen und globalen Label können ebenfalls indirekt spezifiziert werden; hier gilt die Einschränkung, daß ein Label nicht mehr als sechs Zeichen umfassen darf.

Die Parameter der folgenden Funktionen können indirekt spezifiziert werden.

- Funktionen, deren Parameter ein Label ist: [XEQ], [GTO].
- Funktionen, deren Parameter eine Register-Adresse ist:

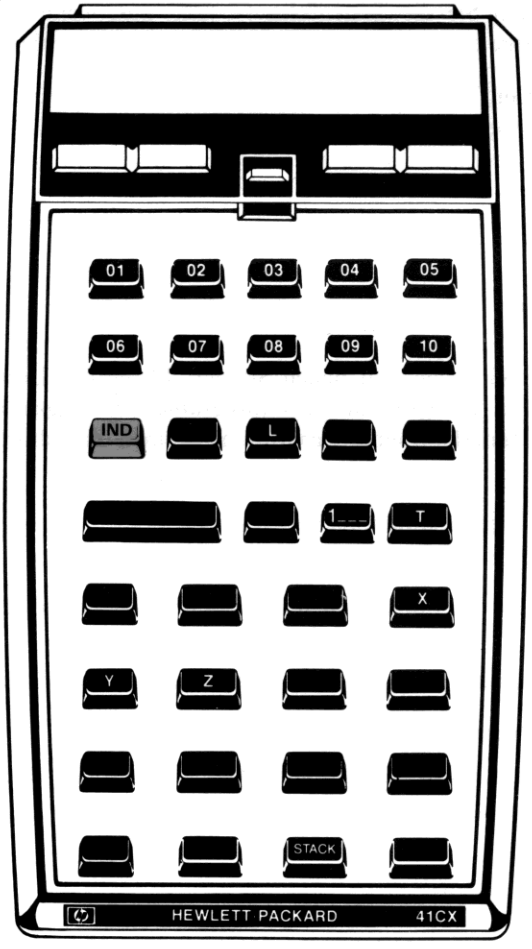
  [STO], [RCL]
  [STO] [+], [STO] [−], [STO] [×], [STO] [÷]
  [ASTO], [ARCL]
  [ISG], [DSE]
  [X< >]
  [VIEW]
  [ΣREG]

- [SF], [CF], [FS?], [FC?], [FS?C], [FC?C].
- Funktionen zum Einstellen des Anzeigeformats: [FIX], [SCI], [ENG].
- [TONE].

Sondertasten

Die nachstehende Abbildung zeigt die Tasten, die eine besondere Bedeutung haben, wenn Sie den Parameter einer in Katalog 3 enthaltenen Funktion spezifizieren.

Spezialtasten zur Spezifikation von Parametern



Stackregister-Adressen. Um ein Stackregister oder das LAST X Register zu spezifizieren, drücken Sie die Taste [.], gefolgt von [X], [Y], [Z], [T] oder [L].

Programmzeilen-Nummern. Führen Sie die Tastenfolge [.] [EEX] aus, wenn Sie eine Zeilennummer über 999 spezifizieren wollen. In der Anzeige erscheint 1___. Geben Sie nun die verbleibenden drei Ziffern ein.

Vereinfachte Parameterspezifikation. Sie können einen einstelligen Parameter (0 bis 9) oder einen zwei- bzw. dreistelligen Parameter (1 bis 10) spezifizieren, indem Sie die entsprechende Taste der beiden oberen Tastenreihen drücken. Beispiel: Wenn ein, zwei oder drei Eingabezeichen angezeigt werden, wird durch Drücken der Taste [Σ+] der Parameter 1, 01 oder 001 spezifiziert. Wenn nur ein Eingabezeichen angezeigt wird, spezifiziert [TAN] den Parameter 0; wenn zwei bzw. drei Eingabezeichen angezeigt werden, spezifiziert [TAN] den Parameter 10 bzw. 010.

Belegen der Tasten des User-Tastenfelds

Der HP-41 verfügt über zwei Funktionen, mit denen Sie den Tasten des User-Tastenfelds globale Label und Funktionen zuordnen können; [ASN] (*assign*) und [PASN] (*programmable assign*). Benutzen Sie die Funktion [ASN], um Tasten manuell zu belegen. Innerhalb eines Programms können Zuordnungen durch die Funktion [PASN] vorgenommen werden.

Zuordnen einer Taste über das Tastenfeld (manuelle Belegung):

1. Führen Sie die Funktion [ASN] aus.
2. Drücken Sie [ALPHA], tasten Sie den Funktionsnamen oder das globale Label ein und drücken Sie dann wieder [ALPHA].
3. Drücken Sie die Taste (oder ■ und die Taste), der die Funktion bzw. das globale Label zugeordnet werden soll.

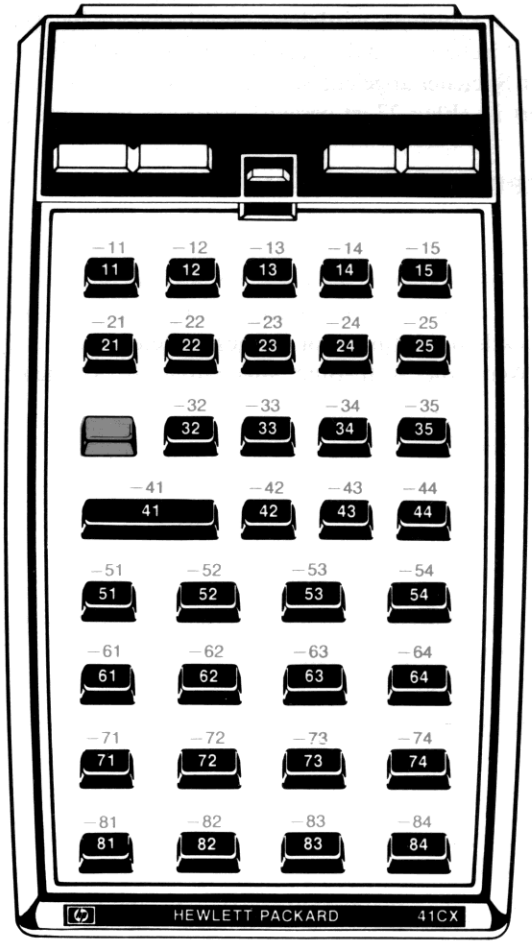
Zuordnen einer Taste in einem Programm:

1. Geben Sie den Funktionsnamen bzw. das globale Label in das Alpha-Register ein.
2. Legen Sie den Tastencode der zu belegenden Taste im X-Register ab. (Die Tastencodes entnehmen Sie bitte der untenstehenden Abbildung.)
3. Führen Sie die Funktion [PASN] aus.

Die untenstehende Abbildung zeigt die Tastencodes des User-Tastenfelds. Beachten Sie bitte die folgenden Erläuterungen:

- Alle Tastencodes sind zweistellig.
- Die Tastencodes der umgeschalteten Tastenpositionen sind negativ.
- Die Modustasten und die Umschalttaste (■) können nicht neu belegt werden.
- Die Taste [R/S] kann neu belegt werden. Ihre Zuordnung überlagert im Ausführungs-Modus die Funktion [RUN] und im Programm-Modus die Funktion [STOP]. Ein laufendes Programm kann jedoch weiterhin durch Drücken von [R/S] angehalten werden.

Das User-Tastenfeld



Wenn Sie einer Taste eine in Katalog 2 oder 3 aufgelistete Funktion oder ein in Katalog 2 aufgelistetes globales Label zuordnen, wird diese Zuordnung im Speicher des User-Tastenfelds aufgezeichnet. (Der Speicher des User-Tastenfelds ist Teil des Hauptspeichers und wird in Abschnitt 12 beschrieben.) Wenn Sie einer Taste jedoch ein in Katalog 1 aufgelistetes globales Label zuordnen, wird diese Zuordnung als Teil des Labels gespeichert. Wenn Sie das Label aus dem Programmspeicher löschen, wird die Tastenzuordnung aufgehoben. Wenn das Programm mit dem zugeordneten Label im erweiterten Speicher abgelegt ist und das User-Tastenfeld beim Abrufen des Programms aus dem erweiterten Speicher aktiviert ist (Flag 27 ist gesetzt), wird die Tastenzuordnung wieder wirksam.

Der Katalog der Tastenfeld-Zuordnungen

Die Tastenfolge [CATALOG] **6** bedingt eine Auflistung aller dem User-Tastenfeld zugeordneten Funktionen und globalen Labels. Die Auflistung erfolgt in der Reihenfolge der Tastencodes, beginnend mit den kleineren Werten (auf dem Tastenfeld von links nach rechts und von oben nach unten). Ein positiver Tastencode wird vor einem negativen aufgelistet (nicht umgeschaltete Taste vor umgeschalteter Taste). Die Katalogausgabe wird durch Drücken von [R/S] angehalten. Wenn die Auflistung angehalten ist, können Sie die folgenden Operationen ausführen:

- Drücken Sie [SST], um die nächste Zuordnung anzuzeigen.
- Drücken Sie [BST], um die vorhergehende Zuordnung anzuzeigen.
- Drücken Sie [f] [C], um die angezeigte Zuordnung zu löschen.
- Drücken Sie [R/S], um die Katalogausgabe fortzusetzen.

Wenn keine Tastenzuordnungen vorhanden sind oder wenn die letzte verbleibende Zuordnung durch [f] [C] gelöscht wird, erscheint die Meldung **CAT EMPTY** in der Anzeige.

Aufheben von Tastenzuordnungen

Sie können folgendermaßen vorgehen, um Tastenzuordnungen des User-Tastenfelds aufzuheben:

- Führen Sie die Funktion [ASN] aus, drücken Sie zweimal die Taste [ALPHA] und drücken Sie dann die Taste, deren Zuordnung gelöscht werden soll.
- Löschen Sie das Alpha-Register, geben Sie den entsprechenden Tastencode in das X-Register ein und führen Sie dann die Funktion [PASN] aus.
- Führen Sie die Tastenfolge [f] [C] aus, wenn die Ausgabe von Katalog 6 angehalten ist und die entsprechende Zuordnung angezeigt wird.

Führen Sie die Funktion [CLKEYS] aus, um alle momentan wirksamen Tastenzuordnungen aufzuheben. Zuordnungen, die als Teil eines globalen Labels im erweiterten Speicher abgelegt sind, werden wieder wirksam, wenn beim Abrufen des Programms das User-Tastenfeld aktiviert ist (Flag 27 ist gesetzt).

Die beiden oberen Reihen des Tastenfelds

Eine besondere Art von Programmlabel, das lokale Alpha-Label, ist darauf zurechtgeschnitten, in Zusammenhang mit den beiden oberen Tastenreihen des User-Tastenfelds benutzt zu werden. Der Name jedes Labels stimmt mit einem Alpha-Zeichen dieser beiden Tastenreihen überein: **A** bis **E** auf der ersten Reihe, **F** bis **J** auf der zweiten Reihe und **a** bis **e** auf der umgeschalteten ersten Reihe. Die Programmierung mit diesen Labeln wird in Abschnitt 20 («Programmverzeigungen») diskutiert. An dieser Stelle wird nur erläutert, unter welchen Bedingungen ein lokales Alpha-Label über das User-Tastenfeld angesprungen werden kann:

- Das User-Tastenfeld ist eingeschaltet.
- Das momentane Programm enthält das lokale Alpha-Label.
- Die Taste, die dem lokalen Alpha-Label entspricht, wurde nicht neu belegt.

Zusammen mit den allgemeinen Regeln für das User-Tastenfeld ergeben diese Bedingungen die folgenden Prioritäten, wenn eine Taste der beiden oberen Reihen des User-Tastenfelds gedrückt wird:

1. Wenn Sie die Taste mit einer Funktion oder einem globalen Label belegt haben, wird die Funktion ausgeführt bzw. die Programmausführung beginnt bei dem globalen Label.
2. Wenn Sie die Taste nicht neu zugeordnet haben und das lokale Alpha-Label innerhalb des momentanen Programms existiert, beginnt die Ausführung bei dem lokalen Alpha-Label.
3. Wenn keine der beiden ersten Bedingungen erfüllt ist, wird die entsprechende Funktion des Normal-Tastenfelds ausgeführt (d.h. die Funktion, die auf bzw. über der Taste steht).

Die Ausführung einer Funktion des Normal-Tastenfelds über das User-Tastenfeld kann beträchtlich mehr Zeit in Anspruch nehmen, da der Computer zuerst die höheren Prioritäten überprüft. Um diese Verzögerung bei der Ausführung einer Funktion des Normal-Tastenfelds zu vermeiden, können Sie das User-Tastenfeld vor Drücken der Taste ausschalten oder die Funktion der entsprechenden Taste des User-Tastenfelds zuordnen.

Funktionsanzeige und Null

Sie können überprüfen, welche Funktion einer Taste momentan zugeordnet ist, ohne die Funktion ausführen zu müssen. Sie brauchen dazu lediglich die gewünschte Taste gedrückt zu halten. Diese Form der Funktionsanzeige ist besonders hilfreich in Verbindung mit dem User-Tastenfeld, wenn Sie nicht sicher sind, welche Tasten neu belegt wurden.

- Lassen Sie die Taste los, wenn die Funktion die Eingabe eines Parameters erfordert (ein oder mehrere Eingabezeichen werden angezeigt). Drücken Sie [←], um die Funktion zu löschen.
- Wenn es sich nicht um eine Parameterfunktion handelt, wird die Funktion ausgeführt, wenn Sie die Taste loslassen. Sie können dies verhindern, indem Sie die Taste so lange gedrückt halten, bis NULL in der Anzeige erscheint.

In vier weiteren Fällen wird eine Programmzeile angezeigt (unter der Voraussetzung, daß Sie die Taste loslassen, bevor NULL angezeigt wird).

- Wenn das User-Tastenfeld eingeschaltet ist und Sie eine Taste drücken, der ein globales Label zugeordnet ist, wird das Label angezeigt und die Programmausführung beginnt bei dem Label.
- Wenn das User-Tastenfeld eingeschaltet ist und Sie eine Taste drücken, die einem lokalen Alpha-Label innerhalb des momentanen Programms entspricht, erscheint XEQ *Label* in der Anzeige und die Programmausführung beginnt bei dem lokalen Alpha-Label.
- Wenn Sie die Taste [R/S] drücken, wird die momentane Programmzeile angezeigt und die Programmausführung beginnt bei der momentanen Zeile.
- Wenn Sie die Taste [SST] drücken, wird die momentane Programmzeile angezeigt und nur diese Zeile wird ausgeführt.

Die Kataloge

Sie können auf sechs Kataloge zugreifen, um den Inhalt des Speichers zu überprüfen. Die Funktion [CATALOG] ist nicht programmierbar; der HP-41 verfügt jedoch über programmierbare Funktionen, die den Katalogen 4 und 5 entsprechen. Im folgenden werden zunächst die allgemeinen Grundlagen der Katalogoperation erläutert; im Anschluß daran wird jeder Katalog im einzelnen vorgestellt.

Grundlegende Katalogoperationen

Führen Sie die Funktion [CATALOG] *n* aus, um die Ausgabe von Katalog *n* zu veranlassen.

Operationen bei laufender Katalogausgabe:

- Durch Drücken jeder Taste außer [R/S] und [ON] wird die Katalogausgabe beschleunigt.
- Durch Drücken von [R/S] wird die Katalogausgabe angehalten.

Operationen bei angehaltener Katalogausgabe:

- Durch Drücken von [SST] wird der nächste Katalogeintrag angezeigt.
- Durch Drücken von [BST] wird der vorhergehende Katalogeintrag angezeigt.
- Durch Drücken von [R/S] wird die Katalogausgabe fortgesetzt.
- Durch Drücken von [←] wird der Katalog-Modus verlassen.

Die Kataloge 4, 5 und 6 verbrauchen soviel Strom wie ein ablaufendes Programm (selbst bei angehaltener Katalogausgabe). Der Computer verläßt daher den Katalog-Modus, wenn für ungefähr zwei Minuten keine Operation ausgeführt wurde.

Wenn sich ein angeschlossener Drucker im Trace-Modus befindet, wird automatisch eine Liste des Katalogs gedruckt.

Die einzelnen Kataloge

Katalog 1: Benutzerprogramme. Eine Auflistung aller globalen Labels und END-Anweisungen. Zusammen mit jeder END-Anweisung wird die Zahl der von dem Programm belegten Bytes angezeigt. Zusammen mit der permanenten END-Anweisung (.END.) wird angezeigt, wieviele Register noch für Programmierzwecke zur Verfügung stehen.

Sie können den Katalog 1 dazu benutzen, ein Programm zum momentanen Programm zu machen. Halten Sie die Katalogausgabe durch Drücken von [R/S] an, wenn das globale Label oder die END-Anweisung des gewünschten Programms angezeigt wird; drücken Sie dann [←], um den Katalog-Modus zu verlassen. (Seite 284.)

Katalog 2: Externe Funktionen. Eine Auflistung aller Funktionen und Programme, die dem Computer über Peripheriegeräte und Einsteck-Module zur Verfügung stehen; auch die Funktionen des erweiterten Speichers und die Zeitfunktionen werden in Katalog 2 aufgeführt. Globalen Labels ist ein ᵀ vorangestellt, um sie von Funktionen zu unterscheiden.

Funktionen und Programme sind ihrem Ursprung nach geordnet. Der Katalog führt normalerweise nur die Haupteinträge (Überschriften) jeder Gruppe auf. Gehen Sie folgendermaßen vor, um alle Einträge anzuzeigen: Halten Sie die Katalogausgabe durch Drücken von [R/S] an und warten Sie, bis die Anzeige zu blinken beginnt; drücken Sie nun die Taste [ENTER↑]. Wenn Sie wieder auf die Anzeige der Überschriften zurückschalten wollen, halten Sie die Katalogausgabe durch Drücken von [R/S] an und drücken Sie dann [ENTER↑]. (Seite 394.)

Katalog 3: Standardfunktionen. Eine alphabetische Auflistung der Standardfunktionen des HP-41. Die Funktionen werden unter ihrem Alpha-Namen aufgeführt, der nicht unbedingt mit der Tastenfeldbezeichnung der Funktion übereinstimmt. Der Alpha-Name wird benötigt, um eine Funktion dem User-Tastenfeld zuzuordnen; in Programmzeilen erscheinen Funktionen unter ihrem Alpha-Namen.

Katalog 4: Verzeichnis des erweiterten Speichers. Eine Auflistung aller im erweiterten Speicher abgelegten Files. Es werden Name, Filetyp und Registeranzahl angezeigt. Nach Auflistung aller Files zeigt der Computer die Anzahl der für einen neuen File verbleibenden Register an. Die programmierbare Version von Katalog 4 ist die Funktion [EMDIR].

Sie können den Katalog 4 dazu benutzen, einen File zum momentanen File zu machen. Halten Sie die Katalogausgabe durch Drücken von [R/S] an, wenn der gewünschte File angezeigt wird; drücken Sie dann [←], um den Katalog-Modus zu verlassen. (Seite 206.)

Katalog 5: Alarmkatalog. Eine Auflistung aller gespeicherten Alarme. Für jeden Alarm werden Zeit, Datum und Meldung angezeigt. Über das Alarmkatalog-Tastenfeld können Sie Alarme löschen, wiederholende Alarme nachstellen und den angezeigten Alarm in Einzelheiten überprüfen. Die programmierbare Version von Katalog 5 ist die Funktion [ALMCAT]. (Seite 255.)

Katalog 6: Zuordnungen des User-Tastenfelds. Eine Auflistung aller dem User-Tastenfeld zugeordneten Funktionen und globalen Labels. Für jede Zuordnung wird der Funktionsname bzw. das globale Label und der Tastencode der belegten Taste angezeigt. Die Tastencodes reichen von 11 ([Σ+]) bis −84 ([VIEW]).

Sie können den Katalog 6 dazu verwenden, Tastenzuordnungen zu löschen. Halten Sie die Katalogausgabe durch Drücken von [R/S] an, wenn die gewünschte Zuordnung angezeigt wird und führen Sie dann die Tastenfolge ■ [C] aus. (Seite 168.)

Fehlermeldungen

Eine unzulässige Operation wird nie ausgeführt. Wenn die unzulässige Operation in einem Programm auftritt, unterbricht der Computer die Programmausführung und zeigt eine Fehlermeldung an.*

- Drücken Sie [←], um eine Fehlermeldung aus der Anzeige zu löschen.
- Wenn Sie eine andere Funktion ausführen wollen, drücken Sie einfach die entsprechende Taste. Es ist nicht nötig, die Fehlermeldung vorher zu löschen.
- Schalten Sie durch Drücken von [PRGM] in den Programm-Modus, um zu untersuchen, welche Anweisung den Fehler verursacht hat. Es wird die Programmzeile angezeigt, die die unerlaubte Operation enthält (wenn durch ein fehlendes Einsteck-Modul ein **NONEXISTENT** Fehler verursacht wurde, erscheint eine **XROM**-Nummer in der Anzeige).

In Anhang A finden Sie eine Liste aller Fehler- und Statusmeldungen. Viele über die Einschubschächte des Computers betriebene Zubehörgeräte verfügen über eigene Meldungen. Schlagen Sie bitte in der Benutzerdokumentation der Zubehörgeräte nach, wenn diese Meldungen in der Anzeige des Computers erscheinen.

* Durch die Flags 24 und 25 kann verhindert werden, daß die Programmausführung durch bestimmte Fehler unterbrochen wird. Diese Flags werden auf Seite 290 beschrieben.

Abschnitt 10

Der automatische Speicherstack

Inhalt

Einführung 174
Berechnungen mit UPN 175
Stack Lift und Stack Drop 175
Benutzen der Taste [ENTER↑] 175
Sperren und Freigeben des Stack Lift 176
Reihenfolge von Eingaben 176
Auffüllen des Stacks 177
Das LAST X Register 179
Fehlerkorrektur 179
Rechnen mit Konstanten 180
Weitere Stackoperationen 180
Austausch von Stackinhalten 181
Rollen des Stacks 181
Speichern und Zurückrufen 182
Registerarithmetik 182
Löschen des Stacks 183

Einführung

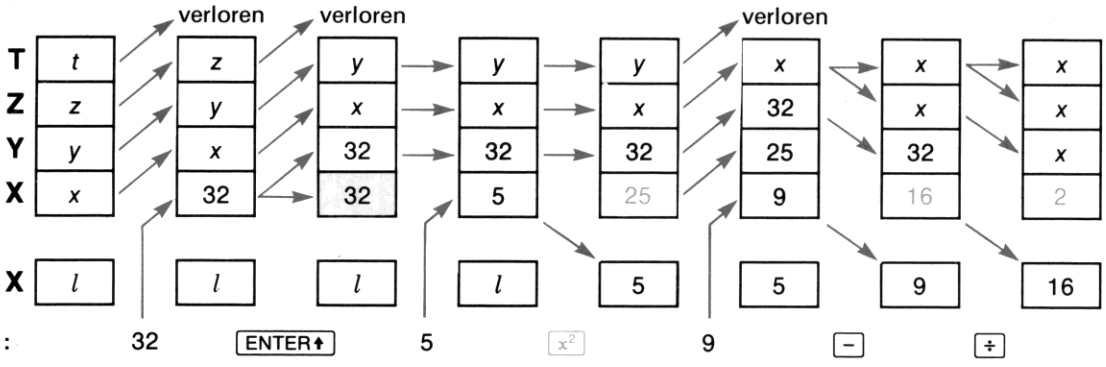
Die numerischen Funktionen verwenden vier Rechenregister, die den sogenannten *automatischen Speicherstack* bilden. Der Speicherstack läßt sich anschaulich als ein Stapel von vier übereinanderliegenden Rechenregistern vorstellen. Wenn Sie Zahlen eingeben und Berechnungen ausführen, werden im Stack automatisch Zahlen nach «oben und unten» verschoben. Diese Operationen basieren auf der Umgekehrten Polnischen Notation (UPN). Durch dieses Logik-System wird die Anzahl der benötigten Tastenfolgen auf ein Mindestmaß reduziert und es werden alle Zwischenergebnisse angezeigt. Schlagen Sie bitte auf Seite 20 nach, wenn Sie nicht mit UPN vertraut sind.

- Im ersten Teil dieses Abschnitts («Berechnungen mit UPN») werden die Grundlagen der Stackoperation erläutert und es wird ein Zahlenbeispiel behandelt. Darüberhinaus wird eine Methode zum Rechnen mit Konstanten vorgestellt, die darauf basiert, daß eine Konstante in allen vier Stack-Registern abgelegt wird.
- Im zweiten Teil («Das LAST X Register») wird ein besonderes Register behandelt, das eng mit den Stack-Registern zusammenhängt. Das LAST X Register dient zur Fehlerkorrektur und bietet eine weitere Möglichkeit zum Rechnen mit Konstanten.
- Im dritten Teil werden weitere Stackoperationen vorgestellt, die Ihnen helfen sollen, den Stack vielseitig einzusetzen. Auch in diesem Teil wird wieder auf den mehrmaligen Gebrauch einer Konstanten eingegangen.

Berechnungen mit UPN

Die untenstehende Abbildung zeigt den Inhalt des automatischen Speicherstacks und des LAST X Registers nach jedem Schritt einer Berechnung. x, y, z, t und l stehen für die Zahlen, die vor Ausführen der Berechnung in den Registern abgelegt sind. Als Beispiel wird der folgende Ausdruck berechnet:

$$\frac{32}{5^2-9}$$



Anhand dieses Beispiels werden die Funktionsweise und die vielfältigen Anwendungsmöglichkeiten des Stacks erläutert.

Stack Lift und Stack Drop

Die automatische Verschiebeoperation des Stackinhalts wird als Stack Lift (Verschieben nach oben) und Stack Drop (Verschieben nach unten) bezeichnet.

Stack Lift. Wenn Sie eine Zahl in das X-Register eingeben, wird der Stack im allgemeinen angehoben. Die im Y- bzw. Z-Register abgelegten Zahlen werden in das Z- bzw. T-Register gehoben und die Zahl im T-Register geht verloren. In dem oben behandelten Beispiel wird der Stack angehoben, wenn die Zahl 32 eingegeben wird, wenn 32 durch Drücken der Taste ENTER↑ in das Y-Register kopiert wird und wenn die Zahl 9 eingegeben wird.

Stack Drop. Der Stack wird im allgemeinen nach unten verschoben, wenn die Zahlen im X- und Y-Register durch eine Funktion zusammengefaßt werden. Die Zahlen im Z- bzw. T-Register werden in das Y- bzw. Z-Register verschoben und die Zahl im LAST X Register geht verloren. Im Beispiel tritt ein Stack Drop auf, wenn die Funktionen − und ÷ ausgeführt werden.

Benutzen der Taste ENTER↑

Durch Drücken der Taste ENTER↑ werden zwei hintereinander eingegebene Zahlen voneinander getrennt (im Beispiel 32 und 5). Dadurch wird die im X-Register abgelegte Zahl (32) in das Y-Register kopiert. Die im X-Register verbleibende Kopie wird durch die als nächstes eingegebene Zahl (5) überschrieben, da ENTER↑ den Stack Lift *sperrt*.

Sperren und Freigeben des Stack Lift

Der Stack Lift wird durch fast alle Funktionen freigegeben. Der Stack wird *nach* der Ausführung einer den Stack Lift freigebenden Funktion angehoben, sobald Sie eine Zahl in das X-Register eingeben. Der Stack Lift wird durch vier Funktionen gesperrt; einige Funktionen verhalten sich neutral.

Den Stack Lift sperrende Funktionen. Die Funktionen [ENTER↑], [CLx], [Σ+] und [Σ−] sperren den Stack Lift. Wenn Sie eine dieser Funktionen ausführen und anschließend eine Zahl eingeben, *ersetzt* diese Zahl den vorhergehenden Inhalt des X-Registers; die Inhalte der Y-, Z- und T-Register bleiben unverändert. Wenn in Stack-Diagrammen das X-Register schattiert dargestellt ist, so bedeutet dies, daß der Stack Lift gesperrt ist und daß die nächste Eingabe den Registerinhalt ersetzt.

Neutrale Funktionen. Die im folgenden aufgeführten Funktionen bewirken weder eine Freigabe noch ein Sperren des Stack Lift; der vorhergehende Zustand bleibt aufrechterhalten.

- Die Modustasten ([ON], [USER], [PRGM] und [ALPHA]).
- Die Löschtaste ([←]) während der Eingabe von Zahlen bzw. Zeichen.
- Die Umschalttaste ([■]).
- Die Kataloge 1, 2, 3 und 6.

Reihenfolge von Eingaben

Die Reihenfolge der Eingabe von Operanden wird durch die beiden im folgenden erläuterten Regeln bestimmt. Beide Methoden reduzieren die Anzahl der benötigten Tastenfolgen; in einigen Fällen müssen Sie sich auf eine Methode festlegen.

Klammerausdrücke. Berechnen Sie bei Klammerausdrücken zuerst den in der innersten Klammerebene stehenden Ausdruck und benutzen Sie dieses Ergebnis dann in einer vereinfachten Darstellung. Wenn zwei Klammerausdrücke berechnet und zusammengefaßt werden müssen, wird das erste Ergebnis durch den automatischen Speicherstack erhalten, während Sie den zweiten Ausdruck berechnen. Das unter «Polynome» behandelte Beispiel veranschaulicht diese Arbeitsweise.

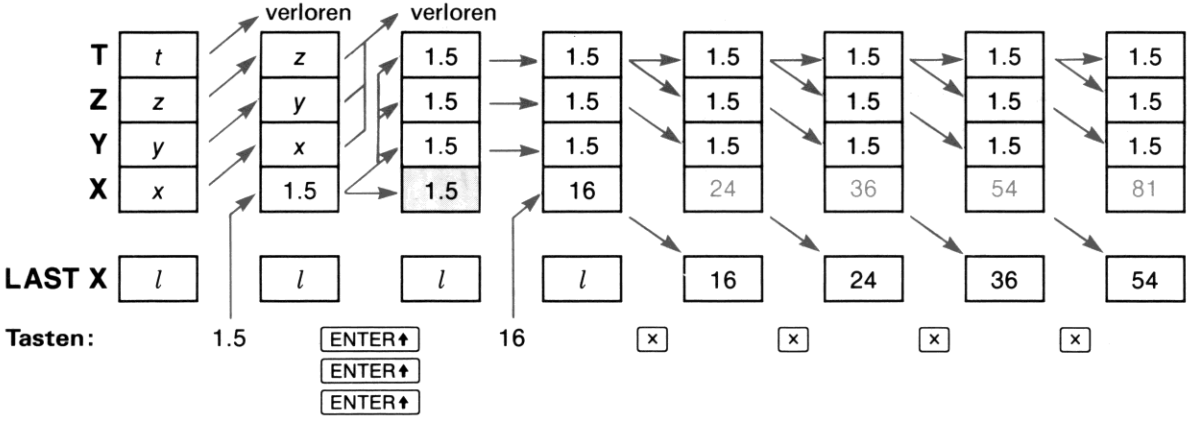
Nicht kommutative Funktionen. Subtraktion und Division werden als *nicht kommutative* Funktionen bezeichnet, da die Reihenfolge der Operanden nicht vertauschbar ist: $5-3 \neq 3-5$ und $5:3 \neq 3:5$. Bei Berechnungen mit nicht kommutativen Funktionen muß zuerst die im Y-Register abzulegende Zahl eingegeben oder berechnet werden. Erst dann kann die im X-Register abzulegende Zahl eingegeben oder berechnet werden. Diese Regel wird im vorhergehenden Beispiel zweimal angewandt.

- Der Zähler (32) wird eingegeben, bevor der Nenner (5^2-9) berechnet wird.
- Es wird zuerst der Ausdruck 5^2 berechnet und dann 9 von dem Ergebnis abgezogen.

Auffüllen des Stacks

Das vorhergehende Beispiel zeigt, wie x aus dem T-Register in die Y- und Z-Register kopiert wird. Diese Eigenschaft des Stack Drop ermöglicht es, eine Konstante im Y-Register zu erhalten (die beiden nächsten Beispiele erläutern dies). Die Anwendung dieser Eigenschaft ist besonders hilfreich, wenn die Konstante bei Ausführung einer nicht kommutativen Funktion (wie z.B. [−] und [÷]) im Y-Register vorhanden sein muß. (Die Funktion [LASTx] legt im Gegensatz dazu die Konstante im X-Register ab.)

Berechnungen mit Wachstumsraten. Berechnen Sie das Wachstum einer Größe mit dem Anfangswert 16, die in einem bestimmten Zeitraum um 50% wächst. Laden Sie zuerst den Stack mit der Wachstumsrate (1.5) und geben Sie dann den Anfangswert (16) in das X-Register ein. Drücken Sie [×], um den Wert nach dem ersten Zeitraum zu berechnen; jedes weitere Drücken von [×] berechnet den Wert für die darauffolgenden Zeiträume.



Polynome. Das Auffüllen des Stacks erleichtert die Berechnung von Polynomen, da die gleiche Variable mehrmals verwendet wird. Benutzen Sie das Horner-Schema zur Umordnung des Polynoms in eine ineinandergeschachtelte Form, die Exponenten größer als eins eliminiert. Berechnen Sie

$$x^4 - 2x^3 + 5x$$

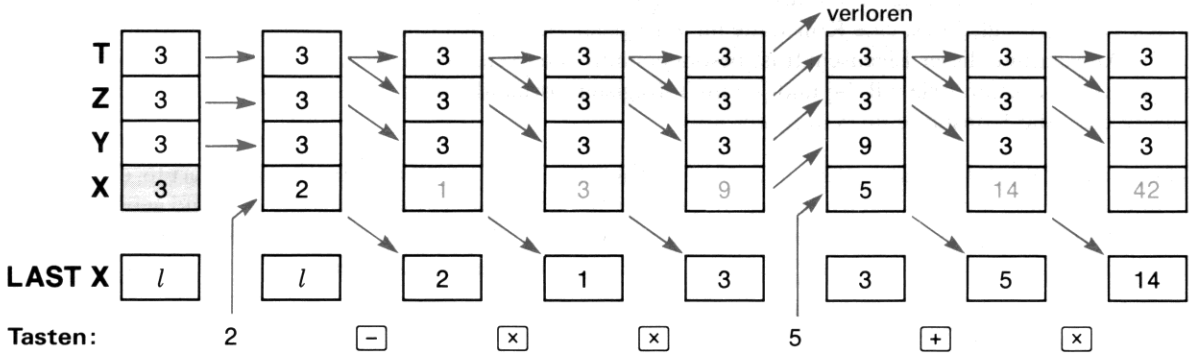
für $x = 3$. Ordnen Sie das Polynom zunächst um und eliminieren Sie die Exponenten.

$$x^4 - 2x^3 + 5x = (x^3 - 2x^2 + 5)x$$

$$= ((x^2 - 2x)x + 5)x$$

$$= (((x - 2)x)x + 5)x$$

Laden Sie den Stack mit der Variablen (Tastenfolge [3] [ENTER↑] [ENTER↑] [ENTER↑]) und führen Sie die unten angegebenen Schritte aus. Beachten Sie, daß die Berechnung bei der innersten Klammerebene beginnt.



Wenn Sie etwas näher mit dem Horner-Schema vertraut sind, können Sie ein Polynom berechnen, ohne es vorher umzuordnen. Betrachten Sie zum Beispiel das Polynom

$$ax^5 + bx^4 + cx^3 + dx^2 + ex + f$$

Die Rechenschritte nach Auffüllen des Stacks mit der Variablen sind:

$$a, [\times], b, [+], [\times], c, [+], [\times], d, [+], [\times], e, [+], [\times], f, [+].$$

- Beachten Sie, daß nach jedem der Koeffizienten (außer dem ersten und dem letzten) die Tasten [+] und [×] gedrückt werden. (Es wird kein Zwischenergebnis zu dem ersten Koeffizienten addiert und der letzte Koeffizient wird nicht mit einer Potenz der Variablen multipliziert.)
- Beginnen Sie mit dem zweiten Koeffizienten, wenn der erste Koeffizient gleich 1 ist. (Die Variable ist schon im X-Register abgelegt.)
- Wenn ein Koeffizient negativ ist, können Sie einen positiven Wert eingeben. Sie müssen dann nach dem Koeffizienten die Taste [−] statt [+] drücken.
- Drücken Sie nur die Taste [×], wenn ein Term keine Potenz von x enthält. (Dadurch wird die Potenz 0 spezifiziert.)

Nichtkumulative Ergebnisse. Sie können auch voneinander unabhängige (nichtkumulative) Berechnungen mit einer Konstanten ausführen. Drücken Sie nach jeder Berechnung die Taste [←], um das X-Register vor Eingabe des nächsten Operanden zu löschen. Dadurch wird der Stack Lift gesperrt und die Konstante im Y-Register wird nicht durch das Ergebnis der vorhergehenden Berechnung ersetzt.

Das LAST X Register

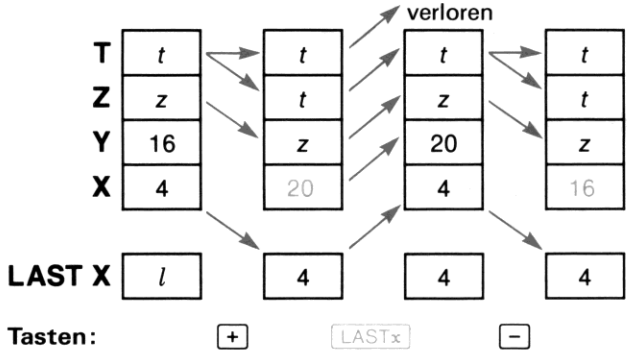
Das LAST X Register enthält den x-Operanden der zuletzt ausgeführten Funktion (außer [CHS]). Drücken Sie die Taste [LASTx], um diese Zahl in das X-Register zurückzurufen. Das LAST X Register ermöglicht es Ihnen, Fehler zu korrigieren und Operanden wiederzuverwenden.

Fehlerkorrektur

Fehler bei Funktionen einer Variablen. Wenn Sie eine falsche einwertige Funktion ausführen, können Sie folgendermaßen vorgehen, um den Fehler zu korrigieren.

1. Drücken Sie [←]. Dadurch wird das falsche Ergebnis durch Null ersetzt und der Stack Lift gesperrt.
2. Drücken Sie [LASTx]. Dadurch wird der Operand in das X-Register zurückgerufen, ohne daß der Inhalt der Y-, Z- und T-Register verändert wird.
3. Führen Sie die korrekte Funktion aus.

Fehler bei Funktionen zweier Variablen. Ein Fehler, der Ihnen bei Ausführen einer zweiwertigen Funktion (z.B. [+] oder [÷]) unterläuft, kann mit Hilfe von [LASTx] und der entsprechenden Umkehrfunktion ([−] oder [×]) korrigiert werden. Die untenstehende Abbildung zeigt, wie ein beim Addieren zweier Zahlen gemachter Fehler korrigiert wird. Drücken Sie zuerst [LASTx] und dann [−].



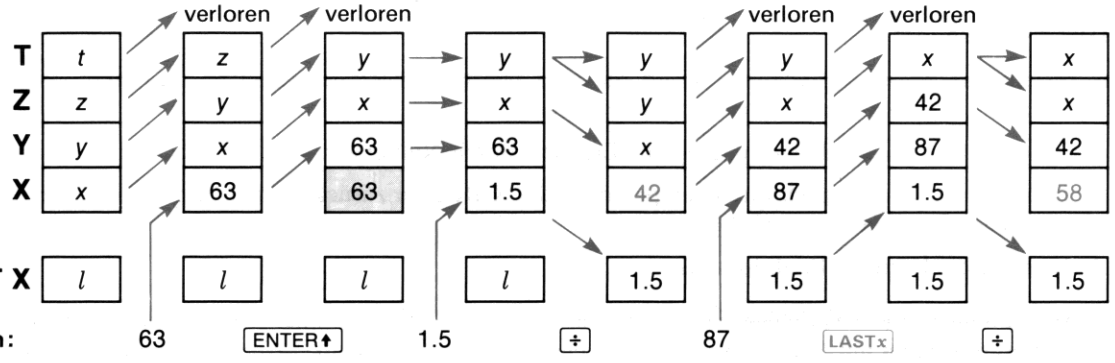
Wie Sie nach der Fehlerkorrektur fortfahren, hängt von der Art des gemachten Fehlers ab:

- Führen Sie [LASTx] nochmals aus, wenn Sie ursprünglich multiplizieren wollten. Dadurch wird die anfängliche Konfiguration des Stacks wiederhergestellt und die Multiplikation kann ausgeführt werden.
- Wenn Sie eine andere Zahl als 16 addieren wollten, können Sie folgendermaßen vorgehen: Drücken Sie [←], um 16 zu löschen, geben Sie die gewünschte Zahl ein, rufen Sie 4 aus dem LAST X Register zurück und führen Sie dann die Addition aus.
- Wenn Sie eine andere Zahl als 4 addieren wollten, geben Sie die gewünschte Zahl ein und führen die Addition aus.

Bei anderen zweiwertigen Funktionen auftretende Fehler sind noch einfacher zu korrigieren. Die Funktion [P→R] kann zum Beispiel durch Ausführen der Funktion [R→P] rückgängig gemacht werden. Die Fehlerkorrektur bei den Funktionen [%] und [%CH] erfolgt wie bei Funktionen einer Variabler. Sie können jeden Fehler korrigieren, indem Sie feststellen, wie sich die ausgeführte Funktion auf den Stack auswirkt; kehren Sie diesen Prozeß dann um.

Rechnen mit Konstanten

Das folgende Beispiel illustriert die mehrfache Verwendung einer Konstanten. Dividieren Sie sowohl 63 als auch 87 durch 1.5. Geben Sie die Konstante (1.5) als zweite Zahl (nach 63) in das X-Register ein; dadurch wird der Stack auf die erste Division vorbereitet und die Konstante automatisch im LAST X Register abgelegt.



Diese Technik ist besonders vorteilhaft, wenn sich die Konstante zur Ausführung von nicht kommutativen Operationen (z.B. [−] und [÷]) im X-Register befinden muß. (Beim Verwenden des Stack Drop zum Rechnen mit Konstanten wird die Konstante im Y-Register abgelegt.)

Weitere Stackoperationen

Die vier Stack-Register können als Registerpaare betrachtet werden. Die meisten Berechnungen werden im X- und Y-Register ausgeführt; die Z- und T-Register verhalten sich wie Speicherregister, die über den Stack Lift und den Stack Drop mit den «aktiveren» X- und Y-Registern verbunden sind. Wenn Sie eine zusätzliche Kopie einer Zahl im X-Register benötigen oder wenn Sie eine Zahl aus dem LAST X Register abrufen (und den alten X-Registerinhalt erhalten wollen), können Sie den ursprünglichen Wert im X-Register kurzzeitig in den höheren Stack-Registern zwischenspeichern und später wiederverwenden.

Sie sollten eine Kettenrechnung sorgfältig überdenken, um die Z- und T-Register optimal einsetzen zu können. Überlegen Sie, wie die Operanden bei jedem Rechenschritt angeordnet sein müssen, arbeiten Sie die Aufgabe von der letzten Berechnung her durch und verwenden Sie die in diesem Abschnitt vorgestellten Operationen, um ein Zwischenergebnis mit einer neu eingegebenen Zahl zu verbinden. Wenn Sie den Stack effektiv einsetzen, können Sie Programmspeicherplatz sparen und die Anzahl der benötigten Speicherregister auf ein Mindestmaß reduzieren.

Austausch von Stackinhalten

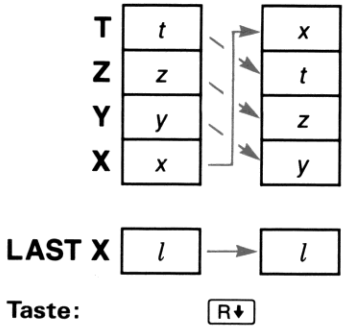
Vertauschen von X- und Y-Register. Wenn Sie die Funktion [x≷y] (*X exchange Y*) ausführen, werden die Inhalte der X- und Y-Register miteinander vertauscht. Diese Funktion bietet verschiedene Einsatzmöglichkeiten:

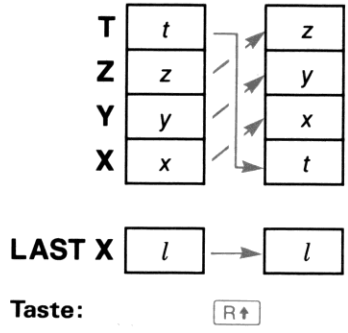
- Anzeige des Inhalts des Y-Registers. Drücken Sie [x≷y], prüfen Sie den angezeigten Wert und drücken Sie dann nochmals [x≷y], um die ursprüngliche Konfiguration des Stacks wiederherzustellen. Diese Technik ist hilfreich, wenn eine Funktion je ein Ergebnis im X- und Y-Register ablegt (statistische Funktionen, Umwandlung zwischen rechtwinkligen und Polarkoordinaten).
- Vertauschen von Zahlen, die sich vor Ausführen einer nichtkommutativen Operation (z.B. Subtraktion und Division) in der falschen Reihenfolge befinden.
- Umordnen von Stackinhalten zusammen mit den Funktionen [R↓] und [R↑]; siehe «Rollen des Stacks» (siehe unten).

Vertauschen des X-Registers mit einem anderen Stack-Register. Um den Inhalt des X-Registers mit dem Inhalt eines anderen Stack-Registers oder dem LAST X Register zu vertauschen, können Sie folgendermaßen vorgehen: Führen Sie die Funktion [X<>] aus, drücken Sie die Taste [·] und spezifizieren Sie dann das gewünschte Register durch [Y], [Z], [T] oder [L]. Ein Beispiel zu dieser Technik finden Sie unter «Registerarithmetik» (Seite 183).

Rollen des Stacks

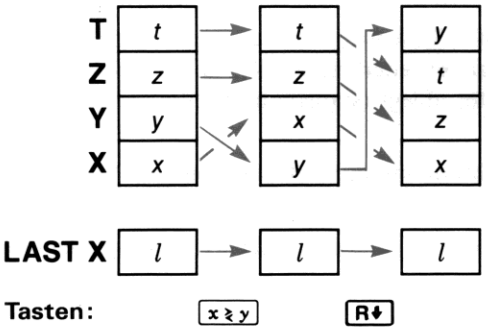
Die Funktionen [R↓] (*roll down*) und [R↑] (*roll up*) verschieben den Inhalt der Stack-Register zyklisch um ein Register nach unten bzw. oben. Dabei gehen keine Werte verloren und es werden keine Kopien von Registerinhalten erzeugt.

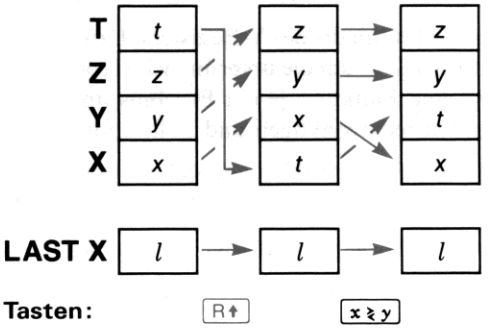




Beachten Sie, daß das LAST X Register von diesen Operationen nicht betroffen wird. Führen Sie [R↓] oder [R↑] viermal hintereinander aus, um den Inhalt aller Stack-Register abzurufen. Jede Zahl wird angezeigt, sobald sie in das X-Register geschoben wird. Wenn die Rolloperation viermal hintereinander ausgeführt wird, ist die ursprüngliche Konfiguration des Stacks wiederhergestellt.

Benutzen Sie [R↓] und [R↑] in Verbindung mit [x≷y], um Inhalte von Stack-Registern untereinander (d.h. nicht mit dem X-Register) zu vertauschen. Mit diesen Funktionen können Sie die Inhalte der Stack-Register in jeder beliebigen Reihenfolge anordnen. Zwei einfache Beispiele:





Speichern und Zurückrufen

Sie können eine in einem Stack-Register abgelegte Zahl kopieren, indem Sie die Funktion [STO] bzw. [RCL] ausführen und ein Stack-Register spezifizieren. Beide Funktionen führen zu dem gleichen Ergebnis: nach Funktionsausführung enthalten das X-Register und das spezifizierte Register die gleiche Zahl.

Speichern. Um die Zahl im X-Register in ein Stack-Register oder in das LAST X Register zu kopieren, können Sie folgendermaßen vorgehen: Führen Sie die Tastenfolge [STO] [·] aus und spezifizieren Sie dann das gewünschte Register durch [Y], [Z], [T] oder [L]. Der Inhalt des spezifizierten Registers geht bei dieser Operation verloren.

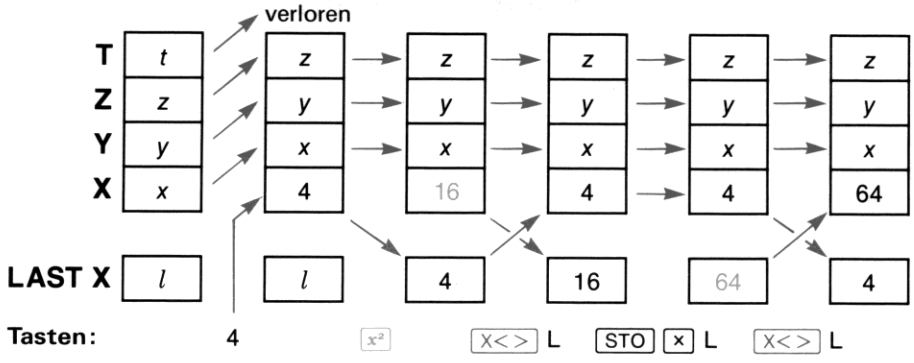
Zurückrufen. Um die in einem Stack-Register bzw. im LAST X Register abgelegte Zahl in das X-Register zu kopieren, können Sie folgendermaßen vorgehen: Führen Sie die Tastenfolge [RCL] [·] aus und spezifizieren Sie dann das gewünschte Register durch [Y], [Z], [T] oder [L]. Die Zahl im T-Register geht verloren, wenn der Stack angehoben wird (außer, der Stack Lift wurde gesperrt).

Registerarithmetik

Die Zahl im X-Register kann mit jeder im Stack abgelegten Zahl durch eine arithmetische Operation verknüpft werden. Führen Sie eine der Tastenfolgen [STO] [+] [·], [STO] [−] [·], [STO] [×] [·] oder [STO] [÷] [·] aus und spezifizieren Sie dann das gewünschte Stack-Register durch [X], [Y], [Z], [T] oder [L]. Denken Sie daran, daß die Reihenfolge der Operanden bei Subtraktion und Division eine entscheidende Rolle spielt; der Operand im spezifizierten Register entspricht der Zahl im Y-Register bei gewöhnlichen Stackberechnungen. Die Registerarithmetik mit Stackregistern unterscheidet sich von gewöhnlichen Stackberechnungen in einigen Punkten.

- Das Ergebnis wird in dem spezifizierten Register abgelegt.
- Die Zahl im X-Register wird nicht verändert (außer wenn Sie dieses Register als Parameter spezifizieren).
- Der Inhalt des LAST X Registers wird nicht verändert (außer wenn Sie dieses Register als Parameter spezifizieren).
- Ein Stack Drop findet nicht statt.

Im folgenden Beispiel wird die Zahl im X-Register in die dritte Potenz erhoben und das Ergebnis im LAST X Register abgelegt. Beachten Sie, daß die übrigen Stack-Register von dieser Operation nicht betroffen werden.



Löschen des Stacks

Wenn Sie die Funktion CLST (*clear stack*) ausführen, werden die Inhalte der X-, Y-, Z- und T-Register durch Null ersetzt. Der Inhalt des LAST X Registers bleibt unverändert.

Abschnitt 11

Numerische Funktionen

Inhalt

Einführung 184
Funktionen einer Variablen 185
Allgemeine Funktionen 185
Funktionen zur Zahlenmanipulation 186
Trigonometrische Operationen 186
Umwandlungen 187
Logarithmen und Exponentialfunktionen 187
Funktionen zweier Variablen 187
Arithmetische Operationen 188
Rechnen mit Zeitwerten 188
Prozentrechnung 188
Umwandlung zwischen kartesischen und Polarkoordinaten 189
Sonstige Funktionen zweier Variablen 189
Statistische Funktionen 190
Die Statistikregister 190
Dateneingabe 191
Mittelwert 192
Standardabweichung 192

Einführung

Dieser Abschnitt behandelt die numerischen Funktionen des HP-41CX. Alle ein- bzw. zweiwertigen Funktionen operieren auf dem Stack; die Wirkungsweise der Funktionen wird durch Stackdiagramme veranschaulicht. Datenpunkte der statistischen Funktionen werden zwar über den Stack eingegeben, die Speicherung erfolgt jedoch in den Statistikregistern des Hauptspeichers. Die Ergebnisse der Statistikfunktionen werden wieder im Stack abgelegt. Die Ergebnisse einiger anderer Funktionen werden nicht notwendigerweise in den Stack zurückgegeben (z.B. Registerarithmetik und [ISG]); auf diese Funktionen wird in diesem Abschnitt nicht eingegangen.

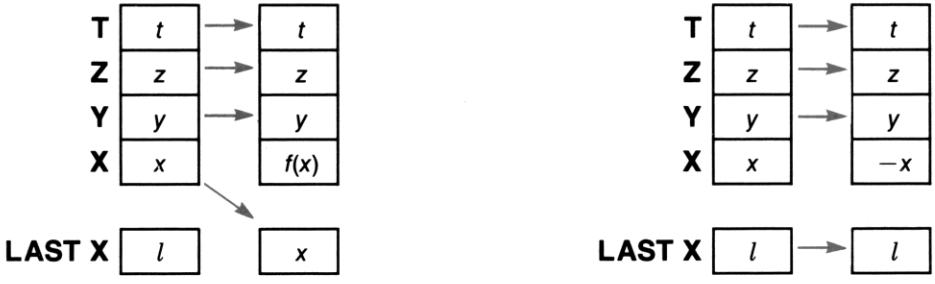
Bei Ausführung von numerischen Funktionen können drei Fehlerbedingungen auftreten:

1. Die Meldung **DATA ERROR** erscheint, wenn Sie versuchen, eine unerlaubte Operation auszuführen (z.B. Division durch 0).
2. Wenn Sie versuchen, eine Berechnung mit einem Alpha-String auszuführen, erscheint die Fehlermeldung **ALPHA DATA** in der Anzeige. Ziffern des Alpha-Tastenfelds werden vom Computer als Zeichen angesehen, es können keine Berechnungen damit ausgeführt werden.
3. Eine Berechnung, deren Ergebnis betragsmäßig größer als $9.999999999 \times 10^{99}$ wäre, hat einen **OUT OF RANGE** Fehler zur Folge. (Ausnahmen hiervon sind die statistischen Summationen [Σ+] und [Σ−].)

Eine Funktion, die einen Fehler zur Folge hat, wird vom HP-41 nicht ausgeführt. Wenn Flag 25 nicht gesetzt ist, unterbrechen die Fehler **DATA ERROR** und **ALPHA DATA** die Programmausführung (wenn der Fehler in einem Programm auftritt) und zeigen die Fehlermeldung an. Wenn weder Flag 24 noch Flag 25 gesetzt ist, unterbricht auch ein **OUT OF RANGE** Fehler die Ausführung und zeigt die Fehlermeldung an.

Funktionen einer Variablen

Bei Ausführung einer einwertigen Funktion wird der im X-Register abgelegte Operand durch das Ergebnis ersetzt; eine Kopie des Operanden bleibt im LAST X Register erhalten. Funktionen einer Variablen haben keinen Einfluß auf den Inhalt der Y-, Z- und T-Register. In der linken Abbildung des Stacks wird das Ergebnis durch $f(x)$ repräsentiert. Das rechte Stackdiagramm zeigt die Funktion [CHS] (*change sign*), die als einzige keine Kopie des Operanden im LAST X Register ablegt.



Funktionen einer Variablen

CHS

Allgemeine Funktionen

Kehrwert ([1/x]).

Quadrat ([x²]) und Quadratwurzel ([√x]).

Fakultät ([FACT]). Diese Funktion berechnet $n! = n(n-1)(n-2)\ldots 1$. Dabei ist n eine positive ganze Zahl.

Funktionen zur Zahlenmanipulation

Betragsfunktion (ABS) und Signumfunktion (SIGN). Die Funktion ABS berechnet $|x|$, den Betrag von x. Die Funktion SIGN liefert die folgenden Ergebnisse: 1 für $x \geq 0$, -1 für $x < 0$. Wenn das X-Register Alpha-Zeichen enthält, liefert SIGN den Wert 0.

Ganzzahliger Anteil (INT) und gebrochener Anteil (FRC). Diese Funktionen trennen den ganzzahligen bzw. gebrochenen Anteil einer Zahl ab. Wenn zum Beispiel im X-Register die Zahl 777.888 abgelegt ist, liefert INT das Ergebnis 777 und FRC gibt 0.888 als Ergebnis zurück.

Rundungsfunktion (RND). Die Funktionen zur Beeinflussung des Anzeigeformats wirken nur auf die Anzeige und haben keinerlei Einfluß auf die interne Zahlendarstellung. Um die interne Darstellung der Zahl im X-Register zu ändern, können Sie folgendermaßen vorgehen:

1. Wählen Sie über das Anzeigeformat die Anzahl der Dezimalstellen der zu rundenden Zahl.
2. Führen Sie die Funktion RND aus.

Beispiel: Wenn Sie eine Zahl auf die nächste ganze Zahl runden wollen, führen Sie zuerst FIX **0** und dann RND aus.

Trigonometrische Operationen

Winkelmodus. Der gewählte Winkelmodus legt fest, wie der Computer die eingegebenen Winkel interpretiert. Der von Ihnen eingestellte Modus bleibt durch den Permanentspeicher erhalten. Diese Funktionen dienen nur zur Wahl des Winkelmodus und haben keinerlei Einfluß auf Zahlen, die sich schon im Computer befinden.

- Führen Sie die Funktion RAD aus, um den *Bogenmaß*-Modus zu wählen. Der **RAD**-Indikator erscheint in der Anzeige. Operanden trigonometrischer Funktionen werden vom Computer als in Radiant spezifizierte Winkelgrößen interpretiert.
- Führen Sie die Funktion GRAD aus, um den *Neugrad*-Modus zu wählen. Der **GRAD**-Indikator erscheint in der Anzeige. Operanden trigonometrischer Funktionen werden als in Neugrad spezifizierte Winkelgrößen interpretiert.
- Führen Sie die Funktion DEG aus, um den *Altgrad*-Modus zu wählen. Weder der **RAD**- noch der **GRAD**-Indikator wird angezeigt. Operanden trigonometrischer Funktionen werden als in Altgrad spezifizierte Winkelgrößen interpretiert. Winkel werden in Dezimalform ausgedrückt, nicht in Stunden-Minuten-Sekunden.

Es gilt: 360 Altgrad = 2π Rad = 400 Neugrad.

Trigonometrische Funktionen. Der HP-41 verfügt über sechs trigonometrische Funktionen:

- SIN (*Sinus*) und SIN^{-1} (*Arcussinus*).
- COS (*Cosinus*) und COS^{-1} (*Arcuscosinus*).
- TAN (*Tangens*) und TAN^{-1} (*Arcustangens*).

Umwandlungen

Umwandlung zwischen Altgrad und Bogenmaß. Führen Sie die Funktion [D-R] (*degrees to radians*) aus, um die angezeigte Zahl von dezimalen Altgrad ins Bogenmaß umzuwandeln. Die Rückumwandlung (vom Bogenmaß in dezimale Altgrad) geschieht durch die Funktion [R-D] (*radians to degrees*).

Umwandlung zwischen Dezimalstunden und Stunden-Minuten-Sekunden. Stundenangaben und Winkelgrößen können sowohl im HMS- (*hours-minutes-seconds*) Format als auch im Dezimalstunden-Format dargestellt werden. Im HMS-Format werden die ersten beiden Stellen nach dem Dezimalpunkt als Minuten und die nächsten beiden Ziffern als Sekunden interpretiert; alle nachfolgenden Ziffern stehen für Bruchteile von Sekunden. Beispiel:

$$HH.MMSSssss = HH \text{ Stunden} + MM \text{ Minuten} + SS.ssss \text{ Sekunden (HMS-Format)}$$
$$= HH + MM/60 + SS.ss/3600 \text{ (Dezimaldarstellung)}$$

Führen Sie die Funktion [HMS] (*to hours-minutes-seconds*) aus, um eine in Dezimaldarstellung gegebene Zahl in das HMS-Format umzuwandeln. Die Rückumwandlung (vom HMS-Format in das Dezimalstunden-Format) geschieht mit der Funktion [HR] (*to decimal hours*).

Umwandlung zwischen Dezimal- und Oktaldarstellung. Führen Sie die Funktion [OCT] (*to octal*) aus, um eine ganze Dezimalzahl in die entsprechende Oktalzahl (Basis 8) umzuwandeln. Die Funktion [DEC] (*to decimal*) wandelt eine ganze Oktalzahl in die entsprechende Dezimalzahl (Basis 10) um.

Logarithmen und Exponentialfunktionen

Dekadischer Logarithmus und Exponentialfunktion zur Basis zehn. Führen Sie die Funktion [LOG] aus, um den dekadischen Logarithmus (Basis 10) der im X-Register abgelegten Zahl zu berechnen. Die Funktion [10^x] potenziert die Zahl 10 mit dem Inhalt des X-Registers.

Natürlicher Logarithmus und natürliche Exponentialfunktion. Drücken Sie [LN], um den natürlichen Logarithmus (Logarithmus zur Basis e) der im X-Register abgelegten Zahl zu berechnen. Die Funktion [e^x] potenziert e mit dem Inhalt des X-Registers.

Der HP-41 verfügt über spezielle Funktionen, um die Ausdrücke $\ln(1+x)$ und $e^x - 1$ für Argumente und Ergebnisse nahe Null zu berechnen (diese Ausdrücke treten in Zusammenhang mit den hyperbolischen Funktionen und deren Umkehrfunktionen sowie in bestimmten finanzmathematischen Berechnungen auf). Diese Ausdrücke werden durch die Funktionen [LN1+X] und [E↑X-1] direkt und mit großer Genauigkeit berechnet.

- [LN1+X] berechnet $\ln(1+x)$.
- [E↑X-1] berechnet $e^x - 1$.

Funktionen zweier Variablen

Die Operanden der zweiwertigen Funktionen werden in den X- und Y-Registern abgelegt; die meisten dieser Funktionen liefern eine einzelne Zahl als Ergebnis und lösen einen Stack Drop aus. (Ausnahmen: Prozentfunktionen und Umwandlungen zwischen kartesischen und Polarkoordinaten.)

Arithmetische Operationen

Stackdiagramme für die Operationen [+], [−], [×] und [÷] finden Sie in Abschnitt 10. Denken Sie daran, die Operanden bei Subtraktion und Division in der richtigen Reihenfolge einzugeben. Für die Zahlen x im X-Register und y im Y-Register gilt:

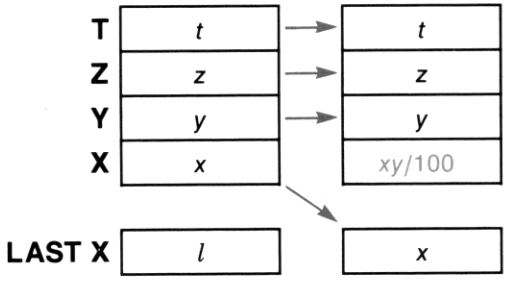
- Eine Subtraktion liefert $y-x$ (nicht $x-y$).
- Eine Division liefert y/x (nicht x/y).

Rechnen mit Zeitwerten

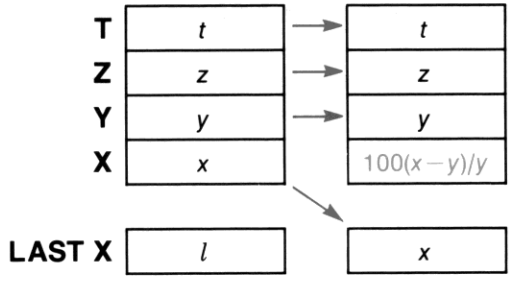
Die Funktionen [HMS+] (*hours-minutes-seconds add*) und [HMS−] (*hours-minutes-seconds subtract*) addieren bzw. subtrahieren im Stunden-Minuten-Sekunden-Format (HMS) gegebene Zahlen. Die Reihenfolge der Eingabe und der auftretende Stack Drop unterscheiden sich nicht von der gewöhnlichen Addition bzw. Subtraktion.

Prozentrechnung

Der HP-41 verfügt über zwei Prozentfunktionen. Beide Funktionen wirken auf die im X-Register abgelegte Zahl unter Verwendung der im Y-Register spezifizierten Basiszahl. Die Basiszahl bleibt unverändert und ein Stack Drop findet nicht statt.



Taste: [%]



Taste: [%CH]

Prozent ([%]). Zum Berechnen von Prozenten ist wie folgt vorzugehen: Legen Sie die Basiszahl im Y-Register ab, geben Sie den Prozentsatz in das X-Register ein und führen Sie die Funktion [%] aus.

Prozentuale Änderung ([%CH]). Die Funktion [%CH] berechnet die prozentuale Differenz zweier Zahlen. Ein positives Ergebnis bedeutet einen relativen Zuwachs und ein negatives Ergebnis bedeutet eine relative Abnahme. Gehen Sie folgendermaßen vor: Legen Sie die erste Zahl (die Basiszahl) im Y-Register ab, geben Sie die zweite Zahl in das X-Register ein und führen Sie die Funktion [%CH] aus. Das Ergebnis stellt die prozentuale Differenz zwischen der zweiten Zahl und der ersten Zahl (Basiszahl) dar.

Prozentualer Anteil. Gehen Sie wie folgt vor, um zu berechnen, welchen prozentualen Anteil eine Zahl an einer anderen Zahl (Basiszahl) hat.

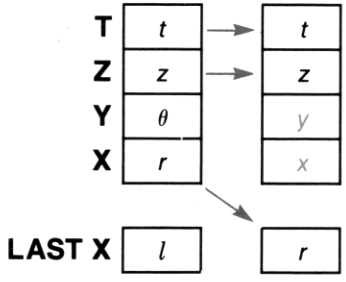
1. Legen Sie die Basiszahl im Y-Register ab und geben Sie die Zahl, deren prozentualer Anteil an der Basiszahl berechnet werden soll, in das X-Register ein.
2. Führen Sie [1/x] aus.
3. Führen Sie [%] aus.
4. Führen Sie [1/x] aus.

Umwandlung zwischen kartesischen und Polarkoordinaten

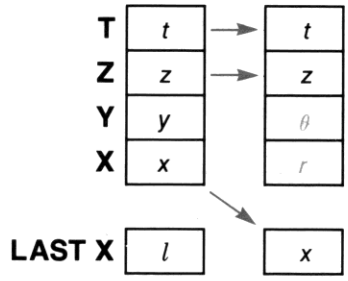
Ein Punkt der Ebene kann sowohl in kartesischen als auch in Polarkoordinaten dargestellt werden. In kartesischen Koordinaten wird ein Punkt P durch die Achsenabstände x (Abszisse) und y (Ordinate) dargestellt; die Parameter des Polarkoordinatensystems sind der Radius r und der Winkel θ. (Eine Illustration dieser beiden Koordinatensysteme finden Sie auf Seite 54.) Der HP-41 verfügt über zwei Funktionen ([P→R] und [R→P]) zur Umwandlung zwischen kartesischen und Polarkoordinaten.

- Die Funktion [P→R] (*polar to rectangular*) wandelt Polarkoordinaten in kartesische Koordinaten um.
- Die Funktion [R→P] (*rectangular to polar*) wandelt kartesische Koordinaten in Polarkoordinaten um. Der Ergebniswinkel θ hat das gleiche Vorzeichen wie die eingegebene y-Koordinate.

Der Winkel θ wird in allen Fällen (Zahleneingabe und Ergebnisausgabe) im momentanen Winkel-Modus dargestellt. Entnehmen Sie die Reihenfolge der Koordinateneingabe bitte der unten abgebildeten Illustration. Beachten Sie, daß kein Stack Drop auftritt. Drücken Sie die Taste [x≷y], um das im Y-Register abgelegte Ergebnis anzuzeigen.



Taste: [P→R]



Taste: [R→P]

Sonstige Funktionen zweier Variablen

Potenzfunktion ([y^x]). Um eine Zahl zu potenzieren, können Sie folgendermaßen vorgehen: Geben Sie die Basiszahl in das Y-Register und den Exponenten in das X-Register ein und führen Sie die Funktion [y^x] aus. Die folgenden Kombinationen von x- und y-Werten sind erlaubt.

- Wenn y positiv ist, darf x jede beliebige Zahl sein.
- Wenn y negativ ist, muß x ganzzahlig sein.
- Wenn y gleich Null ist, muß x positiv sein.

Jede andere Kombination hat den Fehler **DATA ERROR** zur Folge.

Wurzelberechnung. Um die n-te Wurzel einer Zahl zu berechnen, können Sie folgendermaßen vorgehen:

1. Geben Sie den Radikand in das Y-Register ein.
2. Geben Sie n in das X-Register ein.
3. Führen Sie die Funktion [1/x] aus.
4. Führen Sie die Funktion [y^x] aus.

Modulo-Funktion [MOD] (Rest bei Division). Für positive ganze Zahlen x (im X-Register) und y (im Y-Register) berechnet die Modulo-Funktion (y mod x) den bei der Division von y durch x entstehenden ganzzahligen Rest und zeigt diesen im X-Register an. Wenn y ohne Rest durch x teilbar ist, gibt der Computer Null als Ergebnis zurück. Der Stack wird wie bei den arithmetischen Funktionen nach unten verschoben (Stack Drop).

Die Modulo-Funktion kann auch auf nicht ganzzahlige Werte von x und y angewandt werden. Die allgemeine Formel lautet: $y \bmod x = y - \langle y/x \rangle$, wobei $\langle y/x \rangle$ die größte ganze Zahl darstellt, die kleiner oder gleich y/x ist. Wenn Sie [MOD] für den Wert $x = 0$ ausführen, wird als Ergebnis y zurückgegeben.

Statistische Funktionen

Die statistischen Funktionen werden in zwei Stufen ausgeführt. Zuerst geben Sie über den Stack die Datenpunkte ein; der Computer berechnet Zwischenergebnisse und speichert diese in den Statistikregistern. In der zweiten Stufe führen Sie die eigentlichen statistischen Berechnungen aus; der Computer berechnet aus den Zwischenergebnissen die gewünschten Statistiken und legt das oder die Ergebnisse im Stack ab. Die Grundlagen der statistischen Operationen werden in Band I auf den Seiten 55 bis 58 erläutert.

Die Statistikregister

In den Statistikregistern (ein Block von sechs Hauptspeicher-Registern) speichert der Computer verschiedene Summen und Produkte, die für statistische Berechnungen benötigt werden. In der Voreinstellung sind die Register R_{11} bis R_{16} als Statistikregister definiert.

- Sie können die Adressen der Statistikregister ändern, indem Sie die Funktion [ΣREG] ausführen. Der Computer erwartet als Parameter die Adresse des ersten Registers aus dem gewünschten Registerblock. Diese Zuordnung wird durch den Permanentspeicher erhalten.
- Führen Sie die Funktion [ΣREG?] aus, um zu überprüfen, welche Register momentan als Statistikregister definiert sind. Der Computer legt die Adresse des ersten Registers aus dem Block im X-Register ab.*
- Führen Sie die Funktion [CLΣ] aus, um alle Statistikregister zu löschen.

* Die Funktion [ΣREG?] überprüft nicht, ob die Statistikregister wirklich existieren (d.h. ob der Datenspeicherung genügend Speicherplatz zugewiesen ist); es wird nur die Adresse des ersten definierten Statistikregisters abgefragt.

Die aus Ihren Eingaben in die X- und Y-Register berechneten statistischen Summen und Produkte werden in den folgenden Registern gespeichert.

Die Statistikregister

Register	Inhalt	
R_{11}	Σx	Summe der x-Werte.
R_{12}	Σx^2	Summe der Quadrate der x-Werte.
R_{13}	Σy	Summe der y-Werte.
R_{14}	Σy^2	Summe der Quadrate der y-Werte.
R_{15}	Σxy	Summe der Produkte der x- und y-Werte.
R_{16}	Σn	Anzahl der akkumulierten Datenelemente (Datenpaare). n erscheint auch in der Anzeige.

Dateneingabe

Akkumulation von Daten. Wenn Sie die Taste [Σ+] drücken, führt der Computer die folgenden Operationen aus:

- Das Ergebnis der Berechnungen mit den in den X- und Y-Registern abgelegten Zahlen wird zu dem Inhalt der ersten fünf Statistikregister addiert. Wenn dadurch der Inhalt eines der Register den Wert $\pm 9.999999999 \times 10^{99}$ überschreitet, wird dies vom Computer nicht als OUT OF RANGE Fehler interpretiert. Das betreffende Register enthält dann die Zahl $\pm 9.999999999 \times 10^{99}$.
- Die Anzahl der Datenpunkte (im sechsten Statistikregister gespeichert) wird erhöht und der momentane Wert wird im X-Register abgelegt.
- Die Zahl, die sich im X-Register befand, wird durch das LAST X Register erhalten.
- Der Stack Lift wird gesperrt; die als nächstes eingegebene Zahl ersetzt den im X-Register abgelegten Wert n.

Sie können statistische Funktionen von einer oder zwei Variablen ausführen (dies wurde bereits in Band I erläutert). Löschen Sie das Y-Register durch die Tastenfolge 0 [ENTER↑], wenn Sie nur x-Werte aufsummieren. Da die Funktionen [Σ+] und [Σ−] den Stack Lift sperren, bleibt das Y-Register gelöscht, während Sie die x-Werte aufsummieren.

Fehlerkorrektur. Um fehlerhafte Daten zu korrigieren, können Sie wie folgt vorgehen:

1. Geben Sie das fehlerhafte Datenpaar ein. Wenn die fehlerhaften Daten zuletzt eingegeben wurden, können sie durch die Funktion [LASTx] zurückgerufen werden. (Der fehlerhafte y-Wert befindet sich noch im Y-Register und der fehlerhafte x-Wert wurde durch das LAST X Register erhalten.)
2. Führen Sie die Funktion [Σ−] aus. Diese Funktion wirkt ähnlich wie [Σ+], mit den folgenden Unterschieden: die Ergebnisse werden von dem Inhalt der ersten fünf Statistikregister subtrahiert (statt addiert) und der Inhalt des sechsten Registers wird zurückgesetzt (statt erhöht).
3. Geben Sie die korrekten Daten ein.
4. Führen Sie die Funktion [Σ+] aus.

Erlaubte Datenwerte. Datenwerte, die sich um einen relativ kleinen Betrag unterscheiden, machen dem Computer die Ausführung einiger statistischer Funktionen unmöglich. Dies können Sie vermeiden, indem Sie Ihre Werte normalisieren; d.h. Sie geben nicht die Werte selbst, sondern stattdessen die Differenz der einzelnen Werte von einem ungefähren Mittelwert der Datenfolge ein. Diese Zahl muß anschließend zum Ergebnis der jeweiligen Berechnung addiert werden. Wenn z.B. die x-Werte aus der Folge 66599, 66600 und 66601 bestehen, sollten Sie die Daten –1, 0 und 1 eingeben; anschließend müssen Sie 66600 zu den in Frage kommenden Statistiken addieren.

Mittelwert

Die Funktion [MEAN] legt das arithmetische Mittel der aufsummierten x-Werte ($\bar{x}$) im X-Register und das arithmetische Mittel der aufsummierten y-Werte ($\bar{y}$) im Y-Register ab. Die Berechnung erfolgt unter Verwendung der folgenden Formeln:

$$\bar{x} = \frac{\Sigma x}{n}, \qquad \bar{y} = \frac{\Sigma y}{n}.$$

Drücken Sie die Taste [$x \gtrless y$], um den Mittelwert der y-Werte anzuzeigen. Die Zahl, die sich vor Ausführung von [MEAN] im X-Register befand, wird durch das LAST X Register erhalten; die vor Funktionsausführung im Y-Register befindliche Zahl geht verloren.

Standardabweichung

Die Funktion [SDEV] legt die Standardabweichung der aufsummierten x-Werte (s_x) im X-Register und die Standardabweichung der aufsummierten y-Werte (s_y) im Y-Register ab. Die Berechnung erfolgt unter Verwendung der folgenden Formeln:

$$s_x = \sqrt{\frac{n\Sigma(x^2) - (\Sigma x)^2}{n(n-1)}}, \qquad s_y = \sqrt{\frac{n\Sigma(y^2) - (\Sigma y)^2}{n(n-1)}}.$$

Drücken Sie die Taste [$x \gtrless y$], um die Standardabweichung der y-Werte anzuzeigen. Die Zahl, die sich vor Ausführung der Funktion [SDEV] im X-Register befand, wird durch das LAST X Register erhalten; die vor Funktionsausführung im Y-Register befindliche Zahl geht verloren.

Teil III
Der Speicher im Detail

Abschnitt 12

Der Hauptspeicher

Inhalt

Aufteilung des Hauptspeichers 194
Der Programmspeicher 196
Programmzeilen 197
Null-Bytes 197
Packen des Speichers 198
Der Alarmspeicher 198
Der Speicher des User-Tastenfelds 198
Der Datenspeicher 199
Zuweisen von Datenspeichern 199
Registeradressen über R_{99} 199
Operationen mit Datenspeicher-Registern 199
Speichern und Zurückrufen 199
Registerarithmetik 201
Registeraustausch 201
Operationen mit Registerblöcken 201
Löschen von Registern 202

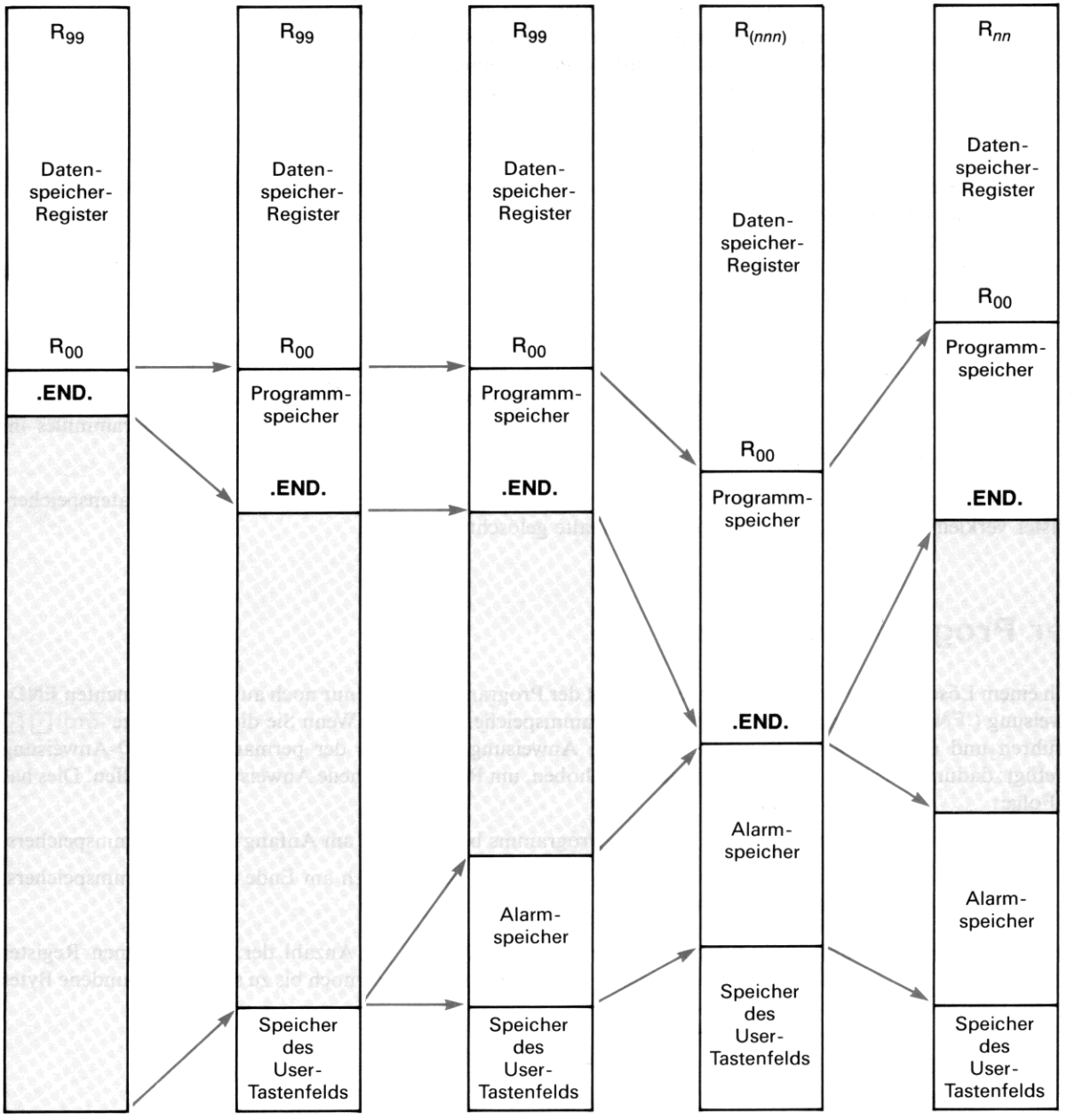
Aufteilung des Hauptspeichers

Die 319 Register des Hauptspeichers können in zwei Hauptgruppen eingeteilt werden.*

- Eine Gruppe besteht aus den Datenspeicher-Registern. Die Anzahl der der Datenspeicherung zugewiesenen Hauptspeicherregister kann über die Funktionen [SIZE] und [PSIZE] verändert werden.
- Die andere Gruppe besteht aus den Registern zur Speicherung von Programmen, Alarmen und Tastenzuordnungen sowie den ungebundenen Registern. Die ungebundenen Register werden nötigenfalls vom Computer automatisch zur Speicherung von Programmen, Alarmen und Tastenzuordnungen verwendet. Die Gesamtzahl der in dieser Gruppe enthaltenen Register ändert sich nur, wenn Sie die Anzahl der Datenspeicher-Register ändern.

* Der Hauptspeicher enthält 320 Register; der Programmspeicher besteht jedoch immer aus mindestens einem Register für die permanente END-Anweisung (.END.).

Die untenstehende Abbildung zeigt fünf verschiedene Konfigurationen des Hauptspeichers – jede Spalte stellt den gesamten Hauptspeicher dar. Die linke Spalte illustriert die Voreinstellung: der Datenspeicherung sind 100 Register zugeordnet; für alle anderen Zwecke bleiben 219 Register. Die Spalten rechts zeigen den Hauptspeicher zu vier späteren Zeitpunkten. Die meisten dieser Details werden vom Computer automatisch berücksichtigt; ein Verständnis der Hauptspeicheraufteilung erleichtert Ihnen jedoch den wirkungsvollen Einsatz Ihres HP-41.



Die erste Spalte zeigt den Einschaltzustand (d.h. die nach einem Löschen des Permanentspeichers vorhandene Aufteilung). Der Datenspeicherung sind 100 Register zugeordnet; das Register mit der höchsten Adresse befindet sich am oberen Ende. Das erste Register nach dem Block der Datenregister enthält die permanente **END**-Anweisung (.END.), die am Ende des Programmspeichers steht. Die ungebundenen Register unterhalb der permanenten **END**-Anweisung dienen zur Speicherung von Programmen, Alarmen und Tastenzuordnungen.

Die zweite Spalte zeigt den Hauptspeicher, nachdem Sie Programme eingegeben und Tastenzuordnungen vorgenommen haben. Die für den Programmspeicher benötigten Register werden von den ungebundenen Registern abgezogen; die permanente **END**-Anweisung wird weiter nach unten geschoben, wenn Sie weitere Programmzeilen eingeben. Je zwei Tastenzuordnungen des User-Tastenfelds belegen ein ungebundenes Register.

Die dritte Spalte zeigt den Hauptspeicher, nachdem Alarme eingestellt wurden.

Die vierte Spalte zeigt den Hauptspeicher, nachdem Sie der Datenspeicherung mehr Register zugewiesen haben, weitere Programme und Alarme gespeichert haben und zusätzliche Tastenzuordnungen vorgenommen haben. Wenn alle Register gebunden sind und eine Operation ausgeführt wird, die zusätzlichen Speicherplatz benötigt, verursacht dies einen Fehler. Wenn Sie versuchen, einen Alarm einzustellen, zeigt der Computer die Fehlermeldung **NO ROOM** an. Andere Operationen haben die Meldungen **PACKING** und anschließend **TRY AGAIN** zur Folge.

Wenn durch das Packen des Speichers nicht genügend ungebundene Register freigesetzt werden, müssen Sie den der Datenspeicherung zugeordneten Registerblock verkleinern oder andere Speicherinhalte löschen. Sie können die Inhalte des Programmspeichers, des Alarmspeichers und des Speichers des User-Tastenfelds über die Kataloge 1, 5 und 6 einsehen. Es besteht auch die Möglichkeit, Daten und Programme in Daten- bzw. Programmfiles im erweiterten Speicher abzulegen.

In der fünften Spalte sind wieder ungebundene Register vorhanden, nachdem Sie die Anzahl der Datenspeicher-Register verkleinert haben bzw. andere Speicherinhalte gelöscht haben.

Der Programmspeicher

Nach einem Löschen des Permanentspeichers besteht der Programmspeicher nur noch aus der permanenten **END**-Anweisung (.END.), die immer das Ende des Programmspeichers darstellt. Wenn Sie die Tastenfolge GTO ⊡ ⊡ ausführen und ein Programm eingeben, wird jede Anweisung genau vor der permanenten **END**-Anweisung eingefügt; dadurch wird das .END. nach unten verschoben, um Platz für die neue Anweisung zu schaffen. Dies hat zur Folge:

- Die erste Anweisung des zuerst eingegebenen Programms befindet sich am Anfang des Programmspeichers.
- Die letzte Anweisung des zuletzt eingegebenen Programms befindet sich am Ende des Programmspeichers, genau vor der permanenten **END**-Anweisung.

In Katalog 1 wird zusammen mit der permanenten **END**-Anweisung die Anzahl der ungebundenen Register angezeigt (.END. **REG** *nnn*). Über die angezeigte Anzahl *nnn* hinaus können noch bis zu sechs ungebundene Bytes (fast ein gesamtes Register) vorhanden sein.

Programmzeilen

In einem Programm wird jede Funktion, jede Zahl und jeder Alpha-String als gesonderte Programmzeile angesehen. Die Anzahl der *Programmzeilen* hängt davon ab, wieviele Funktionen, Zahlen und Alpha-Strings das Programm enthält. Die Anzahl der von diesen Programmzeilen belegten *Registern* und *Bytes* hängt von den einzelnen Funktionen und von der Länge der Ziffern- bzw. Alpha-Strings ab.

- Funktionen belegen zwischen einem und vier Bytes; die genaue Zahl hängt von der jeweiligen Funktion ab (auch ein gegebenenfalls benötigter Parameter ändert die Bytezahl). Im Funktionsverzeichnis (am Ende dieses Bandes) können Sie nachschlagen, wieviele Bytes von jeder Funktion belegt werden.
- Funktionen, die die Spezifikation eines globalen Labels als Parameter erfordern, belegen pro Zeichen des Parameters ein zusätzliches Byte.
- Zahlen belegen ein Byte pro Ziffer und ein weiteres Byte für jedes [·], [CHS] oder [EEX], das mit der Zahl eingegeben wird.
- Alpha-Strings belegen ein Byte pro Zeichen und ein zusätzliches Byte für den gesamten String.

Null-Bytes

Normalerweise folgt das erste Byte einer Programmanweisung unmittelbar auf das letzte Byte der vorhergehenden Anweisung. In einigen Fällen befinden sich jedoch sogenannte «Null-Bytes» zwischen Programmanweisungen. Null-Bytes resultieren aus den folgenden Operationen:

- **Löschen einer Anweisung.** Wenn Sie eine Anweisung löschen, werden die zu löschenden Bytes durch Null-Bytes ersetzt.
- **Einfügen von Programmanweisungen.** Wenn an der Einfügestelle nicht schon Null-Bytes vorhanden sind, fügt der Computer sieben Null-Bytes ein und schiebt alle nachfolgenden Anweisungen im Programmspeicher um sieben Bytes nach unten. Die neue Anweisung ersetzt die Null-Bytes; wenn eine Anweisung weniger als sieben Bytes belegt, bleiben die restlichen Null-Bytes erhalten.
- **Zahlen als Programmzeilen.** Der HP-41 setzt ein Null-Byte vor einen Byte-String, der eine Zahl repräsentiert. Dies geschieht für den Fall, daß die vorhergehende Programmzeile auch eine Zahl enthält. Das Null-Byte wirkt als eine Art Puffer zwischen den beiden Zahlen und verhindert, daß der Computer die beiden Programmzeilen als eine Zahl interpretiert.

Packen des Speichers

Wenn Ihr Programm abgeschlossen ist, sind nur die Null-Bytes von Nutzen, die Zahlen in aufeinanderfolgenden Programmzeilen trennen. Führen Sie die Tastenfolge GTO . . aus, um nicht benötigte Null-Bytes zu eliminieren. Dadurch wird der Programmspeicher gepackt und die belegten Bytes in allen Programmen rücken zusammen, um die unnötigen Null-Bytes zu ersetzen. (Der Speicher des User-Tastenfelds wird ebenfalls gepackt; dies wird später erläutert.) Der Speicher wird in den folgenden Fällen gepackt:

- Sie führen die Funktion PACK aus.
- Sie führen die Tastenfolge GTO . . aus.
- Sie löschen Programme durch CLP oder PCLPS.
- Es sind nicht genügend ungebundene Register vorhanden, um bestimmte Operationen auszuführen. Solche Operationen sind zum Beispiel: Erhöhen der Anzahl der Datenspeicher-Register, Eingabe einer Programmzeile oder Belegen einer Taste mit einer Funktion.

Der Alarmspeicher

Wenn Sie den ersten Alarm speichern, benötigt der Computer zwei ungebundene Register, um den Alarmspeicher abzugrenzen. (Diese Register werden zusätzlich zu den von dem Alarm belegten Registern benötigt.) Diese beiden Register werden erst dann wieder freigegeben, wenn Sie alle gespeicherten Alarme löschen. Die Anzahl der von einem Alarm belegten Register hängt von der Alarmart ab.

- Jeder Alarm benötigt ein Register zur Speicherung der nächsten Auslösezeit und des dazugehörigen Datums.
- Ein wiederholender Alarm benötigt ein zusätzliches Register zur Speicherung des Wiederholungsintervalls.
- Ein Alarm, der das Alpha-Register benutzt, benötigt ein zusätzliches Register für jeden Teilstring mit bis zu sieben Zeichen; dies gilt unabhängig davon, ob es sich um einen Meldealarm oder um einen Steueralarm handelt. (Der Aufpfeil ↑ zählt hierbei als ein Zeichen.) Beispiel: Ein Meldealarm, dessen Meldung aus 18 Zeichen besteht, belegt 2 Register für die ersten 14 Zeichen und ein weiteres Register für die letzten vier Zeichen; die Meldung belegt also insgesamt drei Register.

Der Speicher des User-Tastenfelds

Wenn Sie eine Tastenzuordnung vornehmen, speichert der Computer diese Information im Speicher des User-Tastenfelds. Auch die Zuordnung von Funktionen bzw. globalen Labels, die dem HP-41 über ein Einsteck-Modul zugänglich sind, werden im Speicher des User-Tastenfelds aufgezeichnet. Wenn Sie einer Taste ein globales Label zuordnen, das in Katalog 1 enthalten ist, wird diese Zuordnung nicht im Speicher des User-Tastenfelds, sondern zusammen mit dem globalen Label im Programmspeicher aufgezeichnet.

Ein Register kann zwei Tastenzuordnungen speichern. Die erste Zuordnung belegt ein Register; die zweite Zuordnung wird zusammen mit der ersten gespeichert. Jede ungeradzahlige Zuordnung belegt ein weiteres Register; dieses wird dann durch die nächste (geradzahlige) Zuordnung aufgefüllt.

Eine Zuordnung wird aufgehoben, wenn Sie der gleichen Taste eine andere Funktion zuordnen oder die Zuordnung durch eine der auf Seite 168 erläuterten Methoden löschen. Wenn Sie die beiden im gleichen Register gespeicherten Zuordnungen löschen und den Hauptspeicher packen, wird das Register wieder zu einem ungebundenen Register.

Der Datenspeicher

Zuweisen von Datenspeichern

Ändern der Speicherzuweisung. Der HP-41 verfügt über zwei Funktionen, mit denen Sie Hauptspeicher-Register der Datenspeicherung zuweisen können. Eine dieser Funktionen kann nur manuell über das Tastenfeld ausgeführt werden; die zweite Funktion kann als Programmanweisung aufgezeichnet werden. Wenn Sie die Anzahl der Datenspeicher-Register verkleinern, geht der Inhalt der Register mit den höchsten Adressen verloren.

- Über das Tastenfeld können sie die Anzahl der Datenspeicher-Register durch Ausführen der Funktion SIZE ändern. Die Funktion erwartet die Anzahl der zuzuweisenden Register als Parameter.
- In einem Programm kann die Anzahl der Datenspeicher-Register verändert werden, indem die Anzahl der zuzuweisenden Register im X-Register abgelegt und anschließend die Funktion PSIZE ausgeführt wird.

Überprüfen der Zuweisung. Führen Sie die Funktion SIZE? aus, um herauszufinden, wieviel Register momentan der Datenspeicherung zugewiesen sind. Der Computer legt die Anzahl der Datenspeicher-Register im X-Register ab.

Beispiel. Die untenstehenden Programmzeilen ändern die Registerzuweisung nur dann, wenn die momentane Anzahl von Datenregistern nicht ausreicht.

01 SIZE?	Legt die Anzahl der momentan zugewiesenen Datenregister im X-Register ab.
02 50	Die von dem Programm benötigte Anzahl von Datenregistern. Die momentane Zuweisung (aus Zeile 01) befindet sich jetzt im Y-Register.
03 X>Y?	Abfrage: Werden mehr Datenregister benötigt als vorhanden sind?
04 PSIZE	Wenn ja, weise die benötigte Registeranzahl der Datenspeicherung zu.

Registeradressen über R_{99}

Wenn mehr als 100 Datenregister definiert sind, können Sie auf die Register mit Adressen größer als 99 *nur* indirekt zugreifen. Um auf diese Besonderheit hinzuweisen, führt dieses Handbuch dreistellige Registeradressen in Klammern auf: zum Beispiel $R_{(120)}$. Es ist oft bequemer, einen Datenfile im erweiterten Speicher zu benutzen, es sei denn, Sie benötigen die Register zur indirekten Adressierung in einer Programmschleife oder Sie führen Speicheroperationen (z.B. Registerarithmetik) aus.

Operationen mit Datenspeicher-Registern

Speichern und Zurückrufen

Der HP-41 ermöglicht den Datentransfer zwischen den Datenregistern, den Stackregistern, dem Alpha-Register und Datenfiles im erweiterten Speicher. In diesem Abschnitt werden die Funktionen erläutert, die Daten zwischen den Datenregistern und den Stackregistern bzw. dem Alpha-Register übertragen. Funktionen, die auf Datenfiles im erweiterten Speicher wirken, werden in Abschnitt 13, «Erweiterter Speicher», beschrieben.

Spezifizieren einer Registeradresse als Parameter. Die meisten Datenregister-Funktionen wirken nur auf ein einziges Register, dessen Adresse als Parameter spezifiziert werden muß. Diese Spezifikation kann auf mehrere Arten erfolgen:

- Geben Sie die zweistellige Adresse ein, um ein Register zwischen R_{00} und R_{99} zu spezifizieren.
- R_{00} bis R_{10} können durch Drücken der entsprechenden Taste der beiden obenliegenden Tastenreihen spezifiziert werden.
- Drücken Sie die Taste [·], um auf ein Stackregister bzw. das LAST X Register zuzugreifen und spezifizieren Sie dann das gewünschte Register durch Drücken von [X], [Y], [Z], [T] oder [L].
- Zur indirekten Adressierung eines Registers ist die Taste ■ zu drücken und das gewünschte Register durch eine der ersten drei Methoden zu spezifizieren.

Speichern. Drücken Sie die Taste [STO], um den Inhalt des X-Registers in ein Datenregister zu kopieren; der Computer erwartet als Parameter die Adresse des Zielregisters. Bei dieser Operation wird der Inhalt des X-Registers nicht verändert; der vorhergehende Inhalt des Datenregisters geht verloren.

Zurückrufen. Drücken Sie die Taste [RCL], um den Inhalt eines Datenregisters in das X-Register zu kopieren; der Computer erwartet als Parameter die Adresse des Quellregisters. Der Inhalt des spezifizierten Datenregisters wird nicht verändert. Wenn der Stack Lift gesperrt ist, ersetzen die abgerufenen Daten den Inhalt des X-Registers; ansonsten wird der Stack angehoben (Stack Lift).

Speichern von Alpha-Zeichen. Um die sechs links außen im Alpha-Register stehenden Zeichen in ein Datenregister zu kopieren, drücken Sie zuerst die Taste [ASTO] und spezifizieren dann die Adresse des Zielregisters. Der Inhalt des Alpha-Registers wird dabei nicht verändert; der vorhergehende Inhalt des Zielregisters geht verloren.

- Ein Interpunktionszeichen zählt als eines der sechs Zeichen.
- Ein im Alpha-Register abgelegter Ziffernstring ist keine Zahl. Wenn Sie Alpha-Ziffern in einem Datenregister abspeichern, *erscheint* der Inhalt zwar als Zahl, es können jedoch keine numerischen Operationen damit ausgeführt werden.
- Bevor Sie mit im Alpha-Register abgelegten Ziffern eine numerische Operation ausführen können, müssen Sie den Ziffernstring durch die Funktion [ANUM] in eine Zahl umwandeln und diese Zahl im X-Register ablegen (siehe Seite 311).
- Wenn Sie durch die Tastenfolge [ASTO] [·] [X] Zeichen aus dem Alpha-Register in das X-Register kopieren, wird der Stack nicht angehoben (*im Gegensatz* zur Funktion [RCL]) und der vorhergehende Inhalt des X-Registers geht verloren.

Um mehr als sechs Zeichen in ein Datenregister zu kopieren, müssen Sie vor erneutem Ausführen von [ASTO] den Inhalt des Alpha-Registers ändern (ansonsten würden Sie die gleichen Zeichen noch einmal abspeichern).

- Führen Sie die Funktion [ASHF] (*Alpha shift*) aus, um die sechs links außen im Alpha-Register stehenden Zeichen nach links aus dem Alpha-Register zu schieben.
- Um die sechs links außen im Alpha-Register stehenden Zeichen an den rechten Registerrand zu schieben, geben Sie zuerst die Zahl 6 in das X-Register ein und führen dann die Funktion [AROT] (*Alpha rotate*) aus.

Zurückrufen von Alpha-Zeichen. Führen Sie die Funktion [ARCL] aus, um Daten aus einem Datenregister in das Alpha-Register zu kopieren; der Computer erwartet die Adresse des Quellregisters als Parameter. Der Inhalt des Quellregisters wird bei dieser Operation nicht verändert; die abgerufenen Daten werden an den Inhalt des Alpha-Registers *angehängt* und der Eingabemodus wird aktiviert. Wenn Sie mit den zurückgerufenen Daten einen neuen Zeichenstring aufbauen wollen, sollten Sie vor der Rückrufoperation die Funktion [CLA] ausführen.

Registerarithmetik

Eine im X-Register abgelegte Zahl kann mit einer in einem Datenregister abgelegten Zahl verbunden werden, ohne daß die gespeicherte Zahl in den Stack zurückgerufen werden muß.

- Die Funktion [STO] [+] **nn** addiert die Zahl im X-Register zu der Zahl in R_{nn} und speichert die Summe in R_{nn}.
- Die Funktion [STO] [−] **nn** subtrahiert die Zahl im X-Register von der Zahl in R_{nn} und speichert die Differenz in R_{nn}.
- Die Funktion [STO] [×] **nn** multipliziert die Zahl im X-Register mit der Zahl in R_{nn} und speichert das Produkt in R_{nn}.
- Die Funktion [STO] [÷] **nn** dividiert die Zahl in R_{nn} durch die Zahl im X-Register und speichert den Quotienten in R_{nn}.

Diese Funktionen verhalten sich wie die Funktion [STO]; der vorhergehende Inhalt von R_{nn} geht verloren und die Zahl im X-Register wird nicht verändert. Dies erlaubt Ihnen, die im X-Register abgelegte Zahl als Konstante zu verwenden, ohne auf das LAST X Register zugreifen zu müssen.

Registeraustausch

Bei Ausführen der Funktionen [STO] und [RCL] wird jeweils eine Zahl dupliziert und eine andere gelöscht. Sie können Daten transferieren, *ohne* dabei Registerinhalte zu kopieren oder zu verlieren. Führen sie dazu die Funktion [X<>] aus und spezifizieren Sie das Datenregister, dessen Inhalt Sie mit dem X-Register vertauschen wollen.

Operationen mit Registerblöcken

Die Funktionen [REGMOVE] und [REGSWAP] erlauben es Ihnen, den Inhalt eines Registerblocks in einen anderen Block zu kopieren bzw. die Inhalte von zwei Registerblöcken zu vertauschen. Bei beiden Funktionen muß als Steuergröße die Zahl *sss.dddnnn* im X-Register spezifiziert werden. Diese Steuergröße definiert den Quell- und den Ziel-Registerblock; dabei gilt:

- R_{sss} ist das erste Register des *Quell*-Registerblocks.
- R_{ddd} ist das erste Register des *Ziel*-Registerblocks.
- *nnn* ist die Register*anzahl* in beiden Registerblöcken. Wenn Sie *nnn* nicht spezifizieren, setzt der HP-41 automatisch *nnn* = 001.

Die Registerblöcke dürfen sich nicht überschneiden; stellen Sie sicher, daß gilt $|sss - ddd| \geqslant nnn$.

Kopieren von Registerinhalten. Um den Inhalt eines Registerblocks (Quelle) in einen anderen Registerblock (Ziel) zu kopieren, können Sie folgendermaßen vorgehen: Legen Sie die Steuerzahl im X-Register ab und führen Sie die Funktion [REGMOVE] aus. Der Inhalt des Quell-Registerblocks wird nicht verändert und der vorhergehende Inhalt des Ziel-Registerblocks geht verloren (wie bei der Funktion [STO]).

Vertauschen von Registerinhalten. Um den Inhalt zweier Registerblöcke miteinander zu vertauschen, können Sie folgendermaßen vorgehen: Legen Sie die Steuerzahl im X-Register ab und führen Sie die Funktion [REGSWAP] aus. Bei dieser Funktion spielt es keine Rolle, welcher Registerblock als Quelle und welcher als Ziel bezeichnet wird.

Löschen von Registern

Löschen eines einzelnen Registers. Speichern Sie Null in einem Register, um dessen Inhalt zu löschen.

Löschen eines Registerblocks. Es ist oft sehr hilfreich, am Anfang eines Programms eine Anweisung zum Löschen aller Datenregister vorzusehen. Die Funktion CLRGX (*clear register by X*) löscht den Inhalt eines Registerblocks. Der Registerblock wird durch die im X-Register abgelegte Steuerzahl *bbb.eeeii* definiert. Dabei gilt:

- R_{bbb} (*Beginn*) ist das erste zu löschende Register (d.h. das Register mit der niedrigsten Adresse).
- R_{eee} (*Ende*) ist das letzte zu löschende Register (d.h. das Register mit der höchsten Adresse).
- Die Zahl *ii* (*Inkrement*) ist nur dann zu spezifizieren, wenn lediglich jedes *ii*te Register gelöscht werden soll. Wenn Sie *ii* nicht spezifizieren, setzt der HP-41 automatisch *ii* = 01.

Das Inkrement *ii* ist zum Beispiel bei Berechnungen mit Matrizen sehr nützlich. Nehmen wir an, Sie haben die Zeilen einer 4 × 4 Matrix elementweise im Speicher aufgezeichnet. Sie können jede Zeile der Matrix löschen, indem Sie das erste und letzte Register spezifizieren; das Inkrement *ii* braucht nicht spezifiziert zu werden. Wenn Sie eine Spalte (deren Elemente nicht in aufeinanderfolgenden Registern gespeichert sind) löschen wollen, können Sie wie folgt vorgehen: Spezifizieren Sie zuerst die Registeradressen, die dem ersten und letzten Spaltenelement entsprechen und setzen Sie dann *ii* = 04; dadurch werden nur die Register dieser einen Spalte gelöscht.

R_{bbb} wirdt selbst dann gelöscht, wenn gilt: $bbb > eee$ oder $bbb + ii > eee$. Das Vorzeichen der Steuerzahl sowie zusätzliche Dezimalstellen werden vom Computer ignoriert.

Löschen aller Register. Führen Sie die Funktion CLRG aus, um alle Datenregister zu löschen.

Abschnitt 13

Erweiterter Speicher

Inhalt

Einführung 205
Files im erweiterten Speicher 205
Fileheader 205
Spezifizieren von Files 206
Katalog des erweiterten Speichers 206
Überprüfen der Filegröße 208
Löschen von Files 208
Operationen mit Programmfiles 208
Speichern eines Programms 208
Laden eines Programms 209
Automatische Wiederherstellung von Tastenzuordnungen 211
Überprüfen der Programmgröße 211
Erzeugen von Daten- und Textfiles 211
Erzeugen von Datenfiles 211
Erzeugen von Textfiles 212
Ändern der Filegröße 213
Löschen von Fileinhalten 213
Zeiger in Daten- und Textfiles 213
Aufbau von Datenfiles 214
Aufbau von Textfiles 214
Zeiger-Operationen 215
Operationen mit Datenfiles 216
Zugriff auf alle Datenregister 217
Zugriff auf einen Block von Datenregistern 218
Zugriff auf das X-Register 220
Operationen mit Textfiles 222
Überprüfen der verfügbaren Bytes 222
Record-Operationen 222
Zeichen-Operationen 224
Durchsuchen eines Files 226
Kopieren von Daten in das Alpha-Register 226
Zugriff auf Massenspeicher-Files 227

Einführung

Der erweiterte Speicher stellt eine Erweiterung des Hauptspeichers dar und verfügt über eine Reihe spezifischer Vorteile. Im erweiterten Speicher abgelegte Informationen sind in sogenannten Files aufgezeichnet; der Computer unterscheidet zwischen drei verschiedenen Filetypen:

- Ein im Hauptspeicher abgelegtes Programm kann in einem Programmfile im erweiterten Speicher abgelegt werden. Wenn Sie das Programm ausführen oder modifizieren wollen, müssen Sie es in den Hauptspeicher zurückladen.
- Registerinhalte können in einem Datenfile gespeichert werden; ein Datenfile besteht aus zusammengefassten Einzelregistern. Durch eine einzige Operation haben Sie Zugriff auf alle Datenregister des Hauptspeichers, auf einen Block von Datenregistern oder auf das X-Register. Der direkte Zugriff auf das X-Register macht den erweiterten Speicher zu mehr als nur einer Erweiterung des Hauptspeichers: der erweiterte Speicher kann alternativ zum Hauptspeicher benutzt werden.
- Alpha-Daten können in einem Textfile (auch ASCII-File genannt) gespeichert werden; ein Textfile ist aus Alpha-Strings zusammengesetzt. Die alphanumerischen Fähigkeiten von Textfiles (Größe und Aufbau von Textfiles, direkter Zugriff auf Textfiles durch den Texteditor) übertreffen die des Hauptspeichers bei weitem.

Pro File belegt der Computer zwei Register zur internen Informationsspeicherung (im sogenannten Header); die Anzahl der verfügbaren Register des erweiterten Speichers hängt also von der Anzahl der Files ab. Ein einzelner File darf bis zu 124 Register belegen; zwei Files dürfen zusammen bis zu 122 Register belegen. Allgemein gilt, daß Ihnen für n Files $126-2n$ Register zur Verfügung stehen.

Sie können die Anzahl der Register des erweiterten Speichers durch Einsetzen von Erweiterten Speicher-Modulen HP 82181A vergrößern (siehe Anhang E); Sie können maximal zwei Erweiterte Speicher-Module mit Ihrem HP-41 benutzen. Jedes Modul vergrößert den erweiterten Speicher um 238 Register. Sie können also maximal auf $124 + 238 + 238 = 600$ erweiterte Speicherregister zugreifen (zum Vergleich: der Hauptspeicher verfügt über 319 Register).

Files im erweiterten Speicher

Fileheader

Am Beginn eines Files steht ein aus zwei Registern bestehender *«Header»*. Dies ist eine Art Überschrift, in der Informationen zu dem File gespeichert sind. Obwohl Sie niemals auf die Header-Register direkt zugreifen können, ist es hilfreich, die Art der abgelegten Informationen zu kennen. Ein Header enthält die folgenden Informationen:

Filename. Wenn Sie einen File erzeugen, geben Sie ihm einen Namen. Der Name kann eine beliebige, maximal sieben Zeichen lange Kombination aus alphanumerischen Zeichen sein (Ausnahme: der Filename darf kein Komma enthalten).* Beim Erzeugen und beim Spezifizieren eines Files geben Sie dessen Namen in das Alpha-Register ein.

Filetyp. Der HP-41 unterscheidet zwischen drei verschiedenen Filetypen: Programmfile, Datenfile und Textfile. Ein Programmfile ist die Kopie eines im Hauptspeicher aufgezeichneten Programms; ein Datenfile ist eine Ansammlung von mehreren Datenregistern; ein Textfile besteht aus alphanumerischen Zeichenketten.

* Der Filename darf nicht aus sieben Bytes mit dem Dezimalwert 255 bestehen.

Filegröße. Unter Filegröße versteht man die Anzahl der von dem File belegten erweiterten Speicherregister. Bei Programmfiles enthält der Header auch die Bytezahl des Programms.

Zeiger. Bei Daten- und Textfiles wird durch einen numerischen Zeiger angedeutet, auf welches Element des Files als nächstes zugegriffen werden kann. Programmfiles verfügen über keinen Zeiger.

Spezifizieren von Files

Sie können nicht auf mehrere Files gleichzeitig zugreifen. Der Computer «merkt sich» den zuletzt aufgerufenen File und macht diesen zum *momentanen File*. Einige Funktionen des erweiterten Speichers wirken nur auf dem momentanen File, andere wirken dagegen auch auf einem im Alpha-Register zu spezifizierenden File.

- Einige Funktionen *erfordern* immer die Spezifikation eines Filenamens im Alpha-Register. (Zum Beispiel Programmfile-Funktionen und Löschfunktionen.)
- Bei einigen Funktionen *kann* ein Filename im Alpha-Register spezifiziert werden. Wenn das Alpha-Register leer ist, wirken diese Funktionen auf den momentanen File.
- Einige Funktionen wirken *nur* auf den *momentanen* File, unabhängig vom Inhalt des Alpha-Registers.

Katalog des erweiterten Speichers

Der HP-41 verfügt über zwei programmierbare Funktionen zum Zugriff auf die Files im erweiterten Speicher. Die Funktion [EMDIR] (*extended memory directory*) listet alle im erweiterten Speicher aufgezeichneten Files auf; diese Funktion eignet sich besonders zur Kommunikation zwischen Benutzer und Computer. Die Funktion [EMDIRX] (*extended memory directory by X*) ruft Informationen über einen im X-Register spezifizierten File ab; [EMDIRX] eignet sich besonders zur Verwendung in Programmen.

Die Funktion [EMDIR]. Über das Tastenfeld kann der Katalog des erweiterten Speichers auch durch die Tastenfolge [CATALOG] 4 abgerufen werden. Wenn Sie die Katalogfunktion einer Taste zuordnen wollen oder als Programmschritt aufzeichnen wollen, müssen Sie den Alpha-Namen benutzen: [EMDIR]. Beachten Sie bei der Ausführung der Funktion [EMDIR]:

- Die Files werden in der Reihenfolge aufgelistet, in der sie erzeugt wurden.
- Der Katalog zeigt den Filenamen, den Filetyp (P = Programm, D = Daten, A = ASCII bzw. Text) und die Anzahl der belegten Register. Beispiel: Ein Datenfile ABC, der 20 Register belegt, erscheint folgendermaßen in dem Katalog:

ABC D020

- Durch Drücken einer beliebigen Taste (außer [R/S] und [ON]) wird die Katalogausgabe beschleunigt.
- Die Katalogausgabe wird durch Drücken der Taste [R/S] angehalten; nochmaliges Drücken von [R/S] setzt die Katalogausgabe fort.
- Bei angehaltenem Katalog können Sie durch Drücken der Taste [SST] den Katalogeintrag des nächsten Files anzeigen; durch Drücken von [BST] können Sie den Katalogeintrag des vorhergehenden Files anzeigen. Wenn der letzte File angezeigt ist, wird [SST] nicht ausgeführt; wenn der erste File angezeigt ist, wird [BST] nicht ausgeführt.

Wenn Sie die Katalogausgabe nicht anhalten, verläßt der Computer den Katalog-Modus automatisch nach Anzeige des letzten Files. Im X-Register wird dabei die Anzahl der für den nächsten File verfügbaren Register abgelegt; diese Zahl berücksichtigt bereits die beiden für einen neuen File benötigten Header-Register (d.h. alle angezeigten Register stehen für Ihre Daten zur Verfügung). Beim Ablegen dieses Werts im X-Register wird der Stack angehoben (sofern der Stack Lift nicht gesperrt ist).

Wenn der Computer automatisch den Katalog-Modus verläßt, wird der momentane File nicht neu definiert. Sie können jeden File zum momentanen File machen, indem Sie die Katalogausgabe bei dem gewünschten File anhalten; beenden Sie dann den Katalog-Modus durch Drücken der Taste [←]. Die Anzahl der verfügbaren Register wird bei dieser Operation *nicht* im X-Register abgelegt.

Wenn der erweiterte Speicher keine Files enthält, zeigt der Computer die Meldung **DIR EMPTY** an und legt die Anzahl der verfügbaren Register im X-Register ab. Dabei wird der Stack angehoben (sofern der Stack Lift nicht gesperrt ist).

Die Funktion [EMDIRX]. Um den Namen und Typ des *n*-ten Files im erweiterten Speicher zu bestimmen, können Sie folgendermaßen vorgehen: Legen Sie die Zahl *n* im X-Register ab und führen Sie die Funktion [EMDIRX] aus. Zum Beispiel *n* = 1 für den File, den Sie als erstes erzeugt haben (dieser File steht auch an erster Stelle im Katalog [EMDIR]).

Wenn der *n*-te File existiert, hat [EMDIRX] die folgenden Auswirkungen:

- Der Name des *n*-ten Files wird im Alpha-Register abgelegt.
- Der Filetyp wird durch einen im X-Register abgelegten Buchstabencode gekennzeichnet:*

  PR = Programmfile.
  DA = Datenfile.
  AS = ASCII (Text) File.
- *n* wird im LAST X Register abgelegt.
- Der *n*-te File wird zum momentanen File.

Wenn der *n*-te File nicht existiert, hat [EMDIRX] die folgenden Auswirkungen:

- Das Alpha-Register wird gelöscht.
- Im X-Register wird Null abgelegt.
- *n* wird im LAST X Register abgelegt.
- Der momentane File ist der gleiche wie vor Ausführen von [EMDIRX].

Beachten Sie, daß die Funktion [EMDIRX] den Stack nicht anhebt. Die Eingabe *n* ist der Betrag des ganzzahligen Anteils der Zahl im X-Register.

* Wenn der Computer den Filetyp nicht erkennt, wird eine positive ganze Zahl im X-Register abgelegt.

Die Funktion EMROOM**.** Führen Sie die Funktion EMROOM (*extended memory room*) aus, um zu überprüfen, wieviele Register Ihnen zur Speicherung eines neuen Files zur Verfügung stehen. Das Ergebnis berücksichtigt bereits die beiden Register für den Header eines neuen Files. Wenn Sie mehr als einen File speichern wollen, müssen Sie für jeden weiteren File zwei Register von der angezeigten Zahl abziehen. Das Ergebnis wird im X-Register abgelegt; dabei wird der Stack angehoben (sofern der Stack Lift nicht gesperrt ist).

Überprüfen der Filegröße

Um die Anzahl der von einem File belegten Register (ausschließlich der Header-Register) abzurufen, ist zunächst der Filename in das Alpha-Register einzugeben und danach die Funktion FLSIZE (*file size*) auszuführen. Das Ergebnis wird im X-Register abgelegt; dabei wird der Stack angehoben (sofern der Stack Lift nicht gesperrt ist). Wenn Sie die Größe des momentanen Files überprüfen wollen, genügt es, vor Ausführen von FLSIZE das Alpha-Register zu löschen.

Löschen von Files

Wenn Sie einen File vollständig (d.h. Fileinhalt und Header) aus dem erweiterten Speicher entfernen wollen, ist der Filename im Alpha-Register abzulegen und die Funktion PURFL (*purge file*) auszuführen. Alle nach dem gelöschten File im erweiterten Speicher stehenden Files rücken auf, um die entstandene Lücke aufzufüllen.

Nach Ausführen von PURFL existiert kein momentaner File; wenn Sie die nächste Funktion des erweiterten Speichers ausführen, müssen Sie immer einen Filenamen spezifizieren (dieser File wird dann zum momentanen File).

Operationen mit Programmfiles

Von den verschiedenen Filetypen sind die Programmfiles am einfachsten zu handhaben. Ein Programmfile besteht aus zwei Header-Registern und einer exakten Kopie des Programms aus dem Hauptspeicher.

Speichern eines Programms

Um ein Programm in einen File zu kopieren, können Sie folgendermaßen vorgehen: Spezifizieren Sie das zu kopierende Programm und den Namen des zu erzeugenden Files im Alpha-Register; führen Sie dann die Funktion SAVEP (*save program*) aus. Das Programm und der Name des neuen Files können auf verschiedene Art und Weise spezifiziert werden.

- Spezifizieren Sie das zu kopierende Programm durch ein in dem Programm vorkommendes globales Label. Trennen Sie das Label durch ein Komma von dem Namen des neuen Files.

Alpha-Register

Label,Filename

- Geben Sie nur das globale Label in das Alpha-Register ein, wenn Sie das Label als Filename verwenden wollen.

Alpha-Register

Label

- Geben Sie ein Komma, gefolgt von dem Namen des neuen Files, in das Alpha-Register ein, wenn Sie das momentane Programm aus dem Hauptspeicher kopieren wollen.

Alpha-Register

,Filename

Wenn ein spezifiziertes globales Label mehrfach existiert, kopiert [SAVEP] das Programm, das dem unteren Ende des Programmspeichers am nächsten ist (d.h. das Programm, das von Katalog 1 als letztes aufgelistet wird) und dieses globale Label enthält.

Der Filename kann aus bis zu sieben Zeichen bestehen; es dürfen alle Zeichen außer Kommata verwendet werden. Das achte Zeichen und alle nachfolgenden Zeichen bzw. ein Komma und alle nachfolgenden Zeichen werden vom Computer ignoriert.

> **Hinweis:** Wenn der neue Filename identisch mit dem Namen eines existierenden Programmfiles ist, wird der Inhalt des existierenden Programmfiles überschrieben. (Dies gilt unabhängig davon, welche der oben erläuterten Methoden Sie anwenden.)

Laden eines Programms

Der HP-41 verfügt über zwei Funktionen zum Kopieren eines Programms aus dem erweiterten Speicher in den Hauptspeicher: [GETP] (*get program*) und [GETSUB] (*get subroutine*). Beide Funktionen legen das kopierte Programm am unteren Ende des Programmspeichers ab; dabei unterscheiden sich [GETP] und [GETSUB] in einem wichtigen Punkt:

- [GETP] *ersetzt* das am unteren Ende des Programmspeichers stehende Programm durch das kopierte (geladene) Programm. Unter Umständen wird die erste Zeile des kopierten Programms zur momentanen Programmzeile.
- [GETSUB] *kopiert* das spezifizierte Programm *hinter* das am unteren Ende des Programmspeichers stehende Programm. [GETSUB] hat keine Auswirkungen auf die momentane Programmzeile.

Ausführung über das Tastenfeld.

- Wenn Sie das letzte Programm im Programmspeicher durch ein Programm aus dem erweiterten Speicher ersetzen wollen, ist der Name des Programmfiles im Alpha-Register abzulegen und danach die Funktion [GETP] auszuführen. Die erste Zeile des geladenen Programms wird zur momentanen Programmzeile; Sie können die Programmausführung durch Drücken der Taste [R/S] starten.
- Um das letzte Programm im Programmspeicher zu erhalten und gleichzeitig die erste Zeile des geladenen Programms zur momentanen Programmzeile zu machen, führen Sie zuerst die Tastenfolge [GTO] [.] [.] und dann die Funktion [GETP] aus. (Die Tastenfolge [GTO] [.] [.] erzeugt ein Null-Programm, das durch das kopierte Programm ersetzt wird.)
- Wenn Sie das letzte Programm im Programmspeicher und die momentane Programmzeile erhalten wollen, können Sie folgendermaßen vorgehen: Legen Sie den Namen des Programmfiles im Alpha-Register ab und führen Sie die Funktion [GETSUB] aus. Dies ist insbesondere nützlich, wenn Sie mehrere Programme nacheinander laden wollen.

Ausführung durch ein Programm. Die Funktion [GETP] hat verschiedene Auswirkungen, je nachdem ob das Programm, das die Anweisung [GETP] enthält, am Ende des Programmspeichers steht oder nicht. Die Auswirkung der Funktion [GETSUB] hängt nicht von der Position des Programms im Programmspeicher ab. Es existieren drei Möglichkeiten, ein Programm anzuweisen, ein anderes Programm aus dem erweiterten Speicher zu laden:

- Wenn ein Programm die Funktion [GETSUB] ausführt, wird das geladene Programm an das Ende des Programmspeichers kopiert; die Programmausführung wird mit der Anweisung nach [GETSUB] fortgesetzt.
- Wenn ein *nicht* am Ende des Programmspeichers stehendes Programm die Funktion [GETP] ausführt, ersetzt das geladene Programm das letzte Programm im Programmspeicher; die Programmausführung wird mit der Anweisung nach [GETP] fortgesetzt.
- Wenn ein am Ende des Programmspeichers stehendes Programm die Funktion [GETP] ausführt, wird dieses Programm durch das geladene Programm ersetzt; die Programmausführung wird mit der ersten Zeile des geladenen Programms fortgesetzt.

Beispiele. Die drei oben erläuterten Möglichkeiten machen Sie sehr flexibel; oft existiert mehr als eine Alternative, ein Programm durch ein anderes Programm laden zu lassen. Die Beispiele erläutern, welche Operationen am besten zu bestimmten Problemstellungen passen. Die Beispiele gehen von einem Programm **HAUPT** im Hauptspeicher (aber nicht am Ende des Programmspeichers) aus. **AA**, **BB** und **CC** sind Programme im erweiterten Speicher, die von **HAUPT** als Unterprogramme aufgerufen werden.

Die ersten vier Beispiele illustrieren die ersten beiden Möglichkeiten; die Funktionen [GETP] und [GETSUB] werden von dem Programm **HAUPT** ausgeführt. Um sicherzustellen, daß **HAUPT** nicht das letzte Programm im Programmspeicher ist, können Sie durch die Tastenfolge [GTO] [.] [.] ein Null-Programm erzeugen, bevor Sie **HAUPT** ausführen. Sie können auch am Anfang von **HAUPT** eine Anweisung vorsehen, die durch [GETSUB] ein nur aus einem globalen Label und einer Endanweisung bestehendes Leer-Programm lädt. (Das Laden eines Leer-Programms ermöglicht es Ihnen, am Programmende von **HAUPT** die Anweisung [PCLPS] einzufügen; dadurch werden das Leer-Programm und alle nachfolgend von **HAUPT** geladenen Programme gelöscht.)

- **HAUPT** ruft die Programme **AA**, **BB** und **CC** als Unterprogramme auf und der Hauptspeicher kann alle drei Programme aufnehmen: Verwenden Sie am Anfang von **HAUPT** dreimal [GETSUB], um **AA**, **BB** und **CC** aufzurufen; benutzen Sie [XEQ] zum Ausführen der Unterprogramme.
- **HAUPT** ruft die Unterprogramme in der Reihenfolge **AA**, **BB**, **CC** auf und der Hauptspeicher kann nicht alle Programme auf einmal aufnehmen: Verwenden Sie [GETSUB] und [XEQ], um **AA** zu laden; verwenden Sie [GETP] und [XEQ], um **BB** und **CC** zu laden.
- Es ist nicht sicher, in welcher Reihenfolge **AA**, **BB** und **CC** aufgerufen werden und der Hauptspeicher kann nicht alle Programme auf einmal aufnehmen: Verwenden Sie [GETSUB], [XEQ] und [PCLPS] zum Laden der Unterprogramme; Sie können auch ein Leer-Programm an das Speicherende setzen und die Unterprogramme mit [GETP] und [XEQ] abrufen.
- **AA** enthält ein zweites globales Label **DD** und die Programmausführung soll mit **DD** fortgesetzt werden: Verwenden Sie [GETSUB], um **AA** zu laden; verwenden Sie [XEQ] **DD**, um die Programmausführung mit **DD** fortzusetzen.

Die nächsten beiden Beispiele erläutern die erste und dritte Möglichkeit; die Funktionen [GETSUB] und [GETP] werden in dem Unterprogramm **AA** verwendet. Die Beispiele gehen davon aus, daß **HAUPT** das Unterprogramm **AA** an das Ende des Programmspeichers kopiert hat.

- **AA** enthält eine Programmverzweigung zu **BB**: Verwenden Sie [GETP] in **AA**, um **BB** zu laden. Dadurch wird **AA** automatisch gelöscht und die Programmausführung mit **BB** fortgesetzt.
- **AA** ruft **BB** als Unterprogramm auf: Verwenden Sie [GETSUB] und [XEQ] in **AA**, um **BB** zu laden. Dadurch bleibt **AA** erhalten und nach Durchlaufen von **BB** kann die Programmausführung wieder mit **AA** fortgesetzt werden.

Automatische Wiederherstellung von Tastenfeldzuordnungen

In Abschnitt 10 wurde erläutert, daß die Zuordnung eines globalen Labels zu einer Taste des User-Tastenfelds zusammen mit dem globalen Label gespeichert wird. (Dies gilt nur für Programme, die von Katalog 1 aufgelistet werden.) Wenn [SAVEP] ein Programm in den erweiterten Speicher kopiert, werden alle bestehenden Zuordnungen zusammen mit den globalen Labels gespeichert. Wenn ein Programm bei eingeschaltetem User-Tastenfeld (d.h. Flag 27 ist gesetzt) durch [GETSUB] oder [GETP] abgerufen wird, werden alle mit dem Programm gespeicherten Zuordnungen wieder wirksam.

Überprüfen der Programmgröße

Mit den Funktionen [RCLPT] und [RCLPTA] können Sie überprüfen, wieviele Bytes ein in einem Programmfile gespeichertes Programm belegt. (Diese Funktionen werden normalerweise vor allem auf Daten- und Textfiles angewandt.) Beide Funktionen legen die Bytezahl im X-Register ab und heben den Stack an (sofern der Stack Lift nicht gesperrt ist.)

- Wenn der gewünschte Programmfile nicht der momentane File ist, legen Sie den Filenamen im Alpha-Register ab und führen die Funktion [RCLPTA] aus.
- Führen Sie die Funktion [RCLPT] aus, um die Größe des momentanen Files zu überprüfen. Sie können wahlweise auch das Alpha-Register löschen und die Funktion [RCLPTA] ausführen.

Erzeugen von Daten- und Textfiles

Sie müssen jedem Daten- bzw. Textfile Speicherplatz im erweiterten Speicher zuordnen. Bevor Sie einen File erzeugen, können Sie überprüfen, wieviele Register im erweiterten Speicher zur Verfügung stehen: Führen Sie die Funktion [EMDIR] (auf dem Tastenfeld als [CATALOG] 4 vorhanden) oder die Funktion [EMROOM] aus. Die daraufhin im X-Register abgelegte Zahl bezeichnet die maximale Registeranzahl, die Sie einem neuen File zuordnen können.

Erzeugen von Datenfiles

Um mit der Funktion [CRFLD] (*create file/data*) einen Datenfile zu erzeugen, können Sie folgendermaßen vorgehen:

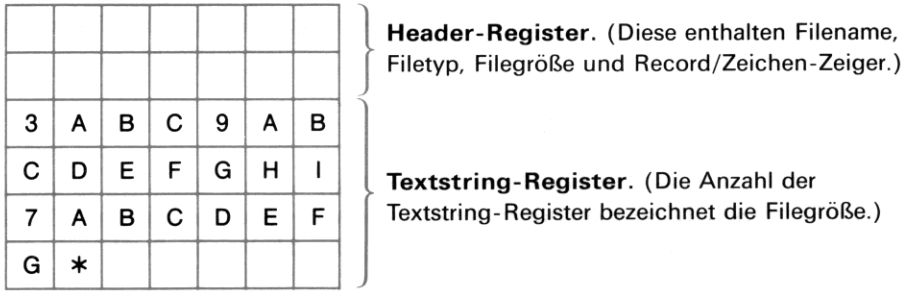
1. Legen Sie den Namen des neuen Files im Alpha-Register ab. Der Name darf aus bis zu sieben beliebigen Zeichen bestehen (Ausnahme: der Filename darf kein Komma enthalten).
2. Legen Sie die Anzahl der Register, die Sie dem File zuordnen wollen, im X-Register ab. (Dabei brauchen Sie die Header-Register nicht zu berücksichtigen.)
3. Führen Sie die Funktion [CRFLD] aus.

Der neue File wird zum momentanen File und der Zeiger ist auf das erste Register positioniert.

Erzeugen von Textfiles

Textfiles werden ähnlich wie Datenfiles erzeugt, aber Sie müssen sich vor Erzeugen des Files überlegen, wieviele Register benötigt werden. Eine ungefähre Abschätzung ist meist ausreichend, da die Filegröße später verändert werden kann. Die in diesem Abschnitt angegebene exakte Formel ist von Nutzen, wenn ein Textfile durch ein Programm erzeugt wird.

Die unten stehende Abbildung zeigt einen Textfile mit drei Records: **ABC**, **ABCDEFGHI** und **ABCDEFG**. Am Anfang jedes Records steht ein zusätzliches Byte, das die Länge des Records angibt (vergleiche XTOA). Ein Record darf bis zu 254 Zeichen enthalten.



Das «*» kennzeichnet das Ende des momentanen Fileinhalts. (Der Dezimalwert dieses Bytes ist 255, daher darf ein Record nicht mehr als 254 Zeichen lang sein.) Dieses Byte und die Bytes zur Kennzeichnung der Recordlänge sind für Sie unsichtbar; wenn Sie den File benutzen, sieht er aus wie in der Abbildung auf Seite 215. Diese zusätzlichen Bytes müssen nur beim Erzeugen eines Files berücksichtigt werden.

Wenn Sie genau wissen, wieviele Records und Zeichen der File enthalten wird, können Sie die benötigte Registeranzahl exakt berechnen.

1. Addieren Sie die Anzahl der Records zu der Zeichenanzahl aller Records.
2. Addieren Sie 1 zu dem Ergebnis.
3. Dividieren Sie das Ergebnis durch 7. Runden Sie einen Rest auf eine ganze Zahl auf.

Nun können Sie mit CRFLAS (*create file/ASCII*) den File erzeugen.

4. Legen Sie den Namen des neuen Files im Alpha-Register ab. Der Name darf aus bis zu sieben beliebigen Zeichen bestehen (Ausnahme: der Name darf kein Komma enthalten).
5. Legen Sie die Anzahl der Register, die Sie dem File zuordnen wollen, im X-Register ab.
6. Führen Sie die Funktion CRFLAS aus.

Der neue File wird zum momentanen File und der Zeiger deutet auf das erste Zeichen des ersten Records.

Ändern der Filegröße

Mit Hilfe der Funktion RESZFL (*resize file*) können Sie die einem Daten- oder Textfile zugeordnete Registeranzahl verändern. Gehen Sie folgendermaßen vor:

1. Stellen Sie sicher, daß der zu ändernde File der momentane File ist.
2. Legen Sie die Anzahl der Register, die Sie dem File zuordnen wollen, im X-Register ab.
3. Führen Sie die Funktion RESZFL aus.

Der ganzzahlige Anteil der Zahl im X-Register bestimmt die neue Größe des momentanen Files. Die Zeigerposition bleibt unverändert.

Vergrößern eines Files. Bevor Sie einen File vergrößern, können Sie durch Ausführen von EMDIR oder EMROOM feststellen, wieviele Register verfügbar sind. Sie können normalerweise zwei Register zu dem Ergebnis addieren, da EMDIR und EMROOM zwei Register für den Header eines neuen Files berücksichtigen, die in diesem Fall nicht benötigt werden. Wenn der Computer jedoch Null als Ergebnis zurückgibt, ist es nicht sicher, ob zwei, ein oder kein Register verbleiben.

Verkleinern eines Files. Wenn Sie einen File um *n* Register verkleinern, geht der Inhalt der letzten *n* Register des Files verloren. Ist im X-Register eine *positive* Zahl spezifiziert, wird RESZFL nur ausgeführt, wenn dadurch kein momentan belegtes Register verloren geht. Ein Register ist belegt, wenn eine der folgenden Bedingungen erfüllt ist:

- Das Register enthält eine von Null verschiedene Zahl (bei Datenfiles).
- Das Register enthält ein Zeichen oder das End-Of-File Byte (bei Textfiles).

Wenn die Zahl im X-Register positiv ist und durch Ausführen von RESZFL belegte Register verlorengehen würden, erscheint die Meldung **FL SIZE ERR** in der Anzeige. Bei Datenfiles können Sie diese Schutzvorkehrung umgehen, indem Sie eine *negative* Zahl im X-Register spezifizieren. In diesem Fall gehen die letzten *n* Register ohne Rücksicht auf ihren Inhalt verloren.

Löschen von Fileinhalten

Die Funktion CLFL (*clear file*) ermöglicht es Ihnen, die in einem File gespeicherten Informationen zu löschen, ohne daß dabei der Inhalt der Header-Register verloren geht.

1. Geben Sie den Namen des zu löschenden Files in das Alpha-Register ein.
2. Führen Sie die Funktion CLFL aus.

Der gelöschte File wird zum momentanen File. Wenn Sie den Inhalt eines Datenfiles gelöscht haben, enthalten alle Register den Wert Null und der Zeiger ist auf das erste Register positioniert. Nach Anwenden von CLFL auf einen Textfile existieren keine Records mehr und der Zeiger ist auf das erste Leer-Byte des Files positioniert.

Zeiger in Daten- und Textfiles

Daten- und Textfiles sind aus einzelnen Registern bzw. Alpha-Strings zusammengesetzt. Die Filezeiger ermöglichen Ihnen den Zugriff auf einzelne Elemente der Files. Dieser Absatz behandelt zuerst den Aufbau von Daten- und Textfiles; anschließend werden die Zeiger-Operationen erläutert.

Aufbau von Datenfiles

Ein Datenfile besteht aus zwei Header-Registern und einem oder mehreren Datenregistern. Der Header enthält den momentanen Wert des Zeigers; dieser Wert bestimmt, auf welches Register als nächstes zugegriffen wird.

Die untenstehende Abbildung zeigt einen Datenfile mit fünf Datenregistern. Beachten Sie, daß der Zeigerwert für das erste Register 000 ist; dieser Wert wird für jedes nachfolgende Register um eins erhöht.

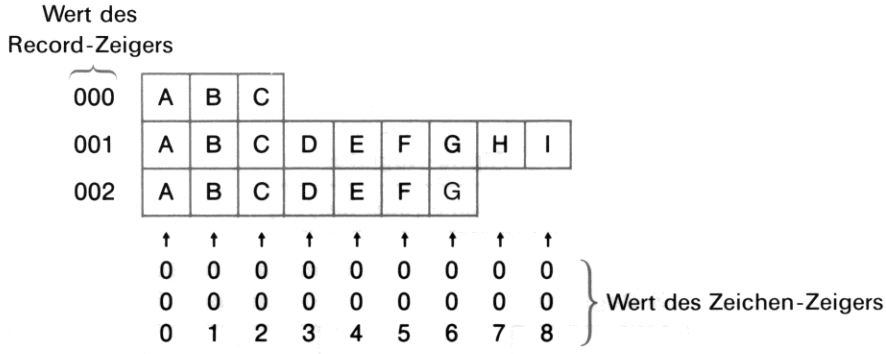
Aufbau von Textfiles

Ein Textfile besteht aus zwei Header-Registern und einem oder mehreren Registern zur Aufnahme Ihrer Records. Jeder Record ist ein Alpha-String aus bis zu 254 Zeichen. Dies bedeutet, daß die einzelnen Records unterschiedlich lang sein können; der Zeiger eines Textfiles (im Header enthalten) besteht daher aus zwei Komponenten:

Record-Zeiger. Der ganzzahlige Anteil eines Textfile-Zeigers ist der Record-Zeiger. Sein Wert weist auf den momentanen Record hin.

Zeichen-Zeiger. Der gebrochene Anteil eines Textfile-Zeigers ist der Zeichen-Zeiger. Sein Wert weist auf das momentane Zeichen des momentanen Records hin. Funktionen, die auf den gesamten Record wirken, ignorieren diesen Teil des Textfile-Zeigers.

Ein Textfile-Zeiger wird als *rrr.ccc* dargestellt, wobei *rrr* der Record-Zeiger und *ccc* der Zeichen-Zeiger ist. Die untenstehende Abbildung zeigt einen Textfile, der drei Records enthält; die Zeigerwerte für die drei A's sind 000.000, 001.000 und 002.000. Die Zeigerwerte für H und I sind 001.007 und 001.008. (Diese Abbildung dient dem besseren Verständnis der Textfile-Zeiger. Die Abbildung auf Seite 212 zeigt, wie der gleiche File in Registern gespeichert wird.)



Zeiger-Operationen

Die Zeiger erlauben Ihnen, auf einzelne Elemente innerhalb eines Daten- bzw. Textfiles zuzugreifen. Der momentane Zeigerwert deutet auf das momentane Register bzw. den momentanen Record und das momentane Zeichen. Sie müssen den Zeiger positionieren, *bevor* Sie eine Funktion auf ein bestimmtes Register (bzw. Record und Zeichen) anwenden können.

Es ist unter Umständen hilfreich, den Zeiger bereits beim ersten Zugriff auf einen File zu positionieren; Sie können dann direkt eine Funktion auf die gewünschten Daten anwenden. Die meisten Funktionen, die den Zeigerwert zum Zugriff verwenden, erhöhen automatisch diesen Wert. Dadurch kann durch einfaches Wiederholen einer Funktion auf aufeinanderfolgende Datenelemente zugegriffen werden.

In einigen Fällen deutet der Zeiger nicht mehr auf ein gespeichertes Datenelement. Wenn eine Funktion zum Beispiel auf das letzte Register eines Datenfiles zugreift und anschließend den Zeiger vorstellt, ist der Zeiger nicht auf ein File-Register positioniert. Wenn Sie diese Funktion wiederholen, zeigt der Computer die Fehlermeldung OUT OF FL an.

Positionieren eines Zeigers. Der HP-41 verfügt über zwei Funktionen zum Verstellen eines Zeigers: [SEEKPTA] (*seek pointer by Alpha*) und [SEEKPT] (*seek pointer*).

Verwenden Sie die Funktion [SEEKPTA], wenn der gewünschte File nicht der momentane File ist.

1. Legen Sie den Namen des Daten- oder Textfiles im Alpha-Register ab.
2. Spezifizieren Sie den gewünschten Zeigerwert im X-Register.
3. Führen Sie die Funktion [SEEKPTA] aus.

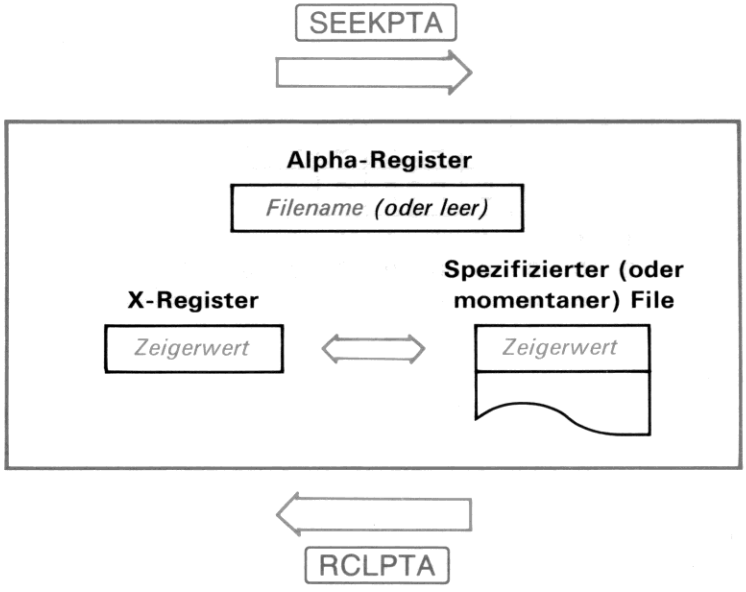
Der im Alpha-Register spezifizierte File wird zum momentanen File; der interne Zeiger ist auf den im X-Register angegebenen Wert positioniert. Bei einem Datenfile positioniert der ganzzahlige Anteil der Zahl im X-Register den Register-Zeiger. Bei einem Textfile positioniert der ganzzahlige Anteil den Record-Zeiger und die ersten drei Ziffern des gebrochenen Anteils positionieren den Zeichen-Zeiger.

Um auf den momentanen File zuzugreifen, können Sie folgendermaßen vorgehen:

- Legen Sie den Zeigerwert im X-Register ab und führen Sie [SEEKPT] aus.

  oder

- Löschen Sie das Alpha-Register, legen Sie den Zeigerwert im X-Register ab und führen Sie [SEEKPTA] aus.



Überprüfen eines Zeigerwerts. Der HP-41 verfügt über zwei Funktionen zum Abrufen des momentanen Zeigerwerts: RCLPTA (*recall pointer by Alpha*) und RCLPT (*recall pointer*). Wenn der gewünschte File nicht der momentane File ist, können Sie folgendermaßen vorgehen:

1. Legen Sie den Filenamen im Alpha-Register ab.
2. Führen Sie die Funktion RCLPTA aus.

Der im Alpha-Register spezifizierte File wird zum momentanen File. Der momentane Zeigerwert wird im X-Register abgelegt; dabei wird der Stack angehoben (sofern der Stack Lift nicht gesperrt ist). Bei einem Datenfile repräsentiert der zurückgegebene Wert den Register-Zeiger. Bei einem Textfile hat dieser Wert das Format *rrr.ccc*; dabei stellt *rrr* den Record-Zeiger und *ccc* den Zeichen-Zeiger dar.

Sie können den momentanen Zeigerwert des momentanen Files überprüfen, indem Sie die Funktion RCLPT ausführen; alternativ dazu können Sie auch das Alpha-Register löschen und dann die Funktion RCLPTA ausführen. Der Zeigerwert wird, wie oben erläutert, im X-Register abgelegt. Die Funktionen RCLPTA und RCLPT können auch auf Programmfiles angewandt werden (siehe Seite 211).

Operationen mit Datenfiles

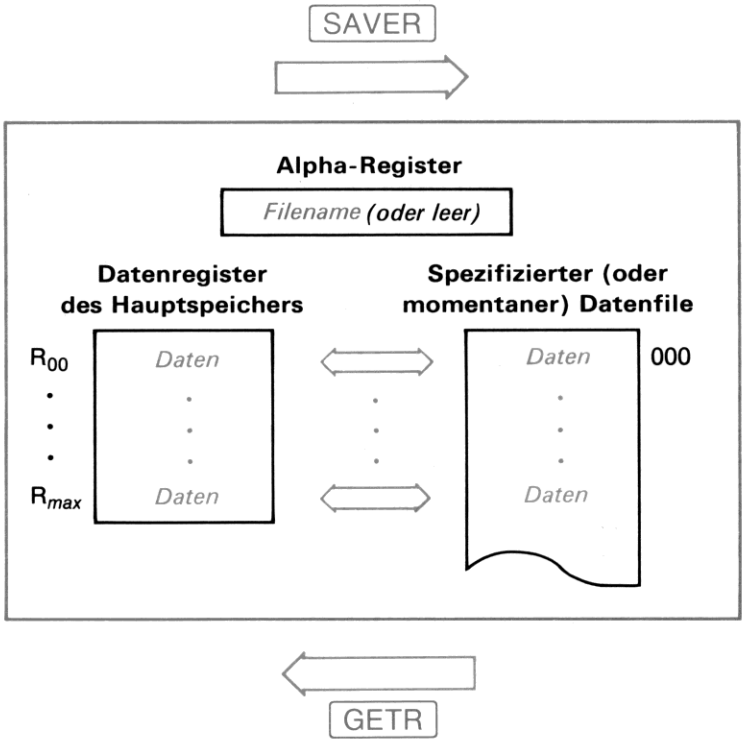
Datenfiles stellen sowohl eine Erweiterung des Hauptspeichers als auch eine Alternative zum Hauptspeicher dar. Um gewisse Operationen wie z.B. Registerarithmetik oder indirekte Adressierung auszuführen, müssen Sie Ihre Daten zunächst in den Hauptspeicher zurückladen. Andererseits gibt es aber auch erweiterte Speicheroperationen, die nicht auf Daten im Hauptspeicher angewandt werden können. Dieser Absatz erläutert zuerst die Funktionen, die auf Datenregister des Hauptspeichers zugreifen; im Anschluß daran werden die Funktionen erläutert, die auf das X-Register zugreifen.

Zugriff auf alle Datenregister

Wenn Sie alle Datenregister des Hauptspeichers in einen Datenfile kopieren wollen, führen Sie die Funktion SAVER (*save registers*) aus. Um einen Datenfile (oder Teile hiervon) in den Hauptspeicher zu kopieren, führen Sie die Funktion GETR (*get registers*) aus. Beide Funktionen übertragen Daten zwischen Registern, deren Adresse und Zeigerwert übereinstimmt: zwischen R_{00} des Hauptspeichers und Register 000 des Datenfiles, zwischen R_{01} und Register 001 usw. Die momentane Zeigerposition hat keinen Einfluß auf SAVER und GETR. Beide Funktionen positionieren den Zeiger auf das Register nach dem zuletzt aufgerufenen Fileregister.

Kopieren der Inhalte aller Datenregister des Hauptspeichers. Um den Inhalt jedes Datenregisters des Hauptspeichers in das korrespondierende Register eines Datenfiles zu kopieren, können Sie folgendermaßen vorgehen: Legen Sie den Namen des Datenfiles im Alpha-Register ab und führen Sie die Funktion SAVER aus. Wenn der gewünschte File bereits der momentane File ist, genügt es, das Alpha-Register vor Ausführen der Funktion SAVER zu löschen.

Wenn weniger Zielregister (im Datenfile) als Quellregister (im Hauptspeicher) vorhanden sind, wird die Kopieroperation nicht ausgeführt; in der Anzeige erscheint die Fehlermeldung END OF FL. Mit der Funktion SAVER können nur alle Hauptspeicher-Register auf einmal kopiert werden.



Kopieren von Daten in alle Datenregister des Hauptspeichers. Um die Register eines Datenfiles in den Hauptspeicher zu kopieren, ist der Name des Datenfiles im Alpha-Register abzulegen und danach die Funktion GETR auszuführen. Wenn der gewünschte File bereits der momentane File ist, genügt es, das Alpha-Register vor Ausführen der Funktion GETR zu löschen.

Die Funktion GETR überträgt Daten zwischen korrespondierenden Registern, beginnend mit dem Fileregister 000 und dem Hauptspeicher-Register R_{00}. Die Kopieroperation wird beendet, wenn keine Register des einen oder anderen Typs verbleiben. (Die Funktion SAVER verursacht im Gegensatz zu GETR einen Fehler, wenn nicht auf alle Hauptspeicher-Register zugegriffen wird.)

Zugriff auf einen Block von Datenregistern

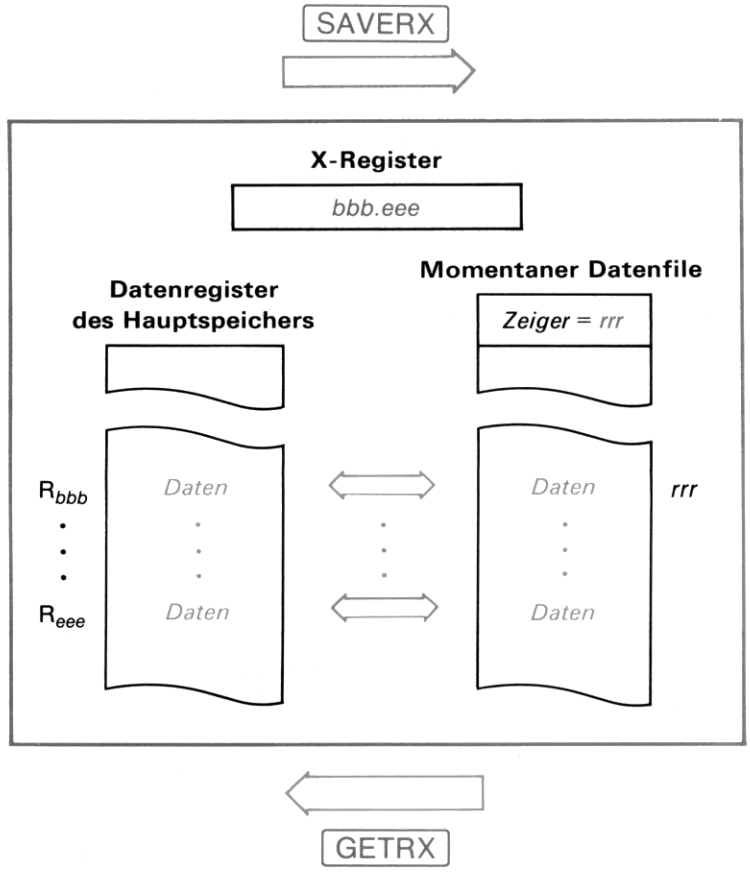
Um Daten zwischen einem Block von Hauptspeicher-Registern und einem gleich großen Registerblock eines Datenfiles zu transferieren, stehen Ihnen die Funktionen SAVERX (*save registers by X*) und GETRX (*get registers by X*) zur Verfügung. Der Datenfile muß der momentane File sein. Die zwei Registerblöcke sind folgendermaßen definiert:

- Der Block von Hauptspeicher-Registern wird durch eine im X-Register abgelegte Steuergröße spezifiziert. Diese Steuerzahl hat das Format *bbb.eee*; dabei ist R_{bbb} das *erste* Register des Blocks und R_{eee} das *letzte* Register des Blocks.
- Der Registerblock des Datenfiles ist durch den momentanen Zeigerwert und die Steuerzahl im X-Register definiert. Das momentane Register ist das erste Register des Blocks; der Registerblock enthält *eee* − *bbb* Register.

Die Funktionen SAVERX und GETRX übertragen Daten zwischen R_{bbb} im Hauptspeicher und dem momentanen Register im Datenfile, zwischen R_{bbb+1} und dem nächsten Datenfile-Register, usw. Beide Funktionen positionieren den Zeiger auf das Register nach dem zuletzt aufgerufenen Fileregister. Wenn der Block von Fileregistern (gezählt vom momentanen Register bis zum Ende des Files) kleiner ist als der spezifizierte Block von Datenregistern, wird die Kopieroperation nicht ausgeführt und in der Anzeige erscheint die Fehlermeldung **END OF FL**.

Kopieren von Daten aus einem Registerblock im Hauptspeicher. Verwenden Sie die Funktion SAVERX, um den Inhalt eines Registerblocks im Hauptspeicher in einen Datenfile zu kopieren.

1. Stellen Sie sicher, daß der Zielfile der momentane File ist und daß der Zeiger auf dasjenige Register positioniert ist, in dem das erste Datenelement abgelegt werden soll.
2. Spezifizieren Sie die Steuerzahl *bbb.eee* im X-Register.
3. Führen Sie die Funktion SAVERX aus.



Kopieren von Daten in einen Registerblock des Hauptspeichers. Verwenden Sie die Funktion GETRX, um den Inhalt eines Registerblocks aus einem Datenfile in den Hauptspeicher zu kopieren.

1. Stellen Sie sicher, daß der Quellfile der momentane File ist und daß der Zeiger auf dasjenige Register positioniert ist, aus dem das erste Datenelement abgerufen werden soll.
2. Spezifizieren Sie die Steuerzahl im X-Register.
3. Führen Sie die Funktion GETRX aus.

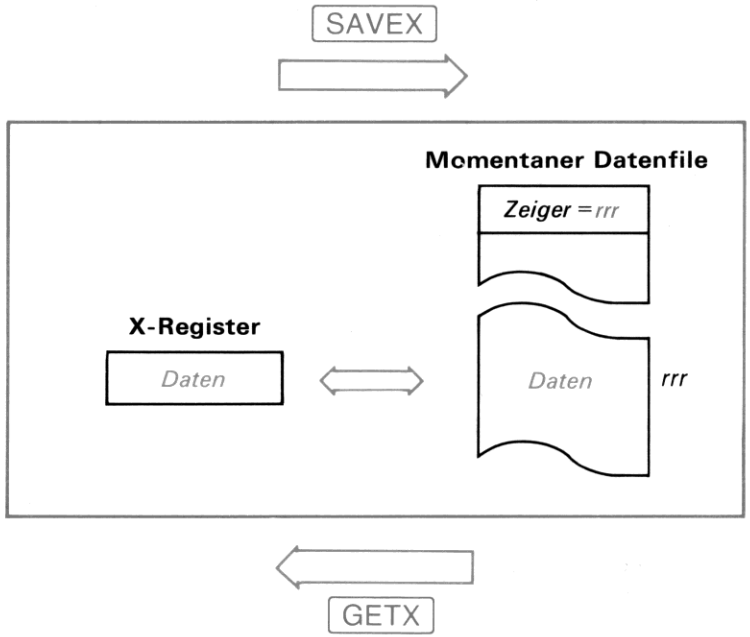
Zugriff auf das X-Register

Um Daten zwischen dem X-Register und einem Datenfile zu transferieren, stehen Ihnen die Funktionen SAVEX (*save X*) und GETX (*get X*) zur Verfügung. Der Datenfile muß der momentane File sein und das Register, auf das Sie zugreifen wollen, muß das momentane Register sein. Beide Funktionen positionieren den Zeiger auf das Register nach dem aufgerufenen Datenfile-Register.

Speichern des Inhalts des X-Registers. Verwenden Sie die Funktion SAVEX, um den Inhalt des X-Registers in ein Datenfile-Register zu kopieren.

1. Stellen Sie sicher, daß der Zielfile der momentane File ist und daß das Zielregister das momentane Register ist.
2. Führen Sie die Funktion SAVEX aus.

Der Inhalt des X-Registers bleibt unverändert und der vorhergehende Inhalt des Zielregisters wird überschrieben (vergleiche die Funktion STO).



Zurückrufen von Daten in das X-Register. Verwenden Sie die Funktion GETX, um den Inhalt eines Datenfile-Registers in das X-Register zurückzurufen.

1. Stellen Sie sicher, daß der Quellfile der momentane File ist und daß das Quellregister das momentane Register ist.
2. Führen Sie die Funktion GETX aus.

Der Inhalt des Quellregisters bleibt unverändert; der Stack wird angehoben und der Inhalt des T-Registers geht verloren, wenn der Stack Lift nicht gesperrt ist (vergleiche die Funktion [RCL]).

Beispiel. Das Beispiel geht davon aus, daß ein Programm den Inhalt des Stacks retten muß, um danach die Stack-Register zur Einstellung eines Alarms zu verwenden; später soll wieder der vorhergehende Zustand des Stacks hergestellt werden. Dies wird durch die drei unten erläuterten Routinen erreicht.

Die erste Routine erzeugt den Datenfile **STACK**; diese Routine wird am Anfang des Programms ausgeführt (d.h. bevor der Stack die zu speichernden Werte enthält).

01 TSTACK	Legt den Filenamen im Alpha-Register ab.
02 4	Legt die Anzahl der Register im X-Register ab.
03 CRFLD	Erzeugt den Datenfile **STACK**, der vier Register enthält.

Die zweite Routine kopiert die im Stack abgelegten Zahlen in den File. Nach Ausführen dieser Routine können die Alarmparameter in den Stack eingegeben werden.

01 TSTACK	Legt den Filenamen im Alpha-Register ab.
02 STO L	Speichert den Inhalt des X-Registers im LAST X Register.
03 CLX	Legt Null im X-Register ab (ohne dabei den Stack anzuheben).
04 SEEKPTA	Deklariert **STACK** als momentanen File und Register 000 als momentanes Register.
05 X<>L	Legt den im LAST X Register zwischengespeicherten Wert wieder im X-Register ab (ohne dabei den Stack anzuheben).
06 R↑ **07 SAVEX** **08 R↑** **09 SAVEX** **10 R↑** **11 SAVEX** **12 R↑** **13 SAVEX**	Diese Schritte verschieben den Stack nach oben und kopieren den jeweiligen Inhalt des X-Registers in ein Register des Files **STACK**. Die Zahl *t* (die sich anfangs im T-Register befand) wird zuerst kopiert; dann werden *z*, *y* und *x* kopiert. Der Zeiger wird automatisch vorgestellt; dadurch wird t in 000, *z* in 001, *y* in 002 und *x* in 003 gespeichert. Diese Reihenfolge vereinfacht die dritte Routine, die die gespeicherten Werte wieder in den Stack zurücklädt.

Die dritte Routine wird nach der Alarmeinstellung ausgeführt; die gespeicherten Zahlen werden wieder in den Stack zurückgeladen.

01 TSTACK	Legt den Filenamen im Alpha-Register ab.
02 CLX	Legt Null im X-Register ab.
03 SEEKPTA	Deklariert **STACK** als momentanen File und Register 000 als momentanes Register.

04 GETX	Ruft *t* in das X-Register zurück, hebt den Stack an und positioniert den Zeiger auf das nächste Register.
05 GETX	Ruft *z* in das X-Register zurück, hebt den Stack an und positioniert den Zeiger auf das nächste Register.
06 GETX	Ruft *y* in das X-Register zurück, hebt den Stack an und positioniert den Zeiger auf das nächste Register.
07 GETX	Ruft *x* in das X-Register zurück, hebt den Stack an und positioniert den Zeiger auf das nächste Register.

Operationen mit Textfiles

Die meisten hier erläuterten Operationen übertragen Daten zwischen einem Textfile und dem Alpha-Register. Alle im Alpha-Register abgelegten Informationen (Zeichen des Alpha-Tastenfelds, durch XTOA erzeugte Bytes) können durch diese Operationen in Textfiles gespeichert und wieder abgerufen werden.

Einfache Operationen sind leichter mit Hilfe des Texteditors ED (siehe Abschnitt 14) auszuführen. Wenn Sie einen Textfile lediglich einsehen oder mit Standardzeichen editieren wollen, können Sie den Editor verwenden, der Ihnen die jeweilige Änderung unmittelbar anzeigt. Programme zur *automatischen* Filemanipulation müssen jedoch die hier erläuterten Funktionen verwenden.

Überprüfen der verfügbaren Bytes

Bevor Sie neue Daten in einem Textfile speichern, können Sie durch Ausführen der Funktion ASROOM (*ASCII room*) überprüfen, wieviele Bytes noch verfügbar sind. Der gewünschte Textfile muß der momentane File sein. Die Funktion ASROOM legt die Anzahl der verfügbaren Bytes im X-Register ab; dabei wird der Stack angehoben (sofern der Stack Lift nicht gesperrt ist). Jeder neue Record belegt ein zusätzliches Byte zur Speicherung der Zeichenzahl. Wenn der File nicht über genügend Speicherplatz verfügt, können Sie ihn durch die Funktion RESZFL (Seite 213) vergrößern.

Record-Operationen

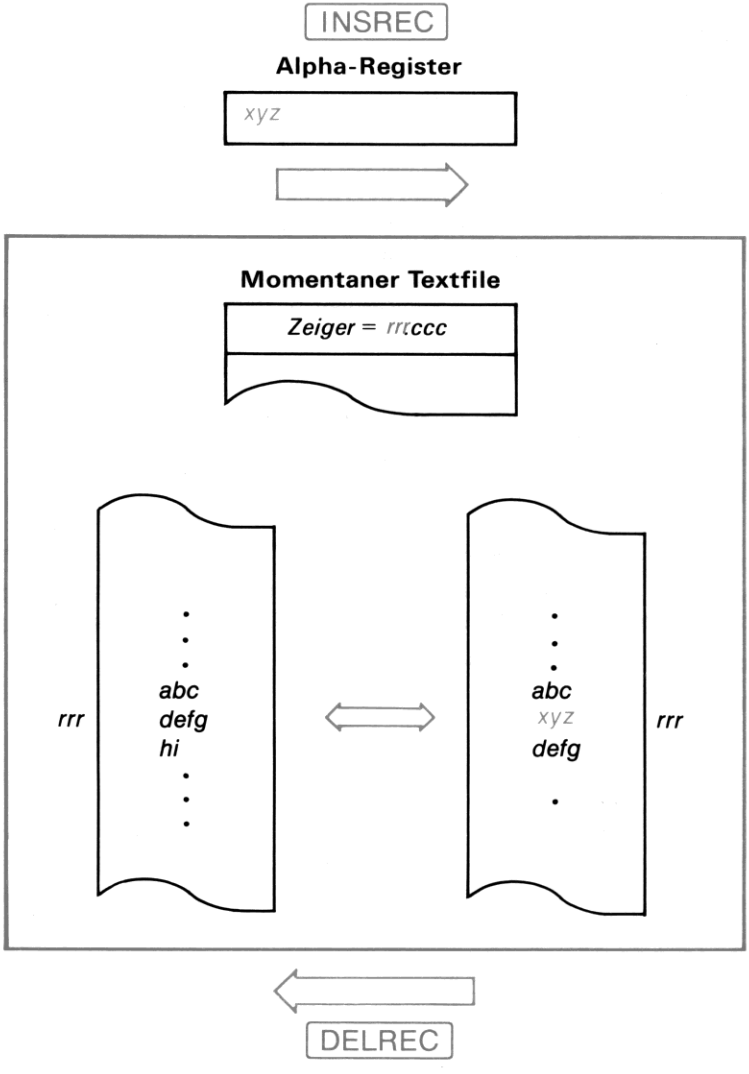
Der HP-41 verfügt über drei Funktionen zur Recordmanipulation: die Funktion APPREC (*append record*) hängt einen Record an einen Textfile an; die Funktion INSREC (*insert record*) fügt einen Record ein; die Funktion DELREC (*delete record*) löscht einen Record. Diese Funktionen wirken nur auf den momentanen File. Die Funktionen APPREC und INSREC erzeugen aus dem Inhalt des Alpha-Registers einen neuen Record; der Zeiger wird auf das erste Byte nach dem letzten Zeichen dieses neuen Records gestellt. Dadurch können Sie direkt anschließend Zeichen an den neuen Record anfügen.

Anhängen eines Records. Um mit der Funktion APPREC einen Record an einen Textfile anzuhängen, können Sie wie folgt vorgehen:

1. Stellen Sie sicher, daß der gewünschte File der momentane File ist.
2. Legen Sie die Daten für den neuen Record im Alpha-Register ab.
3. Führen Sie die Funktion APPREC aus.

Einfügen eines Records. Um mit der Funktion INSREC einen Record in einen Textfile einzufügen, folgen Sie den nachstehenden Anleitungen.

1. Stellen Sie sicher, daß der gewünschte Textfile der momentane File ist und daß der Record-Zeiger auf die gewünschte Position des neuen Records gestellt ist. (Die Position des Zeichen-Zeigers spielt keine Rolle.)
2. Legen Sie die Daten für den neuen Record im Alpha-Register ab.
3. Führen Sie die Funktion INSREC aus.



Löschen eines Records. Mit der Funktion [DELREC] können Sie wie folgt einen Record aus einem Textfile löschen:

1. Stellen Sie sicher, daß der gewünschte Textfile der momentane File ist und daß der zu löschende Record der momentane Record ist. (Die Position des Zeichen-Zeigers spielt keine Rolle.)
2. Führen Sie die Funktion [DELREC] aus.

Der Wert des Record-Zeigers wird nicht verändert; der momentane Record ist jetzt der Record, der nach dem gelöschten Record stand. Der Zeichen-Zeiger ist auf 000 gesetzt.

Zeichen-Operationen

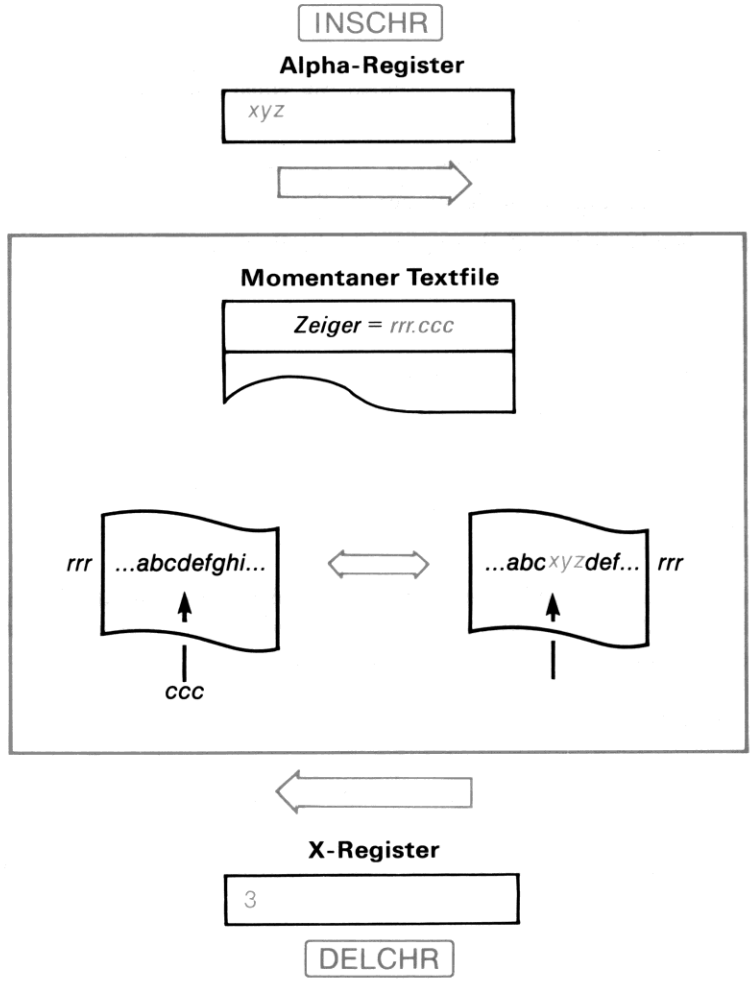
Der HP-41 verfügt über drei Funktionen zur Zeichenmanipulation: die Funktion [APPCHR] (*append characters*) hängt Zeichen an einen Record an; die Funktion [INSCHR] (*insert characters*) fügt Zeichen ein; die Funktion [DELCHR] (*delete characters*) löscht Zeichen. Diese Funktionen wirken nur auf dem momentanen Record des momentanen Textfiles. Die Funktionen [APPCHR] und [INSCHR] tragen den Inhalt des Alpha-Registers in den Record ein; der Zeiger wird auf das Byte nach dem letzten hinzugefügten Zeichen positioniert. Dadurch können Sie jede der beiden Funktionen sofort wiederholen, nachdem Sie neue Daten im Alpha-Register abgelegt haben.

Anhängen von Zeichen. Um mit der Funktion [APPCHR] Zeichen an einen Record anzuhängen, können Sie wie folgt vorgehen:

1. Stellen Sie sicher, daß der gewünschte Textfile der momentane File ist und daß der gewünschte Record der momentane Record ist. (Die Position des Zeichen-Zeigers spielt keine Rolle.)
2. Legen Sie die anzuhängenden Zeichen im Alpha-Register ab.
3. Führen Sie die Funktion [APPCHR] aus.

Einfügen von Zeichen. Zum Einfügen von Zeichen in einen Record ist die Funktion [INSCHR] zu verwenden:

1. Stellen Sie sicher, daß der gewünschte Textfile der momentane File ist und daß der gewünschte Record der momentane Record ist. Der Zeichen-Zeiger ist auf die gewünschte Einfügestelle zu positionieren.
2. Legen Sie die einzufügenden Zeichen im Alpha-Register ab.
3. Führen Sie die Funktion [INSCHR] aus.



Hinweis: Nach Ausführung der Funktion INSCHR würde der Zeichen-Zeiger in der obigen Abbildung weiterhin auf *d* zeigen.

Löschen von Zeichen. Die Funktion DELCHR dient zum Löschen von Zeichen aus einem Record:

1. Stellen Sie sicher, daß der gewünschte Textfile der momentane File ist und daß der gewünschte Record der momentane Record ist. Der Zeichen-Zeiger ist auf das erste zu löschende Zeichen zu positionieren.
2. Legen Sie die Anzahl der zu löschenden Zeichen im X-Register ab.
3. Führen Sie die Funktion DELCHR aus.

Der Wert des Zeichen-Zeigers wird nicht verändert; das momentane Zeichen ist jetzt das Zeichen, das nach dem zuletzt gelöschten Zeichen stand.

Durchsuchen eines Files

Mit Hilfe der Funktion POSFL (*position in file*) können Sie einen File nach einer bestimmten Zeichenkette durchsuchen.

1. Stellen Sie sicher, daß der zu durchsuchende File der momentane File ist.
2. Positionieren Sie den Zeiger auf die Stelle, an der die Suche beginnen soll.
3. Legen Sie den zu suchenden String (Zielstring) im Alpha-Register ab.
4. Führen Sie die Funktion POSFL aus.

Zielstring wird gefunden:

- Der Zeiger wird auf das erste Zeichen des Strings positioniert.
- Dieser Zeigerwert wird im X-Register abgelegt.

Zielstring wird nicht gefunden:

- Die Zeigerposition wird nicht verändert.
- Im X-Register wird die Zahl –1 abgelegt.

In beiden Fällen bleibt der Inhalt des Alpha-Registers erhalten; der Stack wird angehoben (sofern der Stack Lift nicht gesperrt ist). Wenn der Zielstring gefunden wurde, können Sie den dazugehörigen Record mit Hilfe der nachfolgend erläuterten Funktionen in das Alpha-Register zurückrufen.

Kopieren von Daten in das Alpha-Register

Die Funktionen GETREC (*get record*) und ARCLREC (*Alpha recall record*) kopieren Zeichen aus einem Record eines Textfiles in das Alpha-Register. Die beiden Funktionen unterscheiden sich nur in ihren Auswirkungen auf das Alpha-Register.

- Die Funktion GETREC löscht das Alpha-Register bevor die Kopieroperation ausgeführt wird.
- Die Funktion ARCLREC hängt die kopierten Zeichen an den vorhergehenden Inhalt des Alpha-Registers an.

Ansonsten bestehen keine Unterschiede zwischen den beiden Funktionen.

- Beide Funktionen beginnen die Kopieroperation mit dem momentanen Zeichen des momentanen Records im momentanen File.
- Beide Funktionen kopieren aufeinanderfolgende Zeichen, bis entweder das letzte Zeichen des Records kopiert ist oder das Alpha-Register voll ist. (Dies bedeutet, daß ARCLREC keinerlei Auswirkungen hat, wenn das Alpha-Register schon vor Funktionsausführung voll ist.)
- Beide Funktionen löschen Flag 17, wenn das letzte Zeichen des Records kopiert wird; Flag 17 wird gesetzt, wenn das letzte Zeichen nicht kopiert wird. (Bei Verwendung von bestimmten HP-IL Peripheriegeräten bestimmt der Status dieses Flags, ob aufeinanderfolgende Ausgaben als getrennte Zeilen betrachtet werden.)
- Beide Funktionen positionieren den Zeichen-Zeiger auf das Zeichen hinter dem zuletzt kopierten Zeichen.

Die Funktionen GETREC und ARCLREC wirken nur auf den momentanen File. Stellen Sie vor Funktionsausführung den Zeiger auf die gewünschte Position. Sie können durch Abfragen von Flag 17 feststellen, ob der Record noch mehr zu kopierende Zeichen enthält.

Zugriff auf Massenspeicher-Files

Sie können permanente Kopien Ihrer Textfiles erstellen, sofern Ihr System eine über ein HP-IL Modul HP 82160A gesteuerte Massenspeichereinheit enthält. Die Funktion SAVEAS (*save ASCII*) kopiert einen Textfile aus dem erweiterten Speicher in den Massenspeicher; die Funktion GETAS (*get ASCII*) kopiert einen Textfile aus dem Massenspeicher in den erweiterten Speicher. Beim Speichern und Zurückladen von Files gelten die folgenden Regeln:

- Der Zielfile muß existieren, bevor der Quellfile kopiert werden kann. Mit der Funktion CREATE (eine Funktion des HP-IL Moduls) können Sie einen Zielfile im Massenspeicher erzeugen. Die Funktion CREATE erzeugt einen Datenfile; wenn Sie die Funktion SAVEAS ausführen, wird dieser Datenfile in einen Textfile umgewandelt.
- Wenn der Zielfile kleiner als der Quellfile ist, unterbricht ein END OF FL Fehler die Kopieroperation, sobald alle Zielfile-Register belegt sind. Um eine unvollständige Kopie des Quellfiles zu erstellen, können Sie Flag 25 (Ignorieren von Fehlern) setzen.
- Spezifizieren Sie die Namen von Quell- und Zielfile im Alpha-Register; die Filenamen sind durch ein Komma zu trennen.

Alpha-Register

Quellfilename,Zielfilename

Wenn beide Files den gleichen Namen haben, genügt es, nur diesen Namen im Alpha-Register abzulegen.

Abschnitt 14

Der Texteditor

Inhalt

Einführung 228
Die Texteditor-Anzeige 229
Das Fenster 229
Indikatoren 230
Texteditor-Operationen [ED] 230
Ein- und Ausschalten des Texteditors 230
Der Einschaltzustand 230
Das Texteditor-Tastenfeld 230
Das numerische Tastenfeld 232
Tasten zur Zeichenkontrolle 232
Tasten zur Recordkontrolle 232
Fehler 233
Verwenden der Funktion [ED] in Programmen 233

Einführung

Dieser Abschnitt dient zur Erläuterung des Texteditors – eines Werkzeugs zur Bearbeitung von Textfiles (ASCII-Files). Dabei wird unterstellt, daß Sie mit den in Abschnitt 13 beschriebenen Textfile-Operationen vertraut sind: Erzeugen, Löschen, Speicheranforderungen.

Das Positionieren des Record/Zeichen-Zeigers war wichtiger Bestandteil der in Abschnitt 13 beschriebenen Operationen (z.B. [APPREC], [APPCHR] und [INSCHR]). Die genannten Operationen benutzen das Alpha-Register als ein Zwischen-Medium beim Zugriff auf einen Textfile. *Dabei ist zu beachten, daß Sie über diese Operationen nicht direkt auf einen Textfile zugreifen können.*

Mit dem Texteditor können Sie einen Textfile direkt betrachten. Die Funktion [ED] erlaubt es Ihnen, Textfiles einzugeben, zu editieren und anzuzeigen, ohne einen Umweg über das Alpha-Register in Kauf nehmen zu müssen. Der Texteditor vereinfacht zwar Textfile-Operationen beträchtlich, er hat aber seine Grenzen: die Operationen von [ED] sind nicht programmierbar (im Gegensatz zu Funktionen wie [DELCHR]), und Sie können nur die auf dem Alpha-Tastenfeld direkt vorhandenen Zeichen eingeben (im Gegensatz zu [APPCHR] oder [INSREC]).*†

Der Texteditor wurde ausführlich in Abschnitt 8 («Speichern von Text, Daten und Programmen in Files») beschrieben. Der vorliegende Abschnitt ist daher sehr kurz gehalten und dient mehr dem Nachschlagen. Des weiteren finden Sie auch einen Eintrag für [ED] im Funktionsverzeichnis unter «Funktionen des erweiterten Speichers», Seite 426.

Die Texteditor-Anzeige

Das Fenster

Nach Ausführen der Funktion [ED] wird die Anzeige zu einem Fenster, das einen Ausschnitt (12 Zeichen) eines Records des spezifizierten Textfiles zeigt. Der Cursor (ein Unterstreichungszeichen) blinkt abwechselnd mit einem Zeichen oder einer Leerstelle. Die Position des Cursors und der angezeigte Record werden durch den momentanen Wert des Record/Zeichen-Zeigers bestimmt.

Die Recordnummer. Zur Orientierung innerhalb des Files finden Sie am linken Ende jedes Records eine zwei- bis dreistellige Recordnummer. Diese Zahl ist nicht Teil des Records.

Kennung für leeren Record (^{T}). Wenn ein leerer Record angezeigt wird, steht hinter der Recordnummer ein «hochgestelltes T» (^{T}).** Dieses Symbol erscheint sowohl in einem neu erzeugten Record als auch in einem Record, aus dem alle Zeichen gelöscht wurden.

Interpunktion. Ein Satzzeichen («.», «,» oder «:») erscheint am rechten Rand einer regulären Zeichenposition und zählt daher nicht als eine der 12 Zeichenstellen, sofern nicht zwei oder mehr Satzzeichen hintereinander stehen. (Aufeinanderfolgende Satzzeichen sind durch eine Zeichenbreite getrennt.) Der Cursor ist ein reguläres Zeichen (ein Unterstreichungszeichen) und kann daher nicht die gleiche Position wie ein Satzzeichen einnehmen. Wenn der Record/Zeichen-Zeiger auf ein Satzzeichen positioniert ist, blinkt nur das Satzzeichen und der Cursor wird nicht angezeigt.

Der Cursor. Der Cursor und damit auch das momentane Zeichen stehen immer ungefähr in der Mitte des Fensters (außer, wenn der Anfang oder das Ende eines Records angezeigt wird). Der Cursor kann eine Stelle über das rechte Ende des Records hinausgeschoben werden; wenn der Cursor sich in dieser Position befindet, können Sie Zeichen an den Record anhängen.

---

* Nur die Funktion [ED], die den Texteditor einschaltet, ist programmierbar. Alle Editor-Operationen selbst sind jedoch nicht programmierbar. Siehe «Verwenden der Funktion [ED] in Programmen» auf Seite 233.

† Um ein Sonderzeichen (z.B. $) in einen Textfile einzugeben, ist wie folgt vorzugehen: Legen Sie das Zeichen mit Hilfe der Funktion [XTOA] im Alpha-Register ab und kopieren Sie es mit [APPCHR] oder [INSCHR] in den Textfile.

** Der Indikator für einen leeren Record ist eine Art Scheinzeichen. Er verhält sich wie ein Zeichen, mit dem Unterschied, daß es aus einem sonst leeren Record nicht gelöscht werden kann. Das Symbol «^{T}» belegt ein Byte des Speichers; ein leerer Record ist also – strenggenommen – nicht leer. (Die Funktion [ASROOM] berücksichtigt zwei Bytes für jeden leeren Record: ein Byte zur Speicherung der Zeichenzahl und ein Byte für den Indikator «^{T}».)

Indikatoren

In Verbindung mit dem Texteditor sind die in der Anzeige erscheinenden Indikatoren neu definiert:

- **1** deutet an, daß der Einfügungs-Modus aktiviert ist (siehe Seite 232).
- **ALPHA** bedeutet, daß das reguläre Alpha-Tastenfeld (einschließlich der Texteditor-Kontrolltasten) eingeschaltet ist. Wenn **ALPHA** nicht angezeigt wird, ist das numerische Tastenfeld eingeschaltet (siehe Seite 232).
- **SHIFT** und **BAT** haben die übliche Bedeutung.
- Alle anderen Indikatoren verschwinden beim Einschalten des Editors aus der Anzeige; ihr Status wird vom Computer gespeichert. Beim Ausschalten des Editors wird wieder der ursprüngliche Zustand hergestellt.

Texteditor-Operationen ([ED])

Ein- und Ausschalten des Texteditors

Um mit dem Texteditor auf einen Textfile zuzugreifen, ist folgendermaßen vorzugehen:

1. Wenn der File noch nicht existiert, ist mit der Funktion [CRFLAS] ein File zu erzeugen, dessen Name und Größe Sie bestimmen können (siehe Seite 212).
2. Geben Sie den Namen des gewünschten Files in das Alpha-Register ein und führen Sie die Funktion [ED] (*editor*) aus. Ein leeres Alpha-Register spezifiziert den momentanen File.
3. In der Anzeige erscheint der momentane Record und der Cursor ist auf das momentane Zeichen positioniert (bestimmt durch den Record/Zeichen-Zeiger). Ein neuer File beginnt mit einem leeren Record (000).

Drücken Sie die Modustaste [ON] ([EXIT] auf dem [ED] Tastenfeld), um den Texteditor auszuschalten. Der Computer schaltet den Texteditor *automatisch* aus, wenn versucht wird, mehr als 254 Zeichen (die Grenze) in einem Record zu speichern oder wenn für einige Minuten keine Eingabe erfolgt ist. Nach Ausschalten des Texteditors haben alle Indikatoren wieder ihre ursprüngliche Bedeutung.

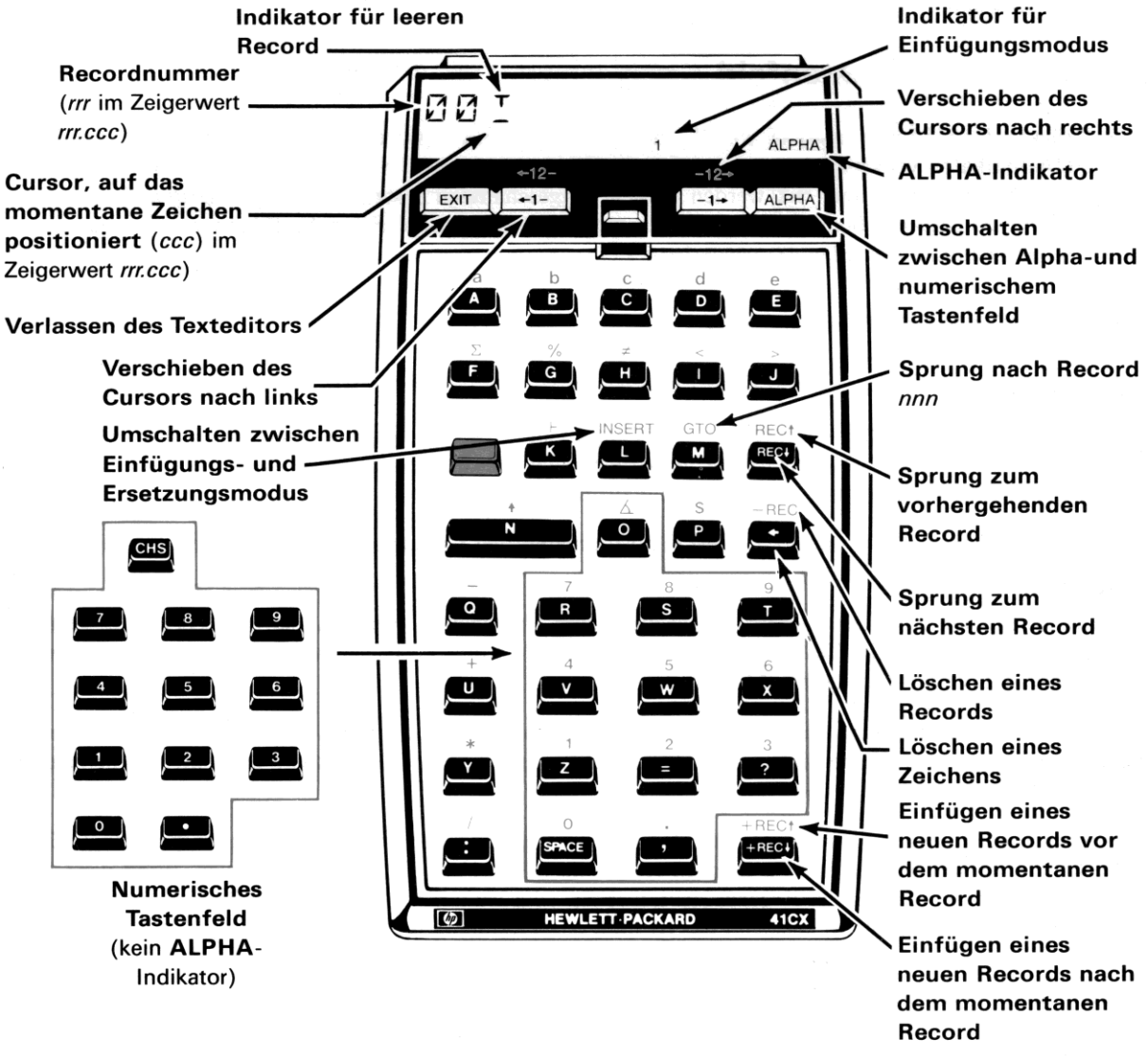
Der Einschaltzustand

Wenn Sie den Texteditor über das Tastenfeld einschalten, ist das Alpha-Tastenfeld aktiviert (**ALPHA**-Indikator wird angezeigt) und der Computer befindet sich im Ersetzungs-Modus (neue Zeichen überschreiben die vorhandenen Zeichen). Wenn der Texteditor von einem Programm eingeschaltet wird, bleibt der vor Ausführen der Funktion [ED] bestehende Status des Alpha-Flags (48) erhalten. Dadurch kann das Programm kontrollieren, welches Tastenfeld eingeschaltet ist.

Das Texteditor-Tastenfeld

Eine kommentierte Illustration des Texteditor ([ED]) Tastenfelds ist auf Seite 231 abgebildet. Weitere Abbildungen des Texteditor-Tastenfelds finden Sie auf der Rückseite Ihres HP-41 und in der Kurzanleitung. Dieses Tastenfeld verfügt sowohl über die alphanumerischen Zeichentasten als auch über die Kontrolltasten zur Ausführung der Editierungs-Funktionen.

Das Texteditor-Tastenfeld



Der Zeichensatz des Texteditor-Tastenfelds ist identisch mit dem des Alpha-Tastenfelds (einschließlich der umgeschalteten Zeichen und der Ziffern). Wenn Sie den Texteditor über das Tastenfeld einschalten, ist dieser Zeichensatz wirksam.

Das numerische Tastenfeld

Durch Drücken der Modustaste [ALPHA] wird das numerische Tastenfeld (siehe Tastenfeld-Abbildung) aktiviert und der **ALPHA**-Indikator verschwindet aus der Anzeige. Das numerische Tastenfeld erleichtert die Eingabe von Ziffern: die Zifferntasten, [.] und [CHS] werden zu nicht umgeschalteten Erstzeichen.* *Der Zugriff auf alle anderen Zeichentasten wird gesperrt.* Beim Verlassen des Texteditor-Modus wird der vor Ausführen von [ED] bestehende Status des Alpha-Indikators (Flag 48) wiederhergestellt.

Tasten zur Zeichenkontrolle

Die Positionen der hier beschriebenen Kontrolltasten können Sie der Tastenfeld-Abbildung entnehmen.

Cursor-Kontrolle. [←1] ([USER]), [1→] ([PRGM]), [←12] ([■] [USER]) und [12→] ([■] [PRGM]). Die Cursorposition repräsentiert den Record/Zeichen-Zeiger. Sie können den Cursor nicht nach links über den Recordanfang und nicht um mehr als eine Stelle über den rechten Rand des Records hinausschieben.

Die Tastenfolgen [REC↓] [REC↑] oder [REC↑] [REC↓] positionieren den Cursor auf das erste Zeichen eines Records (unabhängig von der Recordlänge).

Einfügen von Zeichen. Die Taste [INSERT] ([LBL]) dient zum Umschalten zwischen Einfügungs- und Ersetzungs-Modus. Beim Einschalten des Texteditors ist der Ersetzungs-Modus aktiviert; wenn der «**1**» Indikator in der Anzeige erscheint, ist der Einfügungs-Modus aktiviert.

Im Ersetzungs-Modus (Einschaltzustand) überschreibt ein von Ihnen eingegebenes Zeichen das Zeichen an der momentanen Cursorposition. Im Einfügungs-Modus wird ein eingetastetes Zeichen vor dem Cursor in den Record eingefügt.

Löschen von Zeichen. Durch Drücken der Taste [←] wird das Zeichen an der momentanen Cursorposition gelöscht; die Zeichen rechts von dem gelöschten Zeichen rücken nach links, um die Lücke aufzufüllen. Wenn der Cursor rechts von dem letzten Zeichen des Records steht, löscht [←] das letzte Zeichen. Wenn der Record nur ein Zeichen enthält und dieses gelöscht wird, erscheint das Symbol «^{T}» (Kennung für leeren Record) in der Anzeige.

Tasten zur Recordkontrolle

Hinzufügen von Records. Die Funktionen [+REC↑] ([VIEW]) und [+REC↓] ([R/S]) fügen vor bzw. nach dem momentanen Record einen leeren Record ein. In der Anzeige erscheint die Kennung für einen leeren Record, das Symbol «^{T}». (Wenn der Einfügungs-Modus aktiviert ist, wird er durch Drücken von [+REC↑] oder [+REC↓] ausgeschaltet.)

---

* Wenn Flag 28 (der Dezimaltrennzeichen-Flag) gesetzt ist, erzeugt die Taste [.] des numerischen Tastenfelds ein [,].

Löschen von Records. Die Funktion [-REC] ([CLx]) löscht den momentanen Record. Der Cursor (d.h. der Record/Zeichen-Zeiger) wird auf das erste Zeichen des nächsten Records positioniert; nach Löschen des letzten Records eines Files wird der Cursor auf das erste Zeichen des vorangehenden (jetzt letzten) Records positioniert.

Positionieren des Cursors auf Records. Die Funktionen [REC↓] ([SST]) und [REC↑] ([BST]) positionieren den Cursor (d.h. den Zeiger) auf das erste Zeichen des nächsten bzw. des vorangehenden Records. Der Cursor läßt sich nicht über den ersten bzw. letzten Record eines Files hinausschieben (gegebenenfalls beginnt die Anzeige zu blinken).

Die Tastenfolge [GTO] *nnn* positioniert den Cursor auf das erste Zeichen des spezifizierten Records. Wie beim normalen [GTO] werden Sie auch hier zur Eingabe einer dreistelligen Zahl *nnn* aufgefordert.

Um den letzten Record eines Files schnell aufzufinden, können Sie die Funktion [GTO] mit einer hohen Adresse verwenden (z.B. [GTO] 999).

Fehler

Wenn der Speicherplatz eines Files erschöpft ist, und Sie versuchen, Records oder Zeichen hinzuzufügen, erscheint die Meldung NO ROOM in der Anzeige und der Computer erzeugt ein akustisches Signal. Die Meldung dient nur zur Warnung und bleibt ungefähr 1 Sekunde lang bestehen. Sie können danach dem File über die Funktion [RESZFL] zusätzliche Register zuweisen (siehe Seite 213).

Die Meldung REC TOO LONG erscheint, wenn Sie versuchen, die maximale Recordlänge (254 Zeichen) zu überschreiten. Zusätzlich bedingt dieser Fehler ein automatisches Verlassen des Texteditor-Modus. (Der Zeigerwert wird nicht verändert; wenn Sie die Funktion [ED] ausführen, wird wieder die vorhergehende Stelle des Records angezeigt.)

Die Meldung FL NOT FOUND wird angezeigt, wenn bei der Ausführung von [ED] der im Alpha-Register spezifizierte File nicht existiert.

Die Meldung NAME ERR erscheint, wenn bei Ausführung von [ED] im Alpha-Register ein unerlaubter Filename spezifiziert ist. Ein Filename darf zum Beispiel kein «,» enthalten; auch darf der Filename nicht aus sieben Bytes mit dem Dezimalwert 255 bestehen (▩▩▩▩▩▩▩).

Die Meldung FL TYPE ERR wird angezeigt, wenn bei Ausführung von [ED] im Alpha-Register der Name eines Daten- oder Programmfiles spezifiziert ist.

Verwenden der Funktion [ED] in Programmen

Die Funktion [ED] ist programmierbar, sämtliche damit verbundenen Editor-Operationen dagegen nicht. Wenn [ED] von einem Programm ausgeführt wird, müssen die notwendigen Eingaben bzw. Modifikationen vom Benutzer über das Tastenfeld ausgeführt werden.

```
04ᵀGEHEIM
05 ED
```

Es reicht aus, das Programm anzuweisen, den Filenamen im Alpha-Register abzulegen und die Funktion [ED] auszuführen. Die Programmausführung wird automatisch unterbrochen und die Texteditor-Anzeige erscheint. (Wenn innerhalb von ungefähr zwei Minuten keine Eingabe erfolgt, schaltet der Computer automatisch den Texteditor aus und setzt die Programmausführung fort.)

```
04 AON
05ᵀGEHEIM
06 0
07 SEEKPTA
08 ED
09 AOFF
```

Wenn beim Einschalten des Texteditors das Alpha-Tastenfeld aktiviert sein soll, können Sie die Anweisung [AON] vorsehen. Sie können das Programm auch anweisen, den Cursor (bzw. damit den Record/Zeichen-Zeiger) zu positionieren. Die Zeilen 06 und 07 positionieren den Zeiger auf Record 000 und Zeichen 000. (Zeiger-Operationen werden in Abschnitt 13 erläutert.)

Nach Ausführen der Funktion [ED] stoppt das Programm und wartet auf Eingaben. Wenn Sie die Texteingabe bzw. Modifikation abgeschlossen haben, können Sie die Programmausführung durch Drücken der Taste [EXIT] fortsetzen. Drücken Sie hierzu *nicht* die Taste [R/S], denn auf dem Texteditor-Tastenfeld ist diese Taste umdefiniert und führt die Funktion [+REC↓] aus.

Um einen Textfile innerhalb eines Programms automatisch zu modifizieren, müssen Sie die programmierbaren Funktionen aus Abschnitt 13 verwenden (z.B. [INSCHR] und [APPREC]).

Teil IV
Zeitfunktionen im Detail

Abschnitt 15

Uhr- und Datumsfunktionen

Inhalt

Einstellen und Korrigieren der Uhrzeit 237
Einstellen der Uhrzeit ([SETIME]) 237
Korrigieren der Uhrzeit ([T+X]) 238
Korrigieren der Ganggenauigkeit ([CORRECT]) 238
Anzeige der Uhr 238
Einschalten der Uhr ([CLOCK] und ■ [ON]) 238
Anzeigeformate der Uhr ([CLK24], [CLK12], [CLKT], [CLKTD]) 239
Operationen mit Zeitwerten 240
Abruf des momentanen Zeitwerts ([TIME]) 240
Anhängen einer Zahl im Zeit-Format an den Inhalt des Alpha-Registers ([ATIME]) 240
Anhängen eines Zeitwerts im 24-Stunden-Format an den Inhalt des Alpha-Registers ([ATIME24]) 241
Datumsfunktionen 241
Datumsformate ([MDY], [DMY]) 242
Einstellen des Datums ([SETDATE]) 242
Abruf des Datums ([DATE]) 242
Anhängen einer Zahl im Datumsformat an den Inhalt des Alpha-Registers ([ADATE]) 243
Berechnungen mit Datumswerten 243
Zulässige Datumsangaben 243
Datumsarithmetik ([DATE+]) 244
Anzahl der Tage zwischen zwei Datumsangaben ([DDAYS]) 244
Wochentag ([DOW]) 244
Gültigkeitsbereiche und Fehlermeldungen 245
Gültigkeitsbereiche 245
Fehlermeldungen 245

Der HP-41 bietet Ihnen eine quarzgesteuerte Uhr und eine Vielzahl von Zeitfunktionen. Die Uhr läuft selbst dann, wenn der Computer ausgeschaltet ist. In diesem Abschnitt werden die Zeitfunktionen behandelt, die mit Uhrzeit und Datum zusammenhängen. Viele dieser Funktionen wurden bereits (mit Beispielen) in Abschnitt 6 besprochen. Die Diskussion in diesem Abschnitt dient der Vollständigkeit und ist mehr zum Nachschlagen gedacht. Eine Zusammenfassung dieser Funktionen finden Sie im Funktionsverzeichnis unter «Zeitfunktionen». Anwendungsbeispiele zu vielen der hier vorgestellten Funktionen finden Sie in Abschnitt 22, «Programme zur Zeiterfassung».

Die Uhr wird getrennt von den übrigen Funktionen des Computers und vom Permanentspeicher verwaltet. Ein Löschen des Permanentspeichers hat daher *keinen Einfluß* auf die Uhr und die Stoppuhr; das eingestellte Datumsformat und die gespeicherten Alarme hingegen *werden mit gelöscht*. (Siehe Anhang G unter «Auswirkungen von Speicherlöschung, Stromunterbrechung und Spannungsabfall».)

In drei Anhängen wird ausführlich auf Zeitfunktionen eingegangen: in Anhang B, «Mehr über überfällige Alarme», in Anhang F, «Zeitspezifikationen» und in Anhang G, «Batterien, Gewährleistung und Serviceinformation». Anhang B dürfte interessant für Sie sein, wenn Sie häufig Alarme einstellen, in Anhang F wird die Ganggenauigkeit des Zeitgebers erläutert und in Anhang G finden Sie Informationen zum Stromverbrauch.

Einstellen und Korrigieren der Uhrzeit

1. Verwenden Sie zum ersten Einstellen der Uhrzeit die Funktion [SETIME] und zum ersten Einstellen des Datums die Funktion [SETDATE].
2. Korrigieren Sie bei der Ausführung von [SETIME] entstandene Fehler mit Hilfe der Funktion [T+X]. Verwenden Sie [T+X] auch bei Zeitzonenumstellungen. (Wird bei der Zeitkorrektur durch [T+X] eine Tagesgrenze überschritten, so ändert sich automatisch auch das Datum.)
3. Warten Sie mindestens eine Woche, bevor Sie die Ganggenauigkeit mit Hilfe der Funktion [CORRECT] nachjustieren.

Wenn Sie beim Einstellen oder Korrigieren von Uhrzeit bzw. Datum einen oder mehrere Alarme überspringen, werden zwei Töne erzeugt und die Alarme werden überfällig. (Siehe «Überfällige Alarme» in Abschnitt 16.)

Einstellen der Uhrzeit ([SETIME])

Mit der Funktion [SETIME] kann die Uhrzeit bis auf eine hundertstel Sekunde genau eingestellt werden. Beachten Sie nebenstehende Tabelle bei der Stundeneinstellung.

Einstellen der Uhrzeit:

1. Geben Sie die Uhrzeit im Format ***±HH.MMSSss*** in das X-Register ein.
2. Führen Sie [SETIME] aus.

Die Zeit wird in dem Moment eingestellt, in dem die Tastenfolge zur Ausführung von [SETIME] abgeschlossen wird. Beim Einstellen der Uhrzeit von Hand kann eine Genauigkeit von ungefähr 0,1 Sekunde erreicht werden. Eine präzisere Einstellung wird mit Hilfe der im nächsten Absatz beschriebenen Funktion [T+X] erzielt.

Zeiteinstellung

Uhrzeit	Einstellung
Mitternacht	0
1	1
2	2
⋮	⋮
10	10
11	11
Mittag	±12
13	–1 oder +13
14	–2 oder +14
⋮	⋮
22	–10 oder +22
23	–11 oder +23
Mitternacht	0

Korrigieren der Uhrzeit ([T+X])

Die Funktion [T+X] (*time plus x*) inkrementiert oder dekrementiert die momentane Uhrzeit um den im X-Register angegebenen Wert. Verwenden Sie [T+X], um lokale Zeitänderungen zu berücksichtigen (Zeitzonen, Sommerzeit) oder die *Einstellgenauigkeit* der Uhrzeit zu korrigieren. Benutzen Sie [T+X] *nicht*, um die *Ganggenauigkeit* des Zeitgebers zu justieren. Korrigieren Sie ein Vor- oder Nachgehen der Uhr mit Hilfe der unten beschriebenen Funktion [CORRECT].

Korrektur der Uhrzeit:

1. Geben Sie die gewünschte Zeitänderung (positiv oder negativ) im Format ***±HHHH.MMSSss*** in das X-Register ein.
2. Führen Sie [T+X] aus.

Wenn durch die Zeitänderung eine Tagesgrenze überschritten wird, so verstellt die Funktion [T+X] automatisch auch das Datum. (Das Einstellen des Datums wird später in diesem Abschnitt erläutert.)

Korrigieren der Ganggenauigkeit ([CORRECT])

Mit der Funktion [CORRECT] (*correct accuracy factor*) wird sowohl die Zeiteinstellung korrigiert als auch der Kompensationsfaktor justiert. Der Kompensationsfaktor berücksichtigt Zeitabweichungen, die durch Schwankungen in der Versorgungsspannung und der Temperatur sowie durch Herstellungstoleranzen verursacht werden. Die Korrektur der Uhrzeit durch den Kompensationsfaktor erfolgt fortlaufend; die Funktion [CORRECT] darf daher nicht dazu verwendet werden, einmalige Zeitabweichungen zu korrigieren (verwenden Sie dazu die Funktion [T+X]). Um den bei Ausführung der Tastenfolge auftretenden Reaktionsfehler gering zu halten, sollte mindestens eine Woche vergehen, bis Sie [CORRECT] erneut verwenden.

Die Funktionsweise von [CORRECT] wird ausführlich in Anhang F («Zeitspezifikationen») erläutert. An dieser Stelle wird daher nur die grundlegende Vorgehensweise beschrieben:

1. Geben Sie die korrekte Uhrzeit im Format ***HH.MMSSss*** in das X-Register ein.
2. Führen Sie [CORRECT] zum richtigen Zeitpunkt aus. (Die in Schritt 1 spezifizierte Zeit wird in dem Augenblick zur momentanen Uhrzeit, in dem die Ausführung von [CORRECT] abgeschlossen wird.)

Diese beiden Operationen stellen die Zeit ein und justieren den intern berechneten Kompensationsfaktor (basierend auf vorhergehenden Ausführungen von [SETIME], [CORRECT] und anderen Funktionen). Die Uhrzeit ist nun auf die spezifizierte Zeit eingestellt; sie wird durch den neu berechneten Kompensationsfaktor laufend korrigiert. (Siehe Anhang F.)

Anzeige der Uhr

Einschalten der Uhr ([CLOCK] und ■ [ON])

- Durch Drücken von ■ [ON] oder Ausführen von [CLOCK] wird die laufende Uhr angezeigt. Nur [CLOCK] ist programmierbar.
- Durch die Tastenfolge ■ [ON] wird der HP-41 ausgeschaltet, bevor die Uhr angezeigt wird. Zusätzlich werden die Flags 12 bis 20 gelöscht (siehe Abschnitt 12).
- Durch Drücken der Taste [←] wird die Anzeige der Uhr gelöscht; es wird dann wieder der Inhalt des X-Registers angezeigt. Durch Drücken der meisten anderen Tasten (einschließlich [ON]) wird die Anzeige der Uhr ebenfalls gelöscht; es wird dann die entsprechende Funktion ausgeführt.

Wie die Uhr angezeigt wird, hängt vom gewählten Anzeigeformat ab. Anzeigeformate werden im nächsten Absatz erläutert.

> **Hinweis:** Die Anzeige der Uhr bedingt einen höheren Stromverbrauch als normal. Siehe Anhang G («Stromverbrauch» und «Spannungsabfall»).

Die Flags 12 bis 20 werden bei jedem Einschalten des Computers gelöscht (siehe Abschnitt 19). Die Tastenfolge ■ ON wirkt sich auf diese Flags genauso aus wie ein Einschalten des HP-41, da der Computer vor Anzeige der Uhr kurz ausgeschaltet wird. Die Funktion CLOCK schaltet den HP-41 *nicht* aus. Benutzen Sie also CLOCK, wenn Sie ein Löschen der Flags 12 bis 20 vermeiden wollen. Sie können CLOCK einer Taste des User-Tastenfelds zuordnen und die Funktion dann durch einen Tastendruck ausführen.

Anzeigeformate der Uhr (CLK24, CLK12, CLKT, CLKTD)

Das von Ihnen eingestellte Anzeigeformat der Uhr bleibt sogar dann erhalten, wenn der Permanentspeicher gelöscht wird. Bei einer Unterbrechung der Stromversorgung (z.B. durch zu langes Herausnehmen der Batterien) wird das von Ihnen gewählte Anzeigeformat gelöscht. Im Einschaltzustand wird nur die Uhrzeit im 12-Stunden-Format angezeigt. Das Anzeigeformat wird folgendermaßen eingestellt:

12- und 24-Stunden-Format. Das 24-Stunden-Format wird durch Ausführen der Funktion CLOCK24 (*clock 24 hour*) eingestellt und das 12-Stunden-Format durch CLOCK12 (*clock 12 hour*). Beim 12-Stunden-Format werden die Vormittagswerte durch AM (*ante meridiem*) und die Nachmittagswerte durch PM (*post meridiem*) gekennzeichnet. Die Anzeigeformate haben *keinen* Einfluß auf das *Eingabeformat* von Zeitwerten oder auf das *Ausgabeformat* von Ergebnissen bei Berechnungen mit Zeitwerten (siehe Tabelle «Zeiteinstellung» am Anfang dieses Abschnitts). Mit anderen Worten: Die Funktionen CLOCK24 und CLOCK12 wirken auf das Anzeigeformat der Uhr, nicht auf das Anzeigeformat des X-Registers.

Nur-Uhrzeit- und Uhrzeit/Datum-Format. Die Funktion CLKT (*clock time*) schaltet die Anzeige auf das Nur-Uhrzeit-Format. Das Uhrzeit/Datum-Format wird durch Ausführen der Funktion CLKTD (*clock time and date*) eingestellt. Die Anzeige des Datums hängt vom eingestellten Datumsformat ab (dies wird später in diesem Abschnitt erläutert).

Beispiel: Anzeige von Uhrzeit und Datum am 21. Januar um 15:15 Uhr. (Das Datum wird im Tag/Monat-Format dargestellt.)

Anzeigeformate der Uhr

Format	CLKT	CLKTD
CLK12	3:15:00 PM	3:15 PM 21.01
CLK24	15:15:00	15:15 21.01

Bei der Anzeige des Datums wird der Sekundenteil der Uhrzeit unterdrückt.

> **Hinweis:** Das Uhrzeit/Datum-Format (CLKTD) verbraucht weniger Strom als das Nur-Uhrzeit-Format (CLKT), da die Anzeige im CLKTD-Format nur jede Minute, im CLKT-Format aber jede Sekunde aktualisiert wird.

Operationen mit Zeitwerten

Durch die Funktion [TIME] wird ein die momentane Uhrzeit repräsentierender Zahlenwert in das X-Register gerufen. Eine Zahl, die sich schon im X-Register befindet, kann durch die Funktionen [ATIME] und [ATIME24] in ein Uhrzeit-Format umgewandelt und an den Inhalt des Alpha-Registers *angehängt* werden.

Abruf des momentanen Zeitwerts ([TIME])

Durch die Funktion [TIME] wird ein die momentane Uhrzeit repräsentierender Zahlenwert im X-Register abgelegt. Dabei wird der Stack angehoben (sofern er nicht gesperrt ist). Die Zahl wird im 24-Stunden-Format mit sechs Dezimalstellen dargestellt: ***HH.MMSSss***. Sie müssen [FIX] 6 ausführen, um alle sechs Dezimalstellen anzuzeigen. Mitternacht wird durch die Zahl 0 dargestellt.

Bei *manueller* Ausführung der Funktion [TIME] erscheint zusätzlich die Uhrzeit im 12- oder 24-Stunden-Format (je nach Einstellung) als Meldung, d.h. *nicht* im X-Register. Die Anzeige dieser Meldung wird durch Drücken der Taste [←] gelöscht; es wird dann der Inhalt des X-Registers angezeigt.

Wenn um 17:05:24 zuerst [TIME] von Hand ausgeführt und dann [←] gedrückt wird, erscheint folgende Anzeige (sechs Dezimalstellen, [FIX] 6):

Zeitwerte

Format	[TIME] (vorübergehende Anzeige)	[←] (X-Register)
[CLK12]	5:05:24 PM	17.052400
[CLK24]	17:05:24	17.052400

Anhängen einer Zahl im Zeit-Format an den Inhalt des Alpha-Registers ([ATIME])

Durch die Funktion [ATIME] (*Alpha time*) wird die Zahl im X-Register im eingestellten Uhrzeit-Format ([CLK12] oder [CLK24]) an den Inhalt des Alpha-Registers angehängt. Diese Funktion ist sehr nützlich für Programme, die mit der momentanen Uhrzeit arbeiten.

Verwenden Sie [ATIME24] (siehe folgenden Absatz) oder [ATIME] zusammen mit [CLK24], um einen Wert als verstrichene Zeit zu kennzeichnen. In diesen Fällen werden die Zusätze **AM** bzw. **PM** und mit den Zahlen 13 bis 23 auftretende Vorzeichenwechsel nicht angezeigt.

Der ganzzahlige (Stunden-) Anteil der Zahl im X-Register bestimmt, wie die Zahl im Alpha-Register formatiert wird:

- [ATIME] akzeptiert jede Zahl im Bereich $-100 < x < 100$. Das Minuszeichen wird ignoriert, *außer* bei –1 bis –11; diese Zahlen werden als Nachmittagswerte (13 bis 23) interpretiert (siehe Tabelle «Zeiteinstellung» am Anfang dieses Abschnitts).
- Für $|x| \leqslant 23$ wird die Zahl gemäß der momentanen Formateinstellung ([CLK12] oder [CLK24]) dargestellt.
- Für $|x| \geqslant 24$ werden die Zusätze **AM** oder **PM** nicht verwendet. (Ein Minuszeichen wird ignoriert.) Auf diese Weise können Sie einen Wert für verstrichene Zeit darstellen (30:12:00 entspricht z.B. 30 Stunden und 12 Minuten).

Beispiel (Anzeige von sechs Dezimalstellen, FIX 6):

Anhängen eines Zeitwerts an den Inhalt des Alpha-Registers

Wert im X-Register	Hängt an den Inhalt des Alpha-Registers an	
	bei CLK12	bei CLK24
−11	11:00:00.00 PM	23:00:00.00
−15.25	3:25:00.00 PM	15:25:00.00
30.125633	30:12:56.33	30:12:56.33

- Die an den Inhalt des Alpha-Registers angehängte Zahl wird nicht gerundet, sondern gemäß dem eingestellten numerischen Anzeigeformat abgeschnitten:

FIX, SCI, ENG	Hängt an den Inhalt des Alpha-Registers an	Beispiel: 10.0637 (10:06:37 bei CLK24)
0	HH	10
1 oder 2	HH:MM	10:06
3 oder 4	HH:MMSS	10:06:37
5 oder mehr	HH:MMSSss	10:06:37.00

Wenn im Alpha-Register nicht mehr genügend Platz ist, um die durch ATIME angehängten Zeichen aufzunehmen, werden die linksstehenden Zeichen aus dem Alpha-Register geschoben. Auf diese Weise wird genügend Platz für die rechts anzuhängenden Zeichen geschaffen.

Wenn das X-Register eine Zahl außerhalb des Bereichs $-100 < x < 100$ enthält, erscheint bei Ausführung der Funktion ATIME die Fehlermeldung DATA ERROR.

Anhängen eines Zeitwerts im 24-Stunden-Format an den Inhalt des Alpha-Registers (ATIME24)

Die Funktion ATIME24 (*Alpha time, 24-hour*) wirkt wie ATIME, mit dem Unterschied, daß die an den Inhalt des Alpha-Registers anzuhängende Zahl immer im 24-Stunden-Format bzw. als verstrichene Zeit dargestellt wird. Das momentane Anzeigeformat der Uhr spielt dabei keine Rolle.

Datumsfunktionen

In diesem Absatz wird unter anderem die Datumsformatierung, das Einstellen und Abrufen des Datums sowie das Anhängen einer Zahl im Datumsformat an den Inhalt des Alpha-Registers erläutert.

Datumsformate (MDY, DMY)

In der Voreinstellung wird das Datum im Format Monat/Tag/Jahr dargestellt. Durch die Funktion MDY (*month/day/year*) wird das Monat/Tag/Jahr-Format und durch die Funktion DMY (*day/month/year*) das Tag/Monat/Jahr-Format eingestellt. Dies ist das einzige Zeitformat, das bei einem Löschen des Permanentspeichers mit gelöscht wird.

Die untenstehende Tabelle zeigt, wie der Computer eine Zahleneingabe als Datum interpretiert und wie ein Datum ausgegeben wird.

Datumsformate

Format-einstellung	X-Register Eingabe/ Ausgabe	Anzeige und Druckerausgabe	Flag 31
MDY	MM.DDYYYY	MM/DD oder MM/DD/YY	gelöscht
DMY	DD.MMYYYY	DD.MM oder DD.MM.YY	gesetzt

Führende und nachlaufende Nullen können bei der Datumseingabe weggelassen werden. Die den 6. Mai 1990 repräsentierende Zahl kann zum Beispiel als 6.05199 eingegeben werden (DMY-Format). Bei Anzeige des Datums als Meldung werden führende und nachlaufende Nullen mit dargestellt.

Einstellen des Datums (SETDATE)

Einstellen des Datums:

1. Geben Sie das Datum in Übereinstimmung mit dem eingestellten Anzeigeformat (siehe oben) im Format ***MM.DDYYYY*** oder ***DD.MMYYYY*** in das X-Register ein. Wie schon erwähnt, können dabei führende und nachlaufende Nullen weggelassen werden.
2. Führen Sie die Funktion SETDATE aus.

Der HP-41 kann auf jedes Datum zwischen dem 1. Januar 1900 und dem 31. Dezember 2199 eingestellt werden.

Wenn das von Ihnen eingegebene Datum im Format nicht mit dem eingestellten Monat/Tag-Format übereinstimmt, erscheint keine Fehlermeldung. Nur wenn Ihre Eingabe ein unzulässiges Datum darstellt, wird eine Fehlermeldung angezeigt. Wenn durch die Einstellung des Datums Alarme überfällig werden, werden zwei Töne erzeugt (siehe Abschnitt 16 unter «Überfällige Alarme»).

Abruf des Datums (DATE)

Durch Ausführen der Funktion DATE wird eine das momentane Datum repräsentierende Zahl im X-Register abgelegt. Dabei wird der Stack angehoben, sofern er nicht gesperrt ist.

- Die ins X-Register zurückgerufene Zahl erscheint je nach Einstellung von Flag 31 im Format ***MM.DDYYYY*** oder ***DD.MMYYYY***.
- Bei manueller Ausführung von DATE erscheint darüberhinaus das Datum und der Wochentag als Meldung. Die Darstellung ist ***MM/DD/YY TAG*** oder ***DD.MM.YY TAG***, wobei ***TAG*** eine aus drei Buchstaben bestehende, englische Abkürzung für den Wochentag ist. (Wenn die Jahreszahl zwischen 2000 und 2199 liegt, werden aller vier Ziffern angezeigt: ***MM/DD/YYYY:TA*** oder ***DD.MM.YYYY TAG***. Dabei ist ***TA*** eine aus zwei Buchstaben bestehende Abkürzung des Wochentags.)

Drücken Sie die Taste [←], um wieder das X-Register anzuzeigen.

Anhängen einer Zahl im Datumsformat an den Inhalt des Alpha-Registers ([ADATE])

Durch die Funktion [ADATE] (*Alpha date*) wird der Inhalt des X-Registers im Datumsformat an den momentanen Inhalt des Alpha-Registers angehängt.

- Die Eingabe sollte, je nach eingestelltem Datumsformat, in der Form ***MM.DDYYYY*** oder ***DD.MMYYYY*** erfolgen. Die eingegebene Zahl muß im Bereich $|x| < 100$ liegen, d.h. Sie können Zeitperioden bis zu 99.999999 (Tage oder Monate) darstellen.
- Die angehängte Zahl wird in das Format ***MM/DD/YYYY*** oder ***DD.MM.YYYY*** umgewandelt. (Ein Minuszeichen wird ignoriert.) Führende oder nachlaufende Nullen werden angezeigt.
- Die angehängte Zahl wird gemäß dem eingestellten numerischen Anzeigeformat abgeschnitten (nicht gerundet):

Anhängen eines Datums an den Inhalt des Alpha-Registers

[FIX], [SCI], [ENG]	Hängt an den Inhalt des Alpha-Registers an		Beispiel: 18.01196 (18. Januar 1960 bei [DMY])
0	MM	oder DD	18
1 oder 2	MM/DD	oder DD.MM	18.01
3 oder 4	MM/DD/YY	oder DD.MM.YY	18.01.60
5 oder mehr	MM/DD/YYYY	oder DD.MM.YYYY	18.01.1960

Bei Darstellung von nur zwei Jahresziffern werden die beiden *letzten* Ziffern der Jahreszahl dargestellt.

Berechnungen mit Datumswerten

Ihnen stehen drei Funktionen zur Verfügung, um Berechnungen mit Datumswerten auszuführen:

- Die Funktion [DATE+] addiert oder subtrahiert eine Anzahl von Tagen von einem gegebenen Datum und findet das resultierende Datum.
- Die Funktion [DDAYS] berechnet die Anzahl von Tagen zwischen zwei Datumsangaben.
- Die Funktion [DOW] berechnet den Wochentag zu einem spezifizierten Datum.

Zulässige Datumsangaben

- Bei Datumsberechnungen (Ein- und Ausgabe) kann jedes Datum zwischen dem 15. Oktober 1582 (Einführung des Gregorianischen Kalenders) und dem 10. September 4320 verwendet werden.
- Die Eingabe (***MM.DDYYYY*** oder ***DD.MMYYYY***) *muß* positiv sein. Hinter ***YYYY*** dürfen nur Nullen stehen.

Bei ungültiger Dateneingabe können die Fehlermeldungen **DATA ERROR** und **DATA ERROR Y** erscheinen. Ein ungültiges Ergebnis bei [DATE+] bewirkt die Fehlermeldung **OUT OF RANGE**.

Datumsarithmetik (DATE+)

Die Funktion DATE+ (*date plus*) berechnet das neue Daten aus dem alten Datum (im Y-Register) und der Anzahl von Tagen (im X-Register), die addiert oder subtrahiert werden soll.

1. Geben Sie das bekannte Datum ein (***MM.DDYYYY*** oder ***DD.MMYYYY***).
2. Geben Sie die Anzahl der Tage ein, die addiert oder subtrahiert werden soll. Verwenden Sie eine positive Zahl für die Addition und eine negative Zahl für die Subtraktion. Nur der ganzzahlige Anteil dieser Zahl wird berücksichtigt.
3. Führen Sie DATE+ aus. Der Stack wird nach unten verschoben; das sich ergebende Datum wird dadurch im Format ***MM.DDYYYY*** oder ***DD.MMYYYY*** im X-Register abgelegt. (Der vorhergehende Inhalt des X-Registers bleibt im LAST X Register erhalten.)

Anzahl der Tage zwischen zwei Datumsangaben (DDAYS)

Die Funktion DDAYS (*delta days*) berechnet die Anzahl von Tagen zwischen zwei Datumsangaben im X- und Y-Register. Das Datum in Y wird vom Datum in X subtrahiert.

1. Geben Sie zuerst das frühere Datum ein und dann das spätere (Eingabeformat: ***MM.DDYYYY*** oder ***DD.MMYYYY***). Wenn Sie das spätere Datum zuerst eingeben, erhalten Sie ein negatives Ergebnis.
2. Führen Sie DDAYS aus. Der Stack verschiebt sich nach unten; die Differenz an Tagen wird dadurch im X-Register abgelegt. (Das zweite Datum bleibt im LAST X Register erhalten.)

Wochentag (DOW)

Die Funktion DOW (*day of week*) berechnet den Wochentag zu dem im X-Register stehenden Datum (***MM.DDYYYY*** oder ***DD.MMYYYY***).

1. Geben Sie das Datum ein.
2. Führen Sie DOW aus.
3. Der Wert im X-Register wird in eine Zahl zwischen 0 (Sonntag) und 6 (Samstag) verwandelt. (Das Datum bleibt im LAST X Register erhalten.)

Bei manueller Ausführung von DOW erscheint die englische Abkürzung für den Wochentag als Meldung. Drücken Sie die ← Taste, um das X-Register (mit dem Zahlenwert für den Wochentag) anzuzeigen.

Gültigkeitsbereiche und Fehlermeldungen

Gültigkeitsbereiche

Alle Zeit- und Datumsfunktionen außer ■ ON sind programmierbar.

Uhrzeit: Minutenwerte (***MM***) und Sekundenwerte (***SS***) müssen im Bereich 00 bis 59 liegen. (Werte für Hundertstelsekunden (***ss***) bewegen sich natürlich zwischen 0 und 99.)

Datum: Es kann jedes Datum vom 1. Januar 1900 (die Voreinstellung) bis zum 31. Dezember 2199 eingestellt werden.

Kalenderfunktionen: Bei *Berechnungen* mit Daten kann jedes Datum zwischen dem 15. Oktober 1582 (Einführung des Gregorianischen Kalenders) und dem 10. September 4320 verwendet werden.

Anzeige: Das Ergebnis einiger Zeitfunktionen wird bis auf die Hundertstelsekunde berechnet; zur vollständigen Anzeige aller Stellen des Ergebnisses ist das Anzeigeformat FIX 6 zu verwenden.

Fehlermeldungen

Die Fehlermeldung DATA ERROR erscheint, wenn die *Eingabe* bei einer Zeitfunktion nicht im richtigen Format erfolgt oder außerhalb des Gültigkeitsbereichs liegt. (ADATE, ATIME, ATIME24, CORRECT, DOW, SETAF, SETDATE, SETIME, SETSW, T+X.)

Die Fehlermeldung **OUT OF RANGE** erscheint, wenn das *Ergebnis* einer Zeitfunktion außerhalb des Gültigkeitsbereichs liegt. (T+X, DATE.)

Abschnitt 16

Alarmfunktionen

Inhalt

Alarmarten: Meldealarme, Steueralarme (↑↑), bedingte Alarme (↑) 247
Meldealarme 247
Steueralarme (↑↑) 248
Bedingte Alarme (↑) 248
Einstellen von Alarmen ([XYZALM]) 250
Parameter der [XYZALM] Funktion 250
Beispiele 251
Zurückrufen von Alarmparametern in den Stack ([RCLALM]) 252
Aktivieren und Bestätigen von Meldealarmen 253
Der Aktivierungszyklus 253
Bestätigen von Meldealarmen 254
Gleichzeitige Alarme 255
Der Alarmkatalog ([ALMCAT], [CATALOG] 5) 255
Löschen von Alarmen aus dem Speicher 258
Löschen eines Meldealarms bei Fälligwerden 258
Allgemeine Löschoperationen 258
Löschen von wiederholenden Steueralarmen 259
Überfällige Alarme 259
Entstehen von überfälligen Alarmen 260
Übergangene überfällige Alarme 260
Automatische Erinnerung an überfällige Alarme 260
Automatische Aktivierung von überfälligen Alarmen 260
Anwendungsprogramme zur Alarmeinstellung 261
Alarmeinstellung mit Hilfe eines Programms 261
Einstellen eines Alarms relativ zur momentanen Zeit 263

Die Alarmoperationen des HP-41 sind äußerst vielseitig: Abhängig von den von Ihnen spezifizierten Parametern kann ein Alarm einfach als persönliche Erinnerung dienen (ein akustischer Alarm mit Meldung) oder als hochentwickelte Steuervorrichtung, um ein Programm zu einem bestimmten Zeitpunkt zu starten. Ein Alarm unterbricht in keinem Fall die Ausführung einer Funktion; er wird verzögert, bis diese Funktion (einschließlich einer Funktion, die von einem Programm ausgeführt wird) abgeschlossen ist.*

Der HP-41 kennt drei Arten von Alarmen, die alle auf ähnliche Weise eingestellt werden. Nach dem Einstellen sind Alarme, zusammen mit Programmen und Tastenzuordnungen, im Hauptspeicher aufgezeichnet. In Abschnitt 6 («Zeitfunktionen») finden Sie eine Einführung in die Alarmoperationen (mit Beispielen). Eine Zusammenfassung der Alarmfunktionen finden Sie im Funktionsverzeichnis unter «Zeitfunktionen».

Ein Programm, das durch einen Alarm ausgeführt wird, kann den Status von Flags oder Registern beeinflussen (wie jedes andere Programm auch). Es ist also unter Umständen anzuraten, das aufgerufene Programm so zu gestalten, daß die betroffenen Daten (wenn Sie sie zu nachfolgenden Berechnungen benötigen) wieder abgespeichert werden.

Alarmarten: Meldealarme, Steueralarme (↑↑), bedingte Alarme (↑)

Das Flußdiagramm auf der übernächsten Seite gibt einen Überblick über die drei verschiedenen Alarmarten und ihre Arbeitsweise. Einer der drei Alarme hat die Funktion eines Weckers (der Meldealarm); die beiden anderen Alarme dienen der Programmausführung (der Steueralarm und der bedingte Alarm, der einen «nichtunterbrechenden» Steueralarm darstellt).

Meldealarme

Ein Meldealarm kann zur Erinnerung an einen Termin benutzt werden. Dieser Alarm erzeugt eine Tonfolge und zeigt blinkend die Meldung an, die bei Einstellung des Alarms im Alpha-Register abgelegt wurde. Die Meldung kann bis zu 24 Zeichen umfassen. Wenn der Alarm mit *leerem* Alpha-Register eingestellt wird, erscheinen Uhrzeit und Datum blinkend in der Anzeige.

Ein Meldealarm wird fällig, wenn die eingestellte Alarmzeit erreicht ist – unabhängig davon, ob der Computer ein- oder ausgeschaltet ist oder ein Programm ausführt. Ein Alarm unterbricht jedoch, wie schon erwähnt, nie die Funktion, die *in dem Moment* des Fälligwerdens ausgeführt wird.

Ein Meldealarm muß bestätigt werden, sonst wird er überfällig (d.h. der Alarm bleibt als nicht beantworteter Alarm im Speicher aufgezeichnet). (Überfällige Alarme werden ausführlich in diesem Abschnitt unter «Überfällige Alarme» und in Anhang B erläutert.) Ein Meldealarm ersetzt jede zuvor angezeigte Meldung, hat sonst aber keine Auswirkungen auf Status oder Operationen des Computers; Stackregister, Datenspeicherregister und Alpha-Register werden nicht beeinflußt.

* Dies gilt auch für die sehr «langen» Funktionen von unbeschränkter Dauer: ablaufende Kataloge, Kataloge 4 bis 6 (immer), Text-Editor (ED) und Stoppuhr (SW). Alarme werden während dieser Operationen nicht aktiviert; sie werden verzögert und erst nach Abschluß der Operation ausgeführt.

Steueralarme (↑↑)

Ein Steueralarm führt ein Programm oder eine programmierbare Katalog 2 Funktion aus, die beim Einstellen des Alarms im Alpha-Register als ↑↑*globales Label* oder ↑↑*Katalog 2 Funktionsname* zu spezifizieren ist.* Die Programmausführung beginnt bei dem globalen Label oder, wenn den beiden Pfeilen (↑↑) kein Label oder Name folgt, bei der momentanen Programmzeile. Bei Erreichen der eingestellten Zeit wird das spezifizierte Programm bzw. die spezifizierte Funktion ausgeführt, *ob der HP-41 ein- oder ausgeschaltet ist.* Die Ausführung beginnt, sobald die momentan ablaufende Funktion abgeschlossen ist.

Wenn ein Steueralarm fällig wird, während gerade ein Programm – einschließlich eines durch einen anderen Steueralarm gestarteten Programms – läuft, wird das momentan ausgeführte Programm *zeitweilig* unterbrochen und das Programm bzw. die Katalog 2 Funktion ausgeführt, auf die sich der Steueralarm bezieht.** Der Computer führt das aufgerufene Programm wie ein Unterprogramm des unterbrochenen Programms aus und benutzt dabei eine Unterprogrammebene. Unterprogramme und Unterprogrammebenen werden in Abschnitt 20 («Programmverzweigungen») erläutert.

Ein Steueralarm muß nicht bestätigt werden; er wird aktiviert und dann selbsttätig gelöscht oder auf die neue Alarmzeit eingestellt.

Bedingte Alarme (↑)

Ein bedingter Alarm führt, wie ein Steueralarm, das im Alpha-Register spezifizierte Programm bzw. die spezifizierte Katalog 2 Funktion aus. Es handelt sich jedoch um einen *nichtunterbrechenden Steueralarm*, d.h. der Alarm wird *nur* aktiviert, wenn der Computer *ausgeschaltet* ist oder die Uhr angezeigt wird.

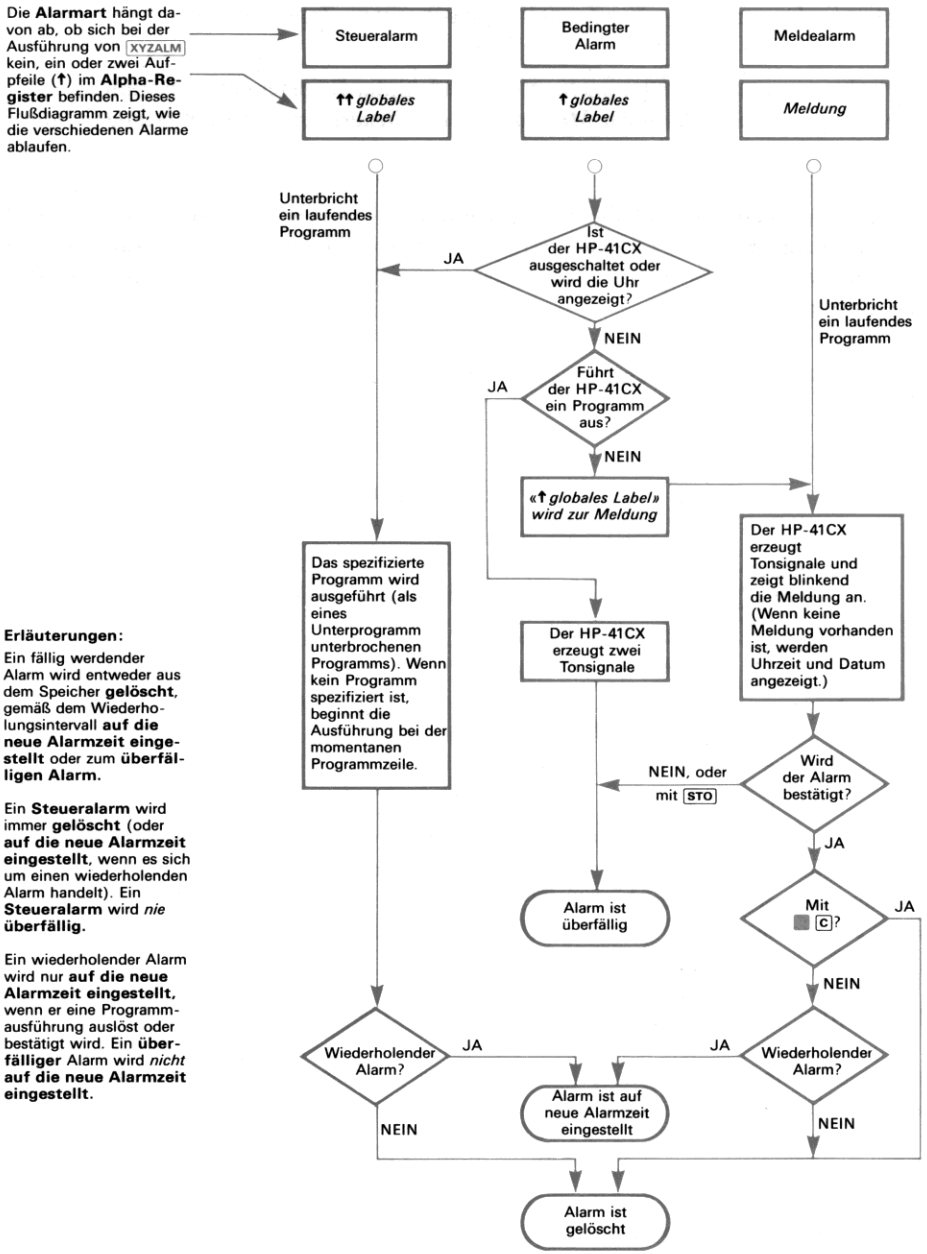
- Wenn ein bedingter Alarm während der Ausführung eines Programms fällig wird, erzeugt der Alarm zwei Töne und wird überfällig.
- Wenn der Computer bei Fälligwerden des Alarms lediglich eingeschaltet ist (kein Programm läuft, Uhr nicht angezeigt), wird der Alarm zum Meldealarm und führt das spezifizierte Programm bzw. die spezifizierte Funktion nicht aus. Der Alarm wird überfällig, wenn er nicht bestätigt wird (wie ein Meldealarm).

Ob ein bedingter Alarm wie ein Steueralarm wirkt, hängt also davon ab, ob der Computer ausgeschaltet ist. Durch einen bedingten Alarm wird ein Programm automatisch ausgeführt, wenn nicht gerade andere Programme oder Berechnungen auf dem HP-41 ablaufen. Ein bedingter Alarm, der wie ein Steueralarm wirkt, braucht nicht bestätigt zu werden. Wenn der bedingte Alarm zum Meldealarm wird, ist er zu bestätigen.

* Katalog 2 Funktionen werden unter «Katalogorganisation der neuen Funktionen» in Anhang I beschrieben.

** Die Programmunterbrechung erfolgt nach Abschluß einer bei Fälligwerden des Steueralarms gerade ausgeführten Funktion. Zusätzlich unterbricht der Alarm nicht, bis der Stack Lift freigegeben ist.

Flußdiagramm zur Alarmeinstellung



Erläuterungen:

Ein fällig werdender Alarm wird entweder aus dem Speicher **gelöscht**, gemäß dem Wiederholungsintervall **auf die neue Alarmzeit eingestellt** oder zum **überfälligen Alarm.**

Ein **Steueralarm** wird immer **gelöscht** (oder **auf die neue Alarmzeit eingestellt**, wenn es sich um einen wiederholenden Alarm handelt). Ein **Steueralarm** wird *nie* **überfällig.**

Ein wiederholender Alarm wird nur **auf die neue Alarmzeit eingestellt,** wenn er eine Programmausführung auslöst oder bestätigt wird. Ein **überfälliger** Alarm wird *nicht* **auf die neue Alarmzeit eingestellt.**

Wenn ein Programm wegen eines Alarms unterbrochen wurde, wird das unterbrochene Programm jetzt fortgesetzt.

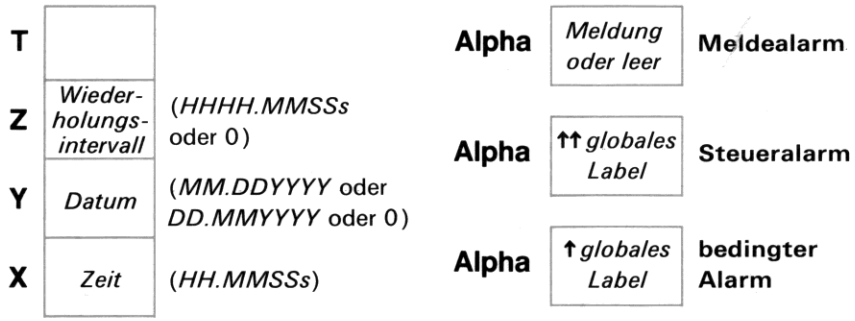
Einstellen von Alarmen (XYZALM)

Alle drei Alarmarten werden mit der programmierbaren Funktion XYZALM (*X, Y, Z alarm*) eingestellt. Die Funktion XYZALM verwendet die Daten im X-, Y-, Z- und Alpha-Register. Zeitwerte können bis auf die Zehntelsekunde genau angegeben werden.

1. Geben Sie die benötigten Parameter ins X-, Y-, Z- und Alpha-Register ein (siehe unten). *Der Inhalt des Alpha-Registers bestimmt, ob der Alarm ein Meldealarm, ein Steueralarm oder ein bedingter Alarm ist.*
2. Führen Sie XYZALM aus.

Eine unerlaubte Dateneingabe bedingt die Fehlermeldung DATA ERROR, die zusätzlich angibt, in welchem Register der Fehler aufgetreten ist (DATA ERROR X usw.). Wenn der Alarm auf einen Zeitpunkt in der Vergangenheit gestellt wird oder wenn sich übergangene, überfällige Alarme im Hauptspeicher befinden, erzeugt der HP-41 zwei Tonsignale.

Parameter der XYZALM Funktion



Z-Register: Wiederholungsintervall des Alarms. Das Wiederholungsintervall spezifiziert die Zeitspanne, nach der sich der Alarm wiederholt. 1 Sekunde ≤ Intervall < 10 000 Stunden. Ein *wiederholender Alarm* (ein Alarm, dessen Wiederholungsintervall von Null verschieden ist) stellt sich gemäß dem Wiederholungsintervall und der ersten Einstellung selbst auf die nächste Aktivierungszeit ein (siehe «Aktivieren und Bestätigen von Meldealarmen» in diesem Abschnitt).

Das Z-Register muß den Wert 0 enthalten, wenn keine Wiederholung gewünscht wird.

> Hinweis: Es ist unter Umständen schwierig, einen wiederholenden Steueralarm mit sehr kurzem Wiederholungsintervall (weniger als 10 Sekunden) zu löschen. Siehe «Löschen von Alarmen aus dem Speicher» in diesem Abschnitt.

Y-Register: Alarmdatum. Das Datum, an dem der Alarm aktiviert wird. Es kann jedes Datum vom 1. Januar 1900 bis zum 31. Dezember 2199 verwendet werden.

Geben Sie Null ein, um den Alarm auf das momentane Datum zu stellen.

X-Register: Alarmzeit. Die Uhrzeit, wann der Alarm fällig («aktiviert») wird, zwischen 0 und +23.59599 (eine Zehntelsekunde vor Mitternacht). Nachmittagswerte (13 bis 23 Uhr) können auch durch die negativen Zahlen –1 bis –11 dargestellt werden. Siehe Tabelle «Zeiteinstellung» in Abschnitt 15.

Alpha-Register: Meldealarm. Ein bis zu 24 Zeichen umfassender Alpha-String. Wenn das Alpha-Register leer ist, werden bei Fälligwerden des Alarms Uhrzeit und Datum angezeigt.

Alpha-Register: Steueralarm. ↑ ↑ *globales Label* oder ↑ ↑ *Katalog 2 Funktionsname* oder nur ↑ ↑.* Das spezifizierte Label bzw. der spezifizierte Name dürfen nur bis zu *sechs* Zeichen umfassen (obwohl Label im Programmspeicher bis zu sieben Zeichen lang sein können), jedes weitere Zeichen wird einfach ignoriert. Beachten Sie, daß im Alarmkatalog jedoch alle sieben Zeichen angezeigt werden.**

Sie können mit einem Steueralarm eine Programmausführung auch an anderer Stelle als dem globalen Label starten, indem Sie *nur* die beiden Pfeile (↑↑) im Alpha-Register ablegen. Wenn der Alarm fällig wird, startet er die Programmausführung *beginnend mit der momentanen Zeile des momentanen Programms*. Diese Alarmspezifikation kann zum Beispiel dazu benutzt werden, ein Programm so zu gestalten, daß es mit der Ausführung an der Stelle fortfährt, an der es sich durch die Anweisung OFF selbst unterbrochen hat.

Alpha-Register: bedingter Alarm. ↑ *globales Label* oder ↑ *Katalog 2 Funktionsname*. Das spezifizierte Label oder der spezifizierte Name kann bis zu sieben Zeichen umfassen.

Sie können einen bedingten Alarm (wie einen Steueralarm) auch so einstellen, daß die Programmausführung statt bei einem globalen Label bei der momentanen Zeile des momentanen Programms beginnt. Geben Sie dazu nur den Pfeil (↑) ohne Label oder Funktionsname ins Alpha-Register ein.

Beispiele

		leer
T		
Z	0	**Alpha**
Y	0	
X	1.01	

Durch Ausführen von XYZALM wird ein nichtwiederholender Meldealarm auf 1:01 Uhr des momentanen Datums eingestellt. Die «Meldung» besteht aus Uhrzeit und Datum.

	Meldung
0	
8.31199	
–10	

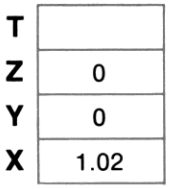
Stellt einen nichtwiederholenden Meldealarm auf 22:00 Uhr am 31. August 1990 (Datumformat DMY).

	Meldung
0.15	
0	
11	

Stellt einen Meldelarm mit 15minütigem Wiederholungsintervall, der das erste Mal um 11:00 Uhr am momentanen Datum fällig wird.

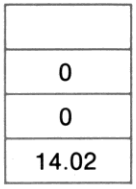
* Der Pfeil (↑) ist das umgeschaltete Zeichen der N Taste des Alpha-Tastenfelds (ENTER↑) Taste des Normal-Tastenfelds).

** Sie können mit einem Steueralarm auch ein Programm oder eine Funktion mit einem aus sieben Zeichen bestehenden Label bzw. Namen ausführen. Schreiben zu diesem Zweck ein kurzes «Aufruf»-Programm, das die gewünschte Funktion bzw. das gewünschte Programm ausführt. Stellen Sie den Alarm dann so ein, daß er das «Aufruf»-Programm ausführt.



↑↑TEST
Alpha

Stellt einen nichtwiederholenden Steueralarm auf 1:02 Uhr des momentanen Datums. Der Alarm ruft das Programm **TEST** auf und führt es aus. Wenn nötig unterbricht der Alarm dazu zeitweilig ein laufendes Programm.



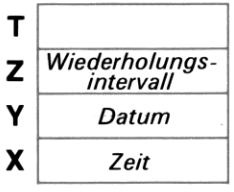
↑TEST

Stellt einen nichtwiederholenden bedingten Alarm auf 14:02 Uhr des momentanen Datums. Der Alarm führt das Programm **TEST** *nur* aus, wenn der HP-41 ausgeschaltet ist oder die Uhr angezeigt wird.

Zurückrufen von Alarmparametern in den Stack (RCLALM)

Die Funktion RCLALM (*recall alarm*) ruft die Parameter eines gespeicherten Alarms in die Stackregister und ins Alpha-Register zurück, wo sie von einem Programm untersucht und verändert werden können. Sie können die zurückgerufenen Werte auch im Massenspeicher oder in anderen Registern speichern. Wenn Sie die meisten Ihrer Alarme löschen, einige aber erhalten wollen, können Sie mit RCLALM bestimmte Alarmwerte bewahren, bevor Sie mit CLALMS alle Alarme löschen (siehe «Allgemeine Löschoperationen» in diesem Abschnitt).

RCLALM ruft die Parameter des spezifizierten Alarms in die X-, Y-, Z- und Alpha-Register zurück. Der Alarm wird durch den Betrag des ganzzahligen Anteils der Zahl im X-Register bestimmt. Diese Alarmnummer stimmt mit der chronologischen Reihenfolge der Alarme im Alarmkatalog überein (siehe «Der Alarmkatalog» in diesem Abschnitt). Gültige Alarmnummern liegen im Bereich von 1 bis 253. Eine Zahl, die größer ist als existierende Alarmnummern, führt zur Fehlermeldung NO SUCH ALM (*no such alarm*).



HH.MMSSss

Alpha-String
Alpha

Meldung oder
↑↑ *globales Label* oder
↑ *globales Label*

Das Ausgabeformat unterscheidet sich vom Eingabeformat bei XYZALM nur dadurch, daß die Zeit immer im 24-Stunden-Format ***HH.MMSSss*** zurückgegeben wird. Das Monat/Tag-Format entspricht der gegenwärtigen Einstellung. Der Inhalt jedes Registers entspricht dem Registerinhalt bei Ausführen von XYZALM für den jeweiligen Alarm.

Bei der Ausführung von RCLALM wird die Alarmnummer (die im X-Register stand) im LAST X Register erhalten und der vorhergehende Inhalt des Y-Registers in das T-Register geschoben.

Aktivieren und Bestätigen von Meldealarmen

Meldealarme, *einschließlich bedingter Alarme, die zu Meldealarmen werden,* folgen einem charakteristischen Aktivierungsschema. Sie *müssen* während des Aktivierungszyklus bestätigt werden (wie untenstehend erläutert); ansonsten werden sie überfällig.

- Ein nichtwiederholender Meldealarm wird durch eine Bestätigung abgestellt *und* aus dem Speicher gelöscht.
- Ein wiederholender Alarm wird durch eine Bestätigung abgestellt und auf die nächste Alarmzeit gestellt. Das Wiederholungsintervall wird ein oder mehrmals zur *ursprünglichen Alarmzeit*, nicht zur Zeit der Bestätigung addiert.

Nichtbestätigte (überfällige) Meldealarme bleiben im Speicher erhalten. Bei Ausschalten des Computers werden sie automatisch wieder aktiviert und erzeugen bei jedem Einschalten des Computers Tonsignale. (Siehe «Überfällige Alarme».)

Steueralarme und bedingte Alarme, die zu Steueralarmen werden, sind selbstbestätigend. Sie führen einfach das spezifizierte Programm oder die spezifizierte Katalog 2 Funktion aus und löschen sich dann automatisch selbst aus dem Speicher (bzw. stellen sich auf die neue Alarmzeit, wenn es sich um wiederholende Alarme handelt).

Der Aktivierungszyklus

Wenn ein Meldealarm fällig wird, läuft das folgende Schema ab:

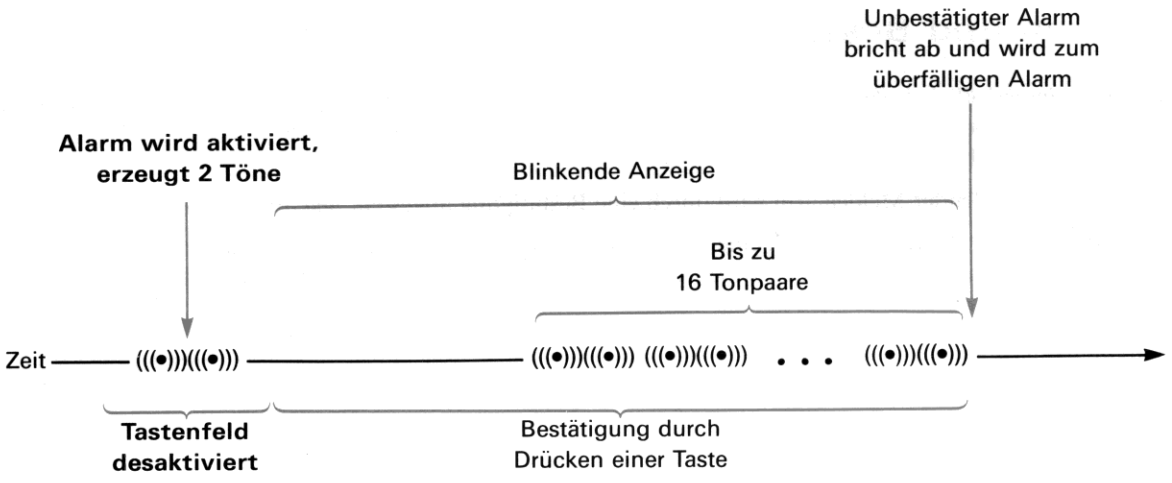
1. Der Alarm erzeugt zwei Tonsignale und zeigt die ersten 12 Zeichen des Alpha-Strings an. Wenn kein Alpha-String angegeben wurde, werden Uhrzeit und Datum angezeigt. (Ein bedingter Alarm zeigt ↑*globales Label* oder ↑*Funktionsname* an.)

   Dieses Phase dauert ungefähr eine Sekunde und während dieser Zeit ist das Tastenfeld gesperrt.

2. Die Anzeige beginnt zu blinken. *Während der Phasen 2 und 3 können Sie den Alarm bestätigen.* Dies geschieht durch Drücken fast jeder beliebigen Taste und wird im nächsten Absatz erläutert.

3. Nachdem die Anzeige fünfmal geblinkt hat, wird der akustische Alarm fortgesetzt und der Computer erzeugt bis zu 16 weitere Tonpaare.

4. Wenn Sie den Alarm jetzt noch nicht bestätigt haben, wird er überfällig. (Siehe «Überfällige Alarme».) Wenn es sich um einen wiederholenden Alarm handelt, wird er nicht auf die neue Alarmzeit eingestellt.



Durch Bestätigen eines Meldealarms wird der obige Aktivierungszyklus gestoppt. Bei einem Meldealarm mit mehr als 12 Zeichen hängt die nachfolgende Anzeige von der Bestätigungsmethode ab (siehe unten).

Bestätigen von Meldealarmen

Solange die Anzeige blinkt und bevor die lange Tonfolge abgeschlossen ist, können Sie einen Meldealarm durch eine der folgenden Methoden bestätigen. Diese Vorgehensweise können Sie auch zur Bestätigung überfälliger Alarme verwenden, wenn diese automatisch reaktiviert werden.

Ergebnis der Alarmbestätigung

Gedrückte Taste		
STO	← oder ON	Jede andere Taste
Stellt den Alarm ab. Nach Loslassen der Taste bleibt die Anzeige noch 3 Sekunden bestehen.* Wenn Sie STO nochmals drücken, bleibt die Anzeige für weitere 3 Sekunden bestehen. Erhält den Alarm als überfälligen Alarm.	Stellt den Alarm ab. Löscht die Alarmanzeige sofort (keine Fortsetzung von 3 Sekunden). Löscht den Alarm aus dem Speicher.†	Stellt den Alarm ab. Nach Loslassen der Taste bleibt die Anzeige noch 3 Sekunden bestehen.* Nochmaliges Drücken einer dieser Tasten verlängert die Anzeige um weitere 3 Sekunden. Löscht den Alarm aus dem Speicher.†

* Wenn die Meldung mehr als 12 Zeichen enthält, werden die ersten 12 angezeigt, solange Sie die Taste gedrückt halten; danach werden die restlichen Zeichen für 3 Sekunden angezeigt.

† Ein wiederholender Alarm wird nicht gelöscht, sondern auf die neue Alarmzeit eingestellt.

Warten Sie mit der Ausführung anderer Funktionen, bis die Alarmmeldung verschwindet. Andernfalls verzögern Sie das Löschen der Alarmmeldung.

Wiederholende Alarme. Das Nachstellen eines wiederholenden Alarms geschieht durch Anhängen des Wiederholungsintervalls an die Alarmzeit, nicht die Bestätigungszeit.

Wenn Sie einen wiederholenden Alarm mit [STO] bestätigen, wird der Alarm nicht nachgestellt, sondern als überfälliger Alarm gespeichert. (Das Wiederholungsintervall bleibt im Alarmkatalog erhalten.) Ein überfälliger wiederholender Alarm wird beim Nachstellen nicht auf einen Zeitpunkt in der Vergangenheit gestellt.

Drücken Sie ■ [C], um einen wiederholenden Meldealarm während seines Aktivierungszyklus aus dem Speicher zu löschen.

Verzögerung der Aktivierung. Wenn bei Fälligwerden eines Alarms ein früherer Meldealarm aktiviert ist, wird der zweite Alarm verzögert, bis der erste Alarm bestätigt wurde oder sein Aktivierungszyklus abgeschlossen ist.

Gleichzeitige Alarme

Wenn mehrere Alarme auf genau die gleiche Zeit eingestellt sind, werden die Alarme in der Reihenfolge aktiviert, in der sie eingestellt wurden. Ein Steueralarm unterbricht jedoch ein von einem Steueralarm oder bedingten Alarm gestartetes Programm. Ein Meldealarm unterbricht zeitweilig ein von einem vorhergehenden Steueralarm gestartetes Programm (wartet aber, wie im vorangegangenen Absatz erläutert, die Beendigung des Aktivierungszyklus eines anderen Alarms ab).

Bei mehreren gleichzeitigen oder sich überschneidenden Steueralarmen unterbrechen aufeinanderfolgende Alarme die von vorhergegangenen Alarmen gestarteten Programme und führen das spezifizierte Programm aus; erst dann kann das vorhergehende Programm ablaufen. Ein bedingter Alarm unterbricht weder ein Programm noch wartet er die Beendigung eines Programms ab; es werden stattdessen zwei Tonsignale erzeugt und der bedingte Alarm wird überfällig. Gleichzeitige Alarme werden in der gleichen Reihenfolge fällig wie überfällige Alarme (dies wird in Anhang B erläutert).

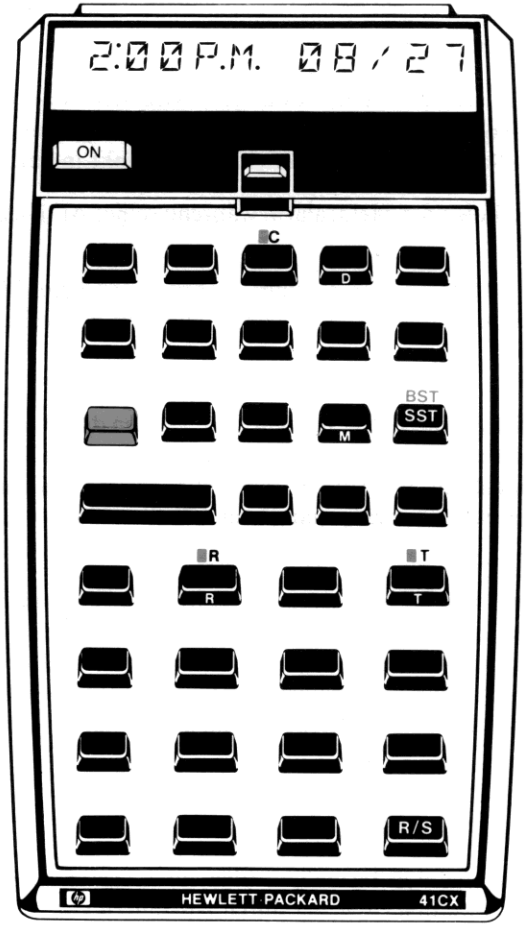
Der Alarmkatalog ([ALMCAT], [CATALOG] 5)

Der Alarmkatalog und das Alarmkatalog-Tastenfeld (das zur Ausführung von speziellen Alarmkatalog-Funktionen umdefinierte Tastenfeld) werden entweder durch Ausführen von [ALMCAT] (*alarm catalog*) oder durch Drücken von [CATALOG] 5 aktiviert. Nur [ALMCAT] ist programmierbar; [CATALOG] 5 ist dagegen schneller manuell auszuführen.

Merkmale des Alarmkatalogs:

- Der Alarmkatalog liefert eine vollständige Liste aller im Speicher des HP-41 aufgezeichneten Alarme (einschließlich überfälliger Alarme).
- Der Alarmkatalog listet alle Alarme in der Reihenfolge der Aktivierungszeit, vom frühsten bis zum spätesten Alarm. (Die Position eines wiederholenden Alarms im Katalog wird bei jedem Nachstellen berichtigt.)
- Für jeden aufgelisteten Alarm wird zuerst Alarmzeit und Datum und anschließend ein Alpha-String (eine Meldung oder ↑↑/↑ aufzurufendes Label) angezeigt.
- Nach Ausgabe des letzten Alarms wird der Alarmkatalog verlassen und wieder der Inhalt des X-Registers angezeigt.

Das Alarmkatalog-Tastenfeld



- Das Aufrufen des Alarmkatalogs gibt den Stack Lift frei.
- Der Alarmkatalog kann, wie andere Kataloge auch, mit [R/S] angehalten und wieder fortgesetzt werden.* Nach Anhalten des Katalogs können Sie ihn mit [SST] oder [BST] schrittweise durchgehen (bei Erreichen von Kataloganfang oder Ende mit [SST]/[BST] blinkt die Anzeige).
- Bei angehaltenem Katalog ist das Tastenfeld zum *Alarmkatalog-Tastenfeld* umdefiniert.

Bei angehaltenem Alarmkatalog können mit den folgenden Tasten des Alarmkatalog-Tastenfelds spezielle Operationen ausgeführt werden. Diese Tasten repräsentieren keine Zeichen des Alpha-Tastenfelds; drücken Sie daher nicht [ALPHA]. Die Buchstaben beziehen sich auf die jeweilige Alarmkatalog-Funktion: **T** (*time*) steht für Zeit, **D** (*date*) für Datum, **M** (*message*) für Meldung und **R** (*repeat interval*) für Wiederholungsintervall.

Das Alarmkatalog-Tastenfeld

Taste(n)	Operation
[T]	Ruft die Aktivierungszeit des momentan angezeigten Alarms zurück.
[f] [T]	Ruft die momentane Uhrzeit zurück.
[D]	Ruft das Aktivierungsdatum des momentan angezeigten Alarms zurück.
[R]	Ruft das Wiederholungsintervall des momentan angezeigten Alarms zurück.
[f] [R]	Stellt den Alarm gemäß dem Wiederholungsintervall auf die nächste Auslösezeit.
[M]	Zeigt den Alpha-String (Meldung auf ↑↑/↑ aufzurufendes Label) des momentan angezeigten Alarms an.
[f] [C]	Löscht den momentan angezeigten Alarm aus dem Speicher.
[←]	Beendet den Alarmkatalog, schaltet das Alarmkatalog-Tastenfeld aus und zeigt wieder den Inhalt des X-Registers an.
[ON]	Schaltet den Computer aus; beendet den Alarmkatalog.

Wird der Computer im Alarmkatalog angehalten und ca. zwei Minuten lang keine Taste gedrückt, verläßt der Computer automatisch den Alarmkatalog.

Die einzelnen Operationen des Alarmkatalog-Tastenfelds sind im Gegensatz zu [ALMCAT] nicht programmierbar. Bei Aufruf von [ALMCAT] in einem Programm wird das Programm nach Ausgabe des Alarmkatalogs fortgesetzt.

* Die allen Katalogen gemeinsamen Operationen sind in Abschnitt 9 aufgeführt. Bei laufendem Katalog führt das Drücken jeder Taste außer [R/S] oder [ON] zu einer Beschleunigung der Katalogausgabe.

Löschen von Alarmen aus dem Speicher

Steueralarme (einschließlich bedingter Alarme, die zu Steueralarmen wurden) löschen sich nach ihrer Aktivierung automatisch selbst. Meldealarme (einschließlich bedingter Alarme, die zu Meldealarmen wurden) werden durch Bestätigen gelöscht. Darüberhinaus gibt es einige allgemeine Löschoperationen, mit denen ein oder mehrere zukünftige und/oder überfällige Alarme gelöscht werden können. Ein rasches Reaktionsvermögen ist erforderlich, wenn Sie einen wiederholenden Alarm mit sehr kurzem Wiederholungsintervall löschen wollen.

Löschen eines Meldealarms bei Fälligwerden

Ein fällig werdender Meldealarm wird, wenn es sich nicht um einen wiederholenden Alarm handelt, durch Bestätigen abgestellt und aus dem Speicher gelöscht. Ein wiederholender Meldealarm wird bei Fälligwerden durch ■ [C] abgestellt und aus dem Speicher gelöscht.

Allgemeine Löschoperationen

Sie können zukünftige und überfällige Alarme (d.h. Alarme, die momentan nicht aktiviert sind) sowie wiederholende Steueralarme auf verschiedene Arten löschen.

- Die Tastenfolge ■ [ON] auf dem Alarmkatalog-Tastenfeld (Katalog 5, siehe oben) ist die einfachste Methode, einen solchen Alarm zu löschen; diese Tastenfolge ist allerdings *nicht programmierbar*.

Die anderen Funktionen zum Löschen von Alarmen ([CLRALMS], [CLALMA] und [CLALMX]) sind programmierbar. [CLALMA] und [CLALMX] sind insbesondere nützlich, wenn Sie einen wiederholenden Alarm durch ein Programm einstellen und später wieder löschen lassen wollen.

Löschen aller Alarme ([CLRALMS]). Die Funktion [CLRALMS] (*clear alarms*) löscht alle Alarme aus dem Speicher. Sie können diese Funktion in Verbindung mit [RCLALM] (Seite 252) benutzen; dabei können die Alarme einzeln zurückgerufen und ihre Parameter abgespeichert werden, bevor zur Rückgewinnung von Alarmspeicherplatz alle Alarme gelöscht werden.

Löschen eines Alarms mit Hilfe des Alpha-Strings. Die Funktion [CLALMA] (*clear alarm by alpha*) löscht den *ersten* Alarm, dessen Alpha-String mit dem String im Alpha-Register übereinstimmt.

Wenn keine Alarme mit gleichen Alpha-Strings (d.h. Meldungen oder durch ↑↑/↑ aufzurufende Label) vorhanden sind, ist [CLALMA] eine sehr sichere Methode zur Alarmlöschung. Wenn zwei oder mehr Alarme den gleichen Alpha-String haben, wird nur der erste (in der Reihenfolge im Alarmkatalogs) gelöscht.

Wenn das Alpha-Register bei Ausführung von [CLALMA] leer ist, wird der erste Alarm ohne Meldung gelöscht.

Wenn kein Alarm mit dem spezifizierten Alpha-String existiert, erscheint die Meldung NO SUCH ALM.

Löschen eines Alarms über die Alarmnummer (CLALMX). Die Funktion CLALMX (*clear alarm by X*) löscht den durch die Zahl im X-Register spezifizierten Alarm. Die Zahl bezieht sich auf die Position (Alarmnummer) des Alarms im Alarmkatalog.

Mehrfach vorhandene Alpha-Strings sind unproblematisch. Sie sollten jedoch daran denken, daß sich eine Alarmnummer jedesmal ändern kann, wenn ein anderer Alarm eingestellt, nachgestellt, aktiviert oder gelöscht wird.

Die Funktion CLALMX interpretiert den Betrag des ganzzahligen Anteils der Zahl im X-Register als Alarmnummer. Wenn z.B. bei Ausführung von CLALMX im X-Register die Zahl 5.1 abgelegt ist, wird der fünfte Alarm im Alarmkatalog gelöscht.

Für $x = 0$ oder $x > 999$ erscheint bei Ausführung von CLALMX die Meldung DATA ERROR.

Wenn kein Alarm mit der spezifizierten Nummer existiert, erscheint die Meldung NO SUCH ALM.

Löschen von wiederholenden Steueralarmen

Normalerweise kann ein wiederholender Steueralarm problemlos mit Hilfe der oben erläuterten Methoden gelöscht werden. Wiederholende Steueralarme mit einem Wiederholungsintervall von weniger als ca. 10 Sekunden (das Minimum ist 1 Sekunde) können jedoch, bedingt durch die Dauer der Löschoperation, schwierig aus dem Speicher zu löschen sein. Obwohl CATALOG 5 schnell ausgeführt werden kann, ist es möglich, daß sich der Alarm wiederholt und nachstellt, bevor Sie ihn löschen können.

Diese Zeitbegrenzung kann auf zwei Methoden umgangen werden:

- Ordnen Sie die Funktion ALMCAT einer Taste des User-Tastenfelds zu, damit sie durch einen einzigen Tastendruck ausgeführt werden kann. Der gesamte Alarmkatalog-Modus ist Teil einer einzigen Funktion und kann nicht durch einen Alarm unterbrochen werden. Nun können Sie mit der Sequenz ■ C den in Frage kommenden Alarm löschen.
- Verwenden Sie die «Tastenbuffer-Technik», um CATALOG 5 oder eine andere Löschfunktion auszuführen. Wenn vor dem Loslassen einer Taste bereits die nächste Taste gedrückt wird, arbeitet der HP-41 ohne Unterbrechung. Der Alarm wird nicht fällig, da er auf eine Unterbrechung zwischen der Ausführung von zwei Funktionen bzw. dem Drücken von zwei Tasten wartet. Durch die Tastenbufferung werden fortlaufend Operationen ausgeführt. (Der Alarm wartet auf das Ende einer Funktionsausführung und wird nicht überfällig.)

Verwenden Sie in Programmen die Löschenfunktionen CLALMA oder CLALMX zum Löschen von wiederholenden Steueralarmen.

Überfällige Alarme

Ein überfälliger Alarm ist ein Alarm, dessen Auslösezeit vor der momentanen Zeit liegt. Im folgenden werden die Grundzüge von Operationen mit überfälligen Alarmen beschrieben. Wenn zwei oder mehr überfällige Alarme gespeichert sind, wird die Reihenfolge ihrer automatischen Aktivierung durch die in Anhang B («Mehr über überfällige Alarme») erläuterten Regeln bestimmt.

Entstehen von überfälligen Alarmen

Ein Alarm wird überfällig, wenn eine der folgenden Bedingungen erfüllt ist.

- Ein Meldealarm wird bei Fälligwerden nicht bestätigt (bzw. durch Drücken der [STO] Taste bestätigt).
- Ein bedingter Alarm (↑) wird aktiviert, während der HP-41 ein Programm ausführt.
- Ein bedingter Alarm wird bei seiner Aktivierung zum Meldealarm (wenn der HP-41 nicht ausgeschaltet ist) und der resultierende Meldealarm wird nicht bestätigt.

Beachten Sie, daß ein Steueralarm normalerweise nicht überfällig werden kann, da er immer aktiviert wird und sich danach entweder selbsttätig nachstellt oder löscht. Ein Steueralarm wird dann überfällig, wenn seine Auslösezeit *übergangen* wird.

Übergangene überfällige Alarme

Ein Alarm wird *übergangen* und damit *überfällig*, wenn eine der folgenden Bedingungen erfüllt ist.

- Ein zukünftiger Alarm wird durch ein Verstellen der Zeit übergangen.
- Ein Alarm wird auf eine in der Vergangenheit liegende Auslösezeit eingestellt.

Übergangene überfällige Alarme entstehen selten. Ein Steueralarm kann nur überfällig werden, wenn er übergangen wird.

Automatische Erinnerung an überfällige Alarme

Wenn überfällige Alarme gespeichert sind, erzeugt der HP-41 bei jedem Einschalten zwei Tonsignale. Dies dient nur zu Ihrer Erinnerung; die Alarme werden nicht aktiviert (haben also auch keine Auswirkung) und können nicht bestätigt werden.

Das Tonsignal wird auch erzeugt, wenn Sie die Zeit verstellen oder die Funktion [XYZALM] ausführen und dabei *übergangene* überfällige Alarme entstehen bzw. schon existieren.

Automatische Aktivierung von überfälligen Alarmen

Wenn Sie den HP-41 ausschalten *oder* die Uhr durch ■ [ON] (*nicht* [CLOCK]) einschalten, aktiviert der Computer automatisch alle überfälligen Alarme.*

Die automatische Aktivierung beginnt mit frühestem Alarm:

- Überfällige Meldealarme zeigen die jeweilige Meldung an.
- Überfällige Steueralarme führen das spezifizierte Programm aus.**

---

* Wenn nach der Ausführung von [CLOCK] die [ON] Taste gedrückt wird, werden die verbleibenden überfälligen Alarme *nicht* aktiviert.

** Vor der automatischen Aktivierung von *Steueralarmen oder bedingten Alarmen* schaltet sich der HP-41 kurzzeitig aus. Deshalb werden nachfolgend solange keine anderen überfälligen Alarme (mit Ausnahme von übergangenen Alarmen) aktiviert, bis wieder [ON] oder ■ [ON] gedrückt wird. (Siehe Anhang B.)

- Ein überfälliger bedingter Alarm wird, wenn keine vorhergehenden überfälligen Steueralarme existieren, automatisch als Steueralarm aktiviert.* Wenn der HP-41 am Ende eines Programms durch die Anweisung OFF ausgeschaltet wird, werden alle bei Ablauf des Programms fällig gewordenen bedingten Alarme aktiviert.

Die automatische Aktivierung von mehreren überfälligen Alarmen wird in Anhang B erläutert.

Selbsttätiges Löschen/Nachstellen von automatisch aktivierten, überfälligen Alarmen. Ein automatisch aktivierter, überfälliger Meldealarm wird durch Bestätigen aus dem Speicher gelöscht bzw. gemäß seinem Wiederholungsintervall nachgestellt. (Durch Drücken von ON wird ein fällig werdender, überfälliger Alarm zwar abgestellt, *nicht* aber gelöscht bzw. nachgestellt. Siehe unten, «Abstellen der Aktivierung von überfälligen Alarmen».)

Auch überfällige Steueralarme oder bedingte Alarme löschen sich selbsttätig bei automatischer Aktivierung (bzw. stellen sich selbsttätig nach). Ein wiederholender Alarm wird durch Addition von Vielfachen des Wiederholungsintervalls zur ursprünglichen Alarmzeit auf die nächste *zukünftige* Auslösezeit eingestellt.

Nach der Bestätigung von überfälligen *Meldealarmen* wird die Funktion ausgeführt, die die automatische Aktivierung ausgelöst hat (d.h. der HP-41 schaltet sich entweder aus oder zeigt die Uhr an).

Abstellen der Aktivierung von überfälligen Alarmen. Durch Drücken der ON Taste während des automatischen Aktivierungszyklus eines überfälligen *Meldealarms* wird die automatische Aktivierung (auch die von nachfolgenden Alarmen) abgebrochen. Der Alarm wird dadurch *nicht* bestätigt. Die unbestätigten, nicht abgearbeiteten Alarme bleiben als überfällige Alarme gespeichert.** Dadurch können Sie die unter Umständen sehr lange Folge von nacheinander aktivierten Alarmen abstellen und wieder die Kontrolle über Ihren HP-41 gewinnen, ohne dabei die gespeicherten Alarme zu löschen. (Diese Methode ist nützlich, wenn Sie die Uhrzeit versehentlich vorstellen und die zukünftigen Alarme dadurch überfällig werden.)

Wenn Sie während der Aktivierung von überfälligen *Steueralarmen oder bedingten Alarmen* die ON Taste drücken, wird der HP-41 dadurch ausgeschaltet, und das von dem betreffenden Alarm gestartete Programm wird gestoppt. (Dadurch werden noch verbleibende überfällige Alarme aktiviert.) Der unterbrochene Steueralarm/bedingte Alarm wird durch die Aktivierung gelöscht bzw. nachgestellt.

Aktivierung eines überfälligen bedingten Alarms (ALMNOW). Ein einzelner überfälliger bedingter (oder Steuer-) Alarm kann durch Ausführung der Funktion ALMNOW (*alarm now*) aktiviert werden. Befinden sich mehrere derartige Programme im Speicher, so wird das *älteste* aktiviert.

Ein bedingter (↑) Alarm kann während des Ablaufens eines Programmes fällig werden, und dadurch überfällig werden. Sie können nach einem solchen Alarm suchen und diesen gegebenenfalls aktivieren (bzw. ein Programm diese Tätigkeit übernehmen lassen), indem Sie die Funktion ALMNOW ausführen. Dabei ist vorausgesetzt, daß sich keine weiteren überfälligen bedingten oder Steueralarme im Speicher befinden. Bei Aufruf von einem Programm wirkt ALMNOW wie ein Unterprogramm.

Anwendungsprogramme zur Alarmeinstellung

Wenn Sie die in diesem Abschnitt beschriebenen Anwendungsprogramme verwenden wollen, müssen Sie sie zunächst anhand der gegebenen Programm-Listings in den Programmspeicher eingeben. Wenn Sie einen optischen Lesestift HP 82153A besitzen, können Sie die Programme über die Strichcode-Listings in Anhang J («Strichcode-Listings der Beispielprogramme») einlesen.

Alarmeinstellung mit Hilfe eines Programms

Das nachstehende Programm **ALMEIN** («**AL**ar**MEIN**stellung») ist nützlich, wenn Sie Schwierigkeiten beim Einstellen von Alarmen haben. **ALMEIN** erleichtert die Einstellung von Alarmen wesentlich; Sie werden nach der benötigten Eingabe gefragt und die eingetasteten Daten werden im richtigen Register abgelegt.

---

* Vor der automatischen Aktivierung von *Steueralarmen oder bedingten Alarmen* schaltet sich der HP-41 kurzzeitig aus. Deshalb werden nachfolgend solange keine anderen überfälligen Alarme (mit Ausnahme von übergangenen Alarmen) aktiviert, bis wieder ON oder ■ ON gedrückt wird. (Siehe Anhang B.)

** Wenn Sie den automatischen Aktivierungszyklus durch Drücken der ON Taste unterbrechen, verhindern Sie damit auch die Aktivierung eines Alarms, der während des automatischen Aktivierungszyklus zum ersten Mal fällig geworden wäre.

Eingabe (nach Aufforderung): Alarmzeit, Meldung (für Meldealarme, Steueralarme und bedingte Alarme), Alarmdatum und Wiederholungsintervall (wenn Wiederholung gewünscht wird).

Ergebnis: Einstellung eines Meldealarms, Steueralarms oder bedingten Alarms.

Benutzerhinweise:

1. Führen Sie **ALMEIN** ([ALMEIN]) aus.
2. Geben Sie bei Anzeige der Eingabeaufforderung ZEIT? die Alarmzeit im Format ***HH.MMSS*** ein. (Wenn keine Eingabe erfolgt, wird das Programm beendet.) Setzen Sie das Programm durch Drücken von [R/S] fort.
3. Geben Sie bei Anzeige der Eingabeaufforderung MELDUNG? die gewünschte Meldung (für einen Meldealarm) oder ↑↑ *globales Label* (für einen Steueralarm) oder ↑ *globales Label* (für einen bedingten Alarm) ein. Geben Sie nichts ein, wenn ein Meldealarm nur Uhrzeit und Datum anzeigen soll. Setzen Sie das Programm durch Drücken von [R/S] fort.
4. Geben Sie bei Anzeige der Eingabeaufforderung DATUM? das Alarmdatum im Format ***MM.DDYYYY*** oder ***DD.MMYYYY*** ein. Wenn Sie nichts eingeben, wird der Alarm auf das momentane Datum gestellt. Setzen Sie das Programm durch Drücken von [R/S] fort.
5. Geben Sie bei Anzeige der Eingabeaufforderung WIEDERHLG? das Wiederholungsintervall im Format ***HHHH.MMSS*** ein. Geben Sie nichts ein, wenn Sie keine Wiederholung wünschen. Setzen Sie das Programm durch Drücken von [R/S] fort.

Wenn Sie **ALMEIN** ausführen, gehen alle zuvor im Stack oder im Alpha-Register gespeicherten Daten verloren und Flag 22 wird gelöscht.

Programm-Listing

```
01 LBLTALMEIN
02 CF 22
03TZEIT?
04 PROMPT
05 FC?C 22
06 RTN
07TDATUM?
08 ASTO T
09TWIEDERHLG?
10 ASTO Y
11TMELDUNG?
12 AVIEW
13 CLA
14 AON
15 STOP
16 AOFF
```

Löscht Flag 22, den numerischen Dateneingabe-Flag.

Hält das Programm an und fragt nach der Alarmzeit. Das Programm wird beendet, wenn Sie keine Zeit eingeben (d.h. der abgefragte Flag 22 bleibt gelöscht).

Speichert die Eingabeaufforderung DATUM? im T-Register und die Eingabeaufforderung WIEDERHLG? im Y-Register.*

Zeigt die Eingabeaufforderung MELDUNG? aus dem Alpha-Register an und hält das Programm bei eingeschaltetem Alpha-Tastenfeld an. (Löscht das Alpha-Register – MELDUNG? wird weiter angezeigt – schaltet das Alpha-Tastenfeld ein und hält das Programm an.) Bei Ausführung von Zeile 13 wird das Alpha-Register gelöscht; wenn Sie keine Meldung eingeben, zeigt der Alarm Uhrzeit und Datum an.

* Bei Ausführung der Parameterfunktionen [ASTO], [VIEW] und [X<>] werden zwei Eingabezeichen angezeigt (z.B. X<>__). Die Sequenz [.] [Z] spezifiziert das Z-Register als Parameter.

17 VIEW T 18 STOP	Zeigt DATUM? an und hält das Programm zur Eingabe an.*
19 FC?C 22 20 0	Wenn Sie kein Datum spezifizieren (Flag 22 bleibt gelöscht) wird 0 verwendet (momentanes Datum).
21 VIEW Z 22 STOP	Zeigt die (vom Y- ins Z-Register geschobene) Eingabeaufforderung WIEDERHLG? an und hält das Programm zur Eingabe an.*
23 FC?C 22 24 0	Wenn Sie kein Wiederholungsintervall spezifizieren, wird 0 verwendet.
25 X<> Z 26 XYZALM 27 END	Legt die Zeit (die in das Z-Register geschoben wurde) wieder im X-Register ab und stellt den Alarm ein.*

Einstellen eines Alarms relativ zur momentanen Zeit

An Hand des folgenden Programmbeispiels werden einige Programmiertechniken aufgezeigt (einschließlich ausgedehnter Stackmanipulationen). Mit Hilfe des Programms **ALMREL** werden Alarme relativ zur momentanen Zeit eingestellt. Anstatt die Uhrzeit der Alarmauslösung zu spezifizieren, geben Sie die Zeitspanne (gerechnet ab der momentanen Uhrzeit) ein, nach der der Alarm aktiviert werden soll.

Eingabe (nach Aufforderung):

1. Zeitspanne, nach der der Alarm aktiviert werden soll, im Format ***HHHH.MMSS***. Diese Zeitspanne darf länger als ein Tag sein.
2. Alpha-String des Alarms (*Meldung* oder ↑↑/↑*aufzurufendes Label*).

Ergebnis: Einstellung eines Meldealarms, eines Steueralarms oder eines bedingten Alarms.

Benutzerhinweise:

1. Führen Sie **ALMREL** (ALMREL) aus.
2. Geben Sie bei Anzeige der Eingabeaufforderung +HH.MMSS? die gewünschte Zeitspanne im Format ***HHHH.MMSS*** ein und setzen Sie dann das Programm durch Drücken von R/S fort.
3. Geben Sie bei Anzeige der Eingabeaufforderung MELDUNG? den Alpha-String des Alarms (oder nichts) ein und drücken Sie R/S.

Das Programm **ALMREL** rechnet die Alarmzeit aus, nachdem Sie in Schritt 3 die R/S Taste gedrückt haben. Die Zeitspanne darf zwischen 0000.0003 (3 Sekunden) und 9999.595999 liegen. Das Programm prüft, ob Ihre Eingabe numerisch ist und weist nicht numerische oder negative Eingaben für ***HH.MMSS*** zurück. (Flag 22 wird bei einer Zahleneingabe gesetzt.)

Bei Ausführung von **ALMREL** gehen alle zuvor im Stack und im Alpha-Register gespeicherten Daten verloren und Flag 22 wird gelöscht.

---

* Bei Ausführung der Parameterfunktionen ASTO, VIEW und X<> werden zwei Eingabezeichen angezeigt (z.B. X<>__). Die Sequenz · Z spezifiziert das Z-Register als Parameter.

Programm-Listing

```
01 LBL"ALMREL
02 CF 22
03" + HH.MMSS?
04 PROMPT
05 FC?C 22
06 RTN
07 X<0?
08 RTN
09"MELDUNG?
10 AVIEW
11 CLA
12 AON
13 STOP
14 AOFF
15 TIME
16 HMS+
17 ENTER↑
18 ENTER↑
19 24
20 /
21 INT
22 DATE
23 X<>Y
24 DATE+
25 LASTX
26 24
27 *
28 ST- Z
29 CLX
30 STO T
31 RDN
32 X<>Y
33 XYZALM
34 END
```

Löscht Flag 22 zur späteren Abfrage.

Hält das Programm an und fragt nach der Zeitspanne. (Setzen Sie das Programm mit [R/S] fort.)

Prüft, ob die Eingabe numerisch ist. Wenn nicht (Flag 22 gelöscht), erfolgt ein Rücksprung zum Programmanfang.

Prüft, ob die Eingabe negativ ist. Wenn ja, erfolgt ein Rücksprung zum Programmanfang.

Hält das Programm an und fragt nach dem Alpha-String des Alarms. Das Programm löscht das Alpha-Register und schaltet das Alpha-Tastenfeld ein (AON). Wenn Sie das Programm fortsetzen, wird das Alpha-Tastenfeld wieder ausgeschaltet (AOFF). Geben Sie die gewünschte Meldung (oder nichts) ein und drücken Sie die [R/S] Taste.

Ruft die momentane Uhrzeit ab.
Addiert die spezifizierte Zeitspanne zur momentanen Uhrzeit.

Legt die Summe (die neue Zeit in Stunden) im X-, Y- und Z-Register ab.

Berechnet, wieviele ganze Tage in der spezifizierten Zeitspanne enthalten sind.

Ruft das momentane Datum ab.
Tageanteil der Zeitspanne in X, momentanes Datum in Y.
Berechnet das Alarmdatum.

Ruft den Tageanteil der Zeitspanne ab, wandelt diese Zahl in Stunden um und subtrahiert das Ergebnis von der Gesamtzeitspanne (in Z). Das Ergebnis ist die Alarmzeit.

Legt 0 im X- und T-Register ab. Nach der zyklischen Verschiebung des Stacks ist die 0 im Z-Register und legt damit fest, daß der Alarm nicht wiederholt wird. Die Alarmzeit befindet sich jetzt in Y (aus Z) und das Alarmdatum in X (aus Y).

Datum in Y, Zeit in X. Der Stack ist jetzt zur Alarmeinstellung angeordnet.

Abschnitt 17

Stoppuhrbetrieb

Inhalt

Das Stoppuhr-Tastenfeld ([SW]) 266
Anzeige der Stoppuhr 267
Funktionen des Stoppuhr-Tastenfelds 268
Die Registerzeiger 270
Allgemeiner Stoppuhrbetrieb mit Zeitnahme 271
Anzeige während der Zeitnahme 271
Begrenzung der Registerzeiger 271
Abruf gespeicherter Zeiten ([RCL]) 272
Anzeige der Differenz gestoppter Zeiten ([ΔSPLIT]) 272
Programmierbare Stoppuhrfunktionen 273
Starten und Anhalten der Stoppuhr ([RUNSW], [STOPSW]) 273
Einstellen und Abrufen der momentanen Stoppuhrzeit ([SETSW], [RCLSW]) 273
Gleichzeitiges Einstellen der Stoppuhr und der Stoppuhrzeiger ([SWPT]) 274
Die Stoppuhr als Countdown Timer 274
Ausdrucken gespeicherter Zeiten 275
Beispiel für ein Stoppuhrprogramm 275

Die *Stoppuhrfunktion* [SW] (oder [SWPT]) stellt einen speziellen Arbeitsmodus des HP-41 dar: der Computer wird nun als Stoppuhr benutzt, wobei dem Tastenfeld spezielle Funktionen zu ihrer Steuerung zugewiesen werden. Der eingebaute Zeitgeber (Timer), der für die Stoppuhr verwendet wird, arbeitet unabhängig von der Uhr. Die Operation der Stoppuhr wird von dem Funktionszustand des Computers nicht beeinflußt. Die Stoppuhr läuft selbst dann weiter, wenn sie nicht mehr angezeigt wird oder wenn der HP-41 ausgeschaltet ist. Andere Funktionen und Programme können ausgeführt werden, ohne daß die Stoppuhr dabei angehalten wird.

Der Stoppuhrbetrieb wird in Abschnitt 6 mit Beispielen eingeführt. Eine Zusammenfassung der Stoppuhrfunktionen finden Sie im Funktionsverzeichnis unter «Zeitfunktionen».

Das Stoppuhr-Tastenfeld ([SW])

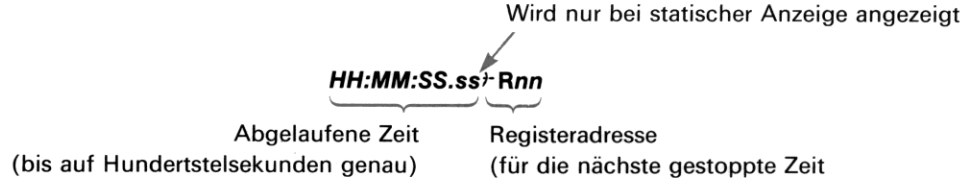
Der HP-41CX wird mit einer Tastenfeld-Schablone geliefert, auf der alle aktiven Funktionen des Stoppuhr-Tastenfeldes eingetragen sind. Ein Diagramm des Tastenfelds ist nachstehend und in der Kurzanleitung zu finden. Solange das Stoppuhr-Tastenfeld eingeschaltet ist, können Sie nur die auf der Schablone eingetragenen Funktionen ausführen.

- Durch Ausführung von [SW] oder [SWPT] wird die Stoppuhr-Anzeige eingeschaltet und das Tastenfeld neu definiert. Durch [SW] werden die Registerzeiger auf Null zurückgesetzt und die Anzeige auf das normale Zeitformat umgeschaltet. Die Funktion [SWPT] positioniert die Registerzeiger auf die von Ihnen spezifizierte Adresse. (Siehe Seite 274.)
- Das Stoppuhr-Tastenfeld wird mit [EXIT] ausgeschaltet, jedoch *halten Sie damit nicht automatisch die laufende Stoppuhr an.* In der Anzeige erscheint wieder der Inhalt des X-Registers. Diese Funktion ist nicht programmierbar.
- Mit der [ON] Taste wird der HP-41 (und damit auch das Stoppuhr-Tastenfeld) ausgeschaltet, *wobei eine laufende Stoppuhr nicht angehalten wird.*

Außer [SW] und [SWPT] ist keine der Funktionen auf dem Stoppuhr-Tastenfeld programmierbar. Es gibt jedoch andere nicht auf dem Stoppuhr-Tastenfeld vorhandene Stoppuhrfunktionen, die programmierbar sind; diese werden in «Programmierbare Stoppuhrfunktionen» erläutert.

Anzeige der Stoppuhr

Bei eingeschaltetem Stoppuhr-Tastenfeld hat die Anzeige folgende Form:



Hinweis: Beachten Sie, daß die Anzeige der Stoppuhr genauso viel Energie verbraucht wie ein laufendes Programm. Schlagen Sie dazu unter «Stromverbrauch» in Anhang G nach. *Bei eingeschaltetem Stoppuhr-Tastenfeld schaltet sich der HP-41 nicht automatisch aus.*

Wenn Sie das Stoppuhr-Tastenfeld ein- bzw. ausschalten, wird weder die angezeigte Zeit gelöscht noch die laufende Stoppuhr angehalten. Jedoch wird beim Einschalten des Stoppuhr-Tastenfelds (mit [SW]) die Zeigeranzeige auf R00 zurückgesetzt.

Durch Drücken von [CLEAR] wird die Stoppuhr auf 00:00:00.00 zurückgesetzt. Die Stoppuhr beginnt automatisch wieder bei Null, wenn sie den Wert 99:59:59.99 überschreitet. Gespeicherte Zeiten können nicht mit Stoppuhrfunktionen gelöscht werden. Jeder neu in einem Register abgelegte Zeitwert ersetzt den zuvor in diesem Register gespeicherten Wert. Gespeicherte Zeiten können im Stoppuhrbetrieb nur gelöscht werden, indem Sie den Registerinhalt durch die Zeit Null ersetzen. Andernfalls müssen Sie das Stoppuhr-Tastenfeld ausschalten und mit [CLRGX] das betreffende Register löschen.

Sie können durch das Drücken einer nicht belegten Taste des Stoppuhr-Tastenfelds die Anzeige festhalten, ohne die Stoppuhr dabei anzuhalten. Die abgelaufene Zeit wird auf eine Zehntelsekunde genau angezeigt. Nur wenn die Stoppuhr angehalten ist, werden alle acht Stellen angezeigt.

Funktionen des Stoppuhr-Tastenfelds

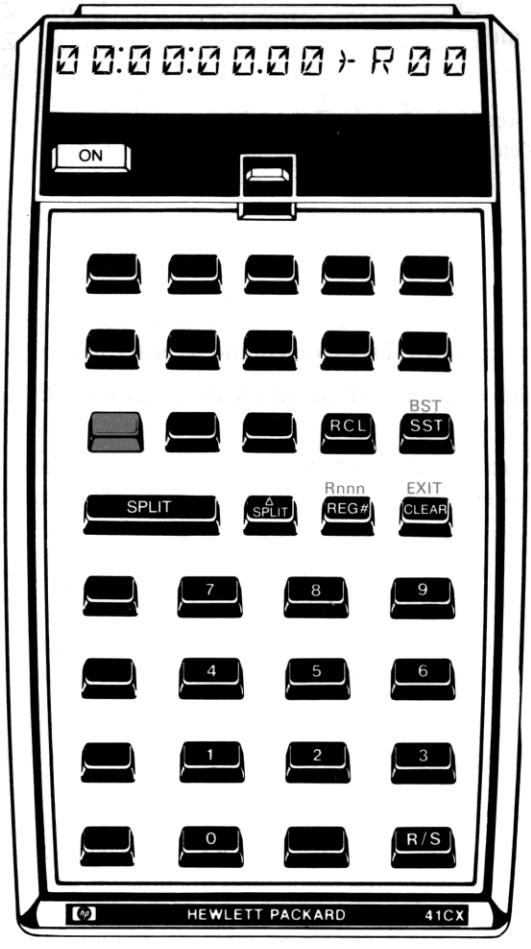
Die folgenden Funktionen sind auf dem Stoppuhr-Tastenfeld vorhanden, das durch die Ausführung von [SW] oder [SWPT] eingeschaltet wird. Zusätzlich dazu werden mit [SW] die Speicher- (⊦) und Rückruf- (=) Zeiger auf Null gesetzt. Im Einschaltzustand ist der Computer auf reguläre Zeitnahme eingestellt (⊦R00).

Das Stoppuhr-Tastenfeld

Taste	Operation
[R/S]	Mit *Run/Stop* wird die Stoppuhr gestartet bzw. angehalten. Die Stoppuhr bzw. die Registerzeiger werden damit nicht zurückgesetzt. (Zusätzlich wird der Rückrufmodus verlassen.)
[CLEAR]	Setzt die angehaltene Stoppuhr auf 00:00:00.00 zurück. Dies beeinflußt nicht die Registerzeiger. (Wenn der Rückrufmodus gesetzt war, wird er durch [CLEAR] verlassen und die Stoppuhr nicht zurückgesetzt.)
[EXIT]	Schaltet das Stoppuhr-Tastenfeld aus, *ohne dabei die Stoppuhr anzuhalten.*
[SPLIT]	Speichert die momentane Stoppuhrzeit in dem angezeigten Datenspeicher-Register (⊦Rnn) und stellt den Zeiger auf das nächste Register. (Gleichzeitig wird dabei auch der Rückrufmodus verlassen.)
[RCL]	Schaltet den Rückrufmodus ein und aus. Im Rückrufmodus können Sie die in dem durch den Registerzeiger spezifizierten Register gespeicherte Zeit abrufen. (Der Rückrufmodus wird ebenso durch [R/S], [SPLIT] und [CLEAR] verlassen.)
[ΔSPLIT]	Schaltet den Delta-Stoppmodus (D) ein und aus, mit dem Sie die Differenz zwischen der letzten Zeit und der im vorhergehenden Register gespeicherten Zeit anzeigen können. Die angezeigte Zeit ändert sich jedesmal, wenn Sie [SPLIT] ausführen. Diese Funktion kann auch im Rückrufmodus benutzt werden. Der Delta-Stoppmodus wird auch verlassen, wenn Sie das Stoppuhr-Tastenfeld ausschalten.
Zifferntasten*	Dienen zur Spezifikation der Speicherregisteradressen (*nn* oder *nnn* mit [Rnnn]). Durch die Angabe eines Zahlenwerts wird der Registerzeiger auf das entsprechende Register positioniert. Beachten Sie, daß Sie immer, dem Anzeigemodus entsprechend, eine zwei- bzw. dreistellige Adresse angeben müssen.
[SST]*	Stellt den Registerzeiger ohne Zeitnahme auf das nächste Register.
[BST]*	Stellt den Registerzeiger ohne Zeitnahme auf das vorhergehende Register.
[REG#]	Wechseltaste zum Einschalten und Ausschalten der Anzeige des Registerzeigers.
[Rnnn]	Wechseltaste zum Umschalten von zwei- auf dreistellige Anzeige des Registerzeigers, um $R_{(100)}$ bis $R_{(319)}$ anzeigen zu können.
Jede andere Taste	Durch Drücken einer nicht belegten Taste des Stoppuhr-Tastenfelds halten Sie, solange die Taste gedrückt ist, die gegenwärtige Zeit in der Anzeige fest. *Die Stoppuhr selbst läuft weiter.*

* Diese Funktionen verändern die Adresse des Registerzeigers unabhängig davon, ob die Stoppuhr läuft oder angehalten ist.

Das Stoppuhr-Tastenfeld



Die Registerzeiger

Die in der Anzeige als R*nn* und D*nn* angedeuteten Registerzeiger geben die Adresse *nn* des momentanen Datenspeicher-Registers an.

Durch die Symbole R und D werden *normale Zeiten* bzw. *Zeitdifferenzen* dargestellt. Die Zeichen ⊦ und = deuten an, ob Zeiten bzw. Zeitdifferenzen gespeichert oder abgerufen werden.

Zeitspeicherung	
⊦R*nn*	Zeit abspeichern
⊦D*nn*	Zeit abspeichern, Anzeige der Differenz
Zeitrückruf	
=R*nn*	Zeitrückruf
=D*nn*	Rückruf einer Zeitdifferenz

Das momentane Register. *Bei jeder Ausführung von* [SW] *werden die Registerzeiger auf 00 gesetzt* (selbst wenn dabei die Zeit nicht auf Null zurückgesetzt wird). Das momentane Register hat folgende Funktionen:

- Bei ⊦R*nn* (reguläre Zeitnahme) zeigt *nn* das Register an, in das die *nächste* (mit [SPLIT]) gestoppte Zeit abgelegt wird.
- Bei ⊦D*nn* (Delta-Zeitnahme) zeigt *nn* das Speicherregister an, in das die nächste gestoppte Zeit abgelegt wird. *Gleichzeitig* stellt dieses Register das zweite von zwei aufeinander folgenden Registern dar, deren Differenz (mit [SPLIT]) berechnet und angezeigt wird.
- Bei =R*nn* (Abruf regulärer Zeitnahmen) stellt *nn* das Register dar, dessen Inhalt gerade angezeigt wird. (Die Anzeige ist selbst bei laufender Stoppuhr angehalten.)
- Bei =D*nn* (Abruf einer Zeitdifferenz) stellt *nn* das zweite von zwei aufeinander folgenden Registern dar, deren Differenz angezeigt wird. (Die Anzeige ist selbst bei laufender Stoppuhr angehalten.)

Verstellen der Registerzeiger. Nach jeder Zeitnahme wird der ⊦ Registerzeiger auf die nächste Speicheradresse gestellt. Mit [SST], [BST] und den Zifferntasten des Stoppuhr-Tastenfelds können Sie den Registerzeiger auch manuell verschieben. Der = Registerzeiger kann nur manuell positioniert werden.

> **Hinweis:** Der Speicherregisterzeiger (⊦) und der Rückrufregisterzeiger (=) werden vom Computer getrennt verwaltet, während die Modi für reguläre Zeitnahme (R) und Zeitdifferenz (D) hingegen unterschiedliche Anzeigeformen für den gleichen Registerzeiger darstellen. Die ⊦ und = Registerzeiger werden beide durch [SW] auf 00 zurückgesetzt.

Dreistellige Anzeige des Registerzeigers ([Rnnn]). Der Registerzeiger wechselt automatisch von zwei- auf dreistellige Anzeige, wenn er von R_{99} auf R_{100} weiterrückt. Mit [Rnnn] können Sie die Anzeige manuell von zwei auf drei Stellen umschalten. (Wenn Sie von drei- auf zweistellige Anzeige des Registerzeigers zurückschalten, wird die ganz links stehende Ziffer der dreistelligen Registeradresse abgeschnitten.) Bei dreistelliger Anzeige des Zeigers wird die Stoppuhrzeit intern weiterhin auf Hundertstelsekunden genau festgehalten, jedoch nur noch auf Zehntelsekunden genau angezeigt.

Unterdrücken der Anzeige des Registerzeigers. Durch Drücken von [REG≠] wird die Anzeige des Registerzeigers unterdrückt, der Zeiger selbst jedoch intern weiter beibehalten. (Durch abermaliges Drücken von [REG≠] wird der Zeiger wieder angezeigt.) Bei dreistelligem Registerzeiger oder bei negativer Stoppuhrzeit können Sie mit dieser Funktion auch die letzte Stelle des Zeitwerts anzeigen.

Allgemeiner Stoppuhrbetrieb mit Zeitnahme

Zeitnahmen werden vom HP-41 automatisch in aufeinanderfolgenden *Datenspeicher-Registern* abgelegt, die auch sonst zur normalen Datenspeicherung verwendet werden. Solange Sie die Stoppuhr nicht (mit [CLEAR]) auf Null zurücksetzen, werden alle aufeinanderfolgenden Zeiten als akkumulierte Zeiten gespeichert. Sie können soviele Zeiten speichern, wie Sie Register zur Verfügung haben. Wenn Sie eine Zeit stoppen und in einem Register ablegen, wird der in diesem Register zuvor gespeicherte Wert überschrieben. Der gespeicherte Zeitwert kann auch bei ausgeschaltetem Stoppuhr-Tastenfeld wie jeder normale Datenwert abgerufen und verändert werden.

Gestoppte Zeiten werden im ***HH:MM:SS.s*** oder ***HH:MM:SS.ss*** Format angezeigt, sind jedoch in der Form ***HH.MMSSss*** gespeichert. Wenn Sie daher bei ausgeschaltetem Stoppuhrmodus eine Zahl abrufen, die eine gestoppte Zeit repräsentiert, wird die Zahl im ***HH.MMSSss*** Format angezeigt. Wenn Sie jedoch bei eingeschaltetem Stoppuhr-Tastenfeld ein Register abrufen, dessen Inhalt keine gestoppte Zeit repräsentiert, versucht der HP-41 diesen Zahlenwert im ***HH:MM:SS.ss*** Format darzustellen. Zahlenwerte, deren Stundenanteil drei- und mehrstellig ist (***HH*** > 99) können nicht verarbeitet werden; dies führt zu der Meldung ERROR nn. (Werte im ***MM***- oder ***HH***-Bereich, die größer als 59 sind, werden *nicht* als Fehler angezeigt.) Die ERROR nn Meldung können Sie löschen, indem Sie ein Register abrufen, das eine Zeit im richtigen Format enthält. Der Rückrufmodus wird durch Drücken der [CLEAR] Taste verlassen.

Das Diagramm auf Seite 76 in Abschnitt 6 zeigt die Betriebsmodi der Stoppuhr.

Anzeige während der Zeitnahme

Wenn Sie bei normaler Zeitnahme (R*nn*) die [SPLIT] Taste gedrückt halten, zeigt die Stoppuhranzeige die gerade genommene Zeit und die Nummer des Registers, in das die Zeit gespeichert wurde, an. (Die Stoppuhr selbst wird dabei nicht angehalten.) Wenn Sie die [SPLIT] Taste loslassen, wird die laufende Zeit wieder angezeigt und der Registerzeiger rückt weiter.

Bei eingeschaltetem Delta-Stoppmodus (D*nn*) laufen die [SPLIT] Zeitnahmen genauso ab. Jedoch wird bei gedrückter [SPLIT] Taste die Differenz zwischen der gerade gestoppten und der zuletzt gestoppten Zeit angezeigt. Die angezeigte Registernummer deutet das Register an, in dem die gerade genommene Zeit abgelegt wird. (Siehe auch «Anzeige der Differenz gestoppter Zeiten».)

Begrenzung der Registerzeiger

Der HP-41 erzeugt einen Warnton, wenn der Registerzeiger (R*nn* oder D*nn*) automatisch oder manuell (aber nicht mit [BST]) auf das letzte verfügbare Speicherregister bewegt wird. (Dies erfolgt unabhängig davon, ob die Stoppuhr läuft oder angehalten ist.)

Wird der Zeiger im Rückrufmodus (=) auf ein nicht existierendes Register bewegt, schaltet sich das Stoppuhr-Tastenfeld automatisch aus und die Meldung NONEXISTENT wird angezeigt. Dies ist auch im Speichermodus (→) der Fall, wenn Sie bei der Ausführung von [SPLIT] versuchen, eine gestoppte Zeit in einem nicht existierenden Speicherregister zu speichern.

Abruf gespeicherter Zeiten ([RCL])

Durch Drücken von [RCL] schalten Sie den Rückrufmodus der Stoppuhr ein bzw. aus. Die Anzeige zeigt nun den Inhalt des bezeichneten Registers an. *Eine laufende Stoppuhr läuft bei eingeschaltetem Rückrufmodus weiter.* Um die in anderen Registern gespeicherten Zeiten anzuzeigen, müssen Sie lediglich die Zeigeradresse ändern (verwenden Sie dazu die Tasten [SST] und [BST] sowie die Zifferntasten).

Wenn Sie im Rückrufmodus [RCL], [SPLIT], [R/S] oder [CLEAR] drücken, wird der Computer auf normale Stoppuhranzeige zurückgeschaltet. Beachten Sie, daß die Registerzeiger für Zeitenspeicherung und -abruf getrennt verwaltet werden. Dadurch werden beim Umschalten zwischen Speicher- (▷) und Rückrufmodus (=) die entsprechenden Zeigeradressen (*nn* oder *nnn*) jedesmal gespeichert. Wenn Sie dann in den vorhergehenden Anzeigemodus zurückschalten, wird der Zeiger wieder auf die ursprüngliche Adresse gestellt. (Siehe «Die Registerzeiger».) Dies gestattet Ihnen z.B., mehrere Zeiten zu stoppen, danach zur Durchsicht der gerade gestoppten Zeiten in den Rückrufmodus zu schalten (während die Stoppuhr weiterläuft), zurückzuspringen und dann mit der Speicherung fortzufahren, wo Sie abgebrochen haben.

Anzeige der Differenz gestoppter Zeiten ([ΔSPLIT])

Durch die Ausführung von [ΔSPLIT] schalten Sie in den Delta-Stoppmodus. Die Stoppuhranzeige zeigt Zeitdifferenzen (D) statt regulärer Zeitnahmen (R) an.

- Eine *Delta-Zeitnahme* stellt die Differenz zwischen der gerade gestoppten Zeit (derjenigen in D*nn*) und der im direkt vorhergehenden Register gespeicherten Zeit an.
- Wenn der Registerzeiger auf D00 steht, wird nur die gerade im Delta-Stoppmodus gestoppte Zeit angezeigt.

Der Delta-Stoppmodus erlaubt Ihnen die Differenz zwischen zwei Zeitnahmen anzuzeigen, ohne dabei die Stoppuhr anhalten zu müssen. Dieser Modus kann sowohl beim Speichern (▷D*nn*) als auch beim Abruf (=D*nn*) von Zeiten verwendet werden.

Im Delta-Stoppmodus können Sie mit [SPLIT] oder [RCL] sehr schnell die Differenz zweier gestoppter Zeiten in benachbarten Registern bestimmen. Die Delta-Zeitfunktion erlaubt Ihnen Zeitabläufe zu stoppen, die sich nicht überschneiden, die jedoch so dicht aufeinanderfolgen, daß Sie keine Zeit haben, die Stoppuhr zurückzusetzen. So können Sie z.B. die Start- und Endzeiten einer Reihe von sehr nahe beieinanderliegenden, sich aber nicht überschneidenden Ereignissen jeweils zusammen festhalten. Durch Abruf der Differenz gepaarter Start- und Endzeiten erhalten Sie somit jeweils die eigentliche Dauer der Ereignisse.

Vergleich Speicherung-Abruf. Delta-Zeitnahmen werden weder gespeichert noch abgerufen, sie werden berechnet. Durch Drücken von [SPLIT] werden auch im Delta-Stoppmodus (▷D*nn*) reguläre Zeiten gestoppt. Die festgehaltene Anzeige (während die [SPLIT] Taste gedrückt ist) *zeigt* jedoch eine Zeitdifferenz an. «Abrufen» von Zeitdifferenzen im Delta-Stoppmodus (=D*nn*) beinhaltet *nicht* das Zurückrufen von Delta-Zeitnahmen, sondern die Anzeige der ermittelten Differenz zweier gespeicherter Zeiten.

Die Funktion von [SPLIT] oder [RCL] (Rückrufmodus) wird daher durch den Delta-Stoppmodus nicht verändert. Bei feststehender Anzeige wird im Delta-Stoppmodus immer eine Zeitdifferenz dargestellt und nicht eine gestoppte Zeit. (Die Anzeige der laufenden Stoppuhr wird nicht verändert.)

Negative Delta-Zeitnahmen und Fehler. Wenn die zweite von zwei aufeinander folgenden Zeitnahmen *nicht* größer ist als die erste, hängt die Anzeige im Delta-Stoppmodus davon ab, ob Zeitnahmen gespeichert oder abgerufen werden: Bei Speicherung gestoppter Zeiten (→*Dnn*) wird nicht eine Zeitdifferenz, sondern nur der Inhalt des momentanen Registers angezeigt. Bei Abruf von Delta-Zeitnahmen (=*Dnn*) wird die Meldung **ERROR =*Dnn*** angezeigt.

> Hinweis: Um die Fehlermeldung ERROR =*Dnn* oder ERROR =*Rnn* zu löschen, müssen Sie die Adresse des Registerzeigers ändern. (Wenn Sie [CLEAR] drücken, schalten Sie zwar den Rückrufmodus aus, die Fehlerbedingung wird jedoch nicht gelöscht.)

Wenn Sie im Delta-Stoppmodus arbeiten und dabei der Inhalt eines der zwei Register, deren Differenz Sie abrufen wollen, nicht eine Zahl im ***HH.MMSSss*** Format enthält, wird ebenfalls die Meldung ERROR =*Dnn* angezeigt (siehe Seite 271).

Programmierbare Stoppuhrfunktionen

Der HP-41 verfügt über sechs programmierbare Stoppuhrfunktionen: Zwei zum Einschalten des Stoppuhr-Tastenfelds ([SW] und [SWPT]) und vier weitere ([RUNSW], [STOPSW], [SETSW] und [RCLSW]) zur Stoppuhrsteuerung bei ausgeschaltetem Stoppuhr-Tastenfeld. Sie können die letzteren vier Funktionen benutzen, um die Stoppuhr zu stellen oder um einen internen Timer während der Ausführung eines Programms laufen zu lassen. (Diese Funktionen können Sie nicht über das Stoppuhr-Tastenfeld ausführen.)

Alle Stoppuhrzeiten, die Sie in das X-Register eingeben oder abrufen, sollten dem ±***HH.MMSSss*** Format entsprechen.

Starten und Anhalten der Stoppuhr ([RUNSW], [STOPSW])

Starten der Stoppuhr. Die Funktion [RUNSW] startet die Stoppuhr.

Anhalten der Stoppuhr. Die Funktion [STOPSW] hält die Stoppuhr an.

Einstellen und Abrufen der momentanen Stoppuhrzeit ([SETSW], [RCLSW])

Einstellen der Stoppuhr. Die [SETSW]-Funktion stellt die Stoppuhr auf die im X-Register spezifizierte Zeit (±***HH.MMSSss***) ein.

Zeiten, die außerhalb des Bereichs von −99.595999 bis +99.595999 liegen, können nicht verarbeitet werden und führen zu der Meldung DATA ERROR. Zusätzliche Ziffern nach ***ss*** werden ignoriert.

Läuft die Stoppuhr bei der Ausführung von [SETSW], wird die Stoppuhr auf die *neue Zeit eingestellt* und läuft danach weiter.

Stoppuhrabruf. Die Funktion [RCLSW] überträgt die gegenwärtige Zeit der Stoppuhr im ***HH.MMSSss*** Format in das X-Register. Dabei wird der Stack angehoben (nur bei freigegebenem Stack Lift).

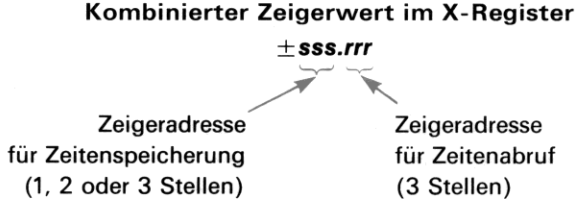
Gleichzeitiges Einstellen der Stoppuhr und der Stoppuhrzeiger ([SWPT])

Die Funktion [SWPT] (*stopwatch and pointers*) entspricht der Funktion [SW], positioniert jedoch zusätzlich die Zeiger der Stoppuhr. Damit kann ein Programm die Zeiger so einstellen, daß das Überschreiben von bereits durch das Programm gespeicherten Daten mit gestoppten Zeiten vermieden wird.

Verwendung von [SWPT]:

- Beim Einschalten des Stoppuhr-Tastenfelds werden die Registerzeiger für Speicherung (⊦R*nn*, ⊦D*nn*) und Abruf (=R*nn*, =D*nn*) von Zeiten auf den im X-Register spezifizierten *sss.rrr*-Wert eingestellt.
- Beim Ausschalten des Stoppuhr-Tastenfelds werden die momentanen Zeigeradressen im *sss.rrr*-Format im X-Register abgelegt. Der zuvor eingegebene *sss.rrr*-Wert wird in das LAST X Register gerettet.

Der Ausgabewert von *sss* kann zur Berechnung der Anzahl der gestoppten Zeiten benutzt werden, *wenn die Zeiten in aufeinanderfolgenden Registern abgelegt wurden* und der Registerzeiger für die Zeitnahmenspeicherung nicht manuell geändert wurde.



Ein *positiver Wert oder der Wert Null* für *sss.rrr* schaltet die Stoppuhr in den regulären Stoppmodus (⊦R*nn*), ein *negativer Wert* in den Delta-Stoppmodus (⊦D*nn*).

Spezifizieren Sie einen Wert zwischen –0.001 und Null, um die Stoppuhr gleichzeitig in den Delta-Stoppmodus zu schalten und beide Zeiger auf Null zu setzen. Wenn beim Ausschalten des Stoppuhr-Tastenfelds beide Zeiger auf Null gesetzt werden und der Delta-Stoppmodus eingeschaltet ist, wird der Wert –0.0000001 im X-Register abgelegt.

Sollte der unwahrscheinliche Fall auftreten, daß beim Ausschalten des Stoppuhr-Tastenfelds ein Zeiger nicht definiert ist, wird der in das X-Register übertragene Zeigerwert (*sss* oder *rrr*) den Wert Null haben. (Dieser Fall tritt z.B. bei unvollständiger Registerzeigeradresse auf.)

Die Stoppuhr als Countdown Timer

Wenn Sie die Stoppuhr (mit [SETSW]) auf eine *negative Zeit* einstellen und dann starten, wird beim Erreichen von 00:00:00.00 ein *Timeralarm* ausgelöst.

- Wenn Stoppuhr-Tastenfeld und -Anzeige *ausgeschaltet* sind, entspricht die Tonfolge des Timeralarms derjenigen eines Meldealarms und TIMER ALARM wird angezeigt. Durch Drücken einer beliebigen Taste können Sie den Alarm beenden. Sie müssen ihn jedoch nicht bestätigen, da er nicht im Speicher festgehalten wird und folglich nicht überfällig werden kann.
- Bei *eingeschaltetem* Stoppuhr-Tastenfeld besteht der Timeralarm lediglich aus zwei Tonzeichen. (Die Anzeige der laufenden Stoppuhr wird nicht unterbrochen.)

Die laufende Stoppuhr hält auf keinen Fall automatisch an. Nach dem Erreichen von Null beginnt sie vorwärts zu zählen. Führen Sie zum Anhalten der Stoppuhr [STOPSW] (nicht über das Stoppuhr-Tastenfeld) aus oder drücken Sie [R/S] (auf dem Stoppuhr-Tastenfeld).

Ausdrucken gespeicherter Zeiten

Wenn Sie einen Drucker HP 82143A oder einen HP-IL Drucker HP 82162A zur Verfügung haben, können Sie eine gespeicherte Zeit ausdrucken, indem Sie die [ATIME24]-Funktion (siehe Abschnitt 15) in Verbindung mit der [PRA]-Funktion (*print Alpha*) ausführen. (Schlagen Sie dazu in Ihrem Drucker- oder HP-IL-Handbuch nach.) Bei eingeschaltetem Stoppuhr-Tastenfeld können Sie keine Zeiten ausdrucken.

Da Zeitdifferenzen (Delta-Zeitnahmen) nicht gespeichert werden, müssen sie vor dem Ausdrucken erst noch einmal berechnet werden. Sie können dazu die Funktion [HMS−] verwenden (jedoch nicht über das Stoppuhr-Tastenfeld):

1. Rufen Sie die spätere Zeit aus dem jeweiligen Speicherregister ab.
2. Rufen Sie die frühere Zeit aus dem jeweiligen Speicherregister ab.
3. Führen Sie [HMS−] aus.
4. Löschen Sie das Alpha-Register (wenn gewünscht).
5. Führen Sie [ATIME24] aus.
6. Führen Sie [PRA] aus.

Zum Ausdrucken normaler Zeiten überspringen Sie einfach die Schritte 1 bis 3 und rufen die gewünschte Zeit ab.

Beispiel für ein Stoppuhrprogramm

Wenn Sie das folgende Anwendungsprogramm verwenden wollen, müssen Sie es zuerst anhand des gegebenen Programm-Listings in den Programmspeicher eingeben. Wenn Sie einen optischen Lesestift HP 82153A besitzen, können Sie das Programm über den Strichcode in Anhang J («Strichcode-Listings der Beispielprogramme») einlesen.

Das folgende Programm bereitet die Stoppuhr für Zeitnahmen vor, schaltet das Stoppuhr-Tastenfeld ein und druckt bei ausgeschaltetem Stoppuhr-Tastenfeld eine bestimmte Anzahl gespeicherter Zeiten aus. (Zur Ausführung dieses Programms muß ein Drucker angeschlossen sein.) Die Zeiten werden im ***HH:MM:SS.ss*** Format ausgedruckt. Die aus jedem Register abgerufenen Zahlenwerte müssen kleiner als 100 sein; ein größerer Wert verursacht einen Fehler.

Eingabe (nach Aufforderung):

1. Die Nummer des ersten Registers, dessen Zeit Sie ausdrucken wollen.
2. Die Nummer des letzten Registers, dessen Zeit Sie ausdrucken wollen.

Ausgabe: Eine Liste der angegebenen Register samt Inhalt. Der Inhalt wird im Zeitformat ausgedruckt.

Benutzeranweisungen:

1. Geben Sie das Programm ZEITEN ein und starten Sie es. Das Programm schaltet automatisch das Stoppuhr-Tastenfeld ein. Nach dem Erscheinen der Stoppuhr-Anzeige können Sie Zeiten stoppen.
2. Beginnen Sie nun die Zeiten zu stoppen. Sie können mit jedem beliebigen Register beginnen, jedoch müssen die gestoppten Zeiten in direkter Reihenfolge in die Register abgelegt werden.
3. Wenn Sie alle Zeiten gestoppt haben, drücken Sie [EXIT].
4. Auf die Anzeige ERSTES REG? hin geben Sie das erste Register an, dessen Inhalt (Zeit) Sie ausdrucken wollen. Wenn Sie mit R_{00} beginnen wollen, müssen Sie keine Zahl eingeben. Drücken Sie anschließend [R/S].
5. Auf die Anzeige LETZT. REG? hin geben Sie das letzte Register an, dessen Inhalt Sie ausdrucken wollen. Drücken Sie dann [R/S].

Programm-Listing

01 LBL^T ZEITEN	
02 STOPSW	Stellt sicher, daß die Stoppuhr angehalten ist.
03 0	
04 SETSW	Setzt die Stoppuhr auf 00:00:00.00 zurück.
05 SW	Schaltet das Stoppuhr-Tastenfeld ein. Nachdem alle gewünschten Zeiten gestoppt wurden und Sie die Stoppuhr ausgeschaltet haben, wird das Programm fortgesetzt.
06^T ERSTES REG?	Wenn keine Zahl eingegeben wird, ist das erste Register R_{00} (aufgrund der Null in Zeile 03).
07 PROMPT	
08^T LETZT. REG?	
09 PROMPT	
10 ADV	Vorschub des Druckerpapiers.
11 RCLFLAG	
12 X<> Z	Hält im Z-Register fest, welche Flags gesetzt sind.
13 LBL 00	
14 FIX 0	Die Registernummer wird ohne Dezimalstellen dargestellt.
15 CF 29	In Verbindung mit [FIX] 0 wird das Dezimaltrennzeichen unterdrückt.
16^T R	Aufbau des Ausgabestrings (R für Register).
17 100	Zeilen 17 bis 23 dienen dem Ausgabeformat: Wenn die Registernummer kleiner als 100 ist, wird ein Freiraum zwischen dem R und der Nummer gelassen, da in diesem Fall die Programmzeile 20 eine Leerstelle an das R in Zeile 16 anfügen läßt. Ist die Registernummer kleiner als 10, wird eine Null vor die Registernummer gesetzt. Dadurch erscheinen auf dem Ausdruck die Registernummern und Inhalte bündig.
18 X<=Y?	
19 GTO 01	
20^T ⊢	
21 SQRT	
22 X>Y?	
23^T ⊢0	

24 LBL 01	
25 RDN	Lädt die Adresse des ersten Registers nach X.
26 ARCL X	Kopiert die Adresse vom X-Register in den String im Alpha-Register.
27 T⊢ =	
28 RCL IND X	Ruft den Wert des Registers ab, dessen Adresse im X-Register steht.
29 FIX 6	
30 ATIME24	Hängt diesen Wert im Zeitformat an den Inhalt des Alpha-Registers an.
31 PRA	Druckt den gesamten Inhalt des Alpha-Registers.
32 RDN	Erhöht die Nummer des ersten Registers um eins und fährt im Programm fort, wenn die Nummer des ersten Registers kleiner oder gleich der Nummer des letzten Registers ist.
33 1	
34 +	
35 X<=Y?	
36 GTO 00	
37 RCL Z	
38 STOFLAG	Stellt die ursprünglichen Flagzustände wieder her.
39 CLX	Löscht das X-Register.
40 END	

Beispiel eines Ausdrucks: Nachfolgend ist ein Ausdruck von Zeiten abgebildet, die in den Registern 00 bis 10 gespeichert sind.

```
            XEQ "ZEITEN"
ERSTES REG?
                        RUN
LETZT. REG?
          10.0000       RUN

R 00 =00:00:01.52
R 01 =00:00:03.31
R 02 =00:00:04.54
R 03 =00:00:06.07
R 04 =00:00:07.71
R 05 =00:00:09.63
R 06 =00:00:11.73
R 07 =00:00:14.09
R 08 =00:00:16.38
R 09 =00:00:18.55
R 10 =00:00:20.27
```

Teil V
Programmierung im Detail

Abschnitt 18

Einfache Programmierung

Inhalt

Laden eines Programms 280
Eintasten eines Programms 280
Kopieren eines ROM-Programms 281
Vergrößern des Programmspeichers 281
Ausführen eines Programms 282
Programmzeilen 282
Nicht programmierbare Operationen 283
Positionieren des Programmzeigers 283
Verwenden von GTO 283
Verwenden von Katalog 1 284
Verwenden von SST und BST 284
Weitere Methoden 285
Modifizieren eines Programms 285
Löschen von Anweisungen 285
Einfügen von Anweisungen 286
Löschen von Programmen 286
Verwenden von CLP 287
Verwenden von PCLPS 287

Laden eines Programms

Eintasten eines Programms

1. Schalten Sie durch Drücken von PRGM den Programm-Modus ein.
2. Führen Sie die Sequenz GTO · · aus, um den Programmzeiger auf das Ende des Programmspeichers zu positionieren.
3. Drücken Sie die Taste LBL und tasten Sie ein globales Label ein.
4. Tasten Sie die Programmanweisungen ein. Verwenden Sie dazu wie im Ausführungs-Modus das Normal-, Alpha- oder User-Tastenfeld.
5. Schließen Sie das Programm durch die Sequenz GTO · · ab (nicht unbedingt erforderlich).

Das Ausführen der Tastenfolge GTO · · hat die folgenden Auswirkungen:

- Der Hauptspeicher wird gepackt. Dadurch wird sichergestellt, daß für das nächste Programm (bzw. Tastenzuordnung oder Alarm) der maximal mögliche Speicherplatz verfügbar ist.

- An das vorausgehende Programm wird eine [END] Anweisung angehängt und es wird ein Null-Programm (bestehend aus der permanenten .END. Anweisung) am Ende des Programmspeichers erzeugt. (Ein Hauptgrund, die Sequenz [GTO] [.] [.] am Ende einer Programmeingabe auszuführen, besteht darin, eine [END] Anweisung an das Programm anzuhängen. In Katalog 1 wird das **END** zusammen mit der von dem Programm belegten Anzahl von Bytes angezeigt.)
- Der Programmzeiger wird auf das Null-Programm positioniert und in der Anzeige erscheint **00 REG nnn**; dabei repräsentiert *nnn* die Anzahl der ungebundenen Register, die noch für Programmierzwecke zur Verfügung stehen. Wenn Sie danach Anweisungen eintasten, werden diese als neues Programm am Ende des Programmspeichers aufgezeichnet.

Die Anzahl der verfügbaren Register wird auch zusammen mit der permanenten **END** Anweisung (.END.) angezeigt. Wenn die letzte Programmzeile angezeigt wird und Sie die Taste [SST] drücken, erscheint .END. **REG nnn** in der Anzeige (dabei bedeutet *nnn* wieder die Anzahl der ungebundenen Register). Sie können fortfahren, Programmanweisungen einzutasten. Um Ihr Programm zur Kontrolle anzuzeigen, können Sie folgendermaßen vorgehen:

- Drücken Sie die Taste [SST], um den Programmzeiger auf Zeile 01 zu positionieren.
- Drücken Sie die Taste [BST], um den Programmzeiger auf die zuletzt eingegebene Programmzeile zu positionieren.

Kopieren eines ROM-Programms

Um ein in einem Festwertspeicher (im Englischen ROM «*Read Only Memory*») wie z.B. einem Applikations-Modul aufgezeichnetes Programm zu verändern, müssen Sie das Programm zuerst in den Programmspeicher kopieren. Führen Sie die Funktion [COPY] aus und spezifizieren Sie ein globales Label des ROM-Programms als Parameter, um das ROM-Programm an das Ende des Programmspeichers zu kopieren.

Vergrößern des Programmspeichers

Wenn der Speicherplatz nicht ausreicht, um eine Anweisung zu speichern oder ein Programm zu kopieren, zeigt der Computer die Meldungen **PACKING** und **TRY AGAIN** an. Wiederholen Sie die gewünschte Operation. Wenn wieder die Meldung **TRY AGAIN** erscheint, müssen Sie dem Programmspeicher mehr Register zuordnen. Dazu stehen Ihnen mehrere Möglichkeiten zur Verfügung:

- Führen Sie die Funktion [SIZE?] aus, um zu überprüfen, wieviele Register der Datenspeicherung zugewiesen sind. Sie können diese Anzahl verringern, indem Sie die Funktionen [SIZE] oder [PSIZE] ausführen.
- Löschen Sie durch Ausführen der Funktion [CLP] oder [PCLPS] Programme aus dem Programmspeicher. Wenn Sie vor der Löschoperation die Funktion [SAVEP] ausführen, werden Kopien der Programme im erweiterten Speicher abgelegt.
- Löschen Sie einen oder mehrere Alarme.
- Löschen Sie eine oder mehrere Programmzeilen; führen Sie dann die Funktion [PACK] bzw. die Tastenfolge [GTO] [.] [.] aus.
- Heben Sie Tastenzuordnungen des User-Tastenfelds auf und führen Sie die Funktion [PACK] bzw. die Tastenfolge [GTO] [.] [.] aus. (Durch Aufheben von Tastenbelegungen, die mit globalen Labelzuordnungen in Katalog 1 aufgelistet werden, lassen sich keine Register freisetzen.)

Ausführen eines Programms

Zur Ausführung eines Programms muß sich der Computer im Ausführungs-Modus befinden. Sie können das Programm dann auf mehrere Arten starten:

- Drücken Sie die Taste [XEQ] und spezifizieren Sie ein globales Label des Programms. Die Programmausführung beginnt bei diesem globalen Label.
- Ordnen Sie ein globales Label einer Taste des User-Tastenfelds zu und drücken Sie diese Taste, wenn der User-Modus aktiviert ist. Die Programmausführung beginnt bei dem globalen Label.
- Positionieren Sie den Programmzeiger auf den Anfang des Programms und drücken Sie die Taste [R/S]. Die Programmausführung beginnt mit der momentanen Programmzeile.
- Positionieren Sie den Programmzeiger auf den Anfang des Programms und drücken Sie die Taste [SST]. Dadurch wird die momentane Programmzeile ausgeführt und der Zeiger wird auf die nächste Zeile positioniert. Die schrittweise Ausführung eines Programms erweist sich als sehr hilfreich bei der Fehlersuche. Überprüfen Sie das Ergebnis nach Ausführung jeder Programmzeile, um festzustellen, an welcher Stelle ein Fehler auftritt.
- Positionieren Sie den Programmzeiger auf den Anfang des Programms, setzen Sie Flag 11 und schalten Sie den Computer aus. Wenn Sie den Computer wieder einschalten, wird das Programm automatisch ausgeführt. Die Programmausführung beginnt mit der momentanen Zeile.
- Stellen Sie einen Steueralarm ein, um das Programm zu einem bestimmten Zeitpunkt auszuführen.

Während ein Programm abläuft, wird der **PRGM**-Indikator angezeigt. Zusätzlich erscheint ein Indikator für laufende Programmausführung (⁾) in der Anzeige; jedesmal, wenn der Computer ein Label findet, rückt das ⁾ um eine Stelle nach rechts. Der Indikator für laufende Programmausführung verschwindet aus der Anzeige, sobald mittels [AVIEW] eine Meldung angezeigt wird oder [ED] das Texteditor-Tastenfeld einschaltet.

Programmzeilen

Im Programm-Modus zeigt der Computer die momentane Zeile im Programmspeicher an. Von Ihnen eingetastete Anweisungen werden automatisch als Programmzeilen aufgezeichnet. Die Programmzeilen sind durchnumeriert; jedes Programm beginnt mit Zeile 01. Jede Zeile enthält eine Anweisung, wahlweise bestehend aus:

- einer Funktion,
- einem Alpha-String mit bis zu 15 Zeichen,
- einer Zahl mit bis zu 10 Ziffern bzw. einer Mantisse mit bis zu 10 Ziffern und einem zweistelligen Exponenten zur Basis 10.

Das Eingeben von Zahlen und Zeichen wird ausführlich in Abschnitt 9, «Das Tastenfeld und die Anzeige», behandelt.

Das hochgestellte T (^{T}) am Anfang einer Programmzeile kennzeichnet den Zeileninhalt als Alpha-String bzw. als globales Label, wenn XEQ, GTO oder LBL vorangeht. Um eine Funktion unter ihrem Alpha-Namen aufzuzeichnen, muß zuvor die Taste [XEQ] gedrückt werden. Wenn [XEQ] nicht gedrückt wird, interpretiert das Programm die eingegebenen Zeichen als Alpha-String. Bei ablaufendem Programm führt der Computer dann nicht die gewünschte Funktion aus, sondern legt den Zeichenstring im Alpha-Register ab.

Nicht programmierbare Operationen

Die unten aufgeführten Operationen können nicht als Programmanweisungen aufgezeichnet werden. Einige haben jedoch ein programmierbares Äquivalent, das hinter der nicht programmierbaren Operation in Klammern angegeben ist.

- Löschoperationen:
  [←], [DEL].
  [CLP] (die Funktion [PCLPS] ist programmierbar).
- Operationen zum Positionieren des Programmzeigers:
  [GTO] [·] [·], [GTO] [·], [SST], [BST].
- Alle Kataloge (die Funktionen [EMDIR] und [ALMCAT] sind programmierbar).
- Die Modustasten:
  [ON] (die Funktion [OFF] ist programmierbar).
  [PRGM].
  [USER] (ein Programm kann jedoch Flag 27 setzen bzw. löschen).
  [ALPHA] (die Funktionen [AON] und [AOFF] sind programmierbar).
- Weitere nicht programmierbare Operationen:
  [COPY], [ON], [PACK], [RUN].
  [ASN] (die Funktion [PASN] ist programmierbar).
  [SIZE] (die Funktion [PSIZE] ist programmierbar).

Positionieren des Programmzeigers

Es existieren verschiedene Methoden, um den Programmzeiger auf eine bestimmte Stelle des Programmspeichers zu positionieren. Einige dieser Methoden erlauben Ihnen den Zugriff auf jedes gespeicherte Programm (d.h. jedes globale Label), andere lassen Sie auf eine beliebige Programmzeile zugreifen. Nicht alle der beschriebenen Operationen wirken sowohl im Ausführungs- als auch im Programm-Modus.

Verwenden von [GTO] [·]

Die folgenden Operationen lassen sich sowohl im Programm- als auch im Ausführungs-Modus verwenden:

- Um den Programmzeiger auf ein globales Label zu positionieren, können Sie folgendermaßen vorgehen: Führen Sie die Tastenfolge [GTO] [·] aus und spezifizieren Sie das gewünschte globale Label. Der Computer beginnt bei dem letzten globalen Label (d.h. dem von Katalog 1 als letztes aufgeführten Label) und durchsucht den Programmspeicher von unten nach oben. Der Programmzeiger wird auf das erste globale Label positioniert, das mit dem spezifizierten Label übereinstimmt.
- Um den Programmzeiger auf die Zeile *nnn* des momentanen Programms zu positionieren, führen Sie die Tastenfolge [GTO] [·] *nnn* aus. Wenn die Zahl *nnn* größer ist als die Nummer der letzten Zeile des Programms, wird der Programmzeiger auf die letzte Zeile positioniert.
- Um den Programmzeiger auf Zeile 1*nnn* (Zeilennummer > 999) zu positionieren, können Sie folgendermaßen vorgehen: Führen Sie die Tastenfolge [GTO] [·] [EEX] aus. In der Anzeige erscheint **GTO .1** ___ und der Computer erwartet die Eingabe von *nnn*.

Verwenden von Katalog 1

Nicht immer kann der Programmzeiger mit Hilfe der Sequenz GTO · auf das gewünschte Programm positioniert werden. Dies gilt unter anderem in den folgenden Fällen:

- Das gewünschte Programm enthält kein globales Label.
- Das globale Label ist zweimal im Programmspeicher enthalten und die Sequenz GTO · positioniert den Programmzeiger nicht auf das gewünschte Label.
- Sie kennen die genaue Schreibweise des globalen Labels nicht.

Um den Programmzeiger mit Hilfe von Katalog 1 auf ein beliebiges globales Label bzw. auf eine beliebige END Anweisung zu positionieren, können Sie wie folgt vorgehen. (Es spielt keine Rolle, ob sich der Computer im Programm- oder im Ausführungs-Modus befindet.)

1. Drücken Sie CATALOG 1, um alle im Programmspeicher aufgezeichneten globalen Label und END Anweisungen anzuzeigen.
2. Durch Drücken einer beliebigen Taste (außer ON und R/S) wird die Katalogausgabe beschleunigt.
3. Drücken Sie R/S, um die Katalogausgabe bei dem gewünschten globalen Label bzw. der gewünschten END Anweisung anzuhalten.
4. Drücken Sie die Taste SST, um den nächsten Katalogeintrag anzuzeigen; die Taste BST zeigt den vorangehenden Katalogeintrag an.
5. Wenn Sie durch Drücken der Taste ← den Katalog-Modus verlassen, wird der Programmzeiger auf das zu dem angezeigten Katalogeintrag gehörende Programm positioniert.

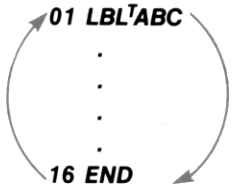
Wenn ein Programm kein globales Label enthält, können Sie den Programmzeiger wie oben beschrieben auf die END Anweisung des Programms positionieren. (Wenn zwei END Anweisungen hintereinander angezeigt werden, gehört die zweite END Anweisung zu einem Programm ohne globales Label.) Sie sollten dann ein globales Label am Programmanfang einfügen: Führen Sie die Tastenfolge GTO · 000 aus, drücken Sie (im Programm-Modus) die Taste LBL und tasten Sie ein globales Label ein.

Verwenden von SST und BST

Im Programm-Modus wird der Programmzeiger durch Drücken der Tasten SST und BST auf die nächste bzw. auf die vorangehende Programmzeile positioniert.

- Drücken Sie die Taste SST, um den Programmzeiger auf die nächste Zeile zu positionieren. Wenn die angezeigte Zeile die letzte Zeile des Programms ist, stellt SST den Zeiger auf die erste Programmzeile (01).
- Drücken Sie die Taste BST, um den Programmzeiger auf die vorangehende Zeile zu positionieren. Wenn die angezeigte Zeile die erste Zeile des Programms ist, stellt BST den Zeiger auf die letzte Programmzeile.

Drücken der Taste SST stellt den Programmzeiger vom Programmende auf den Programmanfang.



Drücken der Taste BST stellt den Programmzeiger vom Programmanfang auf das Programmende.

Weitere Methoden

Wenn sich der Computer im Ausführungs-Modus befindet, stehen Ihnen die folgenden Möglichkeiten zur Positionierung des Programmzeigers zur Verfügung:

Positionieren auf ein globales Label. Drücken Sie die Taste [GTO] und spezifizieren Sie das gewünschte globale Label.

Positionieren auf ein zugeordnetes globales Label. Wenn das globale Label einer Taste des User-Tastenfelds zugeordnet ist, können Sie folgendermaßen vorgehen: Drücken Sie die Taste [R/S], während Sie (bei aktiviertem User-Modus) die zugeordnete Taste gedrückt halten. Lassen Sie die zugeordnete Taste vor der Taste [R/S] los.

Positionieren auf ein numerisches Label des momentanen Programms. Die Tastenfolge [GTO] *nn* positioniert den Programmzeiger auf das Label *nn* ([LBL] *nn*). Der Computer durchsucht das momentane Programm und positioniert den Zeiger auf das erste Label, das mit dem spezifizierten Label übereinstimmt (siehe auch Abschnitt 20, «Programmverzweigungen»).

Positionieren auf den Anfang des momentanen Programms. Drücken Sie die Taste [RTN], um den Programmzeiger auf Zeile 00 des momentanen Programms zu positionieren. Wenn Sie jetzt in den Programm-Modus schalten, erscheint 00 REG *nnn* in der Anzeige; dabei steht *nnn* für die Anzahl der verfügbaren, ungebundenen Register. Wenn Sie eine Eingabe vornehmen, zeichnet der Computer diese als Zeile 01 auf. Dies ist der einfachste Weg, am Programmanfang eine Zeile einzufügen.

Modifizieren eines Programms

Zum Löschen bzw. Hinzufügen von Programmzeilen muß sich der Computer im Programm-Modus befinden.

Löschen von Anweisungen

Löschen von einzelnen Zeilen. Um eine einzelne Anweisung zu löschen, können Sie folgendermaßen vorgehen: Positionieren Sie den Programmzeiger auf die zu löschende Zeile und drücken Sie die Taste [←]. Die momentane Programmzeile wird gelöscht, der Programmzeiger wird auf die vorangehende Zeile positioniert und alle nachfolgenden Zeilennummern werden um eins erniedrigt.

Wenn Sie mehrere aufeinanderfolgende Zeilen löschen wollen, beginnen Sie mit der letzten zu löschenden Zeile (höchste Zeilennummer). In dem nachstehenden Beispiel werden die Zeilen 02 bis 04 gelöscht. Zuerst wird der Programmzeiger auf Zeile 04 positioniert. Drücken der Taste [←] löscht Zeile 04 und positioniert den Programmzeiger auf Zeile 03; nochmaliges Drücken von [←] löscht Zeile 03 und positioniert den Zeiger auf Zeile 02; wenn Sie [←] ein drittes Mal drücken, wird Zeile 02 gelöscht und der Zeiger wird auf Zeile 01 positioniert.

	01 LBL"FLAECHE			
	02 X↑2	01 LBL"FLAECHE		
Momentane	03 PI	02 X↑2	01 LBL"FLAECHE	
Programmzeile	04 *	03 PI	02 X↑2	01 LBL"FLAECHE
(angezeigt)	05 END	04 END	03 END	02 END

Löschen von mehreren Zeilen. Um mehrere aufeinanderfolgende Zeilen auf einmal zu löschen, können Sie folgendermaßen vorgehen:

1. Positionieren Sie den Programmzeiger auf die erste zu löschende Zeile (niedrigste Zeilennummer).
2. Führen Sie die Funktion [DEL] (*delete*) aus.
3. Spezifizieren Sie die Anzahl der zu löschenden Zeilen. Drücken Sie die Taste [EEX], um mehr als 1000 Zeilen zu löschen. In der Anzeige erscheint DEL 1___ und der Computer erwartet als Eingabe die drei verbleibenden Stellen.

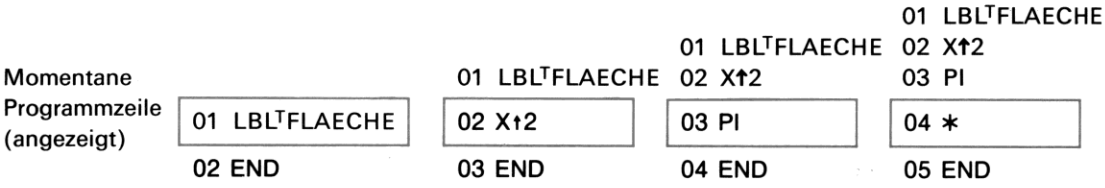
In dem vorausgegangenen Beispiel wurden die Zeilen 02, 03 und 04 eine nach der anderen gelöscht. Eine Alternative dazu wäre, den Programmzeiger auf Zeile 02 zu positionieren und die Funktion [DEL] 003 auszuführen. Dadurch werden die Zeilen 02, 03 und 04 gelöscht und der Programmzeiger wird auf die vorangehende Zeile (Zeile 01) positioniert. Alle nachfolgenden Zeilennummern werden um drei erniedrigt.

Wenn Sie die Funktion [DEL] *nnn* ausführen und weniger als *nnn* Zeilen nach der momentanen Programmzeile vorhanden sind, werden alle Zeilen ab der momentanen Zeile gelöscht (die [END] Anweisung wird *nicht* gelöscht).

Einfügen von Anweisungen

Um eine Anweisung in ein Programm einzufügen, können Sie folgendermaßen vorgehen: Positionieren Sie den Programmzeiger auf die Zeile, die der neuen Zeile vorangehen soll und geben Sie die neue Anweisung ein. (Um eine Zeile zu ersetzen, können Sie die alte Zeile durch Drücken von [←] löschen; dadurch wird der Programmzeiger korrekt positioniert.) Die neue Anweisung wird zur momentanen Zeile und alle nachfolgenden Zeilennummern werden um eins erhöht.

Wenn Sie mehrere Zeilen einfügen wollen, beginnen Sie mit der ersten einzufügenden Zeile (niedrigste Zeilennummer). Die untenstehende Abbildung zeigt, wie die im vorhergehenden Beispiel gelöschten Zeilen wieder eingefügt werden. Zuerst wird der Programmzeiger auf Zeile 01 positioniert. Jede eingegebene Anweisung wird hinter der momentanen Zeile eingefügt; die neue Anweisung wird dann selbst zur momentanen Zeile.



Löschen von Programmen

Der HP-41 verfügt über zwei Funktionen zum Löschen von Programmen. Die nicht programmierbare Funktion [CLP] (*clear program*) dient zum Löschen eines einzelnen Programms; mit der programmierbaren Funktion [PCLPS] (*programmable clear programs*) können mehrere Programme gleichzeitig gelöscht werden.

Verwenden von CLP

Führen Sie die Funktion CLP aus und spezifizieren Sie das globale Label des zu löschenden Programms. Der Computer führt automatisch die folgenden Operationen aus:

1. Der Programmspeicher wird von unten nach oben nach dem spezifizierten globalen Label durchsucht. Die Suche beginnt bei dem letzten globalen Label (d.h. bei dem von Katalog 1 als letztem aufgeführten Label).
2. Alle Anweisungen (Zeile 01 bis END) des ersten Programms, das das spezifizierte globale Label enthält, werden gelöscht.
3. Der Hauptspeicher wird gepackt.

Um das momentane Programm zu löschen, brauchen Sie lediglich CLP mit leerem Alpha-Register auszuführen.

Verwenden von PCLPS

Um ein Programm und alle nachfolgenden Programme aus dem Programmspeicher zu löschen, können Sie folgendermaßen vorgehen:

1. Legen Sie ein globales Label des ersten zu löschenden Programms im Alpha-Register ab.
2. Führen Sie die Funktion PCLPS aus.

Wenn Sie das Alpha-Register löschen und die Funktion PCLPS ausführen, wird das momentane Programm und alle nachfolgenden Programme gelöscht.

Abschnitt 19

Flags

Inhalt

Einführung 288
Flagtypen 289
Benutzerflags (00 bis 10) 289
Steuerflags (11 bis 29) 289
Systemflags (30 bis 55) 291
Zusammenfassung der Flagzustände 292
Flags und das X-Register 292
Flags und Zahlen als Bytes 294
Verwenden von [X<>F] 295
Mehrfache Kopien von Flags 295
Verwenden von [RCLFLAG] und [STOFLAG] 296

Einführung

Ein Flag ist ein Schalter, der zwei Zustände annehmen kann: gesetzt (*set*) oder gelöscht (*clear*). Die Zustände der Flags können als «ja/nein» (wie bei einer Entscheidung) oder «1/0» (wie bei einer binären Stelle) interpretiert werden. Die Flags des HP-41 lassen sich in drei verschiedene Gruppen einteilen:

Benutzerflags. Sie können die Benutzerflags setzen, löschen und abfragen; der Status dieser Flags wird nur von Ihren Anweisungen beeinflußt.

Steuerflags. Sie können Steuerflags setzen, löschen und abfragen. Beim Einschalten stellt der Computer den voreingestellten Status einiger Flags wieder her. Beim Ausführen mancher Operationen ändert der HP-41 den Status bestimmter Flags.

Systemflags. Sie können Systemflags nur abfragen, nicht aber setzen oder löschen.

Die Benutzer- und Steuerflags (00 bis 29) können von Ihnen gesetzt und gelöscht werden.

- Um einen Flag zu setzen, drücken Sie die Taste [SF] (*set flag*); spezifizieren Sie dann die gewünschte Flagnummer.
- Um einen Flag zu löschen, drücken Sie die Taste [CF] (*clear flag*); spezifizieren Sie dann die gewünschte Flagnummer.

Sie können den Status der Flags 00 bis 55 abfragen, indem Sie die Taste [FS?] (*flag set?*) drücken und dann die gewünschte Flagnummer spezifizieren. Wenn die Meldung YES in der Anzeige erscheint, ist der Flag gesetzt; wenn NO angezeigt wird, ist der Flag gelöscht. Das Abfragen von Flags (z.B. durch [FS?]) wird vor allem zur Steuerung der Programmausführung verwendet (siehe Abschnitt 20, «Programmverzweigungen»).

Flagtypen

Benutzerflags (00 bis 10)

Die Benutzerflags sind ausschließlich für Ihren Gebrauch bestimmt; die Bedeutung dieser Flags hängt von Ihrer Definition ab. Beispiel: Ein Programm kann den Benutzer fragen, ob als Währungseinheit einer Berechnung die Deutsche Mark oder der Schweizer Franken verwendet werden soll; die Antwort des Benutzers wird als Status eines der Benutzerflags gespeichert. Zum Überprüfen der gewählten Währungseinheit kann zu jedem Zeitpunkt der Flagstatus abgefragt werden.

Der Status der Benutzerflags wird vom Permanentspeicher aufrechterhalten. Wenn Sie einen Flag setzen oder löschen, bleibt dieser Zustand so lange gültig, bis Sie den Flagstatus wieder explizit ändern. Wenn einer der ersten fünf Benutzerflags gesetzt wird, erscheint der entsprechende Indikator (**0**, **1**, **2**, **3** oder **4**) in der Anzeige.

Die ersten acht Flags können zur Darstellung der acht Bits eines Bytes verwendet werden; dieses Byte kann in eine Zahl umgewandelt und im X-Register abgelegt werden. Diese Operation wird im Anschluß an die Steuer- und Systemflags erläutert.

Steuerflags (11 bis 29)

Die Steuerflags haben eine besondere Bedeutung für den Computer (siehe unten). Der Status dieser Flags beeinflußt die Arbeitsweise Ihres HP-41. Sie können diese Flags verändern und damit einen gewünschten Operationsmodus auswählen. Einige dieser Flags werden vom Computer verändert, um damit auf bestimmte Betriebsbedingungen hinzuweisen; Sie können dann den Status dieser Flags überprüfen.

Flag 11: Automatische Ausführung. Flag 11 ermöglicht die automatische Ausführung von Programmen. Wenn Sie vor Ausschalten des Computers Flag 11 setzen, werden beim Wiedereinschalten die folgenden Operationen ausgeführt:

- Ein Tonsignal wird erzeugt.
- Die Programmausführung beginnt mit der momentanen Programmzeile.
- Flag 11 wird gelöscht.

Flags 12 bis 20: Steuerung von Peripheriegeräten. Diese Flags wirken auf die Operationen der vom Computer gesteuerten Peripheriegeräte und werden bei jedem Einschalten des Computers gelöscht. Die genaue Bedeutung dieser Flags hängt vom Typ der angeschlossenen Einheiten ab; entnehmen Sie Einzelheiten bitte der zu Ihren Peripheriegeräten gehörenden Benutzerdokumentation.

Flag 21: Druckersteuerung. Bei Anwesenheit eines Druckers erlaubt Flag 21 einem Programm, die Ausführung von Funktionen wie VIEW und AVIEW zu steuern. Siehe Anhang D, «Druckeroperationen».

Flags 22 und 23: Dateneingabe. Diese Flags ermöglichen es einem Programm, die nach einer Eingabeaufforderung erfolgte Dateneingabe zu identifizieren.

- Flag 22 ist gesetzt, wenn Zahlen in das X-Register eingetastet werden.
- Flag 23 ist gesetzt, wenn Zeichen in das Alpha-Register eingetastet werden.

Diese Flags werden beim Einschalten des Computers automatisch gelöscht. Wenn eine Abfrageoperation vorgenommen werden soll, ist es zu empfehlen, die Flags vor jedem Anzeigen einer Eingabeaufforderung zu löschen.

Flags 24 und 25: Ignorieren von Fehlern. Normalerweise unterbricht eine Fehlerbedingung die Programmausführung. Die Flags 24 und 25 ermöglichen es, unnötige Programmunterbrechungen zu vermeiden; auftretende Fehlerbedingungen können in die Programmierung mit einbezogen werden.

- Wenn Flag 24 gesetzt ist, ignoriert der Computer *alle* Bereichsfehler (**OUT OF RANGE**). Dieser Fehler resultiert normalerweise (Ausnahme: statistische Summationen) aus Berechnungen, für deren Ergebnis x gilt: $|x| > 9.999999999 \times 10^{99}$. Bei gesetztem Flag 24 wird $\pm 9.999999999 \times 10^{99}$ als Näherung für den genauen Wert zurückgegeben und die Programmausführung wird fortgesetzt.

  Flag 24 wird bei jedem Einschalten des Computers gelöscht. Wenn Flag 24 gesetzt ist, wird dieser Status aufrechterhalten, bis entweder der Flag gelöscht wird oder der Computer ausgeschaltet wird. Benutzen Sie Flag 25, wenn Sie bei Auftreten einer Fehlers nicht den Näherungswert ($\pm 9.999999999 \times 10^{99}$) verwenden wollen, sondern ein von Ihnen vorgesehenes Unterprogramm ausführen wollen.

- Wenn Flag 25 gesetzt ist, ignoriert der Computer *nur einen* Fehler von jedem Typ; dabei wird Flag 25 automatisch gelöscht. Die Anweisung, die den Fehler verursacht, wird nicht ausgeführt. Flag 25 wird bei jedem Einschalten des Computers gelöscht.

  Wenn Flag 24 und Flag 25 gesetzt sind, ist Flag 24 für einen **OUT OF RANGE** Fehler zuständig – Flag 25 wird *nicht* gelöscht. Wenn Flag 25 gesetzt und Flag 24 gelöscht ist, wird bei bereichsüberschreitenden Ergebnissen (**OUT OF RANGE**) *nicht* der Näherungswert $\pm 9.999999999 \times 10^{99}$ in dem betreffenden Register abgelegt.

  Zum Auffinden eines Fehlers können Sie wie folgt vorgehen: Setzen Sie Flag 25 vor einer Anweisung und fragen Sie den Flag nach Ausführen der Anweisung ab. (Sie sollten Flag 25 nach der Anweisung abfragen *und löschen*; sonst kann es passieren, daß der Computer einen nicht vorhergesehenen Fehler ignoriert.) Flag 25 ermöglicht es, bei Auftreten eines Fehlers eine Programmverzweigung statt einer Programmunterbrechung vorzusehen.

Flag 26: Steuerung von akustischen Signalen. Bei gesetztem Flag 26 erzeugen [BEEP], [TONE], Alarme und die Stoppuhr akustische Signale. Flag 26 wird bei jedem Einschalten des Computers gesetzt. (Dies ist der einzige Flag, der im Einschaltzustand *gesetzt* ist.) Sie können die akustischen Signale des Computers abstellen, indem Sie Flag 26 löschen.

Flag 27: User-Tastenfeld. Flag 27 ist gesetzt, wenn das User-Tastenfeld aktiviert ist – d.h. wenn der **USER**-Indikator angezeigt wird. Ein Programm kann diesen Flag abfragen und verändern; damit kann auch ein Programm das User-Tastenfeld verwenden (die Modustaste [USER] ist nicht programmierbar). Der Status von Flag 27 wird vom Permanentspeicher aufrechterhalten.

Flags 28 und 29: Ziffern- und Dezimaltrennzeichen. Diese Flags steuern die Verwendung von Punkt und Komma in numerischen Anzeigen; der Status dieser Flags wird vom Permanentspeicher aufrechterhalten. (Siehe Abschnitt 9 unter «Anzeigeformat».)

Programmbeispiele. Die Programme **ZE** und Σ in Abschnitt 22 verwenden die Flags 22 und 23 (Dateneingabe), wenn die voreingestellten Werte für Uhrzeit, Datum und Tätigkeit angezeigt werden. Der Benutzer kann danach einen davon abweichenden Wert eingeben und dann die Taste [R/S] drücken; um die angezeigten Werte zu bestätigen, reicht es aus, die Taste [R/S] zu drücken. Um herauszufinden, ob der Benutzer einen neuen Wert eingibt, löscht das Programm den entsprechenden Dateneingabeflag vor Anzeige des voreingestellten Werts; wenn das Programm fortgesetzt wird, überprüft der Computer den Status dieses Flags. Wenn vor Drücken der Taste [R/S] ein Wert eingegeben wurde, ist der Dateneingabeflag gesetzt.

Die beiden Programme verwenden Flag 25 (Ignorieren eines Fehlers), wenn auf Records des Textfiles **ZDATEN** zugegriffen wird. Um ohne Kenntnis der Anzahl der Records auf alle Records zugreifen zu können, gehen die Programme folgendermaßen vor: Eine Programmschleife greift auf einen Record zu, setzt Flag 25, greift auf den nächsten Record zu und fragt im Anschluß daran Flag 25 ab. Wenn kein weiterer Record existiert, wird durch diesen Fehler Flag 25 gelöscht. Das Programm kann also durch Abfragen von Flag 25 feststellen, wann die Programmschleife abzubrechen ist.

Systemflags (30 bis 35)

Die Systemflags dienen vor allem internen Steueraufgaben; der Benutzer kann diese Flags nur sehr begrenzt verwenden. Sie können die Systemflags abfragen, aber einige dieser Flags werden immer als *gelöscht* gemeldet. Sie können Systemflags nicht direkt verändern; Sie können jedoch den Status der Flags, die Einstellungen des Benutzers repräsentieren, speichern und später wiederherstellen. In diesem Absatz wird der Gebrauch einiger Systemflags erläutert.

Einstellungsflags. Einige der von Ihnen gewählten Betriebsmodi werden vom Computer in Flags gespeichert. Dadurch kann ein Programm angewiesen werden, einen Betriebsmodus einzustellen und später den ursprünglichen Zustand wiederherzustellen. Die dazu benötigten Funktionen ([RCLFLAG] und [STOFLAG]) werden am Ende dieses Abschnitts erläutert.

Systemflags werden auch dazu verwendet, den Betriebsmodus mancher vom Computer gesteuerten Peripheriegeräte zu speichern; Einzelheiten hierzu entnehmen Sie bitte der Benutzerdokumentation zu dem betreffenden Peripheriegerät. Die unten aufgeführten Flags repräsentieren einstellbare Betriebsmodi:

- Flag 31 repräsentiert das Datumsformat (siehe Seite 242).
- Die Flags 36 bis 39 repräsentieren die Anzahl der angezeigten Ziffern (siehe Seite 160).
- Die Flags 40 und 41 repräsentieren das Anzeigeformat (siehe Seite 160).
- Die Flags 42 und 43 repräsentieren den Winkelmodus (siehe Seite 186).

Betriebsbedingungen kennzeichnende Flags. Die folgenden Flags liefern Informationen über interne Betriebsbedingungen des Computers:

- Flag 44 wird gesetzt, wenn durch Ausführen der Funktion [ON] die Abschaltautomatik desaktiviert wird.
- Flag 48 ist gesetzt, wenn das Alpha-Tastenfeld eingeschaltet ist (d.h. wenn der **ALPHA**-Indikator angezeigt wird).

- Flag 49 wird gesetzt (und der **BAT**-Indikator wird angezeigt), wenn die Batteriespannung auf einen bestimmten Wert abfällt. Ein lange laufendes Programm kann Flag 49 gelegentlich abfragen und die Funktion [OFF] ausführen, wenn Flag 49 gesetzt ist. Wenn die Batterien erschöpft sind und das Programm weiterläuft, kann der Speicherhinhalt des Computers verloren gehen.
- Flag 50 ist gesetzt, wenn eine Meldung angezeigt wird.
- Flag 55 ist gesetzt, wenn ein Drucker angeschlossen ist. Das Zusammenwirken von Flag 55 und Flag 21 (Druckersteuerung) wird in Anhang D («Druckeroperationen») beschrieben.

Zusammenfassung der Flagzustände

Die nachstehende Tabelle gibt Auskunft über den Flagstatus nach einem Löschen des Permanentspeichers («Voreinstellung») und über den Status beim Einschalten des Computers («Einschalten»). Über den Status (*gesetzt* bzw. *gelöscht*) hinaus können die beiden folgenden Zustände auftreten:

P = wird vom Permanentspeicher aufrechterhalten.
? = abhängig von anderen Bedingungen.

Flags und das X-Register

Daten können aus verschiedenen Gründen zwischen Flags und dem X-Register transferiert werden:

1. Vom Benutzer gewählte Einstellungen (z.B. Anzeigeformat) können gespeichert und abgerufen werden. Der Computer speichert solche Einstellungen in Steuer- und Systemflags.
2. Sie können mehrfache Kopien einer Gruppe von Benutzerflags aufzeichnen; dabei definiert immer nur jeweils eine Kopie den momentanen Flagstatus.
3. Sie können die in Benutzerflags aufgezeichnete Information in eine Zahl umwandeln und umgekehrt.

Zur Übertragung zwischen Flags und X-Register stehen zwei Funktionen zur Verfügung:

- Die Funktion [X<>F] (*X exchange flags*) vertauscht den Status der Flags 00 bis 07 mit einer Zahl zwischen 0 und 255 im X-Register. Dadurch kann der Status der Flags 00 bis 07 als Zahl gespeichert und durch nochmaliges Ausführen der Funktion [X<>F] wiederhergestellt werden. Die gespeicherte Zahl kann auch weiterverwendet werden (z.B. in Berechnungen oder zu Programmverzweigungen). Mit der Funktion [X<>F] kann nicht auf Steuer- oder Systemflags zugegriffen werden.
- Mit den Funktionen [RCLFLAG] (*recall flags*) und [STOFLAG] (*restore flags*) kann der Status der Flags 00 bis 43 gespeichert und wiederhergestellt werden (d.h. diese Funktionen erlauben den Zugriff auf Benutzer-, Steuer- und Systemflags). Diese Funktionen ermöglichen Ihnen jedoch nicht die Umwandlung des Flagstatus in eine Zahl.

Zusammenfassend läßt sich folgendes sagen: Nur die Funktionen [RCLFLAG] und [STOFLAG] können den Status von Steuer- und Systemflags speichern und wiederherstellen; nur die Funktion [X<>F] kann den Status von Benutzerflags in eine verwendbare Zahl umwandeln; alle drei Funktionen eignen sich zu Operationen mit mehrfachen Kopien von Benutzerflags. Im folgenden Absatz wird zuerst die Funktion [X<>F] und ihre Verwendung im Zusammenhang mit Kopien von Benutzerflags erläutert. Im Anschluß daran werden die Funktionen [RCLFLAG] und [STOFLAG] vorgestellt. Am Ende dieses Abschnitts finden Sie einen Vergleich dieser Funktionen im Hinblick auf Operationen mit mehrfachen Kopien von Benutzerflags.

Zusammenfassung der Flagzustände

Flagnummer	Bezeichnung bzw. Aufgabe	Status	
		Voreinstellung	Einschalten
00–10	Benutzerflags	gelöscht	P
11	Automatische Ausführung	gelöscht	gelöscht
12–20	Steuerung v. Peripheriegeräten	gelöscht	gelöscht
21	Druckersteuerung	?	?
22	Eingabe von Zahlen	gelöscht	gelöscht
23	Eingabe von Alpha-Zeichen	gelöscht	gelöscht
24	Ignorieren von Bereichsfehlern	gelöscht	gelöscht
25	Ignorieren einer Fehlers	gelöscht	gelöscht
26	Steuerung von akustischen Signalen	gesetzt	gesetzt
27	User-Tastenfeld	gelöscht	P
28	Dezimaltrennzeichen	gesetzt	P
29	Zifferntrennzeichen	gesetzt	P
31	Datumsformat	gelöscht	P
36	Ziffernanzahl	gelöscht	P
37	"	gesetzt	P
38	"	gelöscht	P
39	"	gelöscht	P
40	Anzeigeformat	gesetzt	P
41	"	gelöscht	P
42	Winkelmodus	gelöscht	P
43	"	gelöscht	P
44	Abschaltautomatik desaktiviert	gelöscht	gelöscht
48	Alpha-Tastenfeld	gelöscht	gelöscht
49	Batterie-Kontrolle	?	?
50	Meldung	gelöscht	gelöscht
55	Angeschlossener Drucker	?	?

Flags und Zahlen als Bytes

Die Informationseinheit «Byte» ist der Schlüssel zur Transformation zwischen dem Status von Flags und einer Zahl. Ein Byte besteht aus acht Binärziffern, sogenannten Bits. Beispiele für Bytes: 00010100, 10001100, 00100000 und 00001111.

- Ein Byte kann als Zusammenstellung von acht Flags aufgefaßt werden; jede 0 oder 1 wird durch den Status eines bestimmten Flags repräsentiert.
- Ein Byte kann auch als Dezimalzahl interpretiert werden, nämlich als Summe von Zweierpotenzen.

Dadurch, daß der Status der Flags 00 bis 07 als Byte interpretiert wird und dieses Byte wiederum eine Zahl darstellt, wird es möglich, den Status von acht Flags in eine eindeutige Zahl zwischen 0 und 255 umzuwandeln. Es kann auch der umgekehrte Weg gegangen werden: Sie können den Status von acht Flags auf einmal spezifizieren, indem Sie eine Zahl zwischen 0 und 255 vorgeben und diese Zahl in den Status der Flags 00 bis 07 umwandeln.

Dezimalwerte und Flags 00 bis 07

Flag-nummer	07	06	05	04	03	02	01	00
Flag-status	gesetzt oder gelöscht	gesetzt oder gelöscht	gesetzt oder gelöscht	gesetzt oder gelöscht	gesetzt oder gelöscht	gesetzt oder gelöscht	gesetzt oder gelöscht	gesetzt oder gelöscht
Binär-wert	2^7 oder Null	2^6 oder Null	2^5 oder Null	2^4 oder Null	2^3 oder Null	2^2 oder Null	2^1 oder Null	2^0 oder Null
Dezimal-wert	128 oder Null	64 oder Null	32 oder Null	16 oder Null	8 oder Null	4 oder Null	2 oder Null	1 oder Null

Beispiele. Für die vier untenstehenden Bytes wird sowohl der Dezimalwert als auch der Status der entsprechenden Flags angegeben. Nicht aufgeführte Flags zwischen 00 und 07 sind als gelöscht zu betrachten.

Gesetzte Flags	Byte	Dezimalwert
04, 02	00010100	16 + 4 = 20
07, 03, 02	10001100	128 + 8 + 4 = 140
05	00100000	32
03, 02, 01, 00	00001111	8 + 4 + 2 + 1 = 15

Verwenden von [X<>F]

Das Ausführen der Funktion [X<>F] hat die folgenden Auswirkungen:

- Der Status der Flags 00 bis 07 wird in eine Zahl zwischen 0 und 255 umgewandelt und im X-Register abgelegt.
- Die Zahl im X-Register, die zwischen 0 und 255 liegen muß, wird in den Status der Flags 00 bis 07 umgewandelt. Das Vorzeichen und der gebrochene Anteil der Zahl im X-Register werden vom Computer ignoriert.

Mehrfache Kopien von Flags

Sie können die Anzahl der Benutzerflags virtuell erhöhen, indem Sie den Status der tatsächlich vorhandenen Flags speichern und wiederherstellen. Wenn Sie den momentanen Status von Flags speichern, erzeugen Sie eine «eingefrorene» Kopie dieser Flags. Später können Sie den Status der Flags mit Hilfe der gespeicherten Kopie wiederherstellen; dabei spielt es keine Rolle, ob die Flags selbst in der Zwischenzeit verändert wurden. Nach Erstellen einer Kopie können Sie die Flags löschen und zur Darstellung von neuer Information verwenden; wenn Sie auch den daraus resultierenden Flagstatus abspeichern, haben Sie Zugriff auf *zwei* verschiedene Flagzustände und damit auf zwei verschiedene Informationssätze.

Die folgenden Schritte veranschaulichen diesen Vorgang. Der erste Informationssatz wird durch Setzen der Flags 00 und 02 spezifiziert und der resultierende («erste») Flagstatus wird in R_{00} abgelegt. Der zweite Informationssatz wird durch Setzen der Flags 01 und 03 spezifiziert; der resultierende («zweite») Flagstatus wird in R_{01} abgespeichert, während der erste Flagstatus wiederhergestellt wird.

01 0	Löscht die Flags 00 bis 07.
02 X<>F	
03 SF 00	Stellt den ersten Flagstatus ein.
04 SF 02	
05 0	Legt den ersten Flagstatus im X-Register ab und löscht die Flags 00 bis 07.
06 X<>F	
07 STO 00	Speichert den ersten Flagstatus in R_{00}.
08 SF 01	Stellt den zweiten Flagstatus ein.
09 SF 03	
10 RCL 00	Ruft den ersten Flagstatus in das X-Register zurück.
11 X<>F	Transferiert den zweiten Flagstatus in das X-Register und stellt den ersten Flagstatus wieder her.
12 STO 01	Speichert den zweiten Flagstatus in R_{01}.

Sie können die Anzahl der Flags vervielfachen, indem Sie für jeden Informationssatz eine Kopie des Flagstatus herstellen. Sie können jedoch nicht auf mehr als einen Informationssatz auf einmal zugreifen. Wenn Sie auf einen anderen Informationssatz zugreifen wollen, müssen Sie zuerst den dazugehörigen Flagstatus wiederherstellen.

Beispiel. Ein Programm ruft fünf Unterprogramme auf; jedes dieser Unterprogramme verwendet mehrere Flags. Zu Beginn der Programmausführung kann das Programm eine Kopie des Flagstatus jedes Unterprogramms erstellen und in verschiedenen Registern abspeichern. Das Unterprogramm kann dann seinen eigenen Flagstatus wiederherstellen, die Flags testen und verändern und zum Schluß den momentanen Status wieder abspeichern. Durch diese Vorgehensweise können Unterprogramme in beliebiger Reihenfolge aufgerufen werden, ohne sich gegenseitig zu beeinflussen.

Beispiel. Eine Untersuchung analysiert das Einkommen von Personen nach folgenden Kriterien: Geschlecht, Alter und Ausbildung. Diese Kriterien können mit Hilfe von Flags spezifiziert werden:

Geschlecht: Flag 00 gesetzt = weiblich, gelöscht = männlich.
Alter: Flag 01 gesetzt = Alter < 20.
Flag 02 gesetzt = $20 \leqslant$ Alter < 35.
Flag 03 gesetzt = $35 \leqslant$ Alter < 50.
Flag 04 gesetzt = $50 \leqslant$ Alter.
Ausbildung: Flag 05 gesetzt = ohne abgeschlossene Berufsausbildung.
Flag 06 gesetzt = Facharbeiterausbildung.
Flag 07 gesetzt = Hochschulstudium.

Jede Person kann nun durch zwei Zahlen charakterisiert werden: Eine Zahl zwischen 0 und 255, die den Status der Flags 00 bis 07 repräsentiert und eine Zahl für das Einkommen.

Um die statistischen Daten einer bestimmten Personengruppe aufzusummieren, kann ein Programm die Flagkopie jeder Person überprüfen und nur die Einkommen der Personen berücksichtigen, die der bestimmten Personengruppe angehören. Wenn Sie zum Beispiel das Einkommen aller erfaßten Männer (unabhängig von Alter und Ausbildung) untersuchen wollen, können Sie folgendermaßen vorgehen: Stellen Sie den einer Person zugeordneten Flagstatus wieder her und fragen Sie Flag 00 ab. Wenn Flag 00 gelöscht ist, gehört die Person der gewünschten Gruppe an und Sie können das entsprechende Einkommen berücksichtigen.

Nehmen wir an, Sie wollen die Untergruppe von Frauen zwischen 35 und 50 mit Hochschulstudium untersuchen. Diese Definition umfaßt den Status aller acht Flags; Sie müssen daher keine einzelnen Flags abfragen, sondern nur die den Flagstatus repräsentierende Dezimalzahl überprüfen. «Frau zwischen 35 und 50 mit Hochschulstudium» bedeutet, daß die Flags 00, 03 und 07 gesetzt sind (d.h. der entsprechende Dezimalwert beträgt 137). Es genügt also, den Dezimalwert der Flagkopie einer Person zu überprüfen; wenn dieser Wert 137 ist, gehört die Person der gewünschten Gruppe an und Sie können das entsprechende Einkommen berücksichtigen.

Verwenden von RCLFLAG und STOFLAG

Die Funktion RCLFLAG (*recall flags*) speichert den Status der Flags 00 bis 43. Der Status einiger oder aller dieser gespeicherten Flags kann mit der Funktion STOFLAG (*restore flags*) wiederhergestellt werden. Diese beiden Funktionen ergänzen sich; das Ergebnis von RCLFLAG kann nur von STOFLAG verwertet werden und STOFLAG wirkt nur auf durch RCLFLAG abgerufene Daten.

Verwenden von RCLFLAG. Wenn Sie die Funktion RCLFLAG ausführen, wird der Status der Flags 00 bis 43 in das X-Register zurückgerufen. Diese Flagdaten können dann in einem Register des Hauptspeichers oder des erweiterten Speichers abgelegt werden. (Die Anzeige der Statusdaten ist ohne Bedeutung.) Die Funktion RCLFLAG hebt den Stack an, sofern der Stack Lift nicht gesperrt ist (vergleiche RCL).

Verwenden von [STOFLAG]**.** Mit der Funktion [STOFLAG] können Sie den Status einiger oder aller Flags zwischen 00 und 43 wiederherstellen.

- Um den Status der Flags 00 bis 43 wiederherzustellen, rufen Sie die Statusdaten in das X-Register zurück und führen Sie die Funktion [STOFLAG] aus.
- Um den Status eines Flagblocks wiederherzustellen, können Sie folgendermaßen vorgehen: Rufen Sie die Statusdaten wie oben erläutert in das X-Register zurück. Legen Sie im X-Register die Steuerzahl *bb.ee* ab; dabei repräsentiert *bb* den ersten Flag und *ee* den letzten Flag des Blocks. (Dadurch werden die Statusdaten in das Y-Register gehoben.) Führen Sie die Funktion [STOFLAG] aus.

Die Funktion [STOFLAG] hat keine Auswirkungen auf den Stack (vergleiche [STO]). Wenn gilt $bb \geqslant ee$, wird nur der Status von Flag *bb* wiederhergestellt.

Beispiel. Ein Programm rundet Zahlen auf die nächste ganze Zahl (unter Verwendung der Funktion [RND] in Zusammenhang mit dem Format [FIX] 0); zur Zahleneingabe und Ergebnisausgabe soll jedoch das vom Benutzer gewählte Anzeigeformat benutzt werden.

1. Nach Beendigung der Zahleneingabe im gewählten Anzeigeformat wird mit der Funktion [RCLFLAG] der Status der Flags 00 bis 43 in das X-Register zurückgerufen. Die Anweisung [STO] *nn* speichert diese Statusdaten in R_{nn}.
2. Die Anweisung [FIX] 0 spezifiziert das Anzeigeformat «keine Dezimalstellen».
3. Die Funktion [RND] rundet die im X-Register abgelegte Zahl.
4. Bevor das Ergebnis ausgegeben wird, ruft die Anweisung [RCL] *nn* die Statusdaten zurück. Die Zahl 36.41 spezifiziert die Flags 36 bis 41 (die Anzeigeformatflags) und die Funktion [STOFLAG] stellt den ursprünglichen Status der Flags 36 bis 41 wieder her.

Dadurch kann die Zahleneingabe und Ergebnisausgabe in einem beliebigen Anzeigeformat vorgenommen werden und trotzdem ist während der Programmausführung das Format [FIX] 0 wirksam.

Mehrfache Kopien von Flags. Sie können die Funktionen [RCLFLAG] und [STOFLAG] anstatt der Funktion [X<>F] verwenden, um mehrfache Kopien von Flags zu erstellen. Beachten Sie dabei folgendes:

- Sie haben Zugriff auf einen beliebigen Block von Flags (einschließlich des gesamten Blocks von Flag 00 bis Flag 10). Bei Verwenden der Funktion [X<>Y] haben Sie nur Zugriff auf einen Block, Flag 00 bis 07.
- Sie können die durch [RCLFLAG] abgerufenen Flagdaten nur dazu verwenden, den vorhergehenden Status der Flags wiederherzustellen. Das letzte Beispiel zu [X<>F] zeigt, wie die mit dem Flagstatus zusammenhängende Zahl (Ergebnis der Funktion [X<>F]) direkt verwendet werden kann.

Abschnitt 20

Programmverzweigungen

Inhalt

Einführung 298
Verzweigen zu einem Label 299
Globale Labels 299
Suchen eines globalen Labels 299
Lokale Labels 299
Suchen eines lokalen Labels 300
Speicherbedarf der GTO Anweisung 300
Aufrufen eines Unterprogramms 301
Verwaltung der Rücksprungadressen 302
Suchen von globalen Unterprogrammlabels 303
Speicheranforderung der XEQ Anweisung 303
Bedingte Funktionen 303
Abfragen von Flags 304
Vergleichsoperationen 304
Programmschleifen 305
Schleifensteuerung durch bedingte Funktionen 305
Spezialfunktionen zur Schleifensteuerung 306

Einführung

Man spricht von einer Programmverzweigung, wenn die Ausführung eines Programms mit einer anderen Zeile als der unmittelbar nächsten fortgesetzt wird (d.h. wenn die Programmzeilen nicht der Reihe nach ausgeführt werden). Der HP-41 verfügt über zwei Arten von Funktionen zum Aufbau von Programmverzweigungen.

- Durch Ausführen von GTO *Label* oder XEQ *Label* wird die Programmausführung an das spezifizierte Label übergeben.
- Nach Ausführen einer Flagabfrage, einer Vergleichsoperation oder einer Funktion zur Schleifensteuerung wird die nächste Programmzeile übersprungen, wenn eine bestimmte Bedingung erfüllt ist.

Die beiden Funktionsarten werden oft kombiniert: Nach einer Flagabfrage kann die Anweisung GTO *Label* erfolgen; dadurch bestimmt der Status des spezifizierten Flags, ob die Programmausführung an das spezifizierte Label übergeben wird.

Verzweigen zu einem Label

Die einzige Aufgabe von Labels besteht darin, bei Programmverzweigungen das anzuspringende Ziel anzugeben. Der HP-41 unterscheidet zwischen zwei verschiedenen Arten von Labels: globale Labels, auf die von jedem im Programmspeicher abgelegten Programm zugegriffen werden kann, und lokale Labels, auf die nur innerhalb eines Programms zugegriffen werden kann. Jedes Label (außer ein lokales Alpha-Label) kann sowohl direkt als auch indirekt spezifiziert werden.

Globale Labels

Globale Labels bestehen aus bis zu sieben Alpha-Zeichen, einschließlich Ziffern. Kommata, Punkte und Doppelpunkte dürfen nicht verwendet werden. Einzelne Buchstaben zwischen A und J und zwischen a und e werden als lokale Alpha-Labels bezeichnet; diese Buchstaben können nicht als globales Label verwendet werden. Alle anderen einzelnen Buchstaben und Ziffern sind zulässige globale Labels. Ein globales Label belegt vier Bytes des Programmspeichers und zusätzlich ein weiteres Byte pro Zeichen.

Programme sind durch ihre globalen Labels gekennzeichnet. Funktionen, die auf ein ganzes Programm wirken (z.B. CLP oder SAVEP) erfordern die Spezifikation eines globalen Labels. Darüberhinaus identifiziert ein globales Label auch die Programmzeile, in der es steht. Von außerhalb kann auf verschiedene Teile eines Programms zugegriffen werden, wenn das Programm mehrere globale Label enthält; jedes dieser globalen Labels kann auch dazu dienen, das gesamte Programm zu identifizieren.

Suchen eines globalen Labels

Wenn der Computer die Anweisung GTO *Label* ausführt, durchsucht er zuerst den Programmspeicher. Die Suche beginnt bei dem letzten globalen Label (d.h. bei dem Label, das von Katalog 1 als letztes aufgelistet wird); der Programmspeicher wird von unten nach oben durchsucht, bis ein Label gefunden wird, das mit dem spezifizierten Label übereinstimmt. Wenn zwei identische globale Labels existieren, wird das höhere Label (das von Katalog 1 zuerst aufgelistet wird) nie gefunden, da die Suche immer bei dem weiter unten stehenden Label beendet wird.

Wenn der Computer den Anfang des Programmspeichers erreicht, ohne das spezifizierte Label gefunden zu haben, setzt er die Suche in Katalog 2 fort. Wenn ein in einem Applikations-Modul bzw. Peripheriegerät abgelegtes Programm das spezifizierte globale Label enthält, wird die Programmausführung auf das Modul bzw. Gerät übertragen und bei dem globalen Label fortgesetzt.

Lokale Labels

Lokale Labels sind interne Marken in einem Programm, die zu Verzweigungen innerhalb des momentanen Programms verwendet werden. Im folgenden werden zunächst die drei Arten von lokalen Labels erläutert und anschließend wird beschrieben, wie der Computer nach lokalen Labels sucht.

Lokale numerische Labels. Es existieren zwei Arten von numerischen Labels. Mit der einen Art kann nur ein begrenzter Teil eines Programms übersprungen werden; mit der anderen Art können beliebig lange Programmteile übersprungen werden.

- Die Labels 00 bis 14 werden als *Kurzformen* bezeichnet, da sie nur ein Byte des Programmspeichers belegen. Verwenden Sie diese Labels nur, wenn die Programmspeicher-Distanz zwischen der Anweisung GTO und dem Label nicht mehr als 112 Bytes beträgt.
- Die Labels 15 bis 99 belegen zwei Bytes des Programmspeichers. Diese Labels können zu Verzweigungen über beliebige Distanzen innerhalb eines Programms verwendet werden.

Lokale Alpha-Labels. Lokale Alpha-Labels belegen zwei Bytes des Programmspeichers und können zu Verzweigungen über beliebige Distanzen innerhalb eines Programms verwendet werden. Diese Labels sind auf manuelle Ausführung zurechtgeschnitten: Bei eingeschaltetem User-Tastenfeld wird jeder Taste der beiden obenliegenden Tastenreihen automatisch ein lokales Alpha-Label zugeordnet (siehe «Die beiden oberen Reihen des Tastenfelds» in Abschnitt 9). Sie können diese Tasten verwenden, um die entsprechenden lokalen Alpha-Labels des momentanen Programms auszuführen.

Programmbeispiel. Das Programm Σ (siehe Abschnitt 22) zeigt LOESCHE TAG-A,JOB-B?E an. Sie haben drei Möglichkeiten, um auf diese Anzeige zu reagieren. Drücken Sie [A], um Tage zu löschen; drücken Sie [B], um Jobs (d.h. Tätigkeiten) zu löschen; drücken Sie [E], um das Programm zu beenden. Wenn Sie eine dieser Tasten drücken, wird die Programmausführung bei dem entsprechenden lokalen Alpha-Label fortgesetzt.

Suchen eines lokalen Labels

Ein lokales Label wird nur innerhalb des momentanen Programms gesucht. Der Computer durchsucht das momentane Programm von oben nach unten; die Suche beginnt bei der Anweisung [GTO]. Wenn das Ende des Programms erreicht wird, bevor das spezifizierte Label gefunden ist, setzt der Computer die Suche am Anfang des Programms fort.

Die Suche nach einem lokalen Label kann, abhängig von der Länge des Programms, sehr viel Zeit in Anspruch nehmen. Um diese Suchzeit zu verkürzen, speichert der Computer beim ersten Ausführen der Anweisung [GTO] die Distanz zwischen [GTO] und dem spezifizierten lokalen Label. Dadurch fällt die Suchzeit bei weiteren Ausführungen dieser [GTO] Anweisung weg.

Programmbeispiel. Aus Gründen der Speicherplatzersparnis verwendet das Programm Σ (siehe Abschnitt 22) soviele lokale numerische Kurzform-Labels wie möglich. Das Programm Σ benötigt 28 lokale Kurzform-Labels; da der HP-41 nur über 15 dieser Labels verfügt, werden die Labels 00 bis 03 mehrmals verwendet. Dies wird dadurch ermöglicht, daß die Labels 00 bis 03 nur in Vorwärtsverzweigungen benutzt werden (in einer Vorwärtsverzweigung steht die Anweisung [GTO] vor der Anweisung [LBL]). Die Anweisung **LBL 00** kommt in fünf verschiedenen Programmzeilen vor; jedes **GTO 00** bezieht sich auf das folgendende **LBL 00** und daher sind Fehlinterpretationen nicht möglich.

Die Kurzform-Labels 04 bis 14 werden in Rückwärtsverzweigungen benutzt (in einer Rückwärtsverzweigung steht die Anweisung [GTO] nach der Anweisung [LBL]). Der Computer durchsucht das Programm von der [GTO] Anweisung nach unten und setzt die Suche dann am Programmanfang fort, bis die [LBL] Anweisung gefunden ist. Um Fehlinterpretationen zu vermeiden, kommt jedes dieser Labels nur einmal vor.

Speicherbedarf der [GTO] Anweisung

Wieviele Bytes des Programmspeichers durch eine [GTO] Anweisung belegt werden, hängt von der Art des spezifizierten Labels ab.

- Eine [GTO] Anweisung, die ein globales Label mit n Zeichen spezifiziert, belegt $n+2$ Bytes.
- Eine [GTO] Anweisung, die ein lokales numerisches Label der langen Form (Labels 15 bis 99) oder ein lokales Alpha-Label spezifiziert, belegt drei Bytes.
- Eine [GTO] Anweisung, die ein lokales numerisches Kurzform-Label (Labels 00 bis 14) spezifiziert, belegt zwei Bytes.

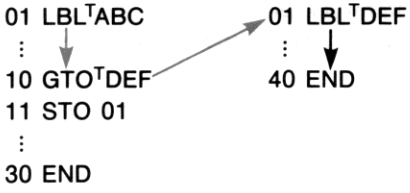
Aufrufen eines Unterprogramms

Durch die Anweisung [XEQ] Label wird ein Unterprogramm aufgerufen. Die Anweisungen [XEQ] Label und [GTO] Label haben die folgenden Gemeinsamkeiten:

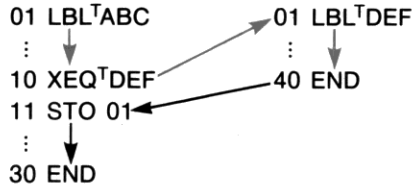
- Beide Anweisungen übergeben die Programmausführung an das spezifizierte Label.
- Alle Arten von Labels, die mit [GTO] verwendet werden können, sind auch im Zusammenhang mit [XEQ] erlaubt.

Die Besonderheit eines Unterprogrammaufrufs liegt in der Reihenfolge der Programmausführung, nachdem [XEQ] die Ausführung an das spezifizierte Label übergeben hat. Die Ausführung der nächsten [RTN] oder [END] Anweisung bedingt die Fortsetzung der Programmausführung mit der Zeile nach der [XEQ] Anweisung (siehe nachstehendes Beispiel).

- Das Programm **ABC** enthält eine *Verzweigung* zum Programm **DEF**. Die Programmausführung wird durch die Anweisung [END] am Ende von **DEF** angehalten.

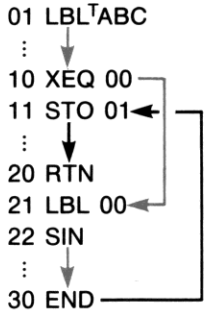


- Das Programm **ABC** ruft das Programm **DEF** als *Unterprogramm* auf. Die Anweisung [END] am Ende von **DEF** transferiert die Programmausführung wieder zu **ABC**.



Das Verwenden von Unterprogrammen spart Speicherplatz. Die Anweisungen eines Unterprogramms kommen nur einmal vor, können aber beliebig oft innerhalb eines Programms aufgerufen werden. Wenn das Unterprogramm mit einem globalen Label beginnt, kann es von jedem anderen Programm aufgerufen werden.

Die Anweisungen [RTN] und [END] übergeben die Programmausführung an die Zeile nach dem Unterprogrammaufruf. Die Anweisung [END] kennzeichnet das Ende des Programms und beeinflußt daher die Suche nach lokalen Labels und die Ausführung von Funktionen, die auf vollständige Programme wirken. Die Anweisung [RTN] kennzeichnet nur das Ende eines Unterprogramms innerhalb eines Programms. In dem folgenden Programm beendet [END] das Unterprogramm und [RTN] schließt die Programmausführung ab. (Das Beispiel ist etwas wirklichkeitsfremd; die Zeilen 22 bis 29 werden nur einmal ausgeführt und daher ist es nicht nötig, ein Unterprogramm zu verwenden.)

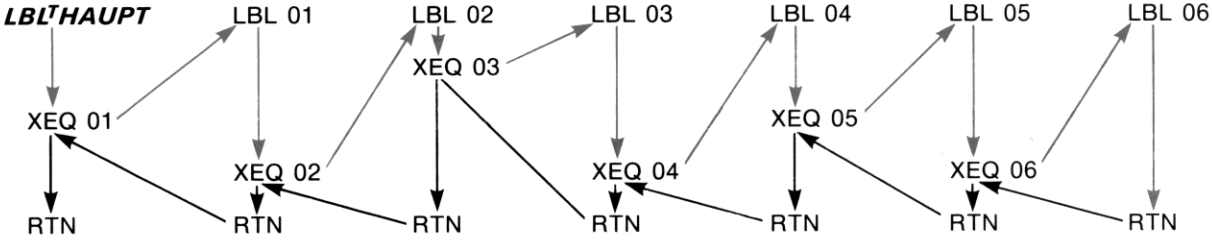


Wenn Sie **ABC** als Unterprogramm eines anderen Programms aufrufen, wird die Programmausführung durch die Anweisung [RTN] in Zeile 20 wieder zu dem aufrufenden Programm transferiert. Wenn ein Programm ein Unterprogramm aufruft, das ein weiteres Unterprogramm aufruft, wird das zweite Unterprogramm abgearbeitet und die Ausführung an das erste Unterprogramm übergeben; nach Abarbeitung des ersten Unterprogramms geht die Ausführung wieder auf das aufrufende Programm über.

Verwaltung der Rücksprungadressen

Wenn durch eine [XEQ] Anweisung ein Unterprogramm aufgerufen wird, «merkt» sich der Computer die Stelle im Programmspeicher, an der die [XEQ] Anweisung steht; dadurch kann die Programmausführung nach Abarbeitung des Unterprogramms wieder an dieser Stelle fortgesetzt werden. Während das Unterprogramm ausgeführt wird, speichert der Computer diese Rücksprungadresse im sogenannten *Unterprogrammrücksprung-Stack* (ein Analogon zum automatischen Speicherstack). Wenn nach Abschluß des Unterprogramms die Programmausführung wieder auf die Zeile nach der [XEQ] Anweisung übergeht, wird die entsprechende Rücksprungadresse aus dem Unterprogrammrücksprung-Stack gelöscht.

Grenzen bei der Verwendung von Unterprogrammen. Wenn ein Unterprogramm ein weiteres Unterprogramm aufruft, werden alle gespeicherten Rücksprungadressen im Unterprogrammrücksprung-Stack nach oben geschoben. Im Stack können bis zu sechs Rücksprungadressen gespeichert werden; Sie können also mit bis zu sechs Unterprogrammebenen arbeiten.



Verlust von Rücksprungadressen. Gespeicherte Rücksprungadressen gehen unter den folgenden Bedingungen aus dem Unterprogrammrücksprung-Stack verloren:

- Wenn beim Aufrufen eines Unterprogramms bereits sechs Rücksprungadressen im Stack gespeichert sind, geht die zuerst gespeicherte Adresse aus dem Stack verloren. In diesem Fall kann die Programmausführung nie zu der [XEQ] Anweisung zurückkehren, die das erste Unterprogramm aufgerufen hat; die Ausführung wird nach Abarbeitung des zuerst aufgerufenen Unterprogramms angehalten, da keine weiteren Rücksprungadressen im Stack abgelegt sind.
- Wenn Sie ein Programm über das Tastenfeld ausführen, gehen alle Rücksprungadressen verloren. Wenn Sie z.B. ein Programm **ABC** anhalten, um ein Programm **DEF** über das Tastenfeld auszuführen, kann der Computer nicht mehr zu **ABC** zurückkehren. Dies gilt auch, wenn ein Alarm das Programm **DEF** ausführt, solange **ABC** angehalten ist. Es spielt in beiden Fällen keine Rolle, ob **ABC** und **DEF** verschiedene Programme sind; der Unterprogrammrücksprung-Stack wird beispielsweise gelöscht, wenn Sie ein lokales Alpha-Label durch Drücken einer Taste des User-Tastenfelds ausführen.

Suchen von globalen Unterprogrammlabels

Wenn der Computer die Anweisung [XEQ] *Label* ausführt, sucht er zuerst, wie bei der Anweisung [GTO], den Programmspeicher ab. Wenn das spezifizierte Label jedoch nicht im Programmspeicher gefunden wird, unterscheidet sich [XEQ] von [GTO]. Die durch [XEQ] veranlaßte Suche erstreckt sich über die Kataloge 1, 2 und 3.

Durchsuchen von Katalog 1. Die Suche beginnt bei dem letzten globalen Label (d.h. dem als letzten von Katalog 1 aufgelisteten Label). Der Programmspeicher wird von unten nach oben durchsucht, bis ein Label gefunden wird, das mit dem spezifizierten Label übereinstimmt. Die Programmausführung wird bei diesem Label fortgesetzt.

Durchsuchen von Katalog 2. Wenn das spezifizierte Label nicht in Katalog 1 gefunden wird, sucht der Computer in Katalog 2 nach einem globalen Label oder nach einem Funktionsnamen in Übereinstimmung mit dem spezifizierten Label. (Katalog 2 wird ausführlich in Anhang H erläutert.) Die Programmausführung wird bei dem gefundenen Label fortgesetzt oder der Computer führt die gefundene Funktion aus.

Durchsuchen von Katalog 3. Wenn das spezifizierte Label auch nicht in Katalog 2 gefunden wurde, sucht der Computer in Katalog 3 nach einer Funktion, deren Name mit dem spezifizierten Label übereinstimmt. Wenn der Computer eine Funktion findet, wird sie ausgeführt. Wenn die Suche bis hierher erfolglos ist, zeigt der Computer die Meldung NONEXISTENT an.

Speicheranforderung der [XEQ] Anweisung

Die Anzahl der von einer [XEQ] Anweisung belegten Bytes im Programmspeicher hängt von der Art des spezifizierten Labels ab.

- Eine [XEQ] Anweisung, die ein globales Label mit n Zeichen spezifiziert, belegt $n+2$ Bytes.
- Eine [XEQ] Anweisung, die ein lokales Label spezifiziert, belegt drei Bytes.
- Eine [XEQ] Anweisung, die eine indirekte Adresse spezifiziert, belegt zwei Bytes.

Bedingte Funktionen

Als bedingte Funktionen bezeichnet man die Funktionen zum Abfragen von Flags und Vergleichsoperationen. Diese Operationen fragen einen Zustand ab und das zurückgegebene Ergebnis lautet entweder *wahr* oder *falsch*.

- Wenn Sie eine bedingte Funktion über das Tastenfeld ausführen, zeigt der Computer entweder YES (Antwort: *wahr*) oder NO (Antwort: *falsch*) an.
- Wenn ein Programm eine bedingte Funktion ausführt, gilt die Regel *Ausführung falls wahr* (*do if true*). Die Programmzeile nach der bedingten Funktion wird ausgeführt, wenn das Ergebnis der bedingten Funktion *wahr* lautet. Wenn das Ergebnis *falsch* lautet, überspringt der Computer die Zeile nach der bedingten Funktion.

Abfragen von Flags

Die folgenden Funktionen erlauben die Abfrage eines Flags.

- [FS?] *nn*. Ist Flag *nn* gesetzt? ($00 \leqslant nn \leqslant 55$)
- [FC?] *nn*. Ist Flag *nn* gelöscht? ($00 \leqslant nn \leqslant 55$)

Zwei weitere Funktionen fragen einen Flag ab und löschen den Flag nach der Abfrage. Diese Funktionen wirken nicht auf Systemflags (30 bis 55), da Systemflags vom Benutzer nicht verändert werden können.

- [FS?C] *nn*. Ist Flag *nn* gesetzt? Lösche Flag *nn*. ($00 \leqslant nn \leqslant 29$)
- [FC?C] *nn*. Ist Flag *nn* gelöscht? Lösche Flag *nn*. ($00 \leqslant nn \leqslant 29$)

Vergleichsoperationen

Vergleich von X mit Null. Die folgenden fünf Funktionen vergleichen die im X-Register abgelegte Zahl mit der Zahl Null:

[X < 0?] [X <= 0?] [x = 0?] [X ≠ 0?] [X > 0?]

Vergleich von X und Y. Die folgenden fünf Funktionen vergleichen die Zahl im X-Register mit der Zahl im Y-Register:

[X < Y?] [x ≤ y?] [x = y?] [X >= Y?] [x > y?]

Zwei dieser Funktionen ([x=y?] und [X≠Y?]) können sowohl Alpha-Daten als auch numerische Daten vergleichen. Wenn sich bei Ausführen der anderen drei Funktionen Alpha-Daten im X- oder Y-Register befinden, erscheint die Fehlermeldung ALPHA DATA.

Indirekter Vergleich von X mit einem anderen Register. Der HP-41 verfügt über sechs Funktionen, die den Inhalt des X-Registers mit dem Inhalt des Registers vergleichen, dessen Adresse im Y-Register spezifiziert ist. Das indirekt spezifizierte Register kann ein Hauptspeicher-Register (R_{00} bis $R_{(319)}$), ein Stackregister oder das LAST X Register sein. Legen Sie die Adresse zwischen 00 und 319 oder die Einzelbuchstaben **X**, **Y**, **Z**, **T** oder **L** im Y-Register ab. Die Vergleichsfunktionen sind:

[X < NN?] [X <= NN?] [X = NN?] [X ≠ NN?] [X >= NN?] [X > NN?]

Die obigen Funktionen erlauben den Vergleich von beliebigen Kombinationen von alphanumerischen Zeichen. Alpha-Strings werden auf der Basis der dazugehörigen ASCII-Codes verglichen und gelten immer als größer als numerische Daten.

Programmschleifen

Eine Programmschleife besteht aus einer Folge von Anweisungen, die mit einem Label beginnt und mit einer Verzweigung zu diesem Label endet. Der einfachste Fall ist eine unendliche Schleife, wie in dem untenstehenden Programm gezeigt.

```
01 LBL"SCHLEIF
02 BEEP
03 GTO" SCHLEIF
04 END
```

Einmal gestartet, würde dieses Programm solange laufen, bis die Batterien verbraucht sind. Unendliche Schleifen sollten im allgemeinen vermieden werden. Schleifen, die sich so oft wiederholen, bis eine bestimmte Bedingung erfüllt ist, sind eine äußerst nützliche und wirkungsvolle Programmiertechnik.

Schleifensteuerung durch bedingte Funktionen

Wenn Sie eine bestimmte Operation so lange ausführen wollen, bis eine bestimmte Bedingung erfüllt ist und dabei nicht genau wissen, wie oft diese Operation wiederholt werden muß, können Sie genau vor der Anweisung [GTO] eine bedingte Funktion vorsehen. Das untenstehende Programm subtrahiert eins von einer Zahl, fragt das Ergebnis ab und wiederholt die Schleife, solange das Ergebnis positiv ist. Sobald die Zahl den Wert Null erreicht (es wird vorausgesetzt, daß mit einem positiven Startwert begonnen wurde), wird die Schleife verlassen und ein akustisches Signal erzeugt.

```
01 LBL"ABC
02 1
03 -
04 X>0?
05 GTO"ABC
06 BEEP
07 END
```

Programmbeispiel. Die Beispielprogramme **ZE** und Σ (siehe Abschnitt 22) verwenden indirekte Schleifensteuerung, um auf den erweiterten Speicher zuzugreifen. Wenn mehrere Records des File **ZDATEN** auf eine bestimmte Art bearbeitet werden müssen und die Anzahl dieser Records nicht genau festliegt, geht das Programm folgendermaßen vor: Die Schleife zum Zugriff auf die einzelnen Records verwendet Flag 25 (Ignorieren eines Fehlers), um die Operation nach dem Abruf des letzten Records zu beenden.

- Nach Verarbeiten eines Records wird Flag 25 gesetzt. Wenn der nächste Fehler auftritt, löscht der Computer Flag 25, aber die Programmausführung wird *nicht* unterbrochen.
- Das Programm versucht bei gesetztem Flag 25, auf den nächsten Record zuzugreifen und fragt dann Flag 25 ab. Wenn Flag 25 gelöscht ist, folgt daraus, daß kein weiterer Record existiert (es liegt eine **END OF FL** Bedingung vor) und der Computer verläßt die Schleife. Wenn Flag 25 gesetzt ist, existiert der nächste Record und der Computer wiederholt die Schleife, um diesen Record zu bearbeiten.

Eine Illustration dieser Technik findet sich in dem Programm Σ. Die Zeilen 353 bis 364 bilden eine Schleife, die auf jeden Zeit-Record zugreift. Innerhalb der Schleife versuchen die Zeilen 356 bis 362, den Zeiger auf den nächsten Record zu positionieren. Wenn die Funktion [SEEKPT] in Zeile 360 einen Fehler verursacht, beendet die Anweisung [GTO] 12 in Zeile 362 die Schleife (siehe Seite 351).

Spezialfunktionen zur Schleifensteuerung

Wenn Sie genau wissen, wie oft eine Schleife durchlaufen werden soll, brauchen Sie nicht die in den vorangegangenen Beispielen erläuterten bedingten Funktionen zu verwenden. In diesem Fall stehen Ihnen zwei Spezialfunktionen zur Verfügung: Die Funktion [ISG] (*inkrement, skip if greater*) erhöht bei jedem Schleifendurchgang einen internen Zähler und überspringt die nächste Programmzeile, wenn der Zähler einen bestimmten Wert überschreitet; die Funktion [DSE] (*decrement, skip if equal*) erniedrigt bei jedem Schleifendurchgang einen internen Zähler und überspringt die nächste Anweisung, wenn der Zähler einen bestimmten Wert erreicht. Beide Funktionen verwenden eine in einem Register abgelegte Steuerzahl. Dieses Register kann ein Hauptspeicher-Register, ein Stackregister oder das LAST X Register sein; das Register kann sowohl direkt als auch indirekt spezifiziert werden.

Die Schleifensteuerzahl hat das Format *iiiii.fffcc*. Dabei gilt:

***iiiii* ist der momentane Wert des Schleifenzählers.** Bei jedem Ausführen von [ISG] bzw. [DSE] wird *iiiii* um den Wert *cc* erhöht (für [ISG]) bzw. erniedrigt (für [DSE]). Der Anteil *iiiii* der Steuerzahl darf bis zu fünf Ziffern enthalten.

***fff* ist der Endwert des Schleifenzählers.** Bei jeder Veränderung von *iiiii* durch [ISG] bzw. [DSE] vergleicht der Computer den resultierenden Wert von *iiiii* mit dem Endwert *fff*. Der Anteil *fff* der Steuerzahl muß aus drei Ziffern bestehen (z.B. 100, 020, 009).

***cc* ist die Schrittweite (Inkrement/Dekrement).** Wenn *cc* gleich Null ist (bzw. nicht spezifiziert wird), benutzt der Computer den voreingestellten Wert 01. Der Anteil *cc* der Steuerzahl muß, sofern er spezifiziert wird, aus zwei Ziffern bestehen (z.B. 30 oder 03).

Wenn der Computer die Funktion [ISG] ausführt, erhöht er zuerst *iiiii* um *cc* und überprüft anschließend, ob der resultierende Wert von *iiiii* größer als der Endwert *fff* ist. Wenn gilt $iiiii > fff$, überspringt der Computer die nächste Anweisung.

Wenn der Computer die Funktion [DSE] ausführt, erniedrigt er zuerst *iiiii* um *cc* und überprüft anschließend, ob der resultierende Wert von *iiiii* kleiner oder gleich dem Endwert *fff* ist. Wenn gilt $iiiii \leq fff$, überspringt der Computer die nächste Anweisung.

Programmbeispiel. Das Programm Σ (siehe Abschnitt 22) verwendet die Funktionen zur Schleifensteuerung, um auf einen Block von aufeinanderfolgenden Datenregistern zuzugreifen. Beim Ablegen von Daten aus dem erweiterten Speicher in diesem Registerblock greift der Computer in der Reihenfolge steigender Adressen auf die Datenregister zu. Wenn die Daten angezeigt werden, greift der Computer in der Reihenfolge fallender Adressen auf die Datenregister zu. (Das Programm muß die Reihenfolge umkehren, um die Daten in chronologischer Folge anzuzeigen.)

Das Füllen der Datenregister in der Reihenfolge steigender Adressen geschieht durch Verwendung der Funktion ISG (in Zeile 213), die die Schleifensteuerzahl (abgelegt in R_{05}) erhöht. Das Anzeigen der Daten in umgekehrter Reihenfolge erfolgt durch die Funktion DSE (in den Zeilen 231 und 260), die die Schleifensteuerzahl erniedrigt.

Σ definiert die Adressen des ersten bzw. letzten Registers des Registerblocks als Anfangs- und Endwert des Schleifenzählers. Bei jedem Schleifendurchlauf stellt der ganzzahlige Anteil der Schleifensteuerzahl die Adresse des Registers dar, auf das zugegriffen werden soll. Diese Verwendung der Schleifensteuerzahl zur indirekten Adressierung (Zeilen 209, 212, 228 und 255) basiert auf dem folgenden Anfangswert der Steuerzahl:

- *iiiii* repräsentiert die Adresse des ersten Registers, auf das indirekt zugegriffen wird.
- *fff* repräsentiert die Adresse des letzten Registers, auf das indirekt zugegriffen wird.
- Für *cc* wird kein Wert spezifiziert, daher verwendet der Computer den voreingestellten Wert 01.

Abschnitt 21

Alpha- und interaktive Funktionen

Inhalt

Einführung 308
Das Alpha-Register und das X-Register 309
Transformation zwischen Zeichen und Zahlen 309
Abrufen einer Zahl aus dem Alpha-Register 311
Manipulieren von Alpha-Strings 312
Durchsuchen und Verschieben des Alpha-Registers 312
Länge eines Strings 313
Beispiel zur Manipulation des Alpha-Registers 314
Eingabeaufforderungen 314
Verwenden von [PROMPT] 314
Verwenden von [PSE] 315
Die Funktionen [GETKEY] und [GETKEYX] 317
Verwenden von [GETKEY] 317
Verwenden von [GETKEYX] 317
Erzeugen von Meldungen und akustischen Signalen 318
Verwenden von [AVIEW] 318
Verwenden von [VIEW] 319
Verwenden von [PSE] 319
Verwenden von [TONE] und [BEEP] 319

Einführung

Dieser Abschnitt behandelt zwei Aspekte des Alpha-Registers: fortgeschrittene Manipulation der im Alpha-Register gespeicherten Daten und Kommunikation zwischen Benutzer und einem ablaufenden Programm.

- Mit Hilfe der fortgeschrittenen Alpha-Registeroperationen können Sie zusätzliche, nicht auf dem Tastenfeld vorhandene Zeichen erzeugen und beliebige Datenbytes speichern.
- Die Kommunikation zwischen Benutzer und einem Programm umfaßt sowohl Funktionen zur Anzeige einer im Alpha-Register abgelegten Meldung als auch Funktionen zur Interpretation der vom Benutzer vorgenommenen Eingaben.

Das Alpha-Register und das X-Register

Der HP-41 verfügt über drei Operationen zum Datentransfer zwischen dem Alpha-Register und dem X-Register.

1. Die Funktion [ARCL] **X** kopiert den Inhalt des X-Registers in das Alpha-Register; die Funktion [ASTO] **X** kopiert sechs Zeichen aus dem Alpha-Register in das X-Register. (Die von der Funktion [ASTO] im X-Register abgelegten Daten bestehen aus Zeichen und können nicht zu Berechnungen verwendet werden.) Die Funktionen [ARCL] und [ASTO], die im allgemeinen auf Datenregister zugreifen, werden in Abschnitt 12 («Der Hauptspeicher») beschrieben.
2. Die Funktion [XTOA] (*X to Alpha*) wandelt die im X-Register spezifizierte Zahl in ein Zeichen um und legt dieses Zeichen im Alpha-Register ab. Die Funktion [ATOX] (*Alpha to X*) wandelt das Alpha-Zeichen wieder in den entsprechenden Dezimalwert um und legt diese Zahl im X-Register ab.
3. Die Funktion [ANUM] (*Alpha number*) durchsucht das Alpha-Register nach einem Ziffernstring und legt diesen String als Zahl im X-Register ab. Dies ist der einzige Weg, um Ziffern in einer zu Berechnungen verwertbaren Form aus dem Alpha-Register abzurufen.

Transformation zwischen Zeichen und Zahlen

Der Computer kann nicht alle Zeichen anzeigen, die im Alpha-Register abgelegt werden können; des weiteren können nicht alle Zeichen, die der Computer anzeigen kann, über das Alpha-Tastenfeld eingegeben werden. Die Funktionen [ATOX] und [XTOA] ermöglichen es, die Fähigkeiten des Alpha-Registers und der Anzeige voll auszunutzen.

Zahlen und Zeichen als Bytes. In Abschnitt 19 wurde erläutert, daß ein Byte sowohl dem Status der Flags 00 bis 07 als auch einer Zahl zwischen 0 und 255 entspricht. Ein Byte entspricht auch einem im Alpha-Register abgelegten Zeichen. Manche Bytes können in der Anzeige nicht voneinander unterschieden werden, aber im Alpha-Register selbst sind diese Bytes eindeutig definiert. Die Fähigkeit des Computers, einzelne Bytes zu spezifizieren, ist sehr nützlich zum Steuern von Peripheriegeräten.

Das Null-Byte 00000000 (Dezimalwert 0) hat im Alpha-Register eine besondere Bedeutung. Daher kann es unter Umständen unmöglich sein, ein Null-Byte aus dem Alpha-Register abzurufen. Diese Einschränkungen werden in Anhang C («Nullzeichen») erläutert. Alle anderen Bytes können unbeschränkt gespeichert, manipuliert und zurückgerufen werden.

Verwenden von [XTOA]. Legen Sie eine Zahl zwischen 0 und 255 im X-Register ab und führen Sie die Funktion [XTOA] aus, um das entsprechende Byte an das rechte Ende des Alpha-Registers anzuhängen. Das X-Register wird nicht verändert.

Wenn das X-Register einen Alpha-String enthält, wird dieser String durch Ausführen von [XTOA] an den Inhalt des Alpha-Registers angehängt. Beachten Sie, daß ein im X-Register abgelegter Alpha-Ziffernstring nicht in das entsprechende Byte umgewandelt wird, sondern unverändert an den Inhalt des Alpha-Registers angehängt wird.

Die folgende Tabelle zeigt alle Anzeigezeichen des HP-41; nicht alle dieser Zeichen sind auf dem Alpha-Tastenfeld vorhanden. Um ein nicht auf dem Tastenfeld vorhandenes Zeichen an den Inhalt des Alpha-Registers anzuhängen, können Sie folgendermaßen vorgehen: Legen Sie den Dezimalcode des Zeichens im X-Register ab und führen Sie die Funktion [XTOA] aus. Die Tabelle zeigt auch die ASCII-Zeichen für die Dezimalcodes bzw. Anzeigezeichen, die mit dem Standard-ASCII-Zeichensatz übereinstimmen. Beachten Sie, daß viele Dezimalcodes zum Aufleuchten aller Anzeigesegmente führen. Alle Dezimalcodes zwischen 128 und 255 führen zum Aufleuchten sämtlicher Anzeigesegmente.

Dezimalcodes der Anzeigezeichen

Code	ASCII	Anzeige	Code	ASCII	Anzeige	Code	ASCII	Anzeige	Code	ASCII	Anzeige
0			32	Leerzeichen		64	@		96	`	
1			33	!		65	A		97	a	
2			34	"		66	B		98	b	
3			35	#		67	C		99	c	
4			36	$		68	D		100	d	
5			37	%		69	E		101	e	
6			38	&		70	F		102	f	
7			39	'		71	G		103	g	
8			40	(		72	H		104	h	
9			41	)		73	I		105	i	
10			42	*		74	J		106	j	
11			43	+		75	K		107	k	
12			44	,		76	L		108	l	
13			45	-		77	M		109	m	
14			46	.		78	N		110	n	
15			47	/		79	O		111	o	
16			48	0		80	P		112	p	
17			49	1		81	Q		113	q	
18			50	2		82	R		114	r	
19			51	3		83	S		115	s	
20			52	4		84	T		116	t	
21			53	5		85	U		117	u	
22			54	6		86	V		118	v	
23			55	7		87	W		119	w	
24			56	8		88	X		120	x	
25			57	9		89	Y		121	y	
26			58	:		90	Z		122	z	
27			59	;		91	[		123	{	
28			60	<		92	\		124	|	
29			61	=		93	]		125	}	
30			62	>		94	^		126	~	
31			63	?		95	_		127		

Verwenden von [ATOX]**.** Bei der Ausführung der Funktion [ATOX] wird der Dezimalwert des links außen im Alpha-Register stehenden Zeichens im X-Register abgelegt. Dieses Byte wird aus dem Alpha-Register gelöscht und der Inhalt des Alpha-Registers wird nach links verschoben. Der Stack wird angehoben, sofern der Stack Lift nicht gesperrt ist. Wenn das Alpha-Register leer ist, liefert die Funktion [ATOX] als Ergebnis den Wert Null.

Am linken Rand des Alpha-Registers stehende Null-Bytes werden von der Funktion [ATOX] ignoriert. Ein in einem Alpha-String enthaltenes Null-Byte geht daher verloren, wenn es an den linken Rand des Alpha-Registers geschoben wird. Sie können überprüfen, wann ein Null-Byte verloren geht, indem Sie die Längen des Alpha-Strings vor und nach dem Verschieben vergleichen (siehe «Länge eines Strings» in diesem Abschnitt).

Programmbeispiel. Die Programme **ZE** und Σ (siehe Abschnitt 22) verwenden die Funktionen [XTOA] und [ATOX], um die täglich mit einer Tätigkeit verbrachte Stundenzahl zu verschlüsseln. Bytes mit einem Dezimalwert zwischen 1 und 240 repräsentieren zwischen 0.1 und 24 Stunden. Dies ist eine sehr effiziente Art der Datenspeicherung; sieben Bytes, die die Daten einer Woche repräsentieren, belegen ein erweitertes Speicherregister des Textfiles **ZDATEN**. Eine Alternative wäre es, die tägliche Stundenzahl in einem Register eines Datenfiles zu speichern; dadurch ließe sich eine größere Genauigkeit erreichen, aber es würde siebenmal soviel Speicherplatz belegt werden.

Abrufen einer Zahl aus dem Alpha-Register

Die Funktion [ANUM] durchsucht das Alpha-Register nach Alpha-Ziffern. Ein gefundener Ziffernstring wird im X-Register abgelegt; dabei wird Flag 22 (numerische Dateneingabe) gesetzt. Das im X-Register abgelegte Ergebnis kann zu Berechnungen verwendet werden. Wenn keine Ziffern gefunden werden, bleiben das X-Register und Flag 22 unverändert.

Wenn Sie zwei Zahlen im Alpha-Register ablegen, ohne diese voneinander zu trennen, behandelt die Funktion [ANUM] den String als eine Zahl. Bei beliebigen Zeichen im Alpha-Register identifiziert [ANUM] zuerst den Ziffernstring und wertet ihn dann aus.

Identifizieren von Ziffernstrings:

- Die Funktion [ANUM] durchsucht das Alpha-Register, beginnend am linken Ende.
- Alpha-Zeichen werden solange ignoriert, bis eine Ziffer gefunden wird. Unmittelbar vor der Ziffer stehende Dezimaltrennzeichen (definiert durch den momentanen Status von Flag 28) und Minuszeichen werden als Teil des Strings betrachtet.
- Alle nachfolgenden Minuszeichen und Dezimaltrennzeichen sowie bis zu 10 Ziffern werden als Teil des Strings betrachtet.
- Der Buchstabe **E** (für «Exponent») wird nur dann als Teil des Strings betrachtet, wenn der String nachfolgend weitere Ziffern enthält.
- Das Zifferntrennzeichen (definiert durch den momentanen Status von Flag 28) wird nur dann als Teil des Strings betrachtet, wenn Flag 29 (Zifferngruppierung) gesetzt ist.
- [ANUM] betrachtet den String als abgeschlossen, wenn das erste Zeichen gefunden wird, das nicht in einem Ziffernstring enthalten sein darf.
- [ANUM] bricht die Suche ab, wenn nach Auffinden von Ziffern ein Null-Byte gefunden wird.

Auswerten von Ziffernstrings:

- Ein einfacher Ziffernstring, z.B. **12345**, stellt genau diese Zahl dar, hier 12345.
- Die nach dem ersten gefundenen **E** stehenden Ziffern (eine bzw. zwei) werden als Exponent interpretiert. Alle nachfolgenden **E**'s werden ignoriert. Wenn der String schon mehr als acht Ziffern enthält, wird ein **E** ignoriert, sofern nicht vor der achten Ziffer ein Dezimalzeichen steht.
- Die als Ergebnis im X-Register abgelegte Zahl ist positiv, wenn der String eine gerade Anzahl an Minuszeichen enthält und negativ, wenn die Anzahl der Minuszeichen ungerade ist. Wenn durch ein **E** ein Exponent definiert ist, wird diese Regel gesondert auf den Exponentenstring angewandt.
- Das erste gefundene Dezimaltrennzeichen trennt den ganzzahligen und den gebrochenen Anteil der Zahl im X-Register. Alle nachfolgenden Dezimaltrennzeichen werden ignoriert. Alle Dezimaltrennzeichen nach einem **E** werden ignoriert.
- Alle Zifferntrennzeichen werden ignoriert. (Zifferntrennzeichen werden nur dann als Teil des Strings betrachtet – und ignoriert – wenn Flag 29 gesetzt ist. Wenn Flag 29 gelöscht ist, betrachtet die Funktion [ANUM] den String als abgeschlossen, sobald ein Zifferntrennzeichen gefunden wird.)

Manipulieren von Alpha-Strings

Zeichen im Alpha-Register können verschiedenen Quellen entstammen. Die Daten können Ihren Textfiles oder einem HP-IL Peripheriegerät entnommen oder über das Alpha-Tastenfeld eingegeben worden sein. Die unten beschriebenen, fundamentalen Operationen ergänzen sich und ermöglichen es Ihnen, komplexe Manipulationen der im Alpha-Register gespeicherten Informationen vorzunehmen; dabei spielt es keine Rolle, wie diese Informationen im Alpha-Register abgelegt wurden.

Durchsuchen und Verschieben des Alpha-Registers

Der HP-41 verfügt über drei Funktionen, um Daten aus dem Alpha-Register in das X-Register zu übertragen. Alle diese Funktionen wirken auf das linke Ende des Alpha-Registers.

- [ASTO] **X** kopiert die sechs links außen stehenden Bytes in das X-Register.
- [ATOX] wandelt das links außen stehende Byte in eine Zahl um und legt diese Zahl im X-Register ab.
- [ANUM] identifiziert den am weitesten links stehenden Ziffernstring und legt den String als Zahl im X-Register ab.

Um alle im Alpha-Register gespeicherten Daten zu verwenden, können Sie den Registerinhalt verschieben und dadurch ein gewünschtes Byte bzw. einen gewünschten String am linken Ende positionieren. Gehen Sie dazu wie folgt vor: Stellen Sie die Position der gewünschten Information mit Hilfe der Funktion [POSA] (*position in Alpha*) fest und verwenden Sie die Funktion [AROT] (*Alpha rotate*), um diese Information an das linke Ende des Alpha-Registers zu schieben.

Durchsuchen des Alpha-Registers. Die Funktion [POSA] durchsucht das Alpha-Register nach dem im X-Register spezifizierten Zeichen oder String. Als Ergebnis wird eine Zahl im X-Register abgelegt, die die Position des Zeichens bzw. des Strings repräsentiert. Das spezifizierte Zeichen bzw. Byte bleibt im LAST X Register erhalten. Der Inhalt des Alpha-Registers wird nicht verändert.

Der von [POSA] zu suchende Zielstring kann spezifiziert werden als:

- der Dezimalwert eines Bytes (0 bis 255);
- ein durch [ASTO] **X** im X-Register abgelegter Alpha-String.

Die Positionen im Alpha-Register werden von links nach rechts gezählt, beginnend bei der Position 0. Die Position eines Alpha-Strings ist definiert als Position des ersten Zeichens in dem String. Wenn das Zeichen bzw. der String gefunden wird, ersetzt eine die Position repräsentierende Zahl die Zielstring-Spezifikation im X-Register. Wenn der gesuchte String bzw. das gesuchte Zeichen nicht gefunden wird, legt der Computer den Wert –1 im X-Register ab.

Wenn der Zielstring mehrmals vorkommt, findet die Funktion [POSA] nur den am weitesten links stehenden Zielstring. Ein Null-Byte (Dezimalwert 0) wird nur gefunden, wenn es hinter einem vom Null-Byte verschiedenen Byte positioniert ist.

Verschieben des Alpha-Registers. Die Funktion [AROT] verschiebt den Inhalt des Alpha-Registers um die im X-Register spezifizierte Stellenzahl. Ein positiver X-Registerinhalt bedingt eine Verschiebung nach links; ein negativer X-Registerinhalt bedingt eine Verschiebung nach rechts. Nur der momentane Inhalt wird verschoben und nicht alle 24 Positionen des Alpha-Registers.

Wenn [AROT] unmittelbar nach [POSA] ausgeführt wird, positioniert dies das gesuchte Zeichen bzw. den gesuchten String (sofern gefunden) auf den linken Rand des Alpha-Registers.

Alle an das linke Ende des Alpha-Registers geschobenen Null-Bytes gehen verloren. Sie können feststellen, ob Null-Bytes verloren gehen, indem Sie immer wieder die Länge des im Alpha-Register abgelegten Strings vergleichen.

Länge eines Strings

Die Funktion [ALENG] legt die Anzahl der im Alpha-Register gespeicherten Zeichen im X-Register ab. Der Inhalt des Alpha-Registers wird dabei nicht verändert; der Stack wird angehoben (sofern der Stack Lift nicht gesperrt ist). Die unten beschriebenen Situationen geben Anregungen, wie die Funktion [ALENG] in Verbindung mit anderen Alpha-Operationen eingesetzt werden kann.

- Wenn Sie den Inhalt des Alpha-Registers mehrmals verschieben, um mit der Funktion [ANUM] Zahlen abzurufen, können Sie mit Hilfe der Funktion [ALENG] überprüfen, wann der Inhalt einmal vollständig durch das Alpha-Register geschoben wurde. Wenn die Stellenanzahl, um die der Inhalt insgesamt verschoben wurde, mit der Zeichenanzahl des Strings übereinstimmt, ist wieder die ursprüngliche Konfiguration hergestellt.
- Wenn Sie mit der Funktion [ATOX] Bytes in das X-Register übertragen und diese einzeln in einem Unterprogramm bearbeiten, können Sie mit der Funktion [ALENG] die Anzahl der Unterprogrammdurchläufe bestimmen.
- Wenn Sie Alpha-Strings verschieben, in denen Null-Bytes enthalten sind, können Sie mit Hilfe der Funktion [ALENG] feststellen, wieviele Null-Bytes (wenn überhaupt) verloren gehen. Vergleichen Sie die Stringlängen vor und nach der Verschiebeoperation; wenn der String nach dem Verschieben weniger Zeichen enthält, sind ein oder mehrere Null-Bytes verloren gegangen.

Beispiel zur Manipulation des Alpha-Registers

Es sei unterstellt, daß ein Peripheriegerät zwei durch ein Leerzeichen getrennte Ziffernstrings im Alpha-Register ablegt. Das Beispiel zeigt, wie die beiden Zahlen abgerufen werden; die erste Zahl wird im X-Register abgelegt und die zweite Zahl im Y-Register. Die Kommentare zu dem Beispiel gehen davon aus, daß im Alpha-Register der Ziffernstring «**123.45 678.90**» vorhanden ist.

01 ANUM	Der erste Ziffernstring 123.45 wird im X-Register abgelegt. Der Inhalt des Alpha-Registers wird nicht verändert.
02 32	Der Dezimalwert für ein Leerzeichen (32) wird im X-Register abgelegt; diese Zahl dient als Operand für die nächsten beiden Operationen. 123.45 wird in das Y-Register gehoben.
03 XTOA	An das Ende des zweiten Ziffernstrings wird ein Leerzeichen angehängt. Dadurch wird sichergestellt, daß die beiden Strings beim Verschieben nicht zusammenrücken. Das Alpha-Register enthält jetzt «**123.45 67.890**».
04 POSA	Die Zahl 6, die die Position des Leerzeichens zwischen den beiden Ziffernstrings repräsentiert, überschreibt den vorhergehenden Inhalt des X-Registers (32). Der Computer zählt ab Position 0; das Leerzeichen steht an der siebten Stelle von links.
05 AROT	Der Inhalt des Alpha-Registers wird um sechs Stellen nach links verschoben. Das Alpha-Register enthält jetzt «**678.90 123.45**».
06 ANUM	Der zweite Ziffernstring (678.90) wird im X-Register abgelegt. 6 wird in das Y-Register gehoben und 123.45 wird in das Z-Register gehoben.
07 RCL Z	Die Zahl 123.45 wird in das X-Register zurückgerufen und 678.90 wird in das Y-Register gehoben.

Eingabeaufforderungen

Der HP-41 verfügt über mehrere Funktionen bzw. Funktionskombinationen zur Programmierung von Eingabeaufforderungen an den Benutzer. Beim Vergleichen der Alternativen ergeben sich die beiden folgenden Problemstellungen:

1. Soll die Programmausführung angehalten werden, bis eine Eingabe erfolgt oder soll die Programmausführung nach einiger Zeit fortgesetzt werden, selbst wenn keine Eingabe vorgenommen wird?
2. Welcher Art können die eingegebenen Daten sein?

Die Alternativen werden anhand dieser beiden Problemstellungen erörtert.

Verwenden von PROMPT

Wenn die Funktion PROMPT ausgeführt wird, zeigt der Computer den Inhalt des Alpha-Registers an und unterbricht die Programmausführung. Die angezeigte Meldung sollte andeuten, was für eine Eingabe erwartet wird: numerische Eingabe, Alpha-Eingabe, eine Operation, ein vom Programm zu interpretierender Tastendruck usw. Bevor Sie sich den Beispielen zuwenden, beachten Sie bitte, daß Ihnen zwei Möglichkeiten zum Fortsetzen der Programmausführung zur Verfügung stehen:

- Durch Drücken der Taste [R/S] wird die Programmausführung mit der Zeile nach der Anweisung [PROMPT] fortgesetzt.
- Durch Drücken einer Taste der beiden obenliegenden Tastenreihen wird die Programmausführung zu dem entsprechenden lokalen Alpha-Label transferiert. (Dies gilt nur, wenn das User-Tastenfeld eingeschaltet ist und der entsprechenden Taste nicht eine andere Funktion zugeordnet ist.) Die Ausführung wird mit dem betreffenden Label fortgesetzt.

Programmbeispiele. Die Programme **ZE** und Σ (siehe Abschnitt 22) verwenden die Funktion [PROMPT] für viele Zwecke. Sehr oft wird der Benutzer mit Hilfe von [PROMPT] aufgefordert, eine angezeigte Zahl (Zeit oder Datum) oder einen Alpha-String (Jobname, d.h. Name einer Tätigkeit) zu bestätigen. Sie haben zwei Möglichkeiten, um diese Aufforderung zu beantworten: entweder Sie geben einen anderen Wert ein und drücken die Taste [R/S] oder Sie drücken nur die Taste [R/S] und bestätigen damit den angezeigten Wert. Die Programme überprüfen mit Hilfe der Flags 22 und 23, ob Sie einen anderen Wert eingetastet haben.

- Wenn Sie eine Zahl in das X-Register eingeben, wird Flag 22 (numerische Eingabe) gesetzt. Durch Abfragen dieses Flags kann festgestellt werden, ob eine Zahl eingegeben wurde.
- Wenn Sie einen Alpha-String in das Alpha-Register eingeben, wird Flag 23 (Alpha-Eingabe) gesetzt. Durch Abfragen dieses Flags kann festgestellt werden, ob ein Alpha-String eingegeben wurde.

Wenn das Programm **ZE** Sie fragt, welche Tätigkeit (Job) Sie beginnen (Zeilen 135 bis 146), löscht das Programm vor Anzeige der Eingabeaufforderung Flag 23. Nach Fortsetzen der Programmausführung überprüft **ZE**, ob Flag 23 gesetzt ist. Darüberhinaus schaltet **ZE** durch Ausführen der Funktion [AON] (*Alpha on*) vor Anzeige der Eingabeaufforderung das Alpha-Tastenfeld ein; die Anweisung [AOFF] (*Alpha off*) nach der Zeile mit [PROMPT] schaltet das Alpha-Tastenfeld wieder aus. Auf diese Weise ist sichergestellt, daß das Alpha-Tastenfeld aktiviert ist, solange die Programmausführung unterbrochen ist.

Wenn das Programm **ZE** versucht, den File **ZDATEN** zu vergrößern und keine erweiterten Speicherregister mehr verfügbar sind, wird mit Hilfe der Funktion [PROMPT] die Meldung **KEIN PLATZ** angezeigt und die Programmausführung unterbrochen (Zeilen 230 und 231). Als Antwort auf diese Meldung erwartet der Computer die Ausführung einer Operation – Sie können beispielsweise einen anderen File verkleinern. Wenn Sie weitere Register des erweiterten Speichers verfügbar gemacht haben, können Sie das Programm durch Drücken von [R/S] wieder starten; **ZE** versucht dann noch einmal, den File **ZDATEN** zu vergrößern.

Das Programm Σ verwendet die Funktion [PROMPT], um die Meldung **LOESCHE TAG-A,JOB-B ?E** anzuzeigen und die Programmausführung zu unterbrechen (Zeilen 304, 305, 306, 307). Durch Drücken von [A], [B] oder [E] wird die Programmausführung bei dem entsprechenden lokalen Alpha-Label fortgesetzt (**LBL A** in Zeile 343, **LBL B** in Zeile 310 oder **LBL E** in Zeile 365).

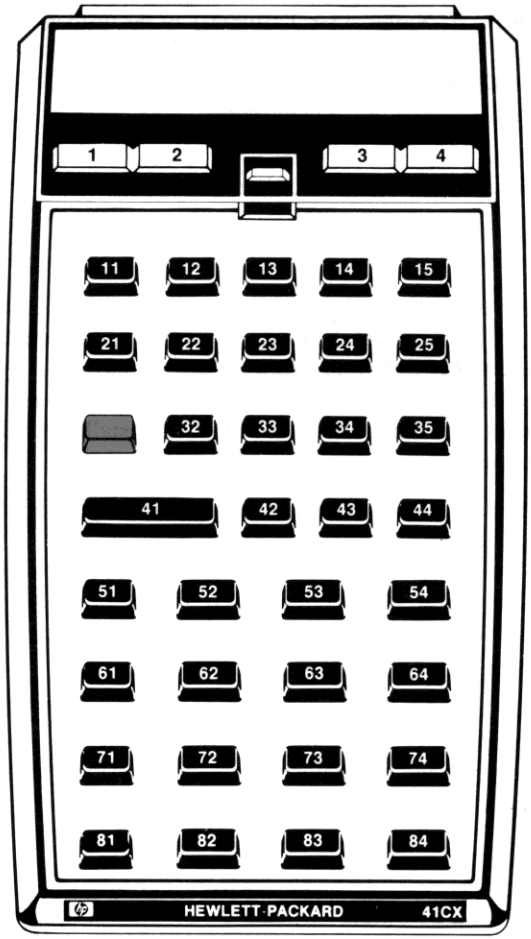
Verwenden von [PSE]

Sie können die Funktion [PSE] (*pause*) in ähnlicher Weise wie die Funktion [PROMPT] verwenden. Beachten Sie die folgenden Unterschiede:

- Die Funktion [PSE] verzögert die Programmausführung für etwas weniger als eine Sekunde. Wenn Sie während dieser Pause eine Zahl oder einen Alpha-String eingeben, wird die Pause wiederholt; wenn Sie eine Funktion ausführen, wird die Programmausführung unterbrochen.
- Normalerweise zeigt die Funktion [PSE] den Inhalt des X-Registers an. Um eine im Alpha-Register abgelegte Meldung anzuzeigen, muß der Funktion [PSE] die Anweisung [AVIEW] oder [AON] vorausgehen.

Mehrere [PSE] Anweisungen hintereinander verlängern die Zeitspanne, die bis zum Beginn der Eingabe verstreichen darf. Bei jeder Ausführung von [PSE] blinkt der **PRGM**-Indikator einmal auf. Die Zahlen- und Zeicheneingabe wird am Ende jeder Pause beendet; wenn Sie Ziffern eingeben, dann etwas länger als eine Sekunde warten und danach weitere Ziffern eingeben, werden die beiden Ziffernstrings als separate Zahlen behandelt.

Tastenpositionen und Tastencodes



[GETKEY] legt einen positiven Tastencode im *X-Register* ab. Drücken von ■ gibt den Tastencode 31 zurück.

[GETKEYX] legt einen positiven *oder negativen* Tastencode im *Y-Register* ab. Mit ■ umgeschaltete Tasten bekommen einen negativen Tastencode zugewiesen.

Die Funktionen [GETKEY] und [GETKEYX]

Zwei Funktionen ([GETKEY] und [GETKEYX]) veranlassen den Computer, die Programmausführung zu verzögern und auf eine Eingabe zu warten. Wenn Sie eine Taste drücken, wird der entsprechende Tastencode (der Tastencode spezifiziert die Position der Taste auf dem Tastenfeld) im Stack abgelegt. Die nachstehende Abbildung zeigt die Tastencodes aller Tasten.

Verwenden von [GETKEY]

Nach Ausführung der Funktion [GETKEY] wartet der Computer 10 Sekunden lang auf das Drücken einer Taste. Wenn Sie innerhalb dieser Zeitspanne eine Taste drücken, legt der Computer den entsprechenden Tastencode im X-Register ab; dabei wird der Stack angehoben (sofern der Stack Lift nicht gesperrt ist). Wenn kein Tastendruck erfolgt, liefert [GETKEY] das Ergebnis Null.

Um eine Meldung anzuzeigen, während [GETKEY] auf einen Tastendruck wartet, können Sie folgendermaßen vorgehen: Legen Sie die Meldung im Alpha-Register ab und führen Sie vor [GETKEY] die Funktion [AVIEW] aus.

Alle von [GETKEY] zurückgegebenen Tastencodes sind zulässige Werte für lokale numerische Label. Sie können die Funktion [GETKEY] mit der Sequenz [XEQ] **IND X** verbinden, um die Programmausführung zu einem dem Tastencode entsprechenden lokalen Label zu transferieren. Diese Technik hat zwar Ähnlichkeit mit der Verwendung von lokalen Alpha-Labels, löscht aber nicht den Unterprogrammrücksprung-Stack.

Ordnen Sie zuerst jeder zulässigen Reaktion einen Buchstaben zu und stellen Sie dann den Code dieser Buchstabentaste fest. Beginnen Sie jedes in Frage kommende Unterprogramm mit einem numerischen Label, das dem Code einer Buchstabentaste entspricht. Wenn der Benutzer [J] für «ja» und [N] für «nein» drücken soll, können Sie folgendermaßen vorgehen: Beginnen Sie das Unterprogramm für «ja» mit [LBL] **25** und das Unterprogramm für «nein» mit [LBL] **41**. Programmieren Sie dann nach [GETKEY] die Anweisung [XEQ] **IND X**. Wenn die Funktion [GETKEY] ausgeführt wird und der Benutzer die Taste [J] oddr [N] drückt, wird entweder die Zahl 25 oder 41 im X-Register abgelegt; die Anwendung [XEQ] **IND X** führt dann das gewünschte Unterprogramm aus.

Verwenden von [GETKEYX]

Die Funktion [GETKEYX] ist der Funktion [GETKEY] ähnlich, bietet aber mehr Flexibilität. [GETKEYX] hat die folgenden Besonderheiten:

Variables Intervall. Eine im X-Register abgelegte Zahl im Format SS.s spezifiziert, wie lange der Computer auf eine Antwort des Benutzers wartet. (Es muß gelten $|SS.s| \leq 99.9$.)

Umgeschaltete und nicht umgeschaltete Tastencodes. Die Funktion [GETKEYX] legt den Tastencode im Y-Register ab. Wenn Sie vor Drücken der gewünschten Taste die Taste ■ drücken, wird der negative Tastencode der als zweites gedrückten Taste zurückgegeben. Durch Drücken von ■ wird das spezifizierte Intervall von neuem gestartet; wenn innerhalb dieses zweiten Intervalls keine Taste gedrückt wird, gibt der Computer das Ergebnis Null zurück.

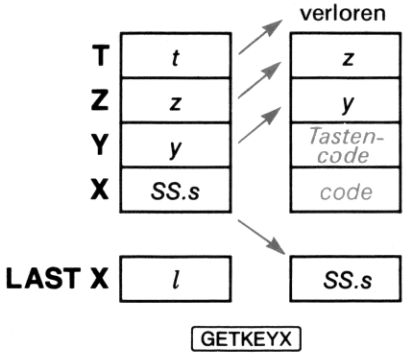
ASCII-Codes. Die Funktion [GETKEYX] legt einen ASCII-Code (falls sinnvoll) im X-Register ab. (Die ASCII-Codes finden Sie unter «Dezimalcodes der Anzeigezeichen» auf Seite 310.) Bei gesetztem Flag 48 (d.h. eingeschaltetem Alpha-Tastenfeld) wird der ASCII-Code einer Zeichentaste des Alpha-Tastenfelds zurückgegeben. Bei gelöschtem Flag 48 wird der ASCII-Code einer Zifferntaste, der Dezimalzeichentaste oder der Taste [CHS] zurückgegeben. In allen anderen Fällen legt [GETKEYX] Null im X-Register ab. Beachten Sie, daß eine Taste, abhängig vom Status von Flag 48, verschiedene ASCII-Codes hat. Der Tastencode 52 entspricht dem Buchstaben R (ASCII-Code 82) des Alpha-Tastenfelds und der Ziffer 7 (ASCII-Code 55) des Normal-Tastenfelds.

Ausführung beim Drücken bzw. Loslassen einer Taste. Das Vorzeichen der im X-Register spezifizierten Zahl *SS.s* bestimmt, ob der Computer beim Drücken oder beim Loslassen einer Taste reagiert.

- Wenn gilt $SS.s \geq 0$, reagiert der Computer beim Loslassen der Taste. Dies ist die einfachere Möglichkeit; die der Taste zugeordnete Funktion wird nie ausgeführt und der Computer liefert ein Ergebnis pro Tastendruck.
- Wenn gilt $SS.s < 0$, reagiert der Computer, wenn die Taste gedrückt wird; dadurch wird es möglich, Tasten zu wiederholen. Ein Programm kann zum Beispiel eine Schleife enthalten, die –0.01 im X-Register ablegt, die Funktion [GETKEYX] ausführt, das durch die Antwort des Benutzers spezifizierte Unterprogramm aufruft und wieder von vorne beginnt. Durch Ausführen der Funktion [GETKEYX] mit einem kleinen negativen Wert von SS.s überprüft der Computer, ob eine Taste niedergedrückt gehalten wird. Wenn der Benutzer eine Taste niedergedrückt hält, ruft die Schleife mit [GETKEYX] das entsprechende Unterprogramm immer wieder auf, bis die Taste losgelassen wird.

  Durch Loslassen der Taste wird der gleiche Effekt erzielt, der normalerweise durch Drücken einer Taste erreicht wird; nur die Tasten [R/S] und [ON] sind auch bei ablaufenden Programmen wirksam.

Die Funktion [GETKEYX] hebt den Inhalt des Y- bzw. Z-Registers in das Z- bzw. T-Register; das Intervall *SS.s* wird aus dem X-Register in das LAST X Register geschoben (siehe untenstehende Abbildung).



Erzeugen von Meldungen und akustischen Signalen

Die im folgenden erläuterten Funktionen erlauben dem Computer, eine Meldung anzuzeigen und akustische Signale zu erzeugen.

Verwenden von [AVIEW]

Wenn ein Programm die Funktion [AVIEW] ausführt, wird der Inhalt des Alpha-Registers angezeigt. Diese Anzeige bleibt bestehen, bis eine andere Meldung angezeigt wird oder die Anzeige durch [CLD] gelöscht wird. Abhängig vom Status der Flags 21 (Druckersteuerung) und 55 (angeschlossener Drucker) kann die Funktion [AVIEW] auch die Programmausführung unterbrechen; näheres hierzu entnehmen Sie bitte Anhang D.

Verwenden von [VIEW]

Die Funktion [VIEW] *nn* zeigt den Inhalt von R_{nn} an, ohne den Registerinhalt in das X-Register zurückzurufen. Das Register, dessen Inhalt Sie überprüfen wollen, kann auch indirekt spezifiziert werden. Wenn ein Programm die Funktion [VIEW] *nn* ausführt, wird der Inhalt von R_{nn} so lange angezeigt, bis die Anzeige durch [CLD] gelöscht wird oder eine andere Meldung angezeigt wird. Auch die Wirkung der Funktion [VIEW] hängt vom Status von Flag 21 ab (vergleiche [AVIEW]).

Verwenden von [PSE]

Die Funktion [PSE] kann dazu verwendet werden, eine Meldung für kurze Zeit anzuzeigen. Wenn [PSE] ausgeführt wird, verzögert der Computer die Programmausführung für etwas weniger als eine Sekunde.

- Wenn die Anzeige schon eine Meldung enthält (statt den Indikator für laufende Programmausführung), wird diese Meldung weiter angezeigt.
- Wenn die Anzeige keine Meldung enthält und das Alpha-Tastenfeld eingeschaltet ist, wird der Inhalt des Alpha-Registers angezeigt.
- In allen anderen Fällen wird der Inhalt des X-Registers angezeigt.

Verwenden von [TONE] und [BEEP]

Die Funktion [TONE] *n* erzeugt ein einzelnes akustisches Signal, dessen Tonhöhe von dem Wert des Parameters *n* abhängt. Die niedrigste Tonhöhe wird für $n = 0$ erreicht, die höchste für $n = 9$.

Die Funktion [BEEP] erzeugt eine voreingestellte Tonfolge, die aus vier akustischen Signalen besteht.

Abschnitt 22

Programme zur Zeiterfassung

Inhalt

Einführung 320
Programmbeispiele 321
Benutzen der Programme 321
Erzeugen der Files 321
Aufbau der ersten Records 322
Verwenden von ZE 323
Verwenden von Σ 325
Files zur Zeiterfassung 328
Der Textfile ZDATEN 329
Der Datenfile ZULETZT 330
Erläuterung von ZE 330
Von ZE verwendete Register und Flags 331
Programm-Listing zu ZE 332
Erläuterung von Σ 339
Von Σ verwendete Register und Flags 339
Programm-Listing zu Σ 340

Einführung

Die in diesem Abschnitt beschriebenen Programme sollen dazu dienen, Ihnen einen Überblick über die Verwendung Ihrer Zeit zu geben. In dem Absatz «Benutzen der Programme» werden die Programme vorgestellt und erläutert. Wenn Sie die Aufgaben dieser Programme verstanden haben, können Sie die kommentierten Programm-Listings durchgehen, um mehr über den Aufbau der Programme zu erfahren.

Sie müssen vielleicht nie so komplexe Programme wie die hier vorgestellten erstellen, aber einige der gezeigten Techniken können sich als nützlich im Hinblick auf Ihre eigenen Anwendungen erweisen. Jedes Programm besteht aus mehreren Hauptteilen, die die Hauptaufgaben des Programms repräsentieren. Jeder dieser Hauptteile ist dann in Routinen unterteilt (Routinen sind Folgen von Programmzeilen, die eine bestimmte Aufgabe ausführen). Sie können viel über die in einer Routine verwendeten Funktionen lernen, indem Sie das Zusammenspiel der Funktionen zur Ausführung der Aufgabe studieren. Einige der vorgestellten Routinen bieten sich vielleicht für Ihre eigenen Programme an.

Programmbeispiele

Die Abschnitte 19, 20 und 21 verweisen auf die im folgenden beschriebenen Programme, um einige Programmierungsfunktionen und -techniken zu erläutern. Unter der Überschrift «Programmbeispiele» werden unter anderem die folgenden Themen diskutiert:

- Flags zur Dateneingabe und zum Ignorieren von Fehlern («Steuerflags» in Abschnitt 19).
- Lokale Alpha-Label und die Suche nach lokalen Alpha-Labels («Lokale Label» in Abschnitt 20).
- Steuerung von Programmschleifen durch bedingte Funktionen und Spezialfunktionen zur Schleifensteuerung («Programmschleifen» in Abschnitt 20).
- Effiziente Datenspeicherung («Transformation zwischen Zeichen und Zahlen» in Abschnitt 21).
- Kommunikation zwischen Benutzer und Programm («Verwenden von PROMPT» in Abschnitt 21).

Benutzen der Programme

Die Zeiterfassung erfolgt in vier Schritten:

- Die Programme ZE (*Zeiterfassung*) und Σ (*Zusammenfassung*) sind in den Programmspeicher zu laden.
- Die Files zur Speicherung der später einzugebenden Informationen sind zu erzeugen. Dies müssen Sie nur einmal tun, es sei denn, Sie löschen die Files oder den Fileinhalt oder Sie löschen den Permanentspeicher.
- Die gespeicherten Informationen sind auf den neuesten Stand zu bringen. Dies müssen Sie immer dann tun, wenn Sie eine Tätigkeit beginnen oder beenden.
- Zusammenfassung der Informationen. Diesen Schritt können Sie ausführen, um zu erfahren, wie Sie Ihre Zeit verbracht haben.

Nachstehend finden Sie Anleitungen, wie jeder dieser Schritte auszuführen ist.

Laden der Programme

In diesem Abschnitt finden Sie Listings der Programme ZE und Σ. Um die Programme einzutasten, gehen Sie vor wie in Abschnitt 18 («Laden eines Programms») beschrieben.

Den Strichcode für die Programme finden Sie in Anhang J («Strichcode-Listings der Beispielprogramme»). Um die Programme mit einem optischen Lesestift HP 82153A einzulesen, folgen Sie bitte der Anleitung im Benutzerhandbuch zum Lesestift.

Erzeugen der Files

Erzeugen des Files ZULETZT. Dieser Datenfile enthält drei Werte, die den Zustand bei Beginn oder Ende der letzten Tätigkeit beschreiben: *letzter Job,* der Name des zuletzt begonnenen oder beendeten Jobs (dieses Handbuch verwendet der Kürze wegen den englischen Ausdruck für Tätigkeit, Job); *letzte Zeit,* die Uhrzeit; und *letztes Datum.* Um mit der Zeiterfassung zu beginnen, müssen Sie diesen File erzeugen und Startwerte für Zeit und Datum darin abspeichern.

1. Legen Sie den String ZULETZT im Alpha-Register ab.
2. Legen Sie den Wert 3 im X-Register ab.
3. Führen Sie die Funktion CRFLD aus. Dadurch wird der Datenfile **ZULETZT** mit drei Registern erzeugt. Der Zeiger ist auf das erste Register positioniert.

4. Legen Sie den Wert 1 im X-Register ab.
5. Führen Sie die Sequenz [STO] 02 aus. Der Startwert für die *letzte Zeit* ist 1.
6. Führen Sie die Funktion [DATE] aus. Das momentane Datum wird im X-Register abgelegt. (Die Anzeige zeigt das Datum als Meldung; das X-Register enthält eine das Datum repräsentierende Zahl).
7. Führen Sie die Sequenz [STO] 03 aus. Der Startwert für das *letzte Datum* ist das momentane Datum.
8. Legen Sie den Wert 1.003 im X-Register ab.
9. Führen Sie die Funktion [SAVERX] aus. Dadurch wird der Inhalt von R_{01}, R_{02} und R_{03} in den File **ZULETZT** kopiert. (Der Startwert für den *letzten Job* ist der momentane Inhalt von R_{01}. Beim ersten Ausführen von ZE können Sie diesen Namen durch einen gewünschten Namen ersetzen.)

Erzeugen des Textfiles ZDATEN (*Zeitdaten*). Dieser Textfile enthält ein Recordpaar für jede spezifizierte Tätigkeit. Um mit der Zeiterfassung zu beginnen, müssen Sie den File nur erzeugen; das Programm ZE legt automatisch die benötigten Records in ZDATEN ab.

1. Geben Sie den String **ZDATEN** in das Alpha-Register ein.
2. Berechnen Sie den Ausdruck $j\,(w+1)$; dabei ist j die Anzahl der Jobs und w die Anzahl der Wochen, über die sich die Zeiterfassung erstrecken soll. Dies ist eine Abschätzung der von ZDATEN benötigten Registeranzahl; diese Abschätzung braucht nicht besonders genau zu sein, da das Programm ZE bei Bedarf automatisch Register an ZDATEN anhängt. Legen Sie die Abschätzung im X-Register ab.
3. Führen Sie die Funktion [CRFLAS] aus. Dadurch wird der Textfile ZDATEN mit der abgeschätzten Registeranzahl erzeugt.

Aufbau der ersten Records

Als nächstes ist das Programm ZE einmal auszuführen, um in ZDATEN zwei Records für Ihre erste Tätigkeit anzulegen. (Ein Record enthält den Jobnamen, der andere enthält die dazugehörenden Zeitdaten.) Wenn Sie nicht wirklich die Arbeit an dem spezifizierten Job aufnehmen, sollten Sie ZE sofort wiederholen, um dem Computer mitzuteilen, daß Sie die Arbeit beenden. Das Programm zeichnet keine Zeit auf, wenn Sie eine Tätigkeit innerhalb von drei Minuten nach Arbeitsbeginn wieder beenden.

1. Führen Sie das Programm ZE aus.
2. Wenn der Computer die Frage **JOBNAME = ?** anzeigt, geben Sie den Namen Ihrer ersten Tätigkeit (d.h. Ihres ersten Jobs) ein und drücken die Taste [R/S]. (Es ist nicht nötig, vor oder nach der Eingabe des Namens die Taste [ALPHA] zu drücken.)
3. Wenn der Computer die Frage **START *Job?*** anzeigt (dabei ist *Job* der soeben von Ihnen eingegebene Name), bestätigen Sie durch Drücken der Taste [R/S].
4. Wenn der Computer die Frage **NEUER JOB?** anzeigt, drücken Sie die Taste [R/S]. Dadurch wird bestätigt, daß der von Ihnen eingegebene Name nicht der falsch geschriebene Name eines bereits existierenden Jobs ist. Der Computer erzeugt jetzt zwei neue Records für Ihren Job.
5. Wenn der Computer die Frage **START JETZT?** anzeigt, bestätigen Sie durch Drücken der Taste [R/S].

Wenn wieder die Zahl Null angezeigt wird, haben Sie das Programm ZE zum ersten Mal erfolgreich ausgeführt. Sie können nun durch nochmalige Ausführung von ZE abbrechen.

6. Drücken Sie [R/S], um das Programm ZE wieder zu starten. (Das Programm ZE ist zweimal hintereinander auszuführen, wenn Sie eine Tätigkeit beenden und sofort die nächste beginnen.)
7. Wenn der Computer die Frage **STOP JETZT?** anzeigt, bestätigen Sie durch Drücken der Taste [R/S].

Wenn wieder die Zahl Null angezeigt wird, sind alle Vorbereitungen getroffen und Sie können ZE gemäß den im folgenden erläuterten Regeln anwenden.

Verwenden von ZE

Führen Sie ZE aus, wann immer Sie eine Tätigkeit beginnen oder beenden. Das Programm bietet Ihnen verschiedene Auswahlmöglichkeiten, abhängig davon, ob Sie einen Job beginnen oder beenden. Diese Auswahlmöglichkeiten werden unter «Arbeitsbeginn» und «Arbeitsschluß» erläutert. Die folgenden Regeln gelten in beiden Situationen.

Allgemeine Regeln

- In den Anweisungen zum Starten und Stoppen kommen drei Variable vor:
- 1. *Letzter Job* ist der Name der Tätigkeit, die beim letzten Ausführen von ZE begonnen oder beendet wurde. Dieser Wert ändert sich nur, wenn Sie einen neuen Jobnamen eingeben.
  2. *Letzte Zeit* ist die Uhrzeit, um die Sie zuletzt einen Job begonnen oder beendet haben. Dieser Wert ändert sich jedes Mal, wenn Sie ZE ausführen. (Die Startzeit wird als negative Zahl gespeichert; wenn Sie die Arbeit beenden, wird der Wert 1 abgespeichert. Beim nächsten Ausführen von ZE zeigt das Vorzeichen des gespeicherten Werts für die *letzte Zeit*, ob Sie das letzte Mal begonnen haben und jetzt aufhören oder umgekehrt.)
  3. *Letztes Datum* ist das Datum, an dem Sie zuletzt einen Job begonnen oder beendet haben. Dieser Wert wird von ZE automatisch verwaltet.
- Wenn der Computer eine Frage anzeigt, verwendet er normalerweise einen voreingestellten Wert für Name und Uhrzeit. Um den angezeigten Wert zu bestätigen, genügt es, die Taste [R/S] zu drücken. Um einen anderen Wert zu wählen, geben Sie diesen Wert ein, bevor Sie die Taste [R/S] drücken.
- Zur Eingabe der Uhrzeit können Sie ein 24-Stunden-Format verwenden:

  $$HH.MM = HH \text{ Stunden } (0 \leqslant HH < 24) \text{ und } MM \text{ Minuten } (0 \leqslant MM < 60).$$

  Sie können auch ein 12-Stunden-Format verwenden und eine Uhrzeit zwischen Mittag und Mitternacht als negative Zahl eingeben:

  $$-\text{HH.MM} = 12 + HH \text{ Stunden } (0 \leqslant HH < 12) \text{ und } MM \text{ Minuten } (0 \leqslant MM < 60).$$

- Wenn der Computer die Frage JOBNAME = ? anzeigt, geben Sie einen Jobnamen ein und drücken die Taste [R/S]. (Sie müssen dazu nicht die Taste [ALPHA] drücken.) Der Computer nimmt jetzt an, daß Sie die Arbeit an diesem Job aufgenommen haben. Dieser Fall tritt beim ersten Ausführen von ZE auf. Der Computer fragt Sie auch nach einem Namen, wenn Sie mit Hilfe des Programms Σ den zuletzt bearbeiteten Job löschen und dann wieder ZE ausführen.
- Wenn der Computer die Meldung KEIN PLATZ anzeigt, sollten Sie folgendermaßen vorgehen: Verkleinern Sie einen im erweiterten Speicher abgelegten File (nicht ZULETZT oder ZDATEN!) um zwei oder mehr Register und drücken Sie die Taste [R/S]. Diese Meldung wird angezeigt, wenn der Computer den File ZDATEN nicht vergrößern kann, da keine Register des erweiterten Speichers mehr verfügbar sind.

Arbeitsbeginn

1. Führen Sie das Programm ZE aus.
2. Wenn der Computer die Frage START *letzter Job?* anzeigt, haben Sie zwei Möglichkeiten:
   - Drücken Sie die Taste [R/S], um zu bestätigen, daß Sie den gleichen Job wie beim letzten Mal beginnen. Der Computer springt zu Schritt 4.
   - Geben Sie einen anderen Jobnamen ein und drücken Sie die Taste [R/S]. (Sie müssen dazu nicht die Taste [ALPHA] drücken.) Wenn Sie den Namen eines existierenden Jobs eingeben (d.h. ein Job, auf den Sie ZE schon angewandt haben), springt der Computer zu Schritt 4.
3. Wenn der Computer den eingegebenen Jobnamen nicht erkennt, zeigt er die Frage NEUER JOB? an. Sie haben nun die folgenden Möglichkeiten:
   - Drücken Sie die Taste [R/S], um zu bestätigen, daß es sich um den Namen einer neuen Tätigkeit handelt. Der Computer schreibt nun zwei Records für diesen neuen Job in den File ZDATEN.
   - Geben Sie einen existierenden Jobnamen ein und drücken Sie die Taste [R/S]. (Sie müssen dazu nicht die Taste [ALPHA] drücken.) Wenn Sie in Schritt 2 einen Jobnamen falsch eingegeben haben, können Sie den Fehler an dieser Stelle korrigieren.
4. Wenn der Computer die Frage START JETZT? anzeigt, haben Sie zwei Möglichkeiten:
   - Drücken Sie die Taste [R/S], um zu bestätigen, daß Sie jetzt beginnen.
   - Geben Sie eine andere Startzeit in das X-Register ein und drücken Sie die Taste [R/S].

Das Programm ist beendet, wenn die Zahl Null in der Anzeige erscheint.

Arbeitsschluß

1. Führen Sie das Programm ZE aus.
2. *a)* Wenn das momentane Datum mit dem *letzten Datum* übereinstimmt, zeigt der Computer die Frage STOP JETZT? an. Sie haben nun die folgenden Möglichkeiten:
   - Drücken Sie die Taste [R/S], um zu bestätigen, daß Sie die Arbeit jetzt beenden.
   - Geben Sie eine andere Stopzeit in das X-Register ein und drücken Sie die Taste [R/S].
2. *b)* Wenn das momentane Datum nicht mit dem *letzten Datum* übereinstimmt, zeigt der Computer die Frage NACH 24:00? an. Sie haben nun die folgenden Möglichkeiten:
   - Drücken Sie die Taste [R/S], um zu bestätigen, daß Sie bis nach Mitternacht gearbeitet haben. Der Computer bringt dann die gespeicherten Zeitdaten dieser Tätigkeit für den Tag, an dem Sie begonnen haben, auf den neuesten Stand (Arbeit von *letzte Zeit* bis Mitternacht). Die Werte für *letzte Zeit* und *letztes Datum* werden geändert, um anzuzeigen, daß Sie die Arbeit um 00:00 Uhr des nächsten Tags begonnen haben. Danach geht der Computer wieder zu Schritt 2*a* oder 2*b* zurück.
   - Geben Sie eine andere Stopzeit in das X-Register ein und drücken Sie die Taste [R/S]. (Diese Technik können Sie anwenden, wenn Sie die Arbeit am gleichen Tag begonnen und beendet haben, aber vergessen haben, das Programm ZE auszuführen.)
3. Wenn Sie eine Stopzeit eingeben, die vor der Startzeit liegt, zeigt der Computer die Meldung START > STOP an und geht zu Schritt 2*a* oder 2*b* zurück.

Das Programm ist beendet, wenn die Zahl Null in der Anzeige erscheint. Wenn Sie jetzt eine neue Tätigkeit beginnen, können Sie die Taste [R/S] drücken, um ZE noch einmal auszuführen.

Verwenden von Σ

Das Programm Σ hat zwei Hauptaufgaben: zusammenfassen und löschen von Zeitdaten. Der Computer fragt Sie zuerst nach den Daten, die zusammengefaßt werden sollen und erstellt anschließend die gewünschte Zusammenfassung. Im Anschluß daran haben Sie verschiedene Auswahlmöglichkeiten, um Zeitdaten für alle Jobs bis zu einem bestimmten Datum zu löschen oder einen Job insgesamt zu löschen.

Sie können eine Zusammenfassung für eine beliebige Zeitspanne, für die Zeitdaten existieren, anfordern. Nehmen wir zum Beispiel an, daß Sie Zeitdaten für einen Monat speichern. Sie könnten dann immer an einem Montag die Zeitdaten der vergangenen Woche zusammenfassen. Am Beginn eines Monats könnten Sie den vergangenen Monat zusammenfassen, alle gespeicherten Zeitdaten für diesen Monat löschen und wieder von vorne beginnen.

Die Anwendungsregeln für die beiden Hauptteile von Σ finden Sie unter «Zusammenfassen von Zeitdaten» und «Löschen von Zeitdaten». Die folgenden Regeln gelten sowohl für das Zusammenfassen als auch für das Löschen von Zeitdaten.

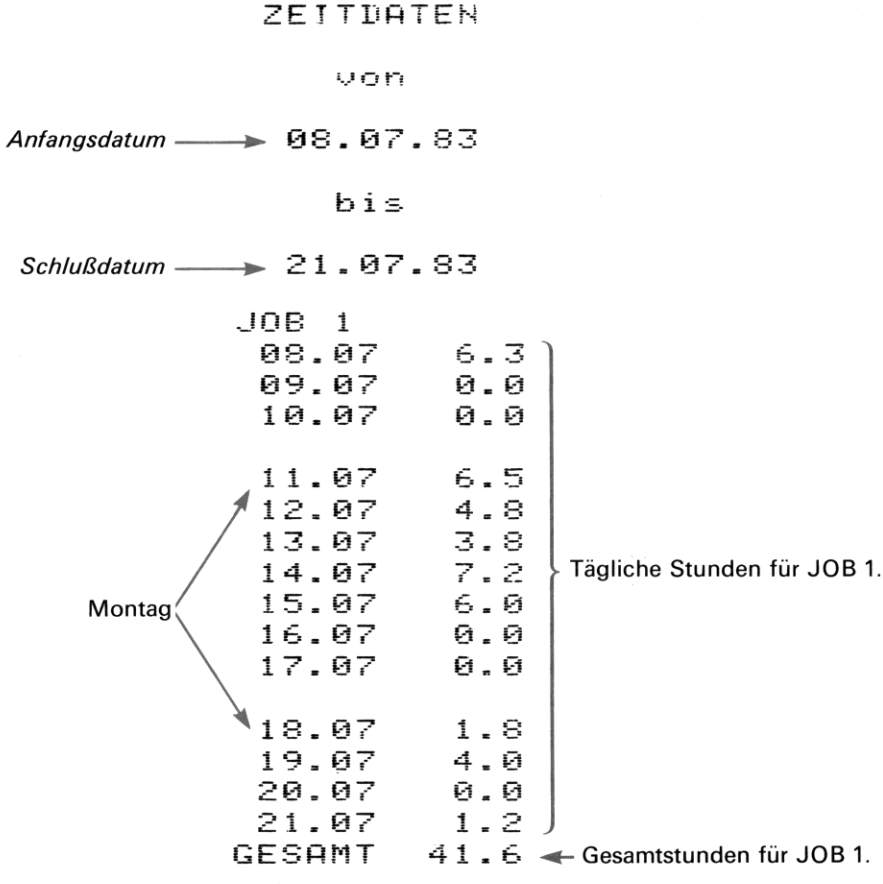
Allgemeine Regeln

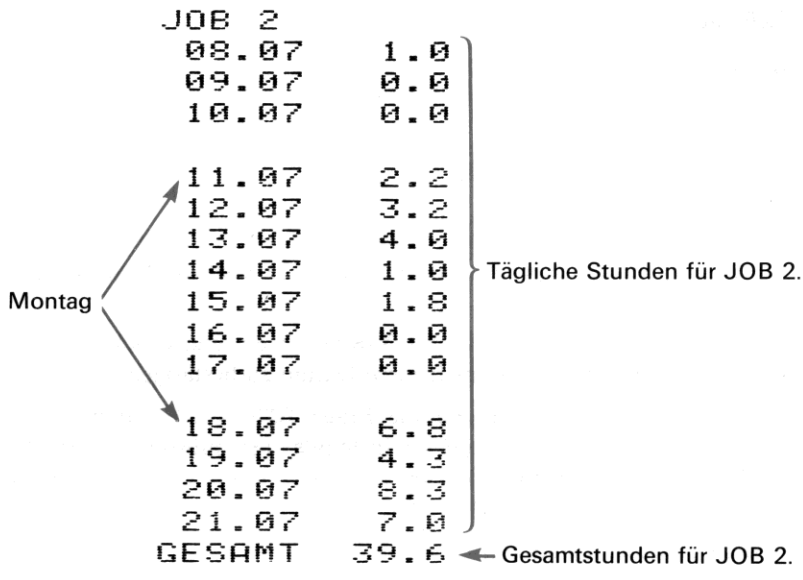
- In den Anweisungen zum Zusammenfassen und Löschen kommen zwei Variable vor.
  1. *Anfangsdatum* ist das Datum der am weitesten zurückliegenden Zeitdaten, die zusammengefaßt oder gelöscht werden sollen. Der voreingestellte Wert ist das früheste Datum, für das Zeitdaten existieren. Beim Zusammenfassen können Sie ein späteres Datum spezifizieren (solange es nicht nach dem *Schlußdatum* liegt).
  2. *Schlußdatum* ist das Datum der jüngsten Zeitdaten, die zusammengefaßt oder gelöscht werden sollen. Der voreingestellte Wert ist das letzte Datum, für das Zeitdaten existieren. Beim Zusammenfassen können Sie ein früheres Datum spezifizieren (solange es nicht vor dem *Anfangsdatum* liegt).
- Beim Anzeigen einer Frage nach einem Datum verwendet der Computer die momentanen Werte von *Anfangsdatum* und *Schlußdatum*. Drücken Sie die Taste [R/S], um das angezeigte Datum zu bestätigen. Um ein anderes Datum zu wählen, geben Sie das gewünschte Datum ein und drücken die Taste [R/S].
- Wenn der Computer die Meldung UNZUL. DATUM anzeigt, repräsentiert die von Ihnen eingegebene Zahl entweder ein Datum vor dem *Anfangsdatum* oder ein Datum nach dem *Schlußdatum*. Möglicherweise entspricht auch der eingegebene Wert nicht dem eingestellten Datumformat ([DMY] oder [MDY]).

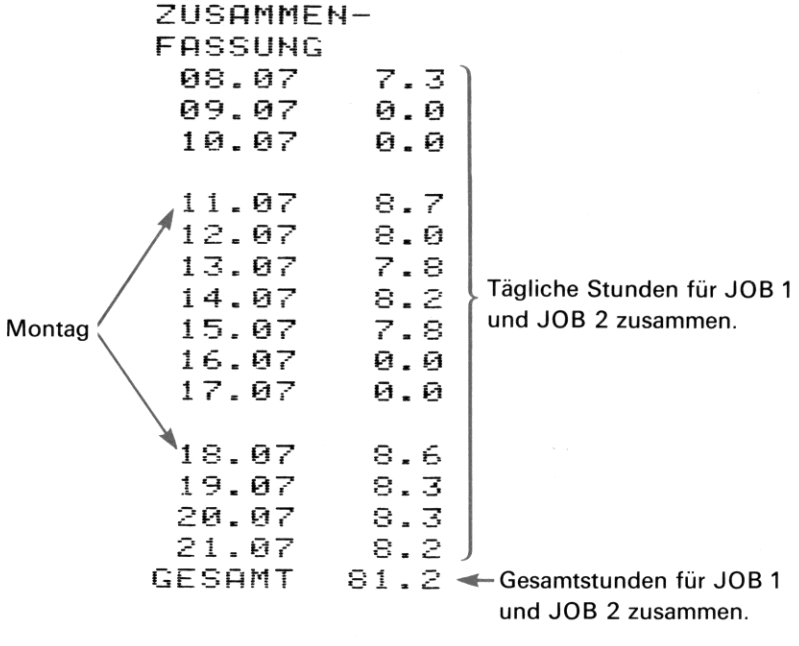
Zusammenfassen von Zeitdaten

1. Wenn Sie einen Drucker HP 82143A oder HP 82162A verwenden, schalten Sie diesen auf MAN (*manual mode*).
2. Führen Sie das Programm Σ aus. (Σ ist [f] [F] auf dem Alpha-Tastenfeld.)
3. Wenn der Computer die Frage VON *Anfangsdatum*? anzeigt, haben Sie die folgenden Möglichkeiten:
   - Drücken Sie die Taste [R/S], um dieses Datum zu bestätigen.
   - Geben Sie ein späteres Datum in das X-Register ein und drücken Sie die Taste [R/S]. Der Computer wiederholt diesen Schritt, um das spätere Datum zu bestätigen.
4. Wenn der Computer die Frage BIS *Schlußdatum*? anzeigt, haben Sie die folgenden Möglichkeiten:
   - Drücken Sie die Taste [R/S], um dieses Datum zu bestätigen.
   - Geben Sie ein früheres Datum in das X-Register ein und drücken Sie die Taste [R/S]. Der Computer wiederholt diesen Schritt, um das frühere Datum zu bestätigen.

5. Wenn der Computer die Meldungen **PACKING** (unter Umständen nur sehr kurz) und **TRY AGAIN** anzeigt, drücken Sie die Taste [R/S]. Dieser Fall tritt auf, wenn das Programm versucht, mehr Hauptspeicher-Register der Datenspeicherung zuzuordnen, als ungebundene Register verfügbar sind. Wenn durch das automatische Packen des Speichers genügend Register freigesetzt worden sind, läuft das Programm nach dem Drücken der Taste [R/S] ohne Schwierigkeiten. Wenn immer noch nicht genügend ungebundene Register verfügbar sind, zeigt der Computer wieder die Meldung **TRY AGAIN** an. In diesem Fall haben Sie zwei Möglichkeiten:
   - Führen Sie Σ erneut aus (beginnend mit Schritt 1) und spezifizieren Sie dieses Mal eine kürzere Zeitspanne als vorher.
   - Löschen Sie ein Programm, Alarm oder Zuordnungen des User-Tastenfelds; führen Sie [PACK] oder [GTO] [·] [·] aus; starten Sie dann erneut das Programm Σ (beginnend mit Schritt 1) und spezifizieren Sie die gleiche Zeitspanne wie vorher.
6. Der Computer liefert Ihnen eine Zusammenfassung der Zeitdaten des spezifizierten Zeitraums. Die nachstehende Abbildung zeigt eine ausgedruckte Zusammenfassung. Wenn kein Drucker angeschlossen ist, zeigt der Computer jede Zeile der Überschrift für ungefähr eine Sekunde an; jede nachfolgende Zeile wird so lange angezeigt, bis Sie die Taste [R/S] drücken. Auf diese Weise können Sie die angezeigten Werte mitschreiben.




JOB 2
08.07 1.0
09.07 0.0
10.07 0.0
11.07 2.2
12.07 3.2
13.07 4.0
14.07 1.0
15.07 1.8
16.07 0.0
17.07 0.0
18.07 6.8
19.07 4.3
20.07 8.3
21.07 7.0
GESAMT 39.6
Montag
Tägliche Stunden für JOB 2.
Gesamtstunden für JOB 2.


ZUSAMMEN-
FASSUNG
08.07 7.3
09.07 0.0
10.07 0.0
11.07 8.7
12.07 8.0
13.07 7.8
14.07 8.2
15.07 7.8
16.07 0.0
17.07 0.0
18.07 8.6
19.07 8.3
20.07 8.3
21.07 8.2
GESAMT 81.2
Montag
Tägliche Stunden für JOB 1 und JOB 2 zusammen.
Gesamtstunden für JOB 1 und JOB 2 zusammen.

Löschen von Zeitdaten

7. Wenn die Zusammenfassung abgeschlossen ist, zeigt der Computer LOESCHE TAG-A,JOB-B?E an. Diese Anzeige bietet Ihnen drei Auswahlmöglichkeiten; treffen Sie Ihre Wahl durch Drücken der Taste [A], [B] oder [E].

   [A] = *Lösche Tage*. Wenn der Computer die Frage BIS *Schlußdatum*? anzeigt, haben Sie die folgenden Möglichkeiten:

   - Drücken Sie die Taste [R/S], um das angezeigte Datum zu bestätigen. Der Computer löscht daraufhin alle bis zu diesem Datum gespeicherten Zeitdaten und zeigt im Anschluß daran wieder LOESCHE TAG-A,JOB-B?E an.
   - Geben Sie ein früheres Datum in das X-Register ein und drücken Sie die Taste [R/S]. Der Computer wiederholt diesen Schritt, um das frühere Datum zu bestätigen.

   [B] = *Lösche Job*. Wenn der Computer die Frage JOBNAME? anzeigt, tasten Sie den Namen des zu löschenden Jobs ein und drücken die Taste [R/S]. (Sie müssen dazu nicht die Taste [ALPHA] drücken.)
   - Wenn der Computer den spezifizierten Job findet, löscht er sowohl den Namen als auch die Zeitdaten dieses Jobs und zeigt wieder LOESCHE TAG-A,JOB-B?E an.
   - Wenn der Computer den spezifizierten Job nicht findet, erscheint zuerst die Meldung FEHL. NAME in der Anzeige und anschließend wieder die Frage LOESCHE TAG-A,JOB-B?E.

   [E] = *Ende*. Wenn Sie diese Taste drücken, wird der vor Ausführen von Σ gültige Flagstatus wiederhergestellt (und damit auch die gewählten Einstellungen). Damit ist das Programm Σ beendet.

8. Führen Sie die Tastenfolge [GTO] [·] [·] oder ein anderes Programm aus. Dies ist notwendig, da den Tasten [A], [B] und [E] bei eingeschaltetem User-Tastenfeld die oben genannten Operationen zugeordnet sind, solange Σ das momentane Programm ist.

Files zur Zeiterfassung

Zur Zeiterfassung werden zwei Files im erweiterten Speicher benutzt. Der Textfile **ZDATEN** enthält je zwei Records für jeden Job. Der Datenfile **ZULETZT** enthält die Information, wann Sie zuletzt einen Job begonnen oder beendet haben.

Der Textfile ZDATEN

In dem Textfile **ZDATEN** sind der Name und die Zeitdaten jedes Jobs in zwei aufeinanderfolgenden Records abgelegt (siehe untenstehende Abbildung).

Recordnummer	Recordinhalt
000	Name des ersten Jobs.
001	Zeitdaten des ersten Jobs.
002	Name des zweiten Jobs.
003	Zeitdaten des zweiten Jobs.
⋮	⋮
n	Name des $(n/2+1)$-ten Jobs.
$n+1$	Zeitdaten des $(n/2+1)$-ten Jobs.
⋮	⋮

Jedes in einem Record für die Zeitdaten abgelegte Zeichen ist ein Byte, dessen Dezimalwert die Stundenzahl repräsentiert, die Sie an einem bestimmten Tag an diesem Job gearbeitet haben. Das erste Byte repräsentiert die Stunden für den momentanen Tag und jedes nachfolgende Byte repräsentiert die Stundenzahl eines vorangehenden Tags. Diese Anordnung erlaubt es, mit Records unterschiedlicher Länge zu arbeiten – das *n*-te Byte in jedem Zeitrecord repräsentiert den gleichen Tag, aber einen anderen Job. Der Zeitrecord eines vor längerer Zeit aufgenommenen Jobs ist länger als der Zeitrecord eines Jobs, der erst kurze Zeit existiert. Wenn Sie einen neuen Job definieren, hängt der Computer die dazugehörigen Records an das Ende von **ZDATEN** an. Dies bedeutet, daß der erste in **ZDATEN** abgespeicherte Job der als erstes definierte Job ist; der Zeitrecord dieses Jobs ist mindestens genauso lang wie die Zeitrecords der übrigen Jobs.

Die tägliche Stundenzahl wird auf eine Dezimalstelle genau berechnet. Dieser Wert wird mit 10 multipliziert und das Ergebnis ist der Dezimalwert des die Stundenzahl repräsentierenden Bytes. Die folgende Tabelle erläutert dieses Verschlüsselungssystem anhand einiger Beispiele:

Zeit (Stunden/ Minuten	Zeit (Dezimalstunden)	Byte (Dezimalwert)
3 Std. 20 Min.	3.3	33
5 Std. 30 Min.	5.5	55
7 Std. 0 Min.	7.0	70
10 Std. 42 Min.	10.7	107
13 Std. 15 Min.	13.3	133

Zwei Bytes bilden eine Ausnahme in diesem System.

- Das Null-Byte (Dezimalwert = 0) kann aus dem Alpha-Register verloren gehen. Daher wird der Wert für «keine Stunden» durch das Byte mit Dezimalwert 254 repräsentiert.
- Das letzte Byte in jedem Zeitrecored ist das Byte mit Dezimalwert 255; dieses Byte identifiziert das Ende eines Records.

Der Datenfile ZULETZT

Der Datenfile **ZULETZT** enthält die Information, wann Sie zuletzt einen Job begonnen oder beendet haben.

Registeradresse	Registerinhalt
000	*Letzter Job* = Name des zuletzt begonnenen oder beendeten Jobs.
001	*Letzte Zeit* = Uhrzeit, um die Sie zuletzt die Arbeit aufgenommen oder beendet haben.
002	*Letztes Datum* = Datum, an dem Sie zuletzt die Arbeit aufgenommen oder beendet haben.

Letzter Job. Da zur Speicherung eines Jobnamens nur ein Register benutzt wird, darf er nur sechs Zeichen enthalten. Wenn Sie die gleiche Tätigkeit mehrmals hintereinander aufnehmen und beenden, brauchen Sie nicht bei jedem Ausführen des Programms ZE den Jobnamen zu spezifizieren. Der Name des Jobs, auf den zuletzt zugegriffen wurde ist im File **ZULETZT** gespeichert.

Ein Jobname darf nicht mit dem Anfang eines anderen Jobnamens übereinstimmen. Beispiel: Wenn ein Job ABCD existiert, dürfen Sie den Namen ABC nicht für einen anderen Job verwenden. (Es ist jedoch zulässig, einen anderen Job BCD zu nennen.)

Letzte Zeit. Das Programm **ZE** speichert einen negativen Wert für *letzte Zeit* bei Arbeitsbeginn und einen positiven Wert bei Arbeitsende. Dadurch wird angedeutet, ob Sie beim nächsten Ausführen von ZE einen Job beginnen oder beenden. Wenn Sie jedesmal bei Arbeitsbeginn und Arbeitsende sofort das Programm ZE ausführen, brauchen Sie den Wert für *letzte Zeit* nie einzugeben.

Letztes Datum. Wenn das in **ZULETZT** gespeicherte Datum nicht mit dem momentanen Datum übereinstimmt, werden automatisch Bytes zur Darstellung des Werts «keine Stunden» in dem Zeitrecord für jeden Job abgelegt. Dadurch bleiben die Zeitdaten immer dem Datum nach geordnet.

Immer wenn Sie einen Job beginnen oder beenden, kopiert das Programm **ZE** die drei Register von **ZULETZT** in die Hauptspeicher-Register R_{01} bis R_{03}. Das Programm verwendet diese Werte zu Berechnungen, bringt die Werte auf den neuesten Stand und speichert die so erhaltenen Werte wieder in den File **ZULETZT** zurück.

Erläuterung von ZE

Das Programm **ZE** besteht aus mehreren Segmenten.

- Die Hauptroutinen rufen den Inhalt von **ZULETZT** aus dem erweiterten Speicher zurück, bestimmen die auszuführenden Operationen, führen die entsprechenden Programmteile aus und kopieren die aktualisierten Werte wieder in den File **ZULETZT** im erweiterten Speicher.
- Die für das Arbeitsende zuständigen Routinen fragen Sie, wann Sie die Arbeit beendet haben, berechnen die Anzahl der Stunden, die Sie gearbeitet haben, bringen den entsprechenden Zeitrecord in **ZDATEN** auf den neuesten Stand und speichern den neuen Wert für die *letzte Zeit*.

- Die für ein neues Datum zuständigen Routinen fügen ein Byte zur Darstellung des Werts «keine Stunden» an jeden Zeitrecord an und speichern den neuen Wert für die *letzte Zeit*.
- Die für den Arbeitsbeginn zuständigen Routinen fragen Sie, welchen Job Sie beginnen und speichern dann die neuen Werte für den *letzten Job* und die *letzte Zeit*.

Darüberhinaus existieren noch drei weitere Unterprogramme am Ende des Programms ZE. Diese Unterprogramme durchsuchen den File **ZDATEN** nach einem spezifizierten Jobnamen, wandeln einen Zeitwert um und vergrößern den File **ZDATEN**.

Von ZE verwendete Register und Flags

Das Programm ZE verwendet die folgenden Datenspeicher-Register und Benutzerflags. Schlagen Sie beim Durcharbeiten des Programm-Listings in diesen Tabellen nach.

Register und Registerinhalte für ZE

Registeradresse	Registerinhalt
00	Flagstatus bei Programmstart.
01	*Letzter Job.*
02	*Letzte Zeit.*
03	*Letztes Datum.*
04	Dezimalwert des Bytes zur Darstellung der Arbeitsstunden.

Durch Benutzerflags dargestellte Zustände in ZE

Flagnummer	Dargestellter Zustand
01	Gelöscht = Jobname in **ZDATEN** gefunden. Gesetzt = Jobname nicht in **ZDATEN** gefunden.
02	Gelöscht = Arbeit wird gerade begonnen. Gesetzt = Arbeit wird gerade beendet.
03	Gelöscht = Gleiches Datum (*letztes Datum* = momentanes Datum). Gesetzt = Neues Datum (*letztes Datum* ≠ momentanes Datum).

Programm-Listing zu ZE

Hauptroutinen. Diese Routinen rufen die beim letzten Ausführen von ZE gespeicherten Daten zurück, bestimmen die auszuführenden Operationen, führen die entsprechenden Programmteile aus, speichern die auf den neuen Stand gebrachten Werte und beenden die Ausführung von ZE.

```
01♦LBL "ZE"
02 SIZE?
03 5
04 X>Y?
05 PSIZE
06 RCLFLAG
07 STO 00
08 FIX 1
09 CF 21
10 "ZULETZT"
11 0
12 SEEKPTA
13 1.003
14 GETRX
15 "ZDATEN"
16 RCLPTA
17 XEQ 50
18 FC? 01
19 GTO 20
20 "JOBNAME = ?"
21 AON
22 PROMPT
23 AOFF
24 ASTO 01
25 1
26 STO 02
```

Die Zeilen **01 bis 26** rufen die Werte für den *letzten Job*, die *letzte Zeit* und das *letzte Datum* zurück. Die Zeilen **02 bis 05** stellen sicher, daß mindestens fünf Register der Datenspeicherung zugeordnet sind. Die Zeilen **06 bis 09** speichern eine Kopie des momentanen Flagstatus, bevor das Anzeigeformat eingestellt und der Flag zur Druckersteuerung gelöscht wird. Die Zeilen **10 bis 14** kopieren den *letzten Job* in R_{01}, die *letzte Zeit* in R_{02} und das *letzte Datum* in R_{03}. (Seit dem letzten Ausführen von **ZE** waren diese Werte in dem File **ZULETZT** gespeichert.) Die Zeilen **15 und 16** machen den File **ZDATEN** zum momentanen File im erweiterten Speicher. Zeile **17** ruft ein Unterprogramm in Zeile 194 auf, das überprüft, ob der File **ZDATEN** den Namen des *letzten Jobs* noch enthält. Wenn dieser Name noch enthalten ist, verzweigt das Programm von Zeile **18** zu Zeile 27. Wenn Sie seit dem letzten Ausführen von ZE den *letzten Job* aus dem File **ZDATEN** gelöscht haben, werden Sie durch die Zeilen **20 bis 23** nach einem gültigen Jobnamen gefragt. Die Zeilen 21 und 23 schalten das Alpha-Tastenfeld automatisch ein und aus. Zeile **24** macht den von Ihnen eingegebenen Namen zum *letzten Namen*. Die Zeilen **25 und 26** speichern einen Wert für die *letzte Zeit*, der andeutet, daß Sie das letzte Mal die Arbeit beendet haben (also jetzt mit der Arbeit neu beginnen).

```
27*LBL 20
28 CF 02
29 RCL 02
30 X>0?
31 SF 02
32 CF 03
33 RCL 03
34 DATE
35 X≠Y?
36 SF 03
37 FC? 02
38 XEQ 01
39 FS? 03
40 GTO 30
41 FS? 02
42 XEQ 40
43 "ZULETZT"
44 0
45 SEEKPTA
46 1.003
47 SAVERX
48 RCL 00
49 STOFLAG
50 CLX
51 STOP
52 GTO "ZE"
```

Die Zeilen **27 bis 52** steuern den Rest der Ausführung des Programms ZE. Die Zeilen **28 bis 31** löschen Flag 02, wenn Sie die Arbeit beenden und setzen Flag 02, wenn Sie mit der Arbeit beginnen. (Wenn Sie das letzte Mal mit der Arbeit begonnen haben, war ein negativer Wert als *letzte Zeit* gespeichert; wenn Sie das letzte Mal die Arbeit beendet haben, war der Wert 1 als *letzte Zeit* gespeichert.) Die Zeilen **32 bis 36** löschen Flag 03, wenn das *letzte Datum* mit dem momentanen Datum übereinstimmt; wenn das *letzte Datum* nicht mit dem momentanen Datum übereinstimmt, wird Flag 03 gesetzt. Die Zeilen **37 bis 42** steuern die allgemeine Programmausführung. Wenn Sie die Arbeit beenden, rufen die Zeilen **37 und 38** das Unterprogramm auf, das in Zeile 58 beginnt. Wenn einer oder mehrere Tage seit der letzten Ausführung von ZE vergangen sind, wird die Programmausführung durch die Zeilen **39 und 40** auf die in Zeile 106 beginnende Routine übertragen (diese Routine fügt ein Byte an jeden Record für Zeitdaten an und springt dann zu Zeile 27). Wenn Sie die Arbeit aufnehmen, rufen die Zeilen **41 und 42** das in Zeile 135 beginnende Unterprogramm auf. Die Zeilen **43 bis 51** beenden die Ausführung von ZE. Die Zeilen **43 bis 47** kopieren die momentanen Werte von *letzter Job*, *letzte Zeit* und *letztes Datum* in den Datenfile **ZULETZT**; dort bleiben diese Werte bis zur nächsten Ausführung von ZE erhalten. Die Zeilen **48 und 49** stellen den vorhergehenden Flagstatus wieder her (und die damit verbundenen Betriebsbedingungen); Zeile **50** löscht die Flagstatus-Daten aus dem X-Register. Zeile **51** hält die Programmausführung an und Zeile **52** enthält einen Sprungbefehl zu Zeile 01; dadurch können Sie ZE durch einfaches Drücken der Taste R/S wiederholen.

Beenden der Arbeit. Diese Routinen bringen bei Arbeitsschluß den Record für die Zeitdaten des *letzten Jobs* auf den neuesten Stand. Normalerweise wird nur die zweite und dritte Routine ausgeführt; die erste Routine wird bei Auftreten eines Fehlers ausgeführt; die vierte Routine wird nur dann ausgeführt, wenn Sie die Arbeit nicht am gleichen Tag beginnen und beenden.

```
53*LBL 00
54 "START > STOP"
55 TONE 5
56 AVIEW
57 PSE
58*LBL 01
59 FS? 03
60 GTO 03
61 TIME
62 "STOP JETZT?"
63 PROMPT
```

Die Zeilen **53 bis 57** zeigen eine Fehlermeldung an. Wenn Sie eine Zeit für das Arbeitsende eingeben, die vor der Zeit des Arbeitsbeginns liegt, übergibt Zeile 72 die Programmausführung an Zeile 53, um eine Fehlermeldung anzuzeigen.

Die Zeilen **58 bis 63** bestimmen die Zeit, zu der Sie die Arbeit beenden. Wenn Sie die Arbeit nicht am gleichen Tag beenden, an dem Sie begonnen haben, wird das Programm mit Zeile 96 fortgesetzt. Die Zeilen **61, 62 und 63** fragen Sie, ob Sie die Arbeit zum momentanen Zeitpunkt beenden und halten dann die Programmausführung an; wenn Sie eine andere Zeit eingeben, ersetzt dieser Wert den im X-Register abgelegten Wert für die momentane Zeit.

```
64*LBL 02
65 X<0?
66 XEQ 60
67 X=0?
68 24
69 RCL 02
70 HMS+
71 X<0?
72 GTO 00
73 HR
74 RND
75 10
76 *
77 STO 04
78 XEQ 50
79 GETREC
80 ATOX
81 RCL 04
82 +
83 254
84 X≠Y?
85 MOD
86 CLA
87 XTOA
88 RCLPT
89 INT
90 SEEKPT
91 1
92 DELCHR
93 INSCHR
94 STO 02
95 RTN
```

Die Zeilen **64 bis 95** aktualisieren das Byte für *das letzte Datum* in dem Record für die Zeitdaten des *letzten Jobs*. Wenn Sie einen negativen Zeitwert (für eine Uhrzeit zwischen 12 und 24 Uhr) eingegeben haben, rufen die Zeilen **65 und 66** ein Unterprogramm in Zeile 216 auf, das diese Zeitwerte in ein 24-Stunden-Format umwandelt. Wenn Sie die Arbeit um Mitternacht beenden, legen die Zeilen **67 und 68** den Wert 24 im X-Register ab. Die Zeilen **69 und 70** subtrahieren die Zeit des Arbeitsbeginns von der Zeit des Arbeitsendes. Zeile 69 ruft den Wert für die *letzte Zeit* zurück (in diesem Fall repräsentiert dieser Wert die Zeit des Arbeitsbeginns und ist daher negativ). In Zeile 70 werden die Zeiten für Arbeitsende und Arbeitsbeginn (negativ) addiert; das Ergebnis ist die von Ihnen gearbeitete Stundenzahl. Wenn das Ergebnis negativ ist, übergeben die Zeilen **71 und 72** die Programmausführung an eine Fehlerroutine in Zeile 53. Die Zeilen **73 bis 77** speichern die gearbeitete Stundenzahl als Dezimalwert eines Bytes. Die Zeilen 73 und 74 wandeln die Zeit aus dem Format Stunden-Minuten-Sekunden in eine Dezimalzahl mit einer Dezimalstelle um. Die Zeilen 75 und 76 wandeln diese Zahl in einen Bytewert um und Zeile 77 speichert diesen Bytewert in R_{04}. Zeile **78** ruft ein Unterprogramm in Zeile 194 auf, das den Record für die Zeitdaten des *letzten Jobs* sucht; die Zeilen **79 und 80** legen den Dezimalwert des ersten Bytes im X-Register ab (dieser Wert repräsentiert die zuvor am *letzten Datum* gearbeitete Stundenzahl). Die Zeilen **81 bis 85** addieren diesen Bytewert zu dem Bytewert der gerade gearbeiteten Stunden; die Zeilen 83, 84 und 85 berichtigen das Ergebnis, falls der vorhergehende Bytewert 254 («keine Stunden») war. Die Zeilen **86 und 87** legen das Byte, dessen Dezimalwert im X-Register spezifiziert ist, im Alpha-Register ab. Die Zeilen 88, 89 und 90 suchen das Byte für das *letzte Datum* und die Zeilen **91, 92 und 93** ersetzen dieses Byte durch das neue, im Alpha-Register abgelegte Byte. Zeile **94** speichert den Wert 1 für die *letzte Zeit*. Zeile **95** übergibt die Programmausführung an Zeile 39, wenn alle Daten berichtigt sind. Wenn Sie bis nach Mitternacht gearbeitet haben, werden die Daten nur bis Mitternacht aktualisiert und die Programmausführung wird wieder mit Zeile 104 fortgesetzt.

```
96*LBL 03
97 24
98 "NACH 24:00?"
99 CF 22
100 PROMPT
101 FS? 22
102 GTO 02
103 XEQ 02
104 0
105 STO 02
```

Die Zeilen **96 bis 105** fragen Sie, ob Sie bis nach Mitternacht gearbeitet haben. Zeile 60 enthält einen Sprungbefehl an diese Stelle, wenn Sie die Arbeit nicht am gleichen Tag beginnen und beenden. Die Zeilen **97 bis 100** fragen, ob Sie bis nach Mitternacht gearbeitet haben; wenn Sie eine andere Uhrzeit eingeben, wird der Wert 24 im X-Register ersetzt und die Programmausführung an Zeile 64 übergeben. **Zeile 103** ruft ein Unterprogramm in Zeile 64 auf, das das Byte für das *letzte Datum* in dem Zeitrecord des *letzten Jobs* aktualisiert (von Arbeitsbeginn bis Mitternacht). Die Zeilen **104 und 105** speichern den Wert Null für die *letzte Zeit*, um die Startzeit «Mitternacht» anzudeuten.

Hinzufügen von Bytes für ein neues Datum. Bei jeder Erstausführung von ZE an einem Tag verwendet das Programm diese Routinen, um ein Byte an den Anfang aller Records zur Zeiterfassung anzuhängen. Dies ist notwendig, um sicherzustellen, daß die Position jedes Bytes mit dem entsprechenden Datum übereinstimmt.

```
106*LBL 30
107 ASROOM
108 2
109 *
110 SF 25
111 SEEKPT
112 FC?C 25
113 GTO 04
114 XEQ 70
115 GTO 30
```

Die Zeilen **106 bis 115** stellen sicher, daß in dem File **ZDATEN** genügend Bytes verfügbar sind, um ein Byte zu jedem Zeitrecord hinzuzufügen. Zeile **107** gibt n zurück, die Zahl der verfügbaren Bytes. Nehmen wir an, daß diese n Bytes zu den ersten n Records für Zeitdaten hinzugefügt werden; der $(n+1)$-te Job würde als erster kein Byte erhalten. Die Zeilen **108 und 109** berechnen $2n$, den Zeigerwert des $(n+1)$-ten Jobs. Die Zeilen **110 und 111** überprüfen, ob der $(n+1)$-te Job existiert. Wenn der Job nicht existiert, sind genügend Bytes verfügbar und der in den Zeilen **112 und 113** enthaltene Springbefehl zu Zeile 116 wird ausgeführt. Wenn der Job existiert, ruft Zeile **114** ein Unterprogramm in Zeile 221 auf, das den File ZDATEN vergrößert; Zeile 115 übergibt die Programmausführung an Zeile 106.

```
116*LBL 04
117 CLA
118 254
119 XTOA
120 1
121 SEEKPT
122 SF 25
```

Die Zeilen **116 bis 122** bereiten das Hinzufügen eines Bytes zu jedem Zeitrecord vor. Die Zeilen **117, 118 und 119** legen das Byte für «keine Stunden» im Alpha-Register ab. Die Zeilen **120 und 121** positionieren den Zeiger auf das erste Zeichen des ersten Records. Zeile **122** setzt den Flag zum Ignorieren von Fehlern; das Programm führt die anschließende Schleife so lange aus, bis Flag 22 durch Auftreten eines Fehlers gelöscht wird.

```
123*LBL 05
124 INSCHR
125 2
126 +
127 SEEKPT
128 FS? 25
129 GTO 05
130 RCL 03
131 1
132 DATE+
133 STO 03
134 GTO 20
```

Die Zeilen **123 bis 134** fügen ein Byte zu jedem Zeitrecord hinzu. Zeile **124** fügt das Byte für «keine Stunden» an der momentanen Zeigerposition ein. Die Zeilen **125, 126 und 127** versuchen, den Zeiger auf das erste Zeichen des nächsten Zeitrecords zu positionieren. Wenn der nächste Zeitrecord existiert, wird der in den Zeilen **128 und 129** enthaltene Sprungbefehl zu Zeile 123 ausgeführt. Wenn alle Zeitrecords ein Byte erhalten haben, addieren die Zeilen **130 bis 133** einen Tag zum *letzten Tag*, um das zu den Zeitrecords hinzugefügte Byte wiederzugeben. Zeile **134** enthält einen Sprungbefehl zu Zeile 27. (Der neue Wert des *letzten Datums* wird mit dem momentanen Datum verglichen, und falls nötig führt das Programm diese Routinen erneut aus.)

Arbeitsbeginn. Die folgenden Routinen untersuchen, welchen Job Sie beginnen und um wieviel Uhr die Arbeit aufgenommen wird. Wenn Sie den gleichen Job beginnen, an dem Sie das letzte Mal schon gearbeitet haben, wird die erste Routine ausgeführt. Wenn Sie den Namen eines existierenden Jobs (d.h. ein Job, die Sie schon mit ZE verwendet haben) eintasten, werden zusätzlich auch die zweite und dritte Routine ausgeführt. Wenn Sie den Namen eines neuen Jobs eingeben, werden auch die vierte, fünfte und sechste Routine ausgeführt. In allen Fällen wird die letzte Routine zur Bestimmung der Startzeit ausgeführt.

```
135♦LBL 40
136 "START "
137 ARCL 01
138 "↳?"
139 CF 23
140 AON
141 PROMPT
142 AOFF
143 FS? 01
144 GTO 07
145 FC? 23
146 GTO 11

147♦LBL 06
148 ASTO 01


149♦LBL 07
150 XEQ 50
151 FC? 01
152 GTO 11


153♦LBL 08
154 CF 23
155 " NEUER JOB?"
156 AON
157 PROMPT
158 AOFF
159 FS? 23
160 GTO 06

161♦LBL 09
162 CLA
163 ARCL 01
164 SF 25
165 APPREC
166 FS?C 25
167 GTO 10
168 XEQ 70
169 GTO 09
```

Die Zeilen **135 bis 146** fragen, ob Sie den gleichen Job beginnen, an dem Sie zuletzt gearbeitet haben. Die Zeilen **136 bis 141** zeigen die Frage **START *letzter Job*?** an. Die Zeilen **140 und 142** schalten automatisch das Alpha-Tastenfeld ein und aus. Wenn Sie eingangs nach einem gültigen Jobnamen gefragt wurden, da der gespeicherte Wert für den *letzten Job* nicht gültig war, wird die in den Zeilen **143 und 144** enthaltene Verzweigung ausgeführt. Wenn Sie bestätigen, daß Sie den *letzten Job* beginnen, übergeben die Zeilen **145 und 146** die Programmausführung an Zeile 183. (Wenn Sie an dieser Stelle einen anderen Jobnamen eingeben, werden die folgenden Routinen ausgeführt.)

Die Zeilen **147 und 148** machen den von Ihnen eingegebenen Jobnamen vorübergehend zum *letzten Job*.

Die Zeilen **149 bis 152** überprüfen den eingegebenen Jobnamen. Zeile **150** ruft ein Unterprogramm in Zeile 194 auf, das Flag 01 löscht, wenn der angegebene Job in **ZDATEN** existiert. Wenn der Name existiert, wird der in den Zeilen **151 und 152** enthaltene Sprungbefehl zu Zeile 183 ausgeführt.

Die Zeilen **153 bis 160** überprüfen, ob Sie absichtlich einen neuen Jobnamen spezifiziert haben. Die Zeilen **154 bis 157** zeigen die Frage **NEUER JOB?** an. Die Zeilen **156 und 158** schalten automatisch das Alpha-Tastenfeld ein und aus. Wenn Sie einen Jobnamen eingeben (weil Sie vorher einen Fehler beim Eintasten eines existierenden Jobnamens gemacht haben), übertragen die Zeilen **159 und 160** die Programmausführung an Zeile 147.

Die Zeilen **161 bis 169** erzeugen einen Record mit dem Namen des neuen Jobs. Die Zeilen **162 und 163** legen diesen Namen im Alpha-Register ab. Die Zeilen **164 bis 167** versuchen, den Namen an den File **ZDATEN** anzuhängen; wenn dies gelingt, wird zu Zeile 170 verzweigt. Wenn in **ZDATEN** nicht genügend Platz ist, um den Namen anzuhängen, ruft Zeile **168** ein Unterprogramm in Zeile 221 zur Vergrößerung von **ZDATEN** auf; im Anschluß daran wird die Programmausführung wieder mit Zeile 161 fortgesetzt und das Programm versucht nochmals, den Namen an **ZDATEN** anzuhängen.

```
170*LBL 10
171 CLA
172 254
173 XTOA
174 1
175 +
176 XTOA
177 SF 25
178 APPREC
179 FS?C 25
180 GTO 11
181 XEQ 70
182 GTO 10
183*LBL 11
184 TIME
185 "START JETZT?"
186 PROMPT
187 X<0?
188 XEQ 60
189 24
190 MOD
191 CHS
192 STO 02
193 RTN
```

Die Zeilen **170 bis 182** erzeugen einen Zeitrecord für den neuen Job. Die Zeilen **171 bis 176** legen ein «keine Stunden» Byte und ein Byte zur Kennzeichnung des Recordendes im Alpha-Register ab. Die Zeilen **177 bis 180** versuchen, diese Bytes an den File **ZDATEN** anzuhängen; wenn dies gelingt, wird zu Zeile 183 verzweigt. Wenn in **ZDATEN** nicht genügend Platz ist, um den neuen Zeitrecord einzutragen, ruft Zeile **181** ein Unterprogramm in Zeile 221 zur Vergrößerung von **ZDATEN** auf. Im Anschluß daran verzweigt Zeile **182** die Programmausführung zu Zeile 170 und es wird nochmals versucht, den neuen Zeitrecord in **ZDATEN** einzutragen.

Die Zeilen **183 bis 193** fragen Sie nach der Startzeit. Die Zeilen **184, 185 und 186** zeigen die Frage **START JETZT?** an. Wenn Sie eine andere Zeit eingeben, ersetzt diese den im X-Register abgelegten Wert für die momentane Zeit. Wenn Sie einen negativen Wert eingegeben haben (zur Kennzeichnung einer Zeit zwischen Mittag und Mitternacht), rufen die Zeilen **187 und 188** ein Unterprogramm in Zeile 216 auf, das diesen Wert in ein 24-Stunden-Format umwandelt. Wenn Sie die Zahl 24 zur Spezifikation von «Mitternacht» eingegeben haben, ersetzen die Zeilen **189 und 190** diesen Wert durch Null. Die Zeilen **191 und 192** deklarieren die Startzeit als neuen Wert der *letzten Zeit*; der Wert ist negativ, da er eine Startzeit repräsentiert. Zeile **193** übergibt die Programmausführung wieder an Zeile 43.

Durchsuchen von ZDATEN nach einem Job. Das Programm kann dieses Unterprogramm aus zwei Gründen aufrufen: Die Zeilen 17 und 150 rufen es auf, um festzustellen, ob ein bestimmter Job existiert, und Zeile 78 ruft es auf, um einen existierenden Job aufzufinden.

```
194*LBL 50
195 CLA
196 ARCL 01
197 SF 01
198 0
199 SEEKPT
```

Die Zeilen **194 bis 199** bereiten das Durchsuchen von **ZDATEN** nach einem Jobnamen vor. Die Zeilen **195 und 196** legen den Jobnamen im Alpha-Register ab. Zeile **197** setzt Flag 01 um anzudeuten, daß der Jobname nicht gefunden wurde; wenn die nächste Routine den Jobnamen findet, wird Flag 01 gelöscht. Die Zeilen **198 und 199** positionieren den Zeiger auf das erste Zeichen des ersten in **ZDATEN** gespeicherten Jobs.

```
200*LBL 12
201 POSFL
202 X<0?
203 RTN
204 ENTER↑
205 INT
206 1
207 +
208 SEEKPT
209 X<>Y
210 2
211 MOD
212 X≠0?
213 GTO 12
214 CF 01
215 RTN
```

Die Zeilen **200 bis 215** durchsuchen **ZDATEN** nach dem Jobnamen. Zeile **201** durchsucht **ZDATEN**, beginnend bei der momentanen Position des Zeigers. Wenn der Jobname nicht gefunden wird, bleibt Flag 01 gesetzt und die Zeilen **202 und 203** übergeben die Programmausführung wieder an Zeile 18 oder 151. Die Zeilen **204 bis 208** positionieren den Zeiger auf das erste Zeichen des nächsten Records, der im allgemeinen der Zeitrecord des Jobs ist. Es ist jedoch möglich, daß die in einem Zeitrecord enthaltenen Bytes zufällig einen Jobnamen bilden. Um sicherzustellen, daß in Zeile 201 wirklich ein Jobname gefunden wurde, überprüfen die Zeilen **209 bis 212**, ob der durch Zeile 201 zurückgegebene Zeigerwert das erste Zeichen in einem geradzahligen Record repräsentiert. Wenn nicht, übergibt Zeile **213** die Programmausführung an Zeile 199. Die Zeilen **214 und 215** löschen Flag 01, um anzudeuten, daß der Jobname existiert und verzweigen dann zu Zeile 18, 79 oder 151.

Umwandlung in ein 24-Stunden-Format

```
216*LBL 60
217 ABS
218 12
219 +
220 RTN
```

Die Zeilen **216 bis 220** wandeln einen negativen Zeitwert (der eine Zeit zwischen Mittag und Mitternacht repräsentiert) in ein 24-Stunden-Format um. Das Programm ruft dieses Unterprogramm in den Zeilen 66 und 188 auf.

Vergrößern von ZDATEN. Das Programm ruft das nachstehende Unterprogramm auf, wenn in **ZDATEN** nicht genügend Platz vorhanden ist, um ein Byte zu jedem Zeitrecord hinzuzufügen bzw. einen neuen Record für einen neuen Job anzuhängen.

```
221*LBL 70
222 "ZDATEN"
223 FLSIZE
224 1
225 +
226 SF 25
227 RESZFL
228 FS?C 25
229 RTN
230 " KEIN PLATZ"
231 PROMPT
232 GTO 70
233 .END.
```

Die Zeilen **221 bis 233** vergrößern **ZDATEN** um ein Register. Dieses Unterprogramm wird von den Zeilen 114, 168 und 181 aufgerufen. Die Zeilen **222 bis 225** berechnen, wieviele Register insgesamt benötigt werden. Die Zeilen **226 bis 229** versuchen, **ZDATEN** auf die benötigte Größe zu erweitern. Wenn dies gelingt, wird wieder auf die Zeilen 115, 169 oder 182 zurückgesprungen. Wenn kein erweitertes Speicherregister mehr verfügbar ist, zeigen die Zeilen **230 und 231** eine Fehlermeldung an und unterbrechen die Programmausführung. Nachdem ein anderer File verkleinert und die Programmausführung durch Drücken der Taste [R/S] fortgesetzt wurde, überträgt Zeile **232** die Programmausführung wieder auf Zeile 221 und das Programm versucht nochmals, **ZDATEN** zu vergrößern.

Erläuterung von Σ

Das Programm Σ stellt zuerst den Zeitraum fest, über den sich die Zusammenfassung erstrecken soll und erstellt dann die Zusammenfassung. Im Anschluß daran ermöglicht es Ihnen Σ, einen Job vollständig zu löschen oder alle Zeitdaten für einen bestimmten Zeitraum aus dem Speicher zu entfernen.

Von Σ verwendete Register und Flags

Das Programm Σ verwendet die folgenden Datenspeicher-Register und Benutzerflags. Schlagen Sie beim Durcharbeiten des Programm-Listings in diesen Tabellen nach.

Register und Registerinhalte für Σ

Register-adresse	Registerinhalt
R_{00}	Flagstatus bei Programmstart.
R_{01}	1.*ccc*; dabei repräsentiert *ccc* den Wert des Zeichenzeigers für das *Schlußdatum*.
R_{02}	Das letzte zusammenzufassende oder zu löschende Datum (*Schlußdatum*).
R_{03}	Das erste zusammenzufassende oder zu löschende Datum (*Anfangsdatum*).
R_{04}	Vorübergehende Datumsspeicherung. *nn* = höchste verwendete Registeradresse.
R_{05}	Schleifensteuerzahl.
R_{06}	Gesamtstundenzahl für einen einzelnen Job.
R_{07}	Gesamtstundenzahl für alle Jobs zusammen.
R_{08} bis R_{nn}	Tägliche Stundenzahl für das Anfangsdatum. … Schlußdatum.

Die in R_{08} bis R_{nn} abgelegten Zahlen haben das Format *xxx.yyy*. Der ganzzahlige Anteil ist der Dezimalwert eines Bytes und repräsentiert *xx.x* Stunden für einen Job. Der gebrochene Anteil ist der Dezimalwert eines Bytes und repräsentiert *yy.y* Stunden für alle Jobs zusammen.

Durch Benutzerflags dargestellte Zustände in Σ

Flagnummer	Dargestellter Zustand
00	Gelöscht = Eingabeaufforderung zur Zusammenfassung. Gesetzt = Eingabeaufforderung zum Löschen.
01	Gelöscht = Eingegebenes Datum ist zulässig. Gesetzt = Eingegebenes Datum ist unzulässig.

Programm-Listing zu Σ

```
01♦LBL "Σ"
02 RCLFLAG
03 STO 00
04 CF 21
05 1
06 STO 01
07 CF 00
```

Die Zeilen **01 bis 07** bilden die Anfangsroutine. Die Zeilen **02 und 03** speichern den momentanen Flagstatus in R_{00}. Zeile 04 desaktiviert einen eventuell angeschlossenen Drucker. Die Zeilen **05 und 06** legen den Zeigerwert für *erster Job/Anfangsdatum* in R_{01} ab. Zeile **07** setzt die Option «Zusammenfassung».

Datumsbestimmung. Diese Routinen bestimmen die Zeitspanne, über die sich die Zusammenfassung erstreckt. Sollten Sie später Tage löschen, verwendet das Programm die Routinen zur Bestimmung der zu löschenden Zeitspanne. Der Status von Flag 00 deutet an, ob das Programm zusammenfaßt oder löscht.

```
08♦LBL 20
09 FIX 4
10 "ZULETZT"
11 2
12 SEEKPTA
13 GETX
14 FC? 00
15 STO 02
16 "ZDATEN"
17 1
18 SEEKPTA
19 255
20 CLA
21 XTOA
22 CLX
23 POSFL
24 FRC
25 1 E3
26 *
27 -
28 DATE+
29 STO 03
30 STO 04
31 FS? 00
32 GTO 00
```

Die Zeilen **08 bis 32** berechnen die voreingestellten Datumswerte. (Wenn das Programm löscht, ruft Zeile 345 diesen Programmteil als Unterprogramm auf.) Die Zeilen **10 bis 13** rufen das *letzte Datum* zurück. Wenn das Programm momentan zusammenfaßt, speichern die Zeilen **14 und 15** den Wert für das *letzte Datum* als voreingestellten Wert für das *Schlußdatum* in R_{02}. Wenn das Programm momentan löscht, bleibt das zuletzt zusammengefaßte Datum als voreingestellter Wert für das *Schlußdatum* in R_{02} erhalten. In beiden Fällen bleibt das *letzte Datum* zur Berechnung des *Anfangsdatums* im Stack erhalten. Um diese Berechnung auszuführen, suchen die Zeilen **16 bis 27** das Byte mit dem Dezimalwert 255 (kennzeichnet das Ende eines Records) im ersten Zeitrecord und verwenden dessen Zeichenzeiger-Wert zur Berechnung der Anzahl der Tage zwischen dem *letzten Datum* und dem *Anfangsdatum*. (Der erste Zeitrecord ist der älteste Zeitrecord.) Die Zeilen **16 bis 26** geben die Anzahl der Bytes zurück, die vor dem Byte zur Kennzeichnung des Recordendes stehen. Zeile **27** subtrahiert diesen Wert von 1 (ist noch aus Zeile 17 im Stack vorhanden); das Ergebnis ist negativ, um in der Vergangenheit liegende Daten anzudeuten. Zeile **28** berechnet das *Anfangsdatum*. Zeile **29** speichert dieses Ergebnis als voreingestellten Wert in R_{03} ab und Zeile **30** legt das Ergebnis zur Vorbereitung der ersten Eingabeaufforderung in R_{04} ab. Wenn das Programm löscht, übergeben die Zeilen **31 und 32** die Programmausführung an Zeile 45; wenn das Programm zusammenfaßt, wird die Programmausführung mit der nächsten Routine fortgesetzt.

```
33*LBL 04
34 "    VON"
35 RCL 04
36 XEQ 01
37 FC? 22
38 GTO 00
39 XEQ 02
40 FC? 01
41 GTO 04
42 RCL 03
43 STO 04
44 GTO 04
```

Die Zeilen **33 bis 44** fragen Sie nach dem Anfangsdatum der Zusammenfassung. Die Zeilen **34 bis 36** fragen Sie unter Verwendung des Unterprogramms in Zeile 62, ob das Datum in R_{04} das gewünschte Datum ist. Wenn Sie das Datum in R_{04} akzeptieren, übergeben die Zeilen **37 und 38** die Programmausführung an Zeile 45. Wenn Sie ein anderes Datum eingeben, ruft Zeile **39** ein Unterprogramm in Zeile 71 auf, um das eingegebene Datum zu überprüfen. (Dieses Unterprogramm legt Ihr Datum in R_{04} ab und löscht Flag 01, wenn das Datum zulässig ist; andernfalls wird Flag 01 gelöscht.) Wenn Ihr Datum zulässig ist, bedingen die Zeilen **40 und 41** eine Verzweigung der Programmausführung nach Zeile 33 und der Computer fragt Sie nach einer Bestätigung des Datums. Wenn das Datum unzulässig ist, legen die Zeilen **42, 43 und 44** wieder den voreingestellten Wert in R_{04} ab und übertragen die Programmausführung danach an Zeile 33.

```
45*LBL 00
46 RCL 04
47 STO 03
48 RCL 02
49 STO 04
```

Die Zeilen **45 bis 49** verknüpfen die Routinen zur Anzeige der Eingabeaufforderungen. Die Zeilen **46 und 47** ersetzen den voreingestellten Wert für das *Anfangsdatum* durch das von Ihnen spezifizierte Datum. Die Zeilen **48 und 49** legen den voreingestellten Wert für das *Schlußdatum* zur Vorbereitung der ersten Eingabeaufforderung in R_{04} ab.

```
50*LBL 05
51 "  BIS"
52 RCL 04
53 XEQ 01
54 FC? 22
55 GTO 00
56 XEQ 02
57 FC? 01
58 GTO 05
59 RCL 02
60 STO 04
61 GTO 05
```

Die Zeilen **50 bis 61** fragen Sie nach dem letzten Datum, das zusammengefaßt oder gelöscht werden soll. Die Zeilen **50 bis 53** fragen Sie unter Verwendung des Unterprogramms in Zeile 62, ob das Datum in R_{04} das gewünschte Datum ist. Wenn Sie das Datum in R_{04} akzeptieren, übergeben die Zeilen **54 und 55** die Programmausführung an Zeile 93. Wenn Sie ein anderes Datum eingeben, ruft Zeile **56** ein Unterprogramm in Zeile 71 auf, um das eingegebene Datum zu überprüfen. Wenn das Datum zulässig ist, übertragen die Zeilen **57 und 58** die Programmausführung an Zeile 50 und der Computer fragt Sie nach einer Bestätigung des Datums. Wenn das Datum unzulässig ist, legen die Zeilen **59 bis 61** wieder den voreingestellten Wert in R_{04} ab und übergeben danach die Ausführung an Zeile 50.

```
62*LBL 01
63 AVIEW
64 PSE
65 "  "
66 ADATE
67 "┣?"
68 CF 22
69 PROMPT
70 RTN
```

Die Zeilen **62 bis 70** fragen Sie nach Datumswerten. Dieses Unterprogramm wird von den Zeilen 36 und 53 aufgerufen. Die Zeilen **63 und 64** zeigen den Inhalt des Alpha-Registers an (BIS oder VON). Die Zeilen **65, 66 und 67** legen das Datum zusammen mit einem Fragezeichen im Alpha-Register ab. Zeile **68** löscht den Flag zur Kennzeichnung einer numerischen Dateneingabe (dadurch kann das Programm später überprüfen, ob Sie ein anderes Datum eingegeben haben). Zeile **69** zeigt den Inhalt des Alpha-Registers an und unterbricht die Programmausführung. Zeile **70** übergibt die Ausführung wieder an Zeile 37 oder 54.

```
 71♦LBL 02
 72 CF 01
 73 STO 04
 74 RCL 02
 75 SF 25
 76 DDAYS
 77 FC?C 25
 78 GTO 03
 79 X<0?
 80 GTO 03
 81 RCL 03
 82 RCL 04
 83 DDAYS
 84 X<0?
 85 GTO 03
 86 RTN

 87♦LBL 03
 88 "UNZUL. DATUM"
 89 AVIEW
 90 TONE 5
 91 SF 01
 92 RTN

 93♦LBL 00
 94 RCL 04
 95 RCL 02
 96 DDAYS
 97 .1
 98 %
 99 ST+ 01
100 FS? 00
101 RTN
102 RCL 03
103 RCL 04
104 STO 02
105 DDAYS
106 8
107 +
108 STO 04
109 .1
110 %
111 6
112 +
```

Die Zeilen **71 bis 86** überprüfen das von Ihnen spezifizierte Datum. Dieses Unterprogramm wird von den Zeilen 39 und 56 aufgerufen. Zeile **72** löscht Flag 01. (Wenn Ihr Datum unzulässig ist, verzweigt das Programm zu Zeile 87 und Flag 01 wird gesetzt.) Zeile **73** legt Ihr Datum in R_{04} ab. Die Zeilen **74, 75 und 76** berechnen, wieviele Tage Ihr Datum vor dem *letzten Datum* liegt. Wenn Sie das Datum nicht in einem zulässigen Format eingegeben haben, verursacht Zeile **76** eine Fehlerbedingung und die Zeilen **77 und 78** übertragen daraufhin die Programmausführung an Zeile 87. Wenn das spezifizierte Datum nach dem *letzten Datum* liegt, übergeben die Zeilen **79 und 80** die Programmausführung an Zeile 87. Die Zeilen **81, 82 und 83** berechnen, um wieviele Tage Ihr Datum nach dem *Anfangsdatum* liegt. Wenn Ihr Datum vor dem *Anfangsdatum* liegt, übergeben die Zeilen **84 und 85** die Programmausführung an Zeile 87. Wenn das Datum allen drei Prüfungen standhält, übergibt Zeile **86** die Ausführung wieder an Zeile 40 oder Zeile 57; Flag 01 bleibt gelöscht um anzudeuten, daß Sie ein zulässiges Datum eingegeben haben.

Die Zeilen **87 bis 92** behandeln die Eingabe einses unzulässigen Datums. Die Zeilen **88, 89 und 90** zeigen eine Fehlermeldung an und erzeugen ein Tonsignal. Die Zeilen **91 und 92** übergeben die Programmausführung wieder an Zeile 40 oder 57; Flag 01 ist gesetzt, um ein unzulässiges Datum anzudeuten.

Die Zeilen **93 bis 112** beenden die Datumauswahl. Die Zeilen **94 bis 99** verändern den in R_{01} gespeicherten Zeigerwert für *erster Job*/*Schlußdatum* gemäß dem von Ihnen spezifizierten *Schlußdatum*. Wenn das Programm löscht, geben die Zeilen **100 und 101** die Ausführung an Zeile 346 zurück. Die Zeilen **103 und 104** ersetzen das in R_{02} abgelegte *letzte Datum* durch das von Ihnen spezifizierte *Schlußdatum*. Die Zeilen **102, 103 und 105** berechnen, wieviele Tage zwischen dem *Anfangs-* und *Schlußdatum* der Zusammenfassung liegen; diese Differenz legt den zur Zusammenfassung benötigten Registerblock fest. R_{08} ist das Register mit der niedrigsten Adresse in dem Block zur Speicherung der täglichen Stundenzahl. Die Zeilen **106, 107 und 108** legen den Wert *nnn* in R_{04} ab (R_{nnn} ist das Register mit der höchsten Adresse in dem Block). Die Zeilen **109 bis 112** berechnen 6.*nnn*; dieser Wert kennzeichnet den Registerblock zur Speicherung der täglichen und der gesamten Stundenzahl.

Vorbereitung der Zusammenfassung. Die folgenden Routinen löschen Register und erzeugen die Überschrift der Zusammenfassung.

```
113♦LBL 06
114 SF 25
115 CLRGX
116 FS?C 25
117 GTO 01
118 RCL 04
119 1
120 +
121 PSIZE
122 X<>Y
123 GTO 06

124♦LBL 01
125 SF 12
126 CF 13
127 SF 21
128 ADV
129 ADV
130 " ZEITDATEN"
131 SF 25
132 PRA
133 FS?C 25
134 GTO 02
135 CF 21
136 AVIEW
137 PSE
```

Die Zeilen **113 bis 123** löschen die Register zur Speicherung der täglichen und der gesamten Stundenzahl. Die Zeilen **114 und 115** versuchen, den Registerblock zu löschen. Wenn genügend Register der Datenspeicherung zugewiesen sind (d.h. wenn R_{nnn} existiert), übergeben die Zeilen **116 und 117** die Programmausführung an Zeile 124. Andernfalls ordnen die Zeilen **118 bis 121** *nnn* + 1 Register der Datenspeicherung zu. Zeile **122** legt 6.*nnn* im X-Register ab und Zeile **123** verzweigt das Programm zu Zeile 113.

Die Zeilen **124 bis 137** beginnen den Aufbau der Kopfzeile der Überschrift der Zusammenfassung. Die Zeilen **125 und 126** wählen den Druckmodus: doppelte Zeichenbreite und Großbuchstaben. Die Zeilen **128 und 129** bedingen einen Papiervorschub um zwei Zeilen. Zeile **130** legt die Überschrift im Alpha-Register ab. Zeile **131** setzt den Flag zum Ignorieren eines Fehlers, um die Druckerfunktion PRA (*print Alpha*) in Zeile **132** vorzubereiten. Bei angeschlossenem und eingeschaltetem Drucker wird nach Ausdruck des Alpha-Registerinhalts die Programmausführung durch die Zeilen **133 und 134** an Zeile 138 übergeben. Bei fehlendem bzw. ausgeschaltetem Drucker löscht Zeile **135** den Flag zur Druckersteuerung (sonst würde die Funktion PRA die Programmausführung unterbrechen); danach zeigen die Zeilen **136 und 137** die Überschrift an.

```
138*LBL 02
139 SF 13
140 "    VON"
141 XEQ 03
142 "  "
143 RCL 03
144 ADATE
145 XEQ 03
146 "    BIS"
147 XEQ 03
148 "  "
149 RCL 02
150 ADATE
151 XEQ 03
152 CF 13
153 SF 21
154 GTO 00
```

Die Zeilen **138 bis 154** schließen die Anzeige bzw. den Ausdruck der Kopfzeile der Zusammenfassung ab. Zeile **139** wählt den Druckmodus «Kleinschreibung». Die Zeilen **140 bis 145** zeigen die Meldung ***von Anfangsdatum*** an bzw. drukken dies aus. Die Zeilen **146 bis 151** zeigen die Meldung ***bis Schlußdatum*** an bzw. drucken dies aus. Die Zeilen 141, 145, 147 und 151 rufen ein Unterprogramm in Zeile 155 auf, das den Inhalt des Alpha-Registers anzeigt bzw. ausdruckt. Zeile ***152*** wählt den Druckmodus «Großbuchstaben». Zeile ***153*** setzt den Flag zur Druckersteuerung; dadurch wird sichergestellt, daß die folgenden AVIEW Anweisungen entweder die Einträge in der Zusammenfassung ausdrucken oder die Ausführung unterbrechen, damit Sie Zeit zum Mitschreiben der angezeigten Ergebnisse haben. Zeile **154** übergibt die Programmausführung an Zeile 161.

```
155*LBL 03
156 ADV
157 AVIEW
158 FC? 21
159 PSE
160 RTN
```

Die Zeilen **155 bis 160** zeigen die Kopfzeile an bzw. drucken sie aus. Dieses Unterprogramm wird in den Zeilen 141, 145, 147 und 151 aufgerufen. Zeile **156** bewirkt einen Papiervorschub (bei angeschlossenem Drucker). Zeile **157** zeigt den Inhalt des Alpha-Registers an; wenn ein angeschlossener Drucker eingeschaltet ist, wird der Inhalt des Alpha-Registers ausgedruckt. Wenn kein Drucker angeschlossen ist bzw. wenn ein angeschlossener Drucker ausgeschaltet ist, verlängern die Zeilen **158 und 159** die Anzeige (Flag 21 wurde in Zeile 135 gelöscht, wenn durch die Druckeranweisung PRA ein Fehler verursacht wurde). Zeile **60** gibt die Ausführung an Zeile 142, 146, 148 oder 152 zurück.

Zurückrufen von Zeitdaten. Die folgenden Routinen überprüfen, ob für einen Job Zeitdaten für die Zeitspanne der Zusammenfassung existieren; diese Daten werden dann aus dem erweiterten Speicher zurückgerufen. Die erste (kurze) Routine wird nur einmal ausgeführt; die zweite Routine wird einmal pro Job ausgeführt und die letzten beiden Routinen werden einmal pro Tag und Job ausgeführt.

```
161*LBL 00
162 FIX 1
163 CLX
164 SEEKPT
```

Die Zeilen **161 bis 164** bereiten den Zugriff auf den ersten Job vor. Zeile **162** wählt ein numerisches Ausgabeformat. Die Zeilen **163 und 164** positionieren den Zeiger in ZDATEN (momentaner File) auf das erste Zeichen des ersten Records (d.h. auf den ersten Buchstaben des ersten Jobnamens).

```
165♦LBL 07
166 RCL 04
167 .1
168 %
169 8
170 +
171 STO 05
172 CLX
173 STO 06
174 GETREC
175 RCLPT
176 INT
177 RCL 01
178 +
179 .001
180 +
181 SF 25
182 SEEKPT
183 FC?C 25
184 GTO 10
185 ADV
186 ADV
187 AVIEW
188 LASTX
189 -
190 SEEKPT
191 GETREC
```

Die Zeilen **165 bis 191** bereiten den Abruf der Zeitdaten eines Jobs aus dem erweiterten Speicher vor. Die Zeilen **166 bis 171** legen die Schleifensteuerzahl 8.*nnn* in R_{05} ab. Die Zeilen **172 und 173** löschen R_{06}; in diesem Register werden die Gesamtstunden für diesen Job aufsummiert. Zeile **174** ruft den Jobnamen in das Alpha-Register zurück. Bevor der Jobname angezeigt (gedruckt) wird, überprüft das Programm, ob für diesen Job Zeitdaten für die Zeitspanne der Zusammenfassung existieren. Die Zeilen **174 bis 178** berechnen den Zeigerwert des *Schlußdatums* in diesen Record. (Wenn der Record kein Byte für das *Schlußdatum* enthält, existieren keine Daten für die Zusammenfassung.) Die Zeilen **179 und 180** stellen den Zeiger um ein Zeichen vor, um sicherzustellen, daß das Byte zur Kennzeichnung des Zeitrecordendes nicht als Datenbyte interpretiert wird. Die Zeilen **181 bis 184** versuchen, den Zeiger auf die berechnete Stelle zu positionieren; wenn dort kein Byte vorhanden ist, wird die Programmausführung an Zeile 243 übergeben. (Wenn der Job zu jung für die Zusammenfassung ist, so gilt dies auch für alle nachfolgenden Jobs.) Die Zeilen **185, 186 und 187** bewirken einen Papiervorschub um zwei Zeilen und zeigen den Jobnamen an bzw. drucken ihn aus. Die Zeilen **188, 189 und 190** stellen den Zeiger auf das *Schlußdatum*; Zeile **191** ruft bis zu 24 Bytes aus dem Zeitrecord dieses Jobs in das Alpha-Register zurück.

```
192♦LBL 08
193 254
194 ATOX
195 X>Y?
196 GTO 02
197 X≠0?
198 GTO 01
199 GETREC
200 GTO 08
```

Die Zeilen **192 bis 200** rufen jedes Byte ab und überprüfen es. Diese Routine und die folgende bilden eine Schleife. Die Zeilen **193 und 194** legen den Wert 254 im Y-Register und den Dezimalwert eines Bytes im X-Register ab. (Um die Routine mehrmals hintereinander auszuführen, überträgt Zeile 194 Bytes in der Reihenfolge des Auftretens im erweiterten Speicher aus dem Alpha- in das X-Register.) Wenn das Byte zur Kennzeichnung des Recordendes (Dezimalwert 255) erscheint, bedingen die Zeilen **195 und 196** ein Verlassen der Schleife, indem sie die Ausführung an Zeile 215 übergeben. Die Zeilen **197 und 198** übertragen bei nicht leerem Alpha-Register die Programmausführung an Zeile 210. Wenn das Alpha-Register leer ist und das Byte zur Kennzeichnung des Recordendes noch nicht gefunden wurde, sind noch weitere Bytes dieses Zeitrecords aus dem erweiterten Speicher abzurufen; Zeile **199** ruft bis zu 24 weitere Bytes zurück und Zeile **200** gibt die Ausführung an Zeile 192 ab.

```
201*LBL 01
202 X=Y?
203 CLX
204 ST+ 06
205 ST+ 07
206 .1
207 %
208 +
209 RCL IND 05
210 FRC
211 +
212 STO IND 05
213 ISG 05
214 GTO 08
```

Die Zeilen **201 bis 214** speichern den Dezimalwert jedes Bytes im Hauptspeicher. Die Zeilen **202 und 203** ersetzen ein Byte für «keine Stunden» (Dezimalwert 254) durch den Dezimalwert 0. Die Zeilen **204 und 205** addieren den Bytewert zum Inhalt von R_{06} (Gesamtstunden für diesen Job) und zum Inhalt von R_{07} (Gesamtstunden für alle Jobs). Für den Bytewert *nnn* legen die Zeilen **206, 207 und 208** den Wert *nnn.nnn* im X-Register ab. Die Zeilen **209 und 210** rufen *.ppp* zurück; dabei ist *ppp* der aufsummierte Bytewert aller Jobs an diesem Tag. Die Zeilen **211 und 212** legen *nnn.qqq* in dem Register für diesen Tag ab; dabei ist $qqq = nnn + ppp$. Die Zeilen **213 und 214** erhöhen die Schleifensteuerzahl und geben die Ausführung an Zeile 192 zurück, wenn die Zeitspanne der Zusammenfassung noch nicht voll abgearbeitet ist.

Anzeige (Ausdruck) der Zeitdaten zu jedem Job. Nachdem ein Job aus dem erweiterten Speicher in den Hauptspeicher zurückgerufen wurde, zeigen die folgenden Routinen die dazugehörigen Zeitdaten an bzw. drucken diese aus. Erst danach wird der nächste Job abgerufen.

```
215*LBL 02
216 8.007
217 RCL 05
218 INT
219 9
220 -
221 +
222 STO 05
223 RCL 02
224 LASTX
225 CHS
226 DATE+
```

Die Zeilen **215 bis 226** bereiten die Anzeige (bzw. den Ausdruck) der täglichen Stunden für den momentanen Job vor. Zeile **216** spezifiziert eine Schleifensteuerzahl, die eine einmalige Ausführung der nächsten Routine zum Zugriff auf R_{08} (tägliche Stunden am *Schlußdatum*) veranlaßt. Die Zeilen **217 bis 220** berechnen, wieviele Register über R_{08} Daten für diesen Job enthalten (in diesen Registern sind die Stunden an Tagen vor dem *Schlußdatum* gespeichert). (Es kann sein, daß keine Zeitdaten für den Job am Anfang der zusammenzufassenden Zeitspanne existieren.) Das Ergebnis *m* wird zweimal verwendet. Zuerst legen die Zeilen **221 und 222** eine Schleifensteuerzahl $(8 + m).007$ in R_{05} ab; dabei ist R_{8+m} das Register mit der höchsten Adresse, das Daten für diesen Job enthält. (Zeile 217 ruft den Endwert der Schleifensteuerzahl der vorhergehenden Routine zurück; auf das von der vorhergehenden Routine zuletzt zugegriffene Register wird von der nächsten Routine als erstes zugegriffen.) Der Wert *m* wird in den Zeilen **223 bis 226** ein zweites Mal verwendet, um *m* Tage vom *Schlußdatum* zu subtrahieren; das Ergebnis ist das *Anfangsdatum* der Zusammenfassung für diesen Job.

```
227♦LBL 09
228 RCL IND 05
229 INT
230 XEQ 00
231 DSE 05
232 GTO 09
233 RCL 06
234 XEQ 01
235 RCLPT
236 INT
237 1
238 +
239 SF 25
240 SEEKPT
241 FS?C 25
242 GTO 07
```

Die Zeilen **227 bis 242** zeigen die Zusammenfassung für diesen Job an bzw. drucken sie aus. Das Programm führt für jeden Tag einmal die Schleife von Zeile 228 bis 232 aus. Die Zeilen **228 und 229** rufen den Dezimalwert des Bytes für die Stundenzahl des Jobs an diesem Tag zurück. Zeile **230** ruft ein Unterprogramm in Zeile 265 auf, das das Datum und die Stunden anzeigt (ausdruckt). Die Zeilen **231 und 232** erniedrigen die Schleifensteuerzahl und übergeben die Ausführung an Zeile 227, wenn weitere Tage zu verarbeiten sind. Zeile **233** ruft den Dezimalwert des Bytes für die Gesamtstunden des Jobs zurück und Zeile **234** ruft ein Unterprogramm in Zeile 279 auf, das die Meldung GESAMT und die Stunden anzeigt (ausdruckt). Die Zeilen **235 bis 240** versuchen, den Zeiger in ZDATEN auf den Namen des nächsten Jobs zu positionieren. Wenn ein weiterer Job in ZDATEN existiert, übergeben die Zeilen **241 und 242** die Programmausführung an Zeile 165.

Anzeige (Ausdruck) der abschließenden Zusammenfassung. Nachdem alle Jobs zusammengefaßt sind, zeigt das Programm eine abschließende Zusammenfassung an (bzw. druckt diese aus), in der die Gesamtstunden an jedem Tag und die Gesamtstunden während der zusammengefaßten Zeitspanne angegeben werden.

```
243♦LBL 10
244 RCL 04
245 .007
246 +
247 STO 05
248 ADV
249 ADV
250 "ZUSAMMEN-
251 "FFASSUNG"
252 AVIEW
253 RCL 03
```

Die Zeilen **243 bis 253** bereiten die Anzeige (den Ausdruck) der abschließenden Zusammenfassung vor. Die Zeilen **244 bis 247** speichern *nnn*.007 in R_{05}; dabei ist R_{nnn} das Register, in dem die tägliche Stundenzahl für das *Anfangsdatum* gespeichert ist. Die Zeilen **248 und 249** bewirken einen Papiervorschub um zwei Zeilen. Die Zeilen **250 bis 252** drucken die Meldung ZUSAMMEN-FASSUNG aus. (Diese Meldung ist für den Druckerbetrieb formatiert und wird in zwei Zeilen untereinander ausgedruckt. Wenn kein Drucker angeschlossen ist, wird die Meldung mit mehreren Leerzeichen zwischen den beiden Worten angezeigt.) Die Zeile **253** ruft das *Anfangsdatum* zurück.

```
254♦LBL 11
255 RCL IND 05
256 FRC
257 1 E3
258 *
259 XEQ 00
260 DSE 05
261 GTO 11
262 RCL 07
263 XEQ 01
264 GTO 03
```

Die Zeilen **254 bis 264** zeigen die abschließende Zusammenfassung an bzw. drucken diese aus. Das Programm durchläuft für jeden Tag einmal die Schleife von Zeile 255 bis Zeile 261. Die Zeilen **255 bis 258** rufen den Bytewert der Gesamtstunden für diesen Tag zurück. Zeile **259** ruft ein Unterprogramm in Zeile 265 auf, das das Datum und die Stunden anzeigt (ausdruckt). Die Zeilen **260 und 261** erniedrigen die Schleifensteuerzahl und übergeben die Ausführung an Zeile 254, wenn weitere Tage verarbeitet werden müssen. Zeile **262** ruft den Bytewert der Gesamtstunden für die gesamte Zeitspanne der Zusammenfassung zurück. Zeile **263** ruft ein Unterprogramm in Zeile 279 auf, das die Meldung GESAMT und die Stunden anzeigt (ausdruckt).

Anzeige- bzw. Druckunterprogramme. Die folgenden drei Routinen zeigen das Datum, die Meldung GESAMT und die Stunden an (bzw. drucken diese Information aus). Das Programm ruft die ersten beiden Routinen als Unterprogramme auf; beide geben die Ausführung an die dritte Routine ab, die ihrerseits die Ausführung wieder an die Zeile nach dem Unterprogrammaufruf zurückgibt.

```
265♦LBL 00
266 X<>Y
267 1
268 RCL Y
269 " "
270 ADATE
271 "├ "
272 DOW
273 X=Y?
274 ADV
275 RDN
276 DATE+
277 X<>Y
278 GTO 02

279♦LBL 01
280 "GESAMT "

281♦LBL 02
282 10
283 /
284 LASTX
285 X>Y?
286 "├ "
287 X↑2
288 X>Y?
289 "├ "
290 ARCL Y
291 AVIEW
292 RCL Z
293 RTN
```

Die Zeilen **265 bis 278** legen das Datum im Alpha-Register ab. Dieses Unterprogramm wird von den Zeilen 230 und 259 aufgerufen; vor seiner Ausführung muß das entsprechende Datum im Y-Register und der Bytewert für die täglichen Stunden im X-Register abgelegt sein. Die Zeilen **266, 267 und 268** legen das Datum in den X- und Z-Registern, den Wert 1 im Y-Register und den Bytewert im T-Register ab. Die Zeilen **269, 270 und 271** legen das Datum im Alpha-Register ab. Die Zeilen **272, 273 und 274** bewirken einen Papiervorschub, wenn das Datum auf einen Montag fällt. Zeile **275** rollt den Wert 1 in das X-Register, das Datum in das Y-Register und den Bytewert in das Z-Register. Zeile **276** berechnet das nachfolgende Datum (und bereitet damit den nächsten Schleifendurchlauf vor). Zeile **277** lädt den Bytewert wieder in das X-Register zurück, wo er vor Ausführung dieser Routine abgelegt war. Zeile **278** übergibt die Programmausführung an Zeile 281.

Die Zeilen **279 und 280** legen den String GESAMT im Alpha-Register ab. Dieses Unterprogramm wird von den Zeilen 234 und 263 aufgerufen.

Die Zeilen **281 bis 293** hängen die berechneten Stunden an den Inhalt des Alpha-Registers an und zeigen die resultierende Zeichenkette an bzw. drucken sie aus. Die Zeilen **282 und 283** wandeln den Bytewert in Stunden und Zehntelstunden um. Die Zeilen **284 bis 289** stellen sicher, daß die Stunden rechtsbündig gedruckt werden. Zeile **290** hängt die Stunden an den Inhalt des Alpha-Registers an und Zeile **291** zeigt den resultierenden Inhalt an bzw. druckt ihn aus. Zeile **292** ruft das nächste Datum in das X-Register zurück (und bereitet damit den nächsten Schleifendurchlauf vor). Zeile **293** übergibt die Ausführung wieder an Zeile 231, 235, 260 oder 264.

Löschen von Tagen oder Jobs

```
294♦LBL 03
295 CF 21
296 CLA
297 11
298 PASN
299 15
300 PASN
301 12
302 PASN
303 SF 27
304♦LBL 12
305 "LOESCHE TAG-A,J"
306 "OB-B?E"
307 PROMPT
308 TONE 5
309 GTO 12
```

Die Zeilen **294 bis 303** bereiten den Computer auf die vorzunehmenden Löschoperationen vor. Zeile **295** löscht den Flag zur Druckersteuerung. Die Zeilen **296** bis **302** heben Belegungen der Tasten [A], [B] und [E] auf. Zeile **303** schaltet das User-Tastenfeld ein.

Die Zeilen **304 bis 309** zeigen die Meldung LOESCHE TAG-A,JOB-B?E an. Als Reaktion können Sie die Taste [A] (*Lösche Tage*), die Taste [B] (*Lösche Job*) oder die Taste [E] (*Ende*) drücken. (Wenn Sie eine dieser Tasten drücken, beginnt die Programmausführung bei dem entsprechenden Label.) Wenn Sie versehentlich die Taste [R/S] drücken, erzeugt Zeile **308** ein Tonsignal und Zeile **309** gibt die Ausführung an Zeile 304 zurück.

Löschen eines Jobs

```
310♦LBL B
311 CF 23
312 "JOBNAME?"
313 AON
314 PROMPT
315 AOFF
316 FC? 23
317 GTO 12
318 CLX
```

Die Zeilen **310 bis 318** bereiten das Löschen eines Jobs vor. Die Zeilen **311 bis 314** fragen Sie nach dem Namen des zu löschenden Jobs. Die Zeilen **313 und 315** schalten automatisch das Alpha-Tastenfeld ein bzw. aus. Wenn Sie die Taste [R/S] drücken, ohne vorher einen Namen eingegeben zu haben, übergeben die Zeilen **316 und 317** die Programmausführung an Zeile 304. Zeile **318** spezifiziert den Zeigerwert für die nächste Routine. (Der Zeigerwert 0 repräsentiert den Anfang des Files **ZDATEN**.)

```
319*LBL 13
320 SEEKPT
321 POSFL
322 X<0?
323 GTO 00
324 2
325 MOD
326 X=0?
327 GTO 01
328 RCLPT
329 INT
330 1
331 +
332 GTO 13

333*LBL 00
334 TONE 5
335 "FEHL. NAME"
336 AVIEW
337 PSE
338 GTO 12

339*LBL 01
340 DELREC
341 DELREC
342 GTO 12
```

Die Zeilen **319 bis 332** suchen den zu löschenden Job. Die Zeilen **320 und 321** durchsuchen **ZDATEN** nach dem Job, den Sie im Alpha-Register spezifiziert haben; die Suche beginnt bei dem im X-Register angegebenen Zeigerwert. Wenn der Name nicht in **ZDATEN** existiert, geben die Zeilen **322 und 323** die Ausführung an eine Fehlerroutine in Zeile 333 ab. Wenn der durch Zeile 321 zurückgegebene Zeigerwert einen zulässigen Jobnamen andeutet, übertragen die Zeilen **324 bis 327** die Ausführung an Zeile 339. Wenn der Zeigerwert keinen Jobnamen andeutet (d.h. wenn die Bytes eines Zeitrecords zufällig einen Namen bilden), berechnen die Zeilen **328 bis 332** den Zeigerwert für den nächsten Record und übertragen die Ausführung an Zeile 319.

Die Zeilen **333 bis 338** bilden eine Fehlerroutine. Zeile **334** erzeugt ein Tonsignal; die Zeilen **335, 336 und 337** zeigen eine Fehlermeldung an; und Zeile **338** übergibt die Programmausführung an Zeile 304.

Die Zeilen **339 bis 342** löschen den Job aus **ZDATEN**. Zeile **340** löscht den Jobnamen und Zeile **341** löscht den Zeitrecord des Jobs. Zeile **342** übergibt die Ausführung an Zeile 304.

Löschen von Tagen

```
343*LBL A
344 SF 00
345 XEQ 20
346 1.001
347 RCL 01
348 X<Y?
349 X<>Y
350 CLA
351 255
352 XTOA
```

Die Zeilen **343 bis 352** bereiten das Löschen von Tagen vor. Die Zeilen **344 und 345** rufen ein Unterprogramm in Zeile 08 zur Bestimmung des ersten und letzten zu löschenden Datums auf. (Die Hauptaufgabe dieses Unterprogramms ist es, die Zeitspanne der Zusammenfassung zu bestimmen; durch Setzen von Flag 00 wird der Ablauf dieses Unterprogramms abgeändert und es wird die zu löschende Zeitspanne bestimmt.) Die Zeilen **346 bis 349** geben 1.*ccc* zurück; dabei stellt *ccc* den Wert des Zeichenzeigers für das letzte zu löschende Datum dar. Zeile **346** spezifiziert den Minimalwert – es muß mindestens ein Byte zur Darstellung eines Tages in jedem Zeitrecord verbleiben – und Zeile **347** ruft den Wert zurück, der mit dem Datum übereinstimmt, das die von Ihnen spezifizierte Zeitspanne abschließt. Die Zeilen **348 und 349** legen den größeren der beiden Werte im X-Register ab. Die Zeilen **350, 351 und 352** legen die Zahl 255 im X-Register ab und erzeugen daraus das Byte zur Kennzeichnung des Recordendes im Alpha-Register.

```
353♦LBL 14
354 DELCHR
355 INSCHR
356 X<>Y
357 2
358 +
359 SF 25
360 SEEKPT
361 FC?C 25
362 GTO 12
363 X<>Y
364 GTO 14
```

Die Zeilen **353 bis 364** löschen Tage aus dem Zeitrecord jedes Jobs. Das Programm durchläuft diese Schleife einmal für jeden Zeitrecord, der Daten innerhalb des angegebenen Zeitintervalls enthält. Zeile **354** löscht alle der Zeitspanne zugeordneten Bytes und Zeile **355** fügt ein neues Byte zur Kennzeichnung des Recordendes ein. Die Zeilen **356 bis 360** versuchen, den Zeiger auf das Byte zu positionieren, das dem letzten zu löschenden Datum im Zeitrecord des nächsten Jobs entspricht. Wenn dieses Byte nicht existiert, übergeben die Zeilen **361 und 362** die Ausführung an Zeile 304. (Dieser Fall tritt auf, wenn keine Records mehr verbleiben oder wenn für diesen Job und alle nachfolgenden Jobs kein Datum innerhalb der spezifizierten Zeitspanne existiert.) Zeil **363** legt wieder die Zahl 255 im X-Register ab und Zeile **364** übergibt die Programmausführung an Zeile 353.

Beenden von Σ

```
365♦LBL E
366 RCL 00
367 STOFLAG
368 CLX
369 .END.
```

Die Zeilen **365 bis 369** schließen Σ ab. Die Zeilen **366 und 367** stellen wieder den Flagstatus her, der vor Ausführung von Σ bestand. Zeile **368** löscht die Flagstatus-Daten der Flags aus dem X-Register.

Anhänge

Fehler- und Statusmeldungen

Dieser Anhang listet alle vom HP-41CX erzeugten Fehler- und Statusmeldungen.

Bei dem Versuch, eine auf dem HP-41 unzulässige Operation auszuführen, erscheint in der Anzeige eine Fehlermeldung und die Operation wird nicht ausgeführt. Sie können die Fehlermeldung durch Drücken von ← löschen. Wenn der Fehler während der Ausführung eines Programms aufgetreten ist, können Sie die fehlerverursachende Programmzeile durch Umschalten in den Programm-Modus anzeigen.

Einige Meldungen sind als Statusmeldungen gekennzeichnet. Eine derartige Meldung ist rein informatorisch und *nicht* das Resultat einer Fehlerbedingung.

Im folgenden beziehen sich die Variablen x, y und z auf die Inhalte des X-, Y- bzw. Z-Registers.

Anzeige	Funktionen	Bedeutung
ALPHA DATA	mathematische Zeit- erweiterte Speicher- Alle übrigen Funktionen, die numerische Daten verwenden	Nichtnumerische Daten wurden für eine Funktion spezifiziert, die numerische Daten erwartet; das X-Register (bzw. das Y- oder Z-Register) enthält Alpha-Daten.
CHKSUM ERR	GETP GETSUB	Der Programmfile oder ein Teil davon ist nicht lesbar und damit verloren.
DATA ERROR	mathematische SDEV	Die gegebenen Operanden bedingen eine unzulässige mathematische Operation (Division durch Null, Wurzel aus einer negativen Zahl).
	Zeit-	Unzulässiger Wert im X-Register.
	y^x	$x \leq 0$ und $y = 0$; oder x nicht ganzzahlig und $y < 0$.
	T+X	$x < -9999{,}595999$ oder $x > 9999{,}595999$; ***MM*** oder ***SS*** > 59.
	TONE	$x \geq 10$ oder $x < 0$.
	MEAN	$n = 0$.

Anzeige	Funktionen	Bedeutung
DATA ERROR (Fortsetzung)	OCT	$\|x\| > 1073741823_{10}$ oder x nicht ganzzahlig.
	DEC	x ist nicht ganzzahlig; oder eine Ziffer in x ist 8 oder 9.
	FACT	$x < 0$ oder nicht ganzzahlig.
	%CH	$y = 0$.
	FIX, SCI, ENG	$\|n\| \geq 10$.
	GETKEYX	$x \geq 100$.
	AROT, POSA, XTOA, X<>F	$x > 255$.
	PSIZE, CLRGX, SEEKPT, SEEKPTA, SWPT	$x > 999$.
	CRFLAS, CRFLD	$x = 0$. (Versuch einen File mit einer Größe von 0 Registern zu erzeugen.)
	EMDIRX, CLALM, RCLALMX, RESZFL	$x = 0$ or $x > 999$.
	STOFLAG	x (oder y, falls x eine Sequenz von Flags in der Form *bb.ee* spezifiziert) wurde nicht durch Ausführung von RCLFLAG erzeugt.
	X=NN?, X≠NN?, X<NN?, X<=NN?, X>NN?, X>=NN?	Das Y-Register enthält von X, Y, Z, T oder L verschiedene Alpha-Daten; oder $y > 999$.
DATA ERROR X	DDAYS	x ist ein unzulässiges oder negatives Datum; oder der Jahranteil enthält mehr als vier Stellen.
	XYZALM	$x \geq 24$ oder unzulässiger ***HH.MMSS*** Wert.
DATA ERROR Y	DATE+, DDAYS, XYZALM	y ist ein unzulässiges oder negatives Datum; oder der Jahranteil enthält mehr als vier Stellen.

Anzeige	Funktionen	Bedeutung
DATA ERROR Z	XYZALM	$z > 10000$ oder unzulässiger ***HHHH.MMSS*** Wert.
DUP FL (*duplicate file*)	erweiterte Speicher-	Im erweiterten Speicher existiert bereits ein File gleichen Namens. ASCII/Text-, Daten- und Programmfiles können nicht die gleichen Namen haben. (Der spezifizierte File wird zum momentanen File.)
END OF FL (*end of file*)	erweiterte Speicher-	Eine nicht vorhandene Zeigeradresse wurde zur Positionierung des Zeigers oder zu einer Schreib-, Lese- oder Löschoperation innerhalb eines Files benutzt. (Bei SAVER und SEEKPTA wird der spezifizierte File zum momentanen File; Zeiger und File bleiben jedoch unverändert.)
	APPCHR APPREC INSCHR INSREC	Nicht genügend Platz, um das Zeichen oder den Record anzuhängen.
	GETAS SAVEAS	Der Filetransfer wurde nicht vollständig ausgeführt, da das Ende des Zielfiles vor dem Ende des Quellfiles gefunden wurde. (Ein Teil des Files wurde übertragen.)
END OF REC (*end of record*)	SEEKPT SEEKPTA	Während der Positionierung des Zeichenzeigers wurde das Ende des Records gefunden.
ERROR = D*nn** (*delta split error in* R_{nn})	SW	Die in R_{nn} oder R_{nn-1} gespeicherte Zahl ist nicht im Format ***HH.MMSS***; oder die in R_{nn} gespeicherte Zeit ist kleiner als die in R_{nn-1} gespeicherte Zeit.
ERROR = R*nn** (*split error in* R_{nn})	SW	Der ganzzahlige Anteil des Wertes in R_{nn} ist größer als 99.
FL NOT FOUND (*file not found*)	erweiterte Speicher-	Der spezifizierte File (bzw. der momentane File) ist im erweiterten Speicher nicht vorhanden. (Nach dem Löschen eines Files ist kein momentaner File definiert.)
FL SIZE ERR (*file size error*)	RESZFL	Die spezifizierte Filegröße *(x)* ist kleiner als die momentane Größe des Files und würde den Verlust von Daten bedingen (siehe Seite 213).
FL TYPE ERROR (*file type error*)	erweiterte Speicher-	Der Typ des spezifizierten Files (oder des momentanen Files) entspricht nicht dem von der Funktion benötigten Typ.

* Diese Fehlermeldung ist rein informatorisch und nicht das Resultat einer echten Fehlerbedingung (siehe Abschnitt 17).

Anzeige	Funktionen	Bedeutung
KEYCODE ERR (*keycode error*)	PASN	Die durch den Tastencode spezifizierte Taste ist nicht zuweisbar.
MEMORY LOST		Der Permanentspeicher wurde zurückgesetzt.
NAME ERR (*file name error*)	erweiterte Speicher-	Fehlender Filename (das Alpha-Register ist leer); oder das Alpha-Register enthält sieben Bytes mit Dezimalwert 255 (unzulässig).
	PCLPS SAVEP	Das spezifizierte Programm ist im Hauptspeicher nicht vorhanden.
NO	Flag- Vergleichs-	*Statusmeldung*. Eine Flag- oder Vergleichsfrage liefert das Ergebnis falsch.
NO DRIVE (*no drive device present*)	GETAS SAVEAS	Der HP-41 enthält kein HP-IL Modul oder die Interfaceschleife enthält keine Massenspeichereinheit.
NONEXISTENT	Speicher- Rückruf- erweiterte Speicher- Vergleichs- CLRGX SWPT	Eines oder mehrere der spezifizierten Register sind nicht als Datenspeicher definiert.
	ASN GTO XEQ Alarm-	Das spezifizierte oder aufgerufene Label existiert nicht. (Falls die Funktion ein globales Label verlangt, kann auch die Angabe eines lokalen Labels zu diesem Fehler führen.)
	ASN XEQ Alarm-	Die aufgerufene Funktion existiert nicht. Bei Alarmen muß die aufgerufene Funktion zusätzlich programmierbar sein. Bei Aufruf einer Funktion aus Katalog 2 muß die Quelleinheit mit angegeben werden.
	STOFLAG	Einer oder mehrere der spezifizierten Flags liegen nicht im Bereich von 0 bis 43.
NO ROOM	erweiterte Speicher-	Der im erweiterten Speicher verfügbare Platz reicht nicht zur Aufnahme des spezifizierten Files oder Programms aus.
	GETP GETSUB	Der im Hauptspeicher verfügbare Platz reicht nicht für das spezifizierte Programm aus.
	PSIZE	(Bei Ausführung als Programmanweisung.) Im Hauptspeicher ist nicht genügend Platz verfügbar.

Anzeige	Funktionen	Bedeutung
NO ROOM (Fortsetzung)	RESZFL	Der erweiterte Speicher enthält nicht genügend Platz zur Vergrößerung des Files.
	XYZALM	Zur Einrichtung des Alarms sind nicht genügend ungebundene Register verfügbar; oder 253 Register sind bereits mit Alarmen belegt.
	Text-Editor*	Nicht mehr genügend Platz für weitere Zeichen oder Records verfügbar.
NO SUCH ALM (*no such alarm*)	Alarm-	Der spezifizierte Alarm existiert nicht.
NULL	alle	*Statusmeldung*. Die spezifizierte Funktion wurde gelöscht.
OUT OF RANGE	numerische	Eine Zahl liegt außerhalb des Bereichs der auf dem HP-41 darstellbaren Zahlen. Overflow = ± 9,999999999 99.
	DATE+ T+X	Das resultierende Datum liegt außerhalb des zulässigen Kalenderbereichs.
PACKING TRY AGAIN	ASN PASN XEQ GTO · · GETP GETSUB Programm-Modus	*Statusmeldung*. Der Programmspeicher wurde gepackt; wiederholen Sie die letzte Operation. Wenn TRY AGAIN wieder erscheint, enthält der Hauptspeicher nicht genügend Platz zur Ausführung der Operation. Versuchen Sie den Speicher neu aufzuteilen.
	SIZE PSIZE	Der Programmspeicher wurde gepackt; wiederholen Sie die Operation. Falls TRY AGAIN wieder erscheint, ist nicht genügend Platz für die Neuaufteilung vorhanden.
PRIVATE	Magnetkartenleser-Anwender-ROMs	Versuch ein privates Programm zu listen; siehe Benutzerhandbuch zum Magnetkartenleser HP 82104A.
RAM	COPY	Versuch ein bereits im Hauptspeicher befindliches Programm dorthin zu kopieren.

* Diese Fehlermeldung ist rein informatorisch und nicht das Resultat einer echten Fehlerbedingung (siehe Abschnitt 14).

Anzeige	Funktionen	Bedeutung
REC TOO LONG	Text-Editor APPCHR INSCHR	Versuch, die maximale Recordlänge (254 Bytes) zu überschreiten. (Der Text-Editor wird verlassen.)
ROM		Versuch, ein im ROM (Festwertspeicher – wie etwa die Applikationsmodule) befindliches Programm zu ändern.
YES	Flag- Vergleichs-	*Statusmeldung*. Das Ergebnis einer Flag- oder Vergleichsabfrage ist wahr.

Anhang B

Mehr über überfällige Alarme

Inhalt

Bedingungen für die Ausführung überfälliger Alarme 361
Aus/Uhr-Bedingung 361
Alarm-Bedingung 361
Verhalten überfälliger Alarme unter der Alarm-Bedingung 362
Änderungen des Betriebszustandes 362
Unterbrechung eines überfälligen Alarms durch einen anderen überfälligen Alarm 363
Alarme und Unterprogrammebenen 363
Beispiel zur Abarbeitung überfälliger Alarme 363

Sobald ein Alarm aus einem der in Abschnitt 16 unter «Überfällige Alarme» beschriebenen Gründen überfällig wird, speichert das System diesen Alarm solange, bis er aktiviert oder gelöscht wird. Dies soll Sie an jeden Alarm erinnern, der nicht in der ursprünglich vorgesehenen Weise bearbeitet wurde. Wenn Sie eine Ansammlung mehrerer überfälliger Alarme im Speicher zulassen, können Sequenzen von automatischen Aktivierungen überfälliger Alarme auftreten. (Wenn sich mehrere übergangene, überfällige Alarme im Speicher befinden, kann die Reihenfolge, in der überfällige Alarme auftreten, recht kompliziert werden.) Wenn Sie die Absicht haben, überfällige oder simultane Alarme in Ihre Anwendungen mit einzubeziehen, werden Ihnen die Informationen in diesem Anhang von Nutzen sein. Simultane Alarme werden in der gleichen Reihenfolge wie überfällige Alarme aktiv.

Überfällige Alarme werden jedesmal automatisch aktiv, wenn Sie den Computer ausschalten oder ■ [ON] drücken. Diese Einrichtung soll Sie daran erinnern, daß ein oder mehrere Alarme überfällig sind. Aus diesem Grund werden bei Fälligkeit eines Alarms zunächst alle eventuell vorhandenen *übergangenen* überfälligen Alarme aktiv, bevor der gerade fällig gewordene Alarm abgearbeitet wird.* Dieser Anhang beschreibt die Regeln für die Aktivierungsfolgen in diesen beiden Fällen. (Eine Beschreibung und Klassifizierung überfälliger Alarme finden Sie in Abschnitt 16.)

* In Abschnitt 16 wird ein übergangener, überfälliger Alarm als ein Alarm definiert, der niemals abgearbeitet (aktiv) wurde, da seine Fälligkeit entweder für einen in der Vergangenheit liegenden Zeitpunkt eingerichtet oder durch eine Zeitänderungsfunktion ([SETIME], [SETDATE], [T+X] oder [CORRECT]) umgangen wurde.

Bedingungen für die Ausführung überfälliger Alarme

Aus/Uhr-Bedingung

Eine Aus/Uhr-Bedingung entsteht durch Ausschalten des Computers oder Anzeige der Uhr (mittels ■ [ON]). Wenn beim Auftreten dieser Bedingung irgendwelche überfälligen Alarme existieren, *versucht* der Computer diese zu aktivieren, beginnend mit dem frühesten Alarm. Sobald ein überfälliger Steueralarm oder ein überfälliger bedingter Alarm angetroffen wird, schaltet der Computer sich kurzzeitig aus, wodurch die Aus/Uhr-Bedingung aufgehoben wird. Beim Wiedereinschalten entsteht die im nächsten Teilabschnitt beschriebene Alarm-Bedingung. Der Steueralarm/bedingte Alarm wird dann unter der Alarm-Bedingung ausgeführt.

Unter der Aus/Uhr-Bedingung können Sie das folgende Verhalten erwarten:

- *Solange keine überfälligen Steueralarme/bedingten Alarme angetroffen werden,* bearbeitet das System alle überfälligen Meldealarme in chronologischer Reihenfolge (beginnend mit dem frühesten Alarm). Jeder Alarm beendet seinen Aktivierungszyklus, bevor der nächste Alarm aktiviert wird. Diese Alarme unterbrechen sich nicht gegenseitig.
- Wenn während der Abarbeitung eines überfälligen Meldealarms die Taste [ON] gedrückt wird, bedingt dies eine Unterbrechung des Alarms ohne Bestätigung. Der Computer schaltet sich aus oder zeigt die Uhr an.
- Wenn keine Steueralarme oder bedingten Alarme überfällig sind und die Taste [ON] nicht während der Ausführung der übrigen Alarme gedrückt wird, schaltet sich der Computer nach Abarbeitung aller überfälligen Alarme aus oder zeigt die Uhr an.

Alarm-Bedingung

Die Alarm-Bedingung entsteht, wenn

- ein zukünftiger Alarm fällig wird, oder
- ein überfälliger Steueralarm oder bedingter Alarm (aus der Aus/Uhr-Bedingung) ein Programm startet oder eine Funktion ausführt.

Unter der Alarm-Bedingung aktiviert der Computer nur die gespeicherten, übergangenen überfälligen Alarme (in chronologischer Reihenfolge). Alle anderen überfälligen Alarme werden ignoriert und bleiben gespeichert. Wird ein zukünftiger Alarm fällig, während übergangene überfällige Alarme vorhanden sind, wird der zukünftige Alarm zu einem übergangenen überfälligen Alarm. Er wird aktiv, sobald alle früheren übergangenen überfälligen Alarme abgearbeitet worden sind.

Verhalten überfälliger Alarme unter der Alarm-Bedingung

Im folgenden finden Sie Definitionen der vier Betriebszustände (oder Modi), die das Verhalten des Computers auf das Eintreffen von Alarmen beeinflussen.

- Aus: der Computer ist ausgeschaltet.
- Zeit: die Uhr wird angezeigt.
- Tastenfeld: der Computer ist eingeschaltet, doch wird weder die Uhr angezeigt noch läuft ein Programm ab.
- Rechnend: ein Programm läuft ab.

Die folgende Tabelle faßt das Verhalten des Computers auf das Fälligwerden eines Alarms in jedem der oben beschriebenen Modi zusammen.

Verhalten überfälliger Alarme

Alarmtyp	Betriebszustand			
	Aus	Zeit	Tastenfeld	Rechnend
Bedingter Alarm (↑)	Programm wird gestartet	Programm wird gestartet	Tonfolge und blinkende Anzeige	2 Töne; Alarm wird überfällig
Steueralarm (↑↑)	Programm wird gestartet			Program wird als Unterprogramm des momentanen Programms gestartet
Meldealarm	Tonfolge und blinkende Anzeige.			

Bei Auftreten der Alarm-Bedingung wird die Reaktion des Computers auf die verschiedenen überfälligen Alarme durch den momentanen Betriebszustand und durch den jeweiligen Alarmtyp bestimmt.

Änderungen des Betriebszustands

Das in einem aktivierten Steueralarm oder bedingten Alarm spezifizierte Programm oder Funktion kann den aktuellen Betriebsmodus des Computers verändern:

- Wenn ein Steueralarm oder ein bedingter Alarm ein Programm startet, schaltet der Computer sofort in den Rechen-Modus. (Die Änderung des Betriebszustands erfolgt vor der Ausführung der ersten Programmanweisung.)
- Eine durch einen Steueralarm oder bedingten Alarm ausgeführte Funktion kann ebenfalls zu einem Moduswechsel führen. Falls beispielsweise die Uhr angezeigt wird, wenn ein Alarm, der die Druckerfunktion PRX (*print X*) ausführt, aktiv wird, schaltet sich der Computer vom Zeit-Modus in den Tastenfeld-Modus. Analog dazu wird der Computer bei Eintreffen eines Alarms, der die Funktion CLOCK ausführt, in den Zeit-Modus geschaltet, selbst wenn zum Zeitpunkt des Eintreffens des Alarms ein Programm abläuft (d.h. der Computer befindet sich im Rechen-Modus).

Unterbrechung eines überfälligen Alarms durch einen anderen überfälligen Alarm

- Ein von einem überfälligen Steueralarm oder bedingten Alarm gestartetes Programm wird von jedem nachfolgenden übergangenen überfälligen Alarm vor der Ausführung der ersten Programmanweisung vorübergehend unterbrochen.
- Ein Meldealarm oder die Ausführung einer durch einen überfälligen Steueralarm/bedingten Alarm gestarteten Funktion wird nicht durch übergangene überfällige Alarme unterbrochen.

Alarme und Unterprogrammebenen

Jeder Programm-Alarm, der einen vorhergehenden Programm-Alarm unterbricht, wirkt intern wie der Aufruf eines Unterprogramms. Wenn mehrere überfällige, jeweils ein Programm startende Steueralarme vorhanden sind (dies ist unwahrscheinlich, da jeder überfällige Steueralarm zu einem übergangenen überfälligen Alarm würde), werden mehrere Unterprogrammebenen benutzt.

Beispiel zur Abarbeitung überfälliger Alarme

Nehmen Sie an, der Computer ist ausgeschaltet, die momentane Zeit ist 9:59, und die folgenden vier Alarme sind programmiert:

ALPHA-Register	Zeit	Status
MELDUNG1	4:00	überfällig
↑ABC	5:00	übergangen überfällig
↑↑XYZ	6:00	übergangen überfällig
MELDUNG2	10:00	zukünftiger Alarm

Hinweis: Die in diesem Beispiel dargestellte Situation ist unwahrscheinlich, da übergangene überfällige Alarme in den meisten Anwendungen nicht vorkommen. Doch soll diese Kombination zusätzliche Aspekte des Alarmverhaltens illustrieren.

Um 10:00 tritt aufgrund des Alarms MELDUNG2 die Alarm-Bedingung ein. Da zu diesem Zeitpunkt zwei übergangene überfällige Alarme vorhanden sind, ergibt sich der folgende Ablauf:

1. Alarm ↑ABC (der älteste *übergangene* überfällige Alarm) startet das Programm ABC. (Dieser Alarm wird zuerst aktiviert, da der frühere Alarm MELDUNG1 kein übergangener überfälliger Alarm ist, und daher nicht aktiviert wird.)
2. Alarm ↑↑XYZ suspendiert sofort das Programm ABC und startet das Programm XYZ als Unterprogramm.

3. Alarm MELDUNG2 (nun ein übergangener überfälliger Alarm) suspendiert seinerseits sofort das Programm XYZ, läßt **MELDUNG2** in der Anzeige blinken und erzeugt eine Tonfolge, sofern er nicht über das Tastenfeld bestätigt wird.
4. Nach der Bestätigung von MELDUNG2 (oder der Beendigung seines Aktivierungszyklus) wird das Programm XYZ ausgeführt. Danach wird die Kontrolle an das Programm ABC übergeben (sofern das Programm XYZ nicht anhält, den Computer ausschaltet oder zuviele Unterprogrammebenen verwendet).
5. Programm ABC wird ausgeführt.

Wenn das Programm XYZ den Computer (durch Ausführung von [OFF]) ausschaltet, wird die Ausführung des Programms ABC nicht wieder aufgenommen. Da der Alarm ↑ABC bereits aktiviert wurde, wird er aus dem Speicher gelöscht.

Wenn der Alarm XYZ ein bedingter Alarm gewesen wäre, hätte er bei der Aktivierung nur zwei Töne erzeugt und wäre zu einem regulären überfälligen Alarm geworden (da der Alarm ↑ABC den Computer in den Rechen-Modus geschaltet hätte). Details finden Sie in der Tabelle «Verhalten überfälliger Alarme» und unter «Änderungen des Betriebszustands». Als generelle Richtlinie gilt: jedesmal wenn ein überfälliger Steueralarm oder bedingter Alarm bei Aktivierung ein Programm startet, wird *jeder* nachfolgende bedingte Alarm nur so aktiviert, daß zwei Töne den Alarm anzeigen und der Alarm selbst zu einem überfälligen Alarm wird.

Wenn der Computer zu Beginn nicht ausgeschaltet gewesen wäre, sondern sich stattdessen im Tastenfeld-Modus befunden hätte, wäre der Alarm ↑ABC wie ein Meldealarm mit der Anzeige **↑ABC** aktiviert worden. Das Programm XYZ wäre danach gestartet (aber nicht als Unterprogramm), von MELDUNG2 unterbrochen und schließlich ausgeführt worden.

Anhang C

Nullzeichen

Inhalt

Nullzeichen und das Alpha-Register 366
Behandlung von Nullzeichen 366

Nullzeichen und das Alpha-Register

Das Nullzeichen hat den Zeichencode 0 und wird durch das Symbol «¯» dargestellt.* Normalerweise erzeugt der Computer keine Nullzeichen. Sie können jedoch unter gewissen Umständen Nullzeichen in Alpha-Datenstrings einbetten.

Da das Nullzeichen gewöhnlich nicht erzeugt wird, benutzt der HP-41 das Nullzeichen intern zu speziellen Zwecken. Aus diesem Grund können Nullzeichen im Alpha-Register gelegentlich zu den in diesem Anhang beschriebenen, unerwarteten Anzeigen führen.

Behandlung von Nullzeichen

Bei der Behandlung von Nullzeichen ist es wichtig, zwischen dem Alpha-*Register* und der Alpha-*Anzeige* zu unterscheiden.

- Das Alpha-Register enthält immer 24 Zeichen; wenn es «leer» ist, enthält es 24 Nullzeichen. Wenn Sie von rechts Zeichen in das Alpha-Register eingeben, ersetzen diese die Nullzeichen. Alle führenden Nullzeichen (die Sie eingegeben haben können oder die bereits vorhanden waren) bleiben erhalten, werden jedoch vom Betriebssystem des Computers ignoriert.
- Die Alpha-Anzeige besteht aus den Zeichen im Alpha-Register *nach* den führenden Nullzeichen. Die Anzeige beginnt mit dem ersten (linksbündigen) Zeichen, das ungleich dem Nullzeichen ist, und enthält alle rechts darauf folgenden Zeichen, *einschließlich aller eingebetteten oder nachlaufenden Nullzeichen.*

Das Betriebssystem des HP-41 definiert den *Anfang eines Alpha-Strings immer als das erste Nicht-Nullzeichen* und ignoriert dabei alle führenden Nullzeichen. Zwischen Nicht-Nullzeichen eingebettete Nullzeichen bleiben erhalten. Wenn jedoch bei einer Rundumverschiebung des Alpha-Strings ein eingebettetes Nullzeichen zu einem führenden Nullzeichen wird, sind dieses Nullzeichen und alle direkt darauf folgenden Nullzeichen verloren.

* Das Nullzeichen sollte nicht mit der Meldung NULL verwechselt werden, die erscheint, wenn eine Funktion gelöscht wird. Ein angezeigtes Nullzeichen wird von den Druckern HP 82143A und HP 82162A als «✦» ausgedruckt (was den Zeichencodes 0 und 10 entspricht).

Anhängen von Zeichen. Wenn Sie ein Zeichen an das Alpha-Register anhängen (mittels [⊢]), unterscheidet sich die Anzeige vom aktuellen Inhalt des Alpha-Registers, *sofern das letzte Zeichen vor der* [⊢] *Operation ein Nullzeichen war.*

Wenn das letzte Zeichen im Alpha-Register ein Nullzeichen ist, behandelt der HP-41 *während der Eingabe der anzuhängenden Zeichen* das Alpha-Register als leer und zeigt nur die anzuhängenden Zeichen an. (Die Eingabeaufforderung (_) ist während der Eingabe der anzuhängenden Zeichen in der Anzeige vorhanden.) Im Alpha-Register bleibt jedoch der ursprüngliche String erhalten, an den im Verlauf der Operation der neu eingegebene String angehängt wird.

Sie können den vollständigen Inhalt des Alpha-Registers durch Drücken von [AVIEW] oder [ALPHA] [ALPHA] anzeigen. Denken Sie daran, daß führende Nullzeichen niemals angezeigt werden.

Löschen von Zeichen. Wenn das letzte Zeichen im Alpha-String ein Nullzeichen ist und Sie bei einer der Operationen [⊢] oder [ARCL] das rechts außen stehende Zeichen mittels [←] löschen, wird das *gesamte Alpha-Register gelöscht.* Der Grund hierfür ist, daß der Computer durch das Löschen des Nullzeichens in den Glauben versetzt wird, daß führende Nullzeichen einem String vorangehen, und daraus schließt, daß das Register leer ist – dies wiederum führt zu einer Löschung des gesamten Registers.

Alpha-Strings in Daten- oder Stackregistern. Wenn Sie einen Nullzeichen enthaltenen Alpha-String in einem Daten- oder Stackregister ablegen, werden bei der Anzeige (oder dem Ausdrucken) des Inhalts des betreffenden Registers keine Nullzeichen mit angezeigt (oder ausgedruckt). Wenn Sie jedoch den Inhalt dieses Registers zuerst mittels [ARCL] in das Alpha-Register zurückrufen und dann den Alpha-Registerinhalt anzeigen, erscheinen *alle* Zeichen des Alpha-Datenstrings in der Anzeige (natürlich mit Ausnahme von eventuell vorhandenen führenden Nullzeichen).

Beim Ausdrucken eines Alpha-Strings in einem Daten- oder Stackregister werden nur die links des ersten Nullzeichens (des ersten Nullzeichens von links) stehenden Zeichen gedruckt. Die verbleibenden Zeichen werden nicht gedruckt.

Ein in einen Alpha-String im X-Register eingebettetes Nullzeichen signalisiert für die Funktion [POSA] (*position in Alpha*) das Ende des mit dem Inhalt des Alpha-Registers zu vergleichenden Strings. (D.h. der Computer benutzt zum Vergleich nur den Teil des X-Register Strings, der sich links des ersten Nullzeichens befindet.)

Filenamen. Enthält das Alpha-Register einen Filenamen, so werden alle in dem Namen eventuell eingebetteten Nullzeichen von den ihn benutzenden Funktionen ignoriert.

Anhang D

Druckeroperationen

Inhalt

Papiervorschub 368
Steuerung von Programmausführung und Anzeige mittels Flag 21 und 55 368
Uhrzeit und Datum auf Programmlisten 369

Papiervorschub

Die programmierbare Funktion [ADV] (*advance*) bewirkt eine Zeilenschaltung auf einem angeschlossenen Drucker. Wenn kein Drucker an den HP-41 angeschlossen ist, hat [ADV] keinerlei Wirkung. (Das gleiche gilt, wenn ein Drucker angeschlossen aber ausgeschaltet ist, oder wenn Flag 21 gelöscht ist.)

Steuerung von Programmausführung und Anzeige mittels Flag 21 und 55

Flag 21 (*Druckersteuerung*) und Flag 55 (*Drucker vorhanden*) werden automatisch bei jedem Einschalten des Computers gesetzt oder gelöscht. Normalerweise sind sie danach beide gesetzt oder gelöscht: gesetzt, wenn ein Drucker angeschlossen ist; gelöscht, wenn kein Drucker vorhanden ist.

Mit Hilfe der Funktionen [VIEW] oder [AVIEW] und durch Manipulation von Flag 21 (der Status dieses Flags kann vom Benutzer verändert werden, der Status von Flag 55 dagegen nicht) können Sie die Anzeige von Meldungen und Ergebnissen während einer Programmausführung steuern; d.h. ob die Ausführung zur Anzeige des Resultats angehalten werden soll, oder ob das Ergebnis nur angezeigt und danach die Ausführung automatisch fortgesetzt werden soll.

Der Status von Flag 21 und 55 bestimmt, wie [VIEW] und [AVIEW] auf ein ablaufendes Programm wirken. Im Normalbetrieb haben beide Flags den gleichen Status, und es gilt:

- Wenn kein Drucker vorhanden ist, bewirken [VIEW] oder [AVIEW] die Anzeige des spezifizierten Registers bzw. des Alpha-Registers. Diese Anzeige bleibt solange erhalten, bis ein nachfolgender Anzeigebefehl neue Daten in die Anzeige schiebt. [VIEW] und [AVIEW] unterbrechen nicht die Programmausführung.
- Wenn ein Drucker vorhanden *und eingeschaltet ist*, reagiert der HP-41 in der oben beschriebenen Weise; zusätzlich werden die angezeigten Daten ausgedruckt.

Es gibt zwei unterschiedliche Anwendungen von [VIEW] und [AVIEW] in einem Programm. 1) Eine Meldung soll Ihnen mitteilen, was das Programm gerade tut – beispielsweise welches Unterprogramm momentan gerade ausgeführt wird. Es besteht jedoch keine Notwendigkeit, derartige Meldungen permanent zu erhalten. 2) Andere Meldungen liefern Ihnen die Resultate Ihres Programms; diese Informationen wollen Sie wahrscheinlich aufzeichnen. Wenn Sie nicht über einen Drucker verfügen, sollte in diesen Fällen die Programmausführung bei der Anzeige der Resultate angehalten werden, so daß Sie die jeweiligen Werte aufschreiben können.

Beachten Sie, daß im Standardbetrieb auch bei nicht vorhandenem Drucker die Programmausführung nicht angehalten wird (um Ihnen eine Abschrift der Daten zu ermöglichen), und daß bei Vorhandensein eines Druckers *sämtliche* angezeigten Daten oder Meldungen auf dem Drucker protokolliert werden. Durch Setzen oder Löschen von Flag 21 vor der Ausführung von [VIEW] oder [AVIEW] können Sie steuern, ob das Programm unabhängig von der Existenz eines Druckers zur Anzeige von Daten und Meldungen anhält oder nicht.

- **Löschen Sie Flag 21 zur Anzeige von Meldungen ohne gleichzeitige Protokollierung.** Wenn Flag 21 bei der Ausführung von [VIEW] oder [AVIEW] gelöscht und kein Drucker vorhanden *bzw. nicht eingeschaltet ist*, werden Meldungen und Ergebnisse ohne Unterbrechung der Programmausführung angezeigt. Dies ist die erste Form des oben beschriebenen Standardbetriebs.

  Wenn ein Drucker angeschlossen und eingeschaltet ist, wird die Meldung angezeigt aber nicht ausgedruckt und die Programmausführung nicht unterbrochen.

- **Setzen Sie Flag 21 zur Protokollierung von Ergebnissen** – unabhängig davon, ob ein Drucker vorhanden ist oder nicht. Wenn Flag 21 bei der Ausführung von [VIEW] oder [AVIEW] gesetzt und kein Drucker vorhanden *oder ausgeschaltet* ist, wird die Programmausführung angehalten, so daß Sie das angezeigte Ergebnis abschreiben können. Drücken Sie anschließend [R/S], um die Programmausführung fortzusetzen.

  Wenn ein Drucker vorhanden ist, wird das Ergebnis ausgedruckt und die Programmausführung nicht unterbrochen. Dies ist die zweite Form des oben beschriebenen Standardbetriebs.

Auf diese Weise können Sie bei angeschlossenem Drucker steuern, welche Anzeigen Sie ausgedruckt haben wollen und welche nicht. Bei nicht angeschlossenem Drucker können Sie entscheiden, bei welchen der angezeigten Ergebnisse und Meldungen Sie die Programmausführung unterbrechen wollen.

Uhrzeit und Datum auf Programmlisten

Wenn Sie ein Programm auf einem HP-IL Drucker (etwa dem HP 82162A) mit Hilfe der Funktionen [PRP] oder [LIST] des HP-IL Moduls HP 82160A auslisten, erscheinen Uhrzeit und Datum in der Anzeige und werden gleichzeitig als Kopf der Programmliste gedruckt. (Uhrzeit und Datum befinden sich zwar in der Anzeige aber nicht im X-Register; Sie können durch Drücken von [←] wieder den Inhalt des X-Registers anzeigen.)

Anhang E

Erweiterte Speicher-Module

Inhalt

Verwendung von Erweiterten Speicher-Modulen 370
Zulässige Konfigurationen 370
Einsetzen von Erweiterten Speicher-Modulen 371
Entfernen von Erweiterten Speicher-Modulen 371
Aufbau des erweiterten Speichers 372

Verwendung von Erweiterten Speicher-Modulen

VORSICHT

Schalten Sie Ihren HP-41CX vor dem Einsetzen oder Entfernen von Modulen immer aus. Die Nichtbeachtung dieser Vorsichtsmaßnahme kann zu Beschädigungen des HP-41CX oder zu Störungen des Systembetriebs führen.

Jedes Erweiterte Speicher-Modul HP 82181A trägt die Beschriftung «X MEMORY» auf dem Gehäuse.

Zulässige Konfigurationen

Solange Sie Ihren HP-41CX nur um ein Erweitertes Speicher-Modul ergänzen wollen, können Sie das Modul in jeden der vier Einschubschächte des Computers einsetzen. Befolgen Sie dabei die im folgenden gegebenen Installationsanweisungen.

Wenn Sie ein zweites Erweitertes Speicher-Modul hinzufügen wollen, müssen die beiden Module in einer der unten gezeigten Anordnungen eingesetzt werden. Sie dürfen zwei Erweiterte Speicher-Module niemals übereinander installieren.

Zulässige Belegungen der Einschubbuchsen bei Verwendung von zwei Erweiterten Speicher-Modulen HP 82181A

X MEMORY	X MEMORY

X MEMORY	X MEMORY

X MEMORY	
	X MEMORY

	X MEMORY
X MEMORY	

Einsetzen von Erweiterten Speicher-Modulen

Beim Einsetzen eines Erweiterten Speicher-Moduls ist wie folgt vorzugehen:

1. **Schalten Sie den HP-41CX aus!**
2. Halten Sie den HP-41CX mit dem Tastenfeld nach oben.
3. Entfernen Sie die Schutzkappe von einem der Einschubschächte.
4. Halten Sie das Modul so, daß die Beschriftung «X MEMORY» zu Ihnen zeigt.
5. Drücken Sie das Modul vorsichtig in den Einschubschacht.

Das Modul ist richtig eingesetzt, sobald Sie es einrasten fühlen.

Entfernen von Erweiterten Speicher-Modulen

Beim Entfernen eines Erweiterten Speicher-Moduls ist wie folgt vorzugehen:

1. **Schalten Sie den HP-41CX aus!**
2. Ziehen Sie mit dem Fingernagel das ausziehbare Griffstück des Moduls etwas vor.
3. Fassen Sie das Modul am Griffstück und ziehen Sie es aus dem Einschubschacht.
4. Verschließen Sie den offenen Einschubschacht mit der Schutzkappe.

Die in einem Erweiterten Speicher-Modul abgelegte Information ist verloren, sobald Sie das Modul aus dem HP-41CX entfernen. **Sie können alle in zwei Erweiterten Speicher-Modulen gespeicherten Daten verlieren,** selbst wenn Sie nur ein Modul entfernen. Der Datenverlust kann minimiert werden, wenn Sie die hier beschriebene Reihenfolge zum Entfernen der Module beachten. In Abhängigkeit von der ursprünglichen Anordnung der Module ist wie folgt vorzugehen:

- Wenn die Module zu verschiedenen Zeitpunkten eingesteckt wurden, so sind sie in der umgekehrten Reihenfolge des Einsteckens zu entfernen.
- Wenn beide Module zum gleichen Zeitpunkt eingesetzt wurden, ist zunächst das Modul in Einschubschacht 2 oder 4 und danach das Modul in Einschubschacht 1 oder 3 herauszunehmen. (Die Numerierung der Einschubschächte ist auf der Rückseite des HP-41CX abgebildet.)

Sie können berechnen, welche Files beim Entfernen eines Erweiterten Speicher-Moduls verlorengehen würden. Gegebenenfalls können Sie dann solange alte, weniger wichtige Files löschen, bis die folgende Prozedur zeigt, daß Ihre wichtigen Files auch nach einem Entfernen eines Moduls erhalten bleiben würden:

- Bestimmen Sie mittels EMDIR die Größe jedes Files (siehe Seite 206).
- Addieren Sie für jeden File zwei zusätzliche Register für den Header.
- Addieren Sie die Gesamtanzahl der Register des ersten Files zu der Gesamtanzahl der Register des zweiten Files, um die Position des letzten Registers im zweiten File zu bestimmen.
- Addieren Sie für jeden nachfolgenden File die Gesamtanzahl der Register zu dieser Summe; der resultierende Wert ist dann jeweils die Position des letzten Registers in dem jeweiligen File.
- Sobald das letzte Register eines Files hinter Register 126 liegt, geht dieser File beim Entfernen des ersten Erweiterten Speicher-Moduls verloren.
- Sobald das letzte Register eines Files hinter Register 364 liegt, geht dieser File beim Entfernen des zweiten Erweiterten Speicher-Moduls verloren.

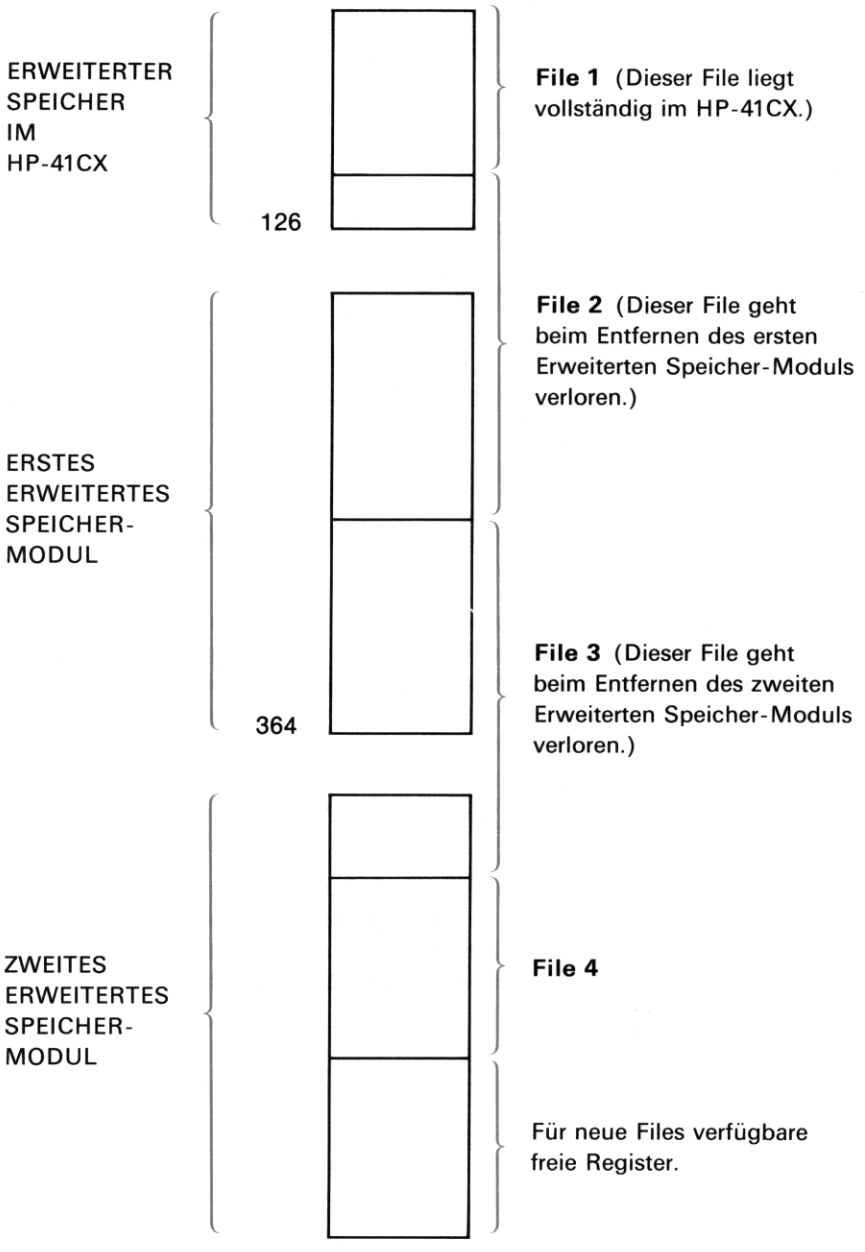
Aufbau des erweiterten Speichers

Im folgenden soll der Aufbau des erweiterten Speichers in einem HP-41CX mit zwei installierten Erweiterten Speicher-Modulen anhand einer Illustration verdeutlicht werden. Dabei wird unterstellt, daß der erweiterte Speicher momentan 4 Files enthält. Der obere Block in der Illustration stellt die 127 bereits im HP-41CX vorhandenen erweiterten Speicherregister dar. File 1 liegt also vollständig im eingebauten erweiterten Speicher des HP-41CX und bleibt daher erhalten, selbst wenn beide Erweiterte Speicher-Module entfernt werden.

Der mittlere Block stellt die 238 Register des ersten Erweiterten Speicher-Moduls dar; d.h. des zuerst eingesetzten und benutzten Moduls bzw. des Moduls in Einschubschacht 1 oder 3. Bei Entfernen dieses Moduls gehen die Files 2, 3 und 4 verloren.

Der untere Block stellt die 238 Register des zweiten Erweiterten Speicher-Moduls dar; d.h. des zuletzt eingesetzten und benutzten Moduls bzw. des Moduls in Einschubschacht 2 oder 4. Beim Entfernen dieses Moduls gehen die Files 3 und 4 verloren.

Aufbau des erweiterten Speichers



Anhang F

Zeitspezifikationen

Inhalt

Der Kompensationsfaktor 374
Zeitkorrektur mittels Kompensationsfaktor ([CORRECT]) 375
Zurückrufen, Einstellen und Löschen des Kompensationsfaktors 376
Berechnung des Kompensationsfaktors 377
Spezifikation von Präzision und Ganggenauigkeit 377
Präzision 377
Ganggenauigkeit 378
Stoppuhrpräzision 378

Der Kompensationsfaktor

Die programmierbaren Zeitkorrekturfunktionen ermöglichen das Nachstellen der momentanen Uhrzeit sowie die Justierung und Überwachung des Kompensationsfaktors.

Wie bei den meisten Instrumenten zur Zeitmessung kann die Ganggenauigkeit des HP-41 Zeitgebers durch Schwankungen in der Versorgungsspannung, der Temperatur sowie durch gewisse Fertigungstoleranzen beeinflußt werden. Solange die Effekte dieser Schwankungen klein und *die sie verursachenden Faktoren relativ konstant sind*, können Sie die resultierenden Zeitveränderungen mit Hilfe des Kompensationsfaktors ausgleichen. Der in den Funktionen [CORRECT], [RCLAF] und [SETAF] verwendete Kompensationsfaktor dient zum *langfristigen* Ausgleich von Variationen in den Idealbedingungen und ist zur Kompensation kurzfristiger Schwankungen in den Umgebungsbedingungen (wie etwa Temperaturen unter dem Gefrierpunkt über mehrere Tage) ungeeignet. In diesen Fällen sollten Sie die Zeit einfach mittels [T+X] neu einstellen.

Der Zeitgeber im HP-41CX ist im wesentlichen identisch mit dem Zeitgeber in einer Quarzkristalluhr. Wenn Sie die Ganggenauigkeit des HP-41CX Zeitgebers mit dem Kompensationsfaktor nachjustieren, entspricht die resultierende Genauigkeit derjenigen einer Quarzkristalluhr. Die unjustierte Genauigkeit des HP-41CX Zeitgebers ist nicht ganz so gut wie die einer Quarzkristalluhr, aber immer noch besser als die einer Uhr mit mechanischem Werk.

Der Kompensationsfaktor ist das in Sekunden gemessene Zeitintervall, nach dem ein Puls (von ca. $9{,}8 \times 10^{-5}$ Sekunden Dauer) zur 10 240 Hz Zeitbasis des Zeitgebers addiert bzw. subtrahiert wird. Die rechtsstehende Tabelle zeigt die Grenzwerte und die Auswirkung des Kompensationsfaktors.

Ein Kompensationsfaktor von –10,5 würde alle 10,5 Sekunden einen Puls von der Zeitbasis subtrahieren. Umgekehrt würde ein Kompensationsfaktor von 0,1 alle 0,1 Sekunden einen Puls zu der Zeitbasis addieren.

Der Kompensationsfaktor

Kompensationsfaktor	Auswirkung	
± 99.9 s ⋮ ± 0.1 s	Addiert/subtrahiert einen Puls alle Sekunden.	↓ Wachsende Häufigkeit der Korrekturen
0.0	Voreinstellung	Keine Korrektur

Sie können den geeigneten Kompensationsfaktor entweder automatisch mittels der [CORRECT] Funktion oder durch manuelle Berechnung ermitteln.

Weitere Informationen bezüglich der Ganggenauigkeit des HP-41CX Zeitgebers finden Sie unter «Spezifikationen von Präzision und Ganggenauigkeit».

Zeitkorrektur mittels Kompensationsfaktor ([CORRECT])

Die Funktion [CORRECT] (*correct the time*) stellt die von Ihnen spezifizierte Zeit ein und justiert dabei automatisch den Kompensationsfaktor. Wenn Sie eine Uhrzeit (***HH.MMSShh***) in das X-Register laden, beinhaltet die Ausführung von [CORRECT] die folgenden Operationen:

- Die Uhr wird auf die gleiche Weise wie bei [SETIME] auf die spezifizierte Zeit eingestellt.
- Der Kompensationsfaktor wird automatisch justiert. Dazu wird ein interner Algorithmus verwendet, der auf der Drift* und der Zeitspanne seit der letzten Ausführung von [SETIME], [SETDATE], [SETAF] (*set accuracy factor*) oder [CORRECT] basiert. Der Zeitgeber verwendet dann den neujustierten Kompensationsfaktor zur automatischen und fortlaufenden Aktualisierung der Zeitbasis.

Wenn Sie [CORRECT] manuell ausführen, kann die Genauigkeit der Zeiteinstellung in Abhängigkeit von Ihrem Tastenanschlag variieren. Die Einstellung wird vorgenommen, wenn die zur Ausführung von [CORRECT] benutzte Taste *losgelassen* wird.** Die Zeitspanne zwischen der letzten Ausführung von [SETIME], [SETDATE], [SETAF] oder [CORRECT] und der der darauf folgenden Ausführung von [CORRECT] muß groß genug sein, um den Reaktionsfehler beim Tastenanschlag vernachlässigbar klein zu machen. In der Regel sollte diese Zeitspanne mindestens 30 Stunden betragen. Verlängerungen der Zeitspanne zwischen den [CORRECT] Ausführungen erhöhen die Wahrscheinlichkeit, einen zuverlässigeren Kompensationsfaktor zu erhalten.***

---

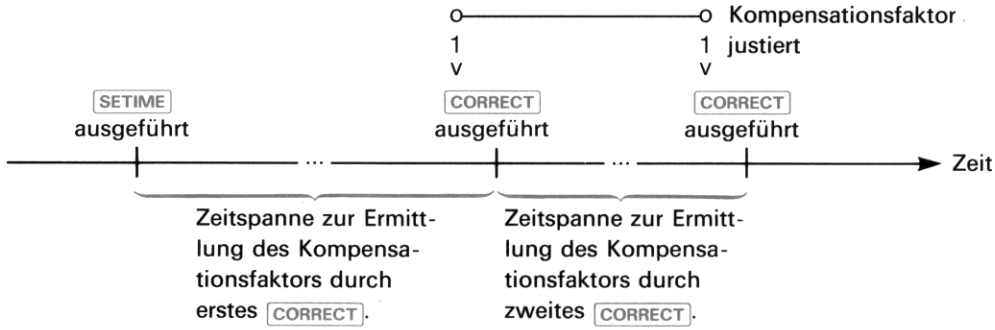
* Die Drift ist die durch Schwankungen in der Versorgungsspannung, der Temperatur sowie durch Materialtoleranzen verursachte Abweichung von der korrekten Zeit. Der Zeitgeber benutzt als Drift die Differenz zwischen der momentanen Uhrzeit bei der Ausführung von [CORRECT] und der im X-Register spezifizierten Uhrzeit.

** Die maximale Genauigkeit eines Tastenanschlags liegt in den meisten Fällen im Bereich von ± 0,1 Sekunden. Sie können diese Anschlagsungenauigkeit verringern, indem Sie [CORRECT] einer Taste zuweisen und nicht die Sequenz [XEQ] [ALPHA] CORRECT [ALPHA] verwenden, da der Computer weniger Zeit zur Lokalisierung und Ausführung von Tasten zugewiesenen Funktionen benötigt.

*** Je länger Sie mit der Ausführung von [CORRECT] warten, um so geringer wird der aus dem Tastenanschlag resultierende Fehler im Verhältnis zum Gesamtfehler aus allen anderen Fehlerquellen sein. Für die meisten Anwendungen ist eine Woche ein realistischer Wert.

Hinweis: Die [CORRECT] Funktionen benutzt die berechnete Drift zur Bestimmung des Kompensationsfaktors. Wenn Sie die langfristige *Genauigkeit* des Zeitgebers mit [CORRECT] verbessern wollen, sollten Sie daher *nicht* [T+X] zur Kompensation von Zeitfehlern verwenden, die durch die normale Drift verursacht werden. Derartige Korrekturen würden von [CORRECT] nicht berücksichtigt werden. Aus diesem Grund führt die Verwendung von [T+X] zur Korrektur von Driftfehlern zu einem weniger zuverlässigen Kompensationsfaktor.

Die durch [CORRECT] bedingte Justierung des Kompensationsfaktors beruht zum Teil auf der Differenz zwischen der aktuellen Zeiteinstellung und der neuen Zeiteinstellung in dem Moment der Ausführung von [CORRECT]. Wenn die Zeit nicht zuvor mit [SETIME] eingestellt wurde, kann die Ausführung von [CORRECT] zu einem ungünstigen Kompensationsfaktor führen. Wenn Sie jedoch die Uhrzeit irgendwann einmal mit [SETIME] eingestellt haben, können Sie [CORRECT] so oft verwenden, wie es Ihnen praktisch erscheint.



Denken Sie daran, daß eine längere Zeitspanne zwischen der Ausführung von [SETIME] oder [CORRECT] und der darauf folgenden Ausführung von [CORRECT] die Zuverlässigkeit des Kompensationsfaktors erhöht.

Zurückrufen, Einstellen und Löschen des Kompensationsfaktors

Die Funktion [RCLAF] (*recall accuracy factor*) lädt den momentanen Kompensationsfaktor in das X-Register. Der Stack wird dabei wie bei einem gewöhnlichen Rückruf einer Zahl aus einem Datenregister angehoben.

Die Funktion [SETAF] (*set accuracy factor*) verwendet den Wert x im X-Register zur Einstellung des Kompensationsfaktors. Der Kompensationsfaktor wird dabei wie folgt entweder auf eine Zehntelsekunde gerundet oder auf Null gesetzt:

- Der Kompensationsfaktor wird auf 0.0 gesetzt, falls $x = 0$ oder $x \geqslant |99{,}95|$ ist.
- Der Kompensationsfaktor wird auf 0.1 gesetzt, falls $0 < x \leqslant 0{,}1$ ($x \neq 0$) gilt und auf -0.1 falls $-0.1 \leqslant x < 0$ gilt.
- Wenn der Wert x im X-Register im Bereich $0.1 < x < 99.949$ oder $-99.95 < x < -0.1$ liegt, wird der Kompensationsfaktor auf die nächste Zehntelsekunde gerundet.

Wenn Sie [SETAF] ausführen, beginnt der Zeitgeber automatisch und fortlaufend mit der Änderung der Zeitbasis entsprechend dem spezifizierten Kompensationsfaktor.

Sie können den Kompensationsfaktor löschen, indem Sie den Wert 0 in das X-Register laden und anschließend [SETAF] ausführen.

Wenn Sie mit einer Unterbrechung in der Stromversorgung Ihres HP-41 rechnen müssen (wie beispielsweise bei einem Austausch der Batterien), ist es sinnvoll, den Kompensationsfaktor zurückzurufen, aufzuschreiben und dann diesen Wert bei wieder gesicherter Stromversorgung als neuen Kompensationsfaktor einzustellen.

Berechnung des Kompensationsfaktors

[CORRECT] bietet ein bequemes Verfahren zur Korrektur der Zeitbasis (durch automatische Berechnung des mittleren aufgelaufenen Fehlers). Wenn Sie jedoch schon nach relativ kurzer Zeit (etwa in einem 36-Stunden-Intervall) einen Kompensationsfaktor einrichten wollen, hat der bei der Ausführung von [CORRECT] auftretende Anschlagfehler einen stärkeren Effekt, als wenn Sie [CORRECT] über größere Zeitintervalle benutzen. Bei kürzeren Zeitintervallen erhalten Sie häufig einen zuverlässigeren Kompensationsfaktor, wenn Sie diesen selbst berechnen und mittels [SETAF] einstellen. Außerdem wird ein mit [CORRECT] ermittelter Kompensationsfaktor nach einer manuellen Änderung der Drift durch [T+X] wahrscheinlich höchst unzuverlässig sein. Nach einer Änderung der Drift mit [T+X] besteht die beste Methode zur Ermittlung eines zuverlässigen Kompensationsfaktors in einer manuellen Berechnung dieses Faktors.

Zur Berechnung des Kompensationsfaktors können Sie die folgende Formel benutzen:

$$AF = \frac{1}{\frac{1}{IAF} - \frac{10240}{86400} ERR_{spd}}$$

wobei
IAF = ursprünglicher Kompensationsfaktor
(Falls IAF Null ist, ersetzen Sie $1/IAF$ durch «0».)
ERR_{spd} = aktuelle Abweichung in Sekunden pro Tag
(Ein «langsamer» Zeitgeber hat eine negative Abweichung, ein «schneller» Zeitgeber eine positive Abweichung.)
10240 = interne Zeitbasis (Anzahl der Pulse pro Sekunde)
86400 = Anzahl der Sekunden pro Tag

Sie sollten den Kompensationsfaktor auf eine Dezimalstelle genau berechnen und ihn anschließend mittels [SETAF] einstellen.

Spezifikation von Präzision und Ganggenauigkeit

Präzision

Die Zeiteinstellung über das Tastenfeld kann auf bis zu 0.1 Sekunden exakt sein, ist jedoch stark von der menschlichen Reaktion abhängig. Sie können die Genauigkeit der *momentanen* Uhrzeit mittels der [T+X] Funktion auf bis zu 0.01 Sekunden erhöhen.

Ganggenauigkeit

Die quarzstabilisierte Zeitbasis sorgt für eine hohe Ganggenauigkeit des Zeitgebers. Wie bei jedem quarzgesteuerten Zeitmeßinstrument ist die eigentliche Stabilität zu jedem Zeitpunkt eine Funktion der Variationen in Betriebstemperatur und Versorgungsspannung. Wenn Sie Ihren HP-41CX unter gleichmäßigen Bedingungen betreiben, kann die Gesamtabweichung durch geeignete Anpassung des Kompensationsfaktors vernachlässigbar klein gehalten werden. Die Gesamtgenauigkeit des Zeitgebers beträgt bei einer Temperatur von 25° C ± 3,02 Sekunden pro Tag (± 35 ppm) und überschreitet diesen Wert (bei anderen Temperaturen) um nicht mehr als ± 1,30 Sekunden pro Tag (± 15 ppm).

Stoppuhrpräzision

Die Stoppuhrpräzision und die Präzision der Differenz zwischen zwei Zeitnahmen beträgt ± 0,01 Sekunden. Zur Erzielung einer maximalen Genauigkeit sollten Zeiten in Abständen von nicht weniger als 0,08 Sekunden gestoppt werden, da ansonsten Fehler aufgrund von Verzögerungen in der internen Verarbeitung auftreten können. Schnell aufeinanderfolgendes Drücken von Tasten des Stoppuhr-Tastenfelds kann vorübergehend zur Unterdrückung von Teilen oder der gesamten Stoppuhr-Anzeige führen; dies hat jedoch keinen Einfluß auf die Arbeitsweise der Stoppuhr selbst.

Anhang G

Batterien, Gewährleistung und Serviceinformation

Inhalt

Die Einschubschächte 381
Batterien und Stromverbrauch 381
Stromverbrauch 381
Stromverbrauch von Peripheriegeräten 382
Auswirkungen von Speicherlöschung, Stromunterbrechung und Spannungsabfall 382
Anzeige abfallender Batteriespannung 383
Einsetzen neuer Batterien 383
Funktionsprüfung 385
Gewährleistung 386
Änderungsverpflichtung 387
Gewährleistungsinformationen 387
Service 388
Service-Zentrale in den Vereinigten Staaten 388
Serviceniederlassungen in Europa 388
Internationale Serviceinformation 389
Reparaturkosten 389
Gewährleistung auf Serviceleistungen 389
Versandanweisungen 390
Sonstiges 390
Benutzerberatung 390
Temperaturspezifikationen 391

Die Einschubschächte

Wenn einer der Einschubschächte (Ports) des Computers kein Einschubmodul enthält, so sollten Sie diesen Schacht immer mit einer der Schutzkappen verschließen. Die Schutzkappen verhindern, daß die Kontakte in den Einschubschächten verschmutzen.

VORSICHT

Stecken Sie keine Finger oder anderen Objekte (mit Ausnahme von HP Modulen oder anderen Einsteck-Zubehörteilen) in die Einschubschächte. Die Nichtbeachtung dieser Vorsichtsmaßnahme kann zu einem Löschen des Permanentspeichers und zu Beschädigungen an Einschubschacht und Computer führen.

Batterien und Stromverbrauch

Die Stromversorgung des HP-41 erfolgt über vier Batterien. In Abhängigkeit von der jeweiligen Auslastung kann der HP-41 mit einem Satz Alkalibatterien bis zu sechs Monate betrieben werden. Der Computer enthält bei der Auslieferung Typ-N-Alkalizellen; Sie können stattdessen aber auch einen wiederaufladbaren Batteriesatz (Nickel-Cadmium-Zellen) verwenden.

Die durch die Batterien ermöglichte Gesamtbetriebsdauer hängt wesentlich davon ab, welche Operationen Sie ausführen und wie oft Sie Peripheriegeräte betreiben. Ohne die Verwendung von Peripheriegeräten liefert ein Satz von vier frischen Alkalibatterien eine reine Programmrechenzeit von 45 bis 70 Stunden (hierbei sei angemerkt, daß die Ausführung von Programmen diejenige Form des Rechenbetriebs mit dem höchsten Stromverbrauch ist). Wenn nur die Anzeige eingeschaltet ist und keine Berechnungen ablaufen, ist der Stromverbrauch minimal.

Die aktuelle Lebensdauer der Batterien hängt davon ab, wie oft Sie den HP-41 und Peripheriegeräte benutzen, ob Sie ihn mehr zur Ausführung von Programmen oder mehr für manuelle Berechnungen verwenden und mit welchen Funktionen Sie arbeiten. Neben dem Betrieb von Peripheriegeräten bedingen die folgenden Operationen einen erhöhten Stromverbrauch: Ausführung von Programmen, Anzeige der Uhr, Anzeige/Benutzung der Stoppuhr und Verwendung von Katalogen. Die Kataloge 4, 5 und 6 nehmen genauso viel Strom wie ein ablaufendes Programm auf, selbst wenn sie angehalten werden (aus diesem Grund verursachen sie ein schnelleres automatisches Abschalten des Computers).

Bei ausgeschaltetem Computer bleibt der Inhalt des Permanentspeichers über einen Zeitraum erhalten, der dem der Haltbarkeit der Batterien außerhalb des Computers entspricht – ungefähr 1½ Jahre bei Alkalibatterien.

Stromverbrauch

Der eingebaute Zeitgeber bezieht die ständig benötigte Versorgungsspannung über die Batterien des Computers. Der aktuelle Stromverbrauch hängt von der jeweiligen Betriebsart des HP-41 ab. Insgesamt kann der Stromverbrauch der einzelnen Zeitfunktionen in drei Klassen eingeteilt werden:

- Hohe Leistungsaufnahme (5 bis 20 mA). Dieser Stromverbrauch wird bei aktiver [ALMCAT] Funktion und bei aktivem Stoppuhr-Tastenfeld (Stoppuhr angezeigt) erreicht und entspricht dem Stromverbrauch eines ablaufenden Programms, einer Katalogoperation oder dem des Text-Editors.

- Mittlere Leistungsaufnahme (0,5 bis 2,0 mA). Dies ist der Stromverbrauch bei angezeigter Uhr ([CLKTD]) und entspricht dem des Computers bei Betriebsbereitschaft (Anzeige eingeschaltet, auf Eingaben wartend). Zur Anzeige von Uhrzeit und Datum ([CLKTD]) wird nur die Hälfte der Energie wie zur Anzeige der reinen Uhrzeit ([CLKT]) benötigt. Der Grund hierfür liegt in der häufigeren Aktualisierung der [CLKT] Anzeige.
- Geringe Leistungsaufnahme (0,01 bis 0,5 mA). Dies ist die Leistungsaufnahme bei ausgeschaltetem Computer. Der Präzisionsoszillator des Zeitgebers schwingt ständig, um die Uhr und die gegebenenfalls laufende Stoppuhr zu betreiben.

Solange der Computer eingeschaltet ist, besteht die typische Auslastung aus einer Mischung der Betriebszustände «bereit» und «rechnend». Die aktuelle Lebensdauer der Batterien hängt daher sehr stark davon ab, wie lange der Computer in jeder der drei Betriebsarten benutzt wird.

Ein frisch aufgeladener Batteriesatz HP 82120A hat eine Kapazität von 65 mAH (Milliampere-Stunden), während ein frischer Satz Alkalibatterien ungefähr 500 mAH liefert.

Stromverbrauch von Peripheriegeräten

Wenn Sie Peripheriegeräte verwenden, die über die Batterien des HP-41 mit Strom versorgt werden (wie etwa der Magnetkartenleser oder der optische Lesestift), reduziert sich die Lebensdauer der Batterien erheblich. Bei häufigem Gebrauch von Peripheriegeräten sollten Sie den HP-41 über den wiederaufladbaren Batteriensatz HP 82120A betreiben.

Auswirkungen von Speicherlöschung, Stromunterbrechung und Spannungsabfall

Speicherlöschung. Von einem Zurücksetzen des Permanentspeichers ([←]/[ON]) bleiben *unberührt*:

- die momentane Uhrzeit und die momentanen Formateinstellungen ([CLK12] bzw. [CLK24] und [CLKT] bzw. [CLKTD]);
- die Stoppuhr.

Ein Zurücksetzen des Permanentspeichers hat die folgenden Auswirkungen:

- Die Inhalte von Hauptspeicher und erweitertem Speicher werden gelöscht.
- Sämtliche Alarme werden gelöscht.
- Sämtliche Tastenzuordnungen des User-Tastenfelds werden gelöscht.
- Sämtliche Flags werden auf ihren voreingestellten Status gesetzt (siehe «Zusammenfassung der Flagzustände» in Abschnitt 19).
- Der Datenspeicherung werden 100 Register zugewiesen.

Siehe auch «Der Permanentspeicher» in Abschnitt 1.

Vorübergehende Stromunterbrechung. Eine Stromunterbrechung (beispielsweise beim Austauschen der Batterien) von ausreichender Dauer kann eine *Totallöschung* des Computers bewirken, von der Permanentspeicher *sowie* Uhrzeit, Datum und Alarme betroffen werden. Uhrzeit und Datum werden in diesem Fall auf den 1. Januar 1900 12:00 zurückgesetzt. Es wird daher empfohlen, nach jeder Stromunterbrechung die aktuelle Uhrzeit und das Datum zu überprüfen. Wenn Uhrzeit und Datum korrekt sind, ist die Zeitinformation im Computer durch den Stromausfall nicht verändert worden. Wenn Uhrzeit und Datum *nicht* korrekt erscheinen, sollten Sie wie folgt vorgehen:

1. Stellen Sie Uhrzeit und Datum (mittels [SETIME] und [SETDATE]) neu ein.
2. Setzen Sie die gewünschten Anzeigeformate für Uhrzeit und Datum ([CLK12] oder [CLK24], [CLKT] oder [CLKTD], [MDY] oder [DMY]).
3. Initialisieren Sie die Stoppuhr und halten Sie sie gegebenenfalls an ([SETSW] [STOPSW]).

Spannungsabfall. Wenn die Batteriespannung nicht zum Betrieb der Zeitanzeige ausreicht, bewirkt die Ausführung von [CLOCK] oder■ [ON] ein Ausschalten des Computers (Uhrzeit und Datum bleiben jedoch intern erhalten). In den meisten Fällen tritt dieses Verhalten jedoch erst auf, wenn der Spannungsabfall bereits durch das Aufleuchten des **BAT**-Indikators signalisiert wird.

Anzeige abfallender Batteriespannung

Wenn die Batteriespannung unter einen bestimmten Wert absinkt, erscheint der **BAT**-Indikator in der Anzeige. Sie können die Lebensdauer der Batterien beträchtlich verlängern, indem Sie eventuell vorhandene Peripheriegeräte ausstecken. (Vor dem Ausstecken sowohl den HP-41 als auch das Peripheriegerät ausschalten!)

Bei Verwendung von Alkalibatterien (und ohne angeschlossene Peripheriegeräte) gilt:

- Der Computer kann nach dem ersten Aufleuchten des **BAT**-Indikators noch ca. 2 bis 7 Stunden zur reinen Programmausführung benutzt werden.*
- Wenn der Computer nach dem ersten Aufleuchten des **BAT**-Indikators ausgeschaltet wird, bleibt der Inhalt des Permanentspeichers noch für ungefähr einen Monat erhalten.

Einsetzen neuer Batterien

Die mit Ihrem HP-41 gelieferten Batterien und die im folgenden gelisteten Ersatzbatterien können *nicht* wieder aufgeladen werden.

WARNUNG

Versuchen Sie nicht, die Batterien aufzuladen; werfen Sie sie nicht in offenes Feuer und lagern Sie sie nicht in der Nähe großer Wärmeausstrahlung. Die Batterien können in diesen Fällen auslaufen bzw. explodieren.

* Die angegebene Zeit gilt für *kontinuierliche* Programmausführung (Betriebsart «rechnend»). Wenn Sie den Computer für manuelle Berechnungen – eine Mischung aus den den Betriebsarten «bereit» und «rechnend» – benutzen, können Sie den HP-41 noch wesentlich länger als angegeben verwenden.

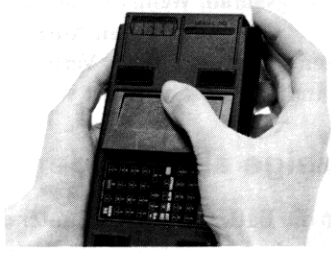
Die folgenden Batterien werden als Ersatzbatterien für Ihren HP-41 empfohlen:

Eveready E90	Mallory MN9100	VARTA 7245
National AM5(s)	Panasonic AM5(s)	

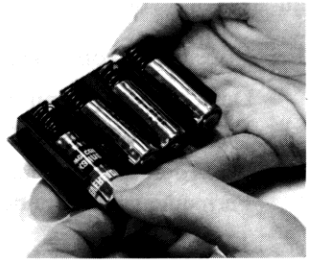
Der Inhalt des Permanentspeichers bleibt nach dem Ausbau der alten Batterien noch für eine kurze Zeit erhalten (sofern Sie den Computer vor dem Entfernen der Batterien ausgeschaltet haben). Sie haben daher normalerweise ausreichend Zeit zum Austauschen der Batterien zur Verfügung, ohne dabei Programme oder Daten zu verlieren. Der Inhalt des Permanentspeichers geht erst verloren, wenn über eine längere Zeitspanne keine Batterien im Computer sind.

Um neue Batterien einzusetzen, verfahren Sie wie folgt:

1. Vergewissern Sie sich, daß der Computer ausgeschaltet ist.
2. Halten Sie den Computer wie in der Abbildung gezeigt, und drücken Sie den Batteriefachdeckel nach oben, bis er herausspringt.



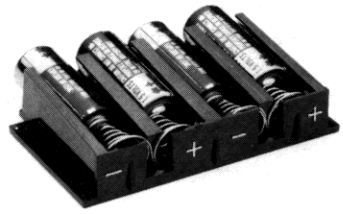
3. Entfernen Sie die Batterien aus dem Batteriefach.



VORSICHT

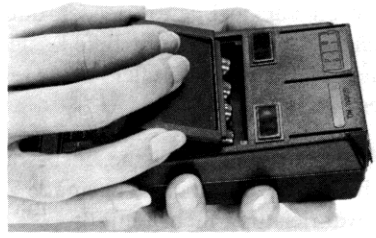
Erneuern Sie im nächsten Schritt *alle vier* Batterien. Wenn Sie nicht alle Batterien austauschen, besteht die Gefahr, daß eine der alten Batterien ausläuft. Achten Sie weiterhin darauf, daß Sie die Batterien nicht verkehrt herum einsetzen. In diesem Fall kann der Inhalt des Permanentspeichers verloren gehen.

4. Setzen Sie die neuen Batterien so ein, daß ihre Polaritätsmarkierungen mit denen an der Batteriehalterung übereinstimmen. Der Computer läßt sich nicht einschalten, wenn eine der Batterien falsch eingelegt ist.



5. Setzen Sie die Batteriehalterung mit den neuen Batterien so in das Batteriefach ein, daß die offenen Batterienenden auf die Einschubschächte zeigen.

6. Drücken Sie das obere Ende der Batteriehalterung bis zum Anschlag in das Gehäuse des Computers und lassen Sie dann das untere Ende einschnappen.



7. Schalten Sie den Computer ein. Wenn der Permanentspeicher aus irgendeinem Grund zurückgesetzt wurde (d.h. der Inhalt ist gelöscht), erscheint in der Anzeige die Meldung MEMORY LOST. Sie können diese Anzeige durch Drücken einer beliebigen Taste löschen.

Funktionsprüfung

Wenn sich Ihr HP-41 nicht einschalten läßt oder anderweitig nicht korrekt zu arbeiten zu scheint, können Sie den Computer mit Hilfe der folgenden Schritte austesten:

1. Vergewissern Sie sich, daß die Batterien einwandfrei eingesetzt und die Batteriekontakte nicht verschmutzt sind.
2. Wenn der Computer nicht auf das Drücken von Tasten reagiert, können Sie ihn folgendermaßen zurücksetzen: Drücken Sie zur gleichen Zeit die Tasten [ON] und [ENTER↑], halten Sie die Tasten kurz niedergedrückt und lassen Sie sie dann gleichzeitig wieder los. Schalten Sie den Computer, wenn nötig, ein und prüfen Sie, ob der HP-41 auf Tastendruck reagiert.
3. Wenn sich der Computer immer noch nicht einschalten läßt, nehmen Sie den Batteriensatz heraus und setzen ihn anschließend wieder ein.

   Setzen Sie neue Batterien ein, wenn sich der Computer immer noch nicht einschalten läßt.

   Wenn der Computer danach noch immer nicht auf Tastendruck reagiert, nehmen Sie die Batterien heraus und lassen Sie den Computer sich über Nacht entladen. Wenn nach dem Wiedereinsetzen der Batterien und dem Einschalten des Computers in der Anzeige die Meldung MEMORY LOST erscheint, ist der Speicher gelöscht und der Computer zurückgesetzt.

4. Wenn der HP-41 auch danach nicht auf Tastendruck reagiert, entfernen Sie den Batteriesatz und schließen Sie die Kontakte im Batteriefach kurz. *Sie brauchen die Kontakte nur für einen Moment kurzzuschließen.* Setzen Sie die Batterien wieder ein. Der Inhalt des Permanentspeichers ist verloren, und Sie müssen unter Umständen die Taste [ON] mehrmals drücken, um den Computer wieder einzuschalten.
5. Wenn der Computer sich immer noch nicht einschalten läßt, sollte er an eine Serviceniederlassung eingesandt werden.

Gewährleistung

Hewlett-Packard gewährleistet, daß der HP-41 frei von Material- und Verarbeitungsfehlern ist, und verpflichtet sich, etwaige fehlerhafte Teile kostenlos instandzusetzen oder auszutauschen, wenn das Gerät – direkt oder über einen HP-Vertragshändler – an Hewlett-Packard eingeschickt wird. Die Gewährleistung beträgt 12 Monate ab Verkaufsdatum.

Weitergehende Ansprüche, insbesondere auf Ersatz von Folgeschäden, können nicht geltend gemacht werden. Schäden, die durch unsachgemäße Veränderungen des Geräts durch Dritte zurückzuführen sind, werden von dieser Gewährleistung nicht umfaßt.

Die Gewährleistung gilt nur in Verbindung mit entweder

a) dem von einem Hewlett-Packard Vertragshändler ausgestellten und unterschriebenen Kaufbeleg, aus dem das Kaufdatum und die Seriennummer des Produkts hervorgehen müssen,

   oder

b) der Originalrechnung von Hewlett-Packard.

Die Ansprüche des Käufers aus dem Kaufvertrag bleiben von dieser Gewährleistungsregelung unberührt.

Änderungsverpflichtung

Sämtliche Produkte werden auf der Grundlage der technischen Daten bei der Herstellung verkauft. Hewlett-Packard übernimmt keine Verpflichtung, einmal verkaufte Geräte zu modifizieren oder dem neuesten technischen Stand anzupassen.

Gewährleistungsinformationen

Wenn Sie bezüglich dieser Gewährleistung Fragen haben, setzen Sie sich bitte mit einem Hewlett-Packard Vertragshändler oder einer Hewlett-Packard Verkaufs- und Serviceniederlassung in Verbindung. Falls Ihnen dies nicht möglich ist, schreiben Sie an:

- In Europa:

  Hewlett-Packard S.A.
  150, route du Nant-d'Avril
  P.O. Box
  CH-1217 Meyrin 2
  Genf
  Schweiz
  Telefon: (022) 83 81 11
  Hinweis: Senden Sie an diese Adresse keine Geräte zur Reparatur.

- In den U.S.A.:

  Hewlett-Packard
  Portable Computer Division
  1000 N.E. Circle Blvd.
  Corvallis, OR 97330
  Telefon: (503) 758-1010

- In anderen Ländern:

  Hewlett-Packard Intercontinental
  3495 Deer Creek Rd.
  Palo Alto, CA 94304
  U.S.A.
  Telefon: (415) 857-1501
  Hinweis: Bitte senden Sie an diese Adresse keine Geräte zur Reparatur.

Service

Hewlett-Packard unterhält ein weltweites Netz von Serviceniederlassungen. Diese Niederlassungen stehen Ihnen jederzeit für eine eventuelle Reparatur zur Verfügung, auch wenn die Gewährleistungsfrist von einem Jahr bereits abgelaufen sein sollte. Reparaturen nach Ablauf der Gewährleistungsfrist sind kostenpflichtig.

Service-Zentrale in den Vereinigten Staaten

In den Vereinigten Staaten erreichen Sie die Service-Zentrale von Hewlett-Packard für Taschenrechner- und Taschencomputer-Produkte unter der folgenden Anschrift:

Hewlett-Packard Company
Corvallis Division Service Department
P.O. Box 999/1000 N.E. Circle Blvd.
Corvallis, Oregon 97330, U.S.A.
Telefon: (503) 757-2000

Serviceniederlassungen in Europa

Hewlett-Packard unterhält Serviceniederlassungen in den folgenden Ländern. Wenn das Land, in dem Sie sich befinden, nicht aufgeführt ist, sollten Sie sich mit dem HP-Händler, bei dem Sie Ihr Gerät erworben haben, in Verbindung setzen.

BELGIEN
HEWLETT-PACKARD BELGIUM SA/NV
Boulevard de la Woluwe 100
Woluwelaan
B-1200 BRÜSSEL
Tel.: (2) 762 32 00

DÄNEMARK
HEWLETT-PACKARD A/S
Datavej 52
DK-3460 BIRKEROD (Kopenhagen)
Tel.: (02) 81 66 40

DEUTSCHLAND
HEWLETT-PACKARD GmbH
Vertriebszentrale
Berner Straße 117
Postfach 560140
D-6000 FRANKFURT 56
Tel.: (0611) 5004-1

FINNLAND
HEWLETT-PACKARD OY
Revontulentie 7
SF-02100 ESPOO 10 (Helsinki)
Tel.: (90) 455 02 11

FRANKREICH
HEWLETT-PACKARD FRANCE
Division Informatique Personnelle
S.A.V. Calculateurs de Poche
F-91947 LES ULIS CEDEX
Tel.: (6) 907 78 25

GROSSBRITANNIEN
HEWLETT-PACKARD Ltd.
King Street Lane
Winnersh, Wokingham
GB BERKSHIRE RG11 5AR
Tel.: (734) 78 47 74

ITALIEN
HEWLETT-PACKARD S.P.A.
Casella postale 3645 (Milano)
Via G. Di Vittorio, 9
I-20063 CERNUSCO SUL NAVIGLIO
(Milan)
Tel.: (2) 90 36 91

NIEDERLANDE
HEWLETT-PACKARD NEDERLAND B.V.
Van Heuven Goedhartlaan 121
NL-1181 KK AMSTELVEEN (Amsterdam)
P.O. Box 667
Tel.: (020) 47 20 21

NORWEGEN
HEWLETT-PACKARD NORGE A/S
P.O. Box 34
Oesterndalen 18
N-1345 OESTERAAS (Oslo)
Tel.: (2) 17 11 80

ÖSTERREICH
HEWLETT-PACKARD GmbH
Wagramerstraße-Lieblgasse
A-1220 WIEN
Tel.: (0222) 23 65 11

OSTEUROPA
Wenden Sie sich an die unter Österreich angegebene Adresse.

SCHWEDEN
HEWLETT-PACKARD SVERIGE AB
Skalholtsgatan 9
Box 19
S-16393 SPÅNGA
Tel.: (0046) 8 750 20 00

SCHWEIZ
HEWLETT-PACKARD (SCHWEIZ) AG
Allmend 2
CH-8967 WIDEN
Tel.: (057) 31 21 11

SPANIEN
HEWLETT-PACKARD ESPAÑOLA S.A.
Calle Jerez 3
E MADRID 16
Tel.: (1) 458 2600

Internationale Serviceinformation

Nicht jede Hewlett-Packard Serviceniederlassung bietet Service für alle Hewlett-Packard Produkte. Wenn Sie jedoch Ihr Gerät bei einem autorisierten Hewlett-Packard Händler gekauft haben, können Sie sicher sein, daß in dem Land des Erwerbs auch Servicemöglichkeiten bestehen.

Wenn Sie sich nicht in dem Land befinden, in dem Sie Ihr Gerät erworben haben, befragen Sie die lokale Hewlett-Packard Serviceniederlassung nach den dort verfügbaren Servicemöglichkeiten. Wenn kein Service verfügbar ist, senden Sie Ihr Gerät an die zuvor aufgeführte Adresse der Service-Zentrale in den Vereinigten Staaten. Unter der gleichen Adresse können Sie auch eine Liste der Serviceniederlassungen in anderen Ländern anfordern.

Sämtliche der durch den Versand entstehenden Kosten gehen zu Ihren Lasten.

Reparaturkosten

Bei Reparaturen außerhalb der Gewährleistungsfrist werden Standardsätze zugrunde gelegt, die den Arbeitslohn und den Materialaufwand beinhalten. In den Vereinigten Staaten unterliegt der gesamte Rechnungsbetrag der lokalen Verkaufssteuer. In europäischen Ländern ist der Rechnungsbetrag mehrwertsteuerpflichtig bzw. unterliegt ähnlichen Steuern. Der jeweilige Steueranteil wird in der Rechnung getrennt ausgewiesen.

Für Produkte, die durch Gewalteinwirkung oder sonstigen Mißbrauch beschädigt worden sind, gelten diese festen Reparatursätze nicht. In diesen Fällen wird die Reparatur individuell nach Arbeitszeit und Materialaufwand berechnet.

Gewährleistung auf Serviceleistungen

Bei Reparaturen außerhalb der Gewährleistungsfrist wird eine Garantie auf das Material und die Arbeitsleistung von 90 Tagen gegeben. Diese Frist gilt ab dem Reparaturdatum.

Versandanweisungen

Wenn Sie Ihr defektes Gerät einsenden, fügen Sie bitte bei.

- Eine vollständig ausgefüllte Service-Karte, einschließlich einer Fehlerbeschreibung.
- Die Originalrechnung oder einen sonstigen Kaufnachweis, sofern die Reparatur in die einjährige Gewährleistungsfrist fällt.

Zur Vermeidung von Transportschäden sollte das Gerät (zusammen mit der Service-Karte, einer kurzen Beschreibung des Fehlers, sowie, falls erforderlich, dem Kaufnachweis) nur in der Originalverpackung oder einer adäquaten Schutzverpackung versandt werden. Transportschäden fallen nicht unter die einjährige Gewährleistung. Das verpackte Gerät sollte entweder direkt oder über Ihren HP-Vertragshändler zur nächsten Hewlett-Packard Serviceniederlassung geschickt werden. (Wenn Sie sich nicht in dem Land befinden, in dem Sie Ihr Gerät erworben haben, beachten Sie bitte den Abschnitt «Internationale Serviceinformation».) Hewlett-Packard empfiehlt Ihnen, den Transport gegebenenfalls versichern zu lassen.

Die Versandkosten zur Hewlett-Packard Serviceniederlassung gehen zu Ihren Lasten, unabhängig davon, ob sich das Gerät noch in der Gewährleistungsfrist befindet oder nicht.

Bei Reparaturen innerhalb der Gewährleistungsfrist übernimmt die Serviceniederlassung die Kosten für den Rückversand. Bei Reparaturen außerhalb dieser Frist sind die Rücksendungskosten im Rechnungsbetrag enthalten.

Sonstiges

Hewlett-Packard bietet keine Service-Verträge an. Entwurf und Ausführung des Produkts und der Elektronik sind geistiges Eigentum von Hewlett-Packard; Service-Handbücher können daher nicht an Kunden abgegeben werden.

Sollten Sie weitere servicebezogene Fragen haben, setzen Sie sich bitte mit Ihrem HP-Vertragshändler oder der nächstgelegenen Hewlett-Packard Serviceniederlassung in Verbindung.

Benutzerberatung

Sollten beim Einsatz Ihres Geräts in bestimmten Anwendungsfällen Fragen auftauchen, so wenden Sie sich bitte an den HP-Vertragshändler, bei dem Sie Ihren HP-41 bezogen haben.

Ein Bezugsquellennachweis für den Fachhandel in der Bundesrepublik Deutschland ist erhältlich über

HEWLETT-PACKARD GmbH
Vertriebszentrale
Bernerstraße 117
Postfach 560140
D-6000 FRANKFURT/M 56

Temperaturspezifikationen

- Betriebstemperatur: 0° bis 45° C
- Lagertemperatur: 0° bis 45° C

*Wenn die Batterien ausgebaut sind oder die Ganggenauigkeit der Uhr ohne Bedeutung ist, gelten die folgenden Temperaturtoleranzen bei der Lagerung Ihres HP-41:**

–20° bis 60° C

* Wenn der Betrieb der Uhr durch vorübergehende extreme Temperaturschwankungen beeinflußt wurde, sollten Sie zur Korrektur *nicht* den Kompensationsfaktor sondern die Funktion [T+X] verwenden (d.h. Sie stellen einfach die korrekte Zeit ein und lassen den Kompensationsfaktor unverändert). Wie in Anhang F erläutert, dient der Kompensationsfaktor zur Berücksichtigung konstanter Einflußgrößen und sollte daher nicht zum Ausgleich ständig wechselnder Bedingungen benutzt werden.

Anhang H

Peripheriegeräte, Erweiterungen und HP-IL

Inhalt

HP-41 Peripheriegeräte 392
Magnetkartenleser HP 82104A 392
Drucker HP 82143A 393
Optischer Lesestift HP 82153A 393
Erweiterungseinheiten 393
Erweiterte Speicher-Module HP 82181A 393
Applikations-Module 393
Hewlett-Packard Interface Loop (HP-IL) und Peripheriegeräte 394
XROM-Funktionen und XROM-Nummern 394
Katalog 2: Katalog der externen Funktionen 394
XROM-Programme und XROM-Funktionen 395
Anzeige von XROM-Funktionen als Programmanweisungen 395
Doppelte XROM-Nummern 397

Ihr HP-41 wird zum *Controller* eines Computersystems, wenn Sie ihn mit Hewlett-Packard Peripheriegeräten verbinden bzw. um Erweiterungseinheiten ergänzen. Zusätzlich können Sie mittels des Hewlett-Packard Interface Loop (HP-IL) Moduls den HP-41 und bis zu 30 andere Einheiten zu einer seriellen Kommunikationsschleife zusammenschließen.

Die vier Einschubschächte (I/O Ports) des Computers dienen zum Einstecken von Systemerweiterungen – jeweils eine Einheit pro Port. (Das HP-IL Modul belegt einen Port, aber jedes weitere HP-IL Peripheriegerät wird danach lediglich über ein Kabel mit dem Modul oder einer anderen HP-IL Einheit verbunden.)

HP-41 Peripheriegeräte

Magnetkartenleser HP 82104A

Der Kartenleser HP 82104A erlaubt das Aufzeichnen von Programmen, Datenregistern und Tastenbelegungen auf Magnetkarten. Des weiteren können Sie mit Hilfe des Kartenlesers auf Magnetkarten gespeicherte Programme, Daten und Tastenbelegungen in den Hauptspeicher eines HP-41 zurückladen.

Mit Hilfe des Kartenlesers können Sie Information sehr einfach und schnell speichern und zurückladen (das mühevolle Eintasten von Programmen und Daten entfällt!). Alle Programme der Benutzerbibliothek werden auf Magnetkarten ausgeliefert. Des weiteren kann der HP 82104A auch Magnetkarten mit HP-97 und HP-67 Programmen lesen; die gelesenen Programme werden dabei automatisch in den internen Code des HP-41 umgesetzt.

Drucker HP 82143A

Mit dem HP 82143A können Sie Ihre Resultate und Programme nahezu geräuschlos auf 24 Zeichen breitem Thermopapier ausdrucken. Der Zeichensatz des Druckers umfaßt Groß- und Kleinbuchstaben, Ziffern und eine doppeltbreite Darstellung aller Zeichen. Zusätzlich können Sie über mehrere Druck-Modi bestimmen, welche Ausgaben gedruckt werden sollen, was insbesondere bei der Überprüfung komplexer Berechnungen und der Fehlersuche in Programmen von Nutzen ist.

Optischer Lesestift HP 82153A

Der optische Lesestift HP 82153A liest in HP-Strichcode verschlüsselte Programme in den Hauptspeicher Ihres HP-41 ein. Auf diese Weise können Sie Programme und Daten wesentlich schneller und exakter als durch manuelles Eintasten eingeben; zusätzlich können auch einzelne Funktionen vom Strichcode in den Computer übertragen werden. Alle Programme der Benutzerbibliothek sind als Strichcode-Listings erhältlich; die HP Lösungsbücher für den HP-41 enthalten ebenfalls Strichcode-Listings der einzelnen Programme.

Erweiterungseinheiten

Erweiterte Speicher-Module HP 82181A

Jedes Erweiterte Speicher-Modul HP 82181A vergrößert den erweiterten Speicher des HP-41 um 238 zusätzliche Speicherregister (1666 Bytes). Der HP-41CX verfügt standardmäßig bereits über 124 erweiterte Speicherregister. Sie können maximal zwei Erweiterte Speicher-Module HP 82181A in Ihrem HP-41 benutzen.

Applikations-Module

Die Applikations-Module sind Festwertspeicher (ROMs – Read Only Memory), die fertige Software zur Lösung von Aufgabenstellungen in speziellen Anwendungen (wie etwa Schaltkreisanalyse oder Finanzmathematik) enthalten. Sie können bis zu vier Applikations-Module in Ihren HP-41 einsetzen. Die in einem Applikations-Modul vorhandenen Programme und Funktionen werden in Katalog 2 gelistet.

Hewlett-Packard Interface Loop (HP-IL) und Peripheriegeräte

Durch Einstecken des HP-IL Moduls HP 82160A in einen der Einschubschächte des HP-41 können Sie ein serielles Peripheriesystem mit bis zu 30 HP-IL kompatiblen Peripherieeinheiten aufbauen. Der HP-41 dient dabei als Controller, der die Aktivitäten der anderen Einheiten in der Schleife überwacht und steuert. Die Zeitmeßfähigkeiten des HP-41CX erlauben die zeitlich gesteuerte Aufnahme von Daten durch HP-IL Einheiten sowie die automatische Ausgabe von Ergebnissen. Das HP-IL Modul enthält die zum Betrieb von HP-IL Druckern und Massenspeichereinheiten benötigten Funktionen.

Als HP-IL Peripheriegeräte sind momentan unter anderem verfügbar: Massenspeicher, Drucker, Plotter und Meßinstrumente sowie ein Video-Interface. Zusätzlich ergänzt das Erweiterte I/O Modul HP 82183A den Funktionssatz des HP-IL Moduls zur Eingabe/Ausgabe-Steuerung, und das Plottermodul HP 82184A bietet Ihnen fortgeschrittene Grafik-Fähigkeiten (einschließlich der Erzeugung von Strichcodes). Eine vollständige und aktuelle Liste der momentan verfügbaren HP-IL Produkte ist bei Ihrem HP-Vertragshändler erhältlich.

XROM-Funktionen und XROM-Nummern

Alle Funktionen und Programme, die durch eine HP-41 Peripherieeinheit implementiert werden und auf die Sie als Anwender zugreifen können, werden intern als «externe ROM» (*XROM*) Funktionen behandelt. Sämtliche externen Einheiten werden in Katalog 2 gelistet; des weiteren können Sie auch eine Liste der einzelnen Funktionen einer bestimmten externen Einheit erzeugen. Jede externe Funktion wird über eine aus zwei Teilen bestehende XROM-Nummer identifiziert.

Katalog 2: Katalog der externen Funktionen

Katalog 2 (siehe auch Abschnitt 9) ist eine nach Einheiten geordnete Liste aller XROM-Funktionen/Programme.* Beim HP-41CX zeigt Katalog 2 nur die Namen der externen Einheiten (die «ROM-Header»), solange Sie nicht die Katalogoperation unterbrechen und [ENTER↑] drücken. Dies bewirkt das Listen der einzelnen Funktionen und Programme, die durch die Einheit mit dem zuletzt angezeigten Namen zur Verfügung gestellt werden. Wenn Sie wieder zur Header-Liste zurückkehren wollen, müssen Sie zunächst die jeweilige Funktionsliste anhalten und anschließend wieder [ENTER↑] drücken.

[SST] und [BST] haben die gleiche Funktion wie bei den übrigen Katalogen. Sie können mit [BST] nicht von der Header-Liste auf die Funktionsliste umschalten; dies ist nur durch [ENTER↑] möglich. Wenn sämtliche Funktionen einer externen Einheit gelistet sind, wird die Katalogoperation mit dem Header der nächsten externen Einheit und den durch diese Einheit implementierten Funktionen fortgesetzt.

---

* Auch einige interne Funktionen (z.B. Zeitfunktionen und Funktionen des erweiterten Speichers) werden von Katalog 2 aufgeführt. Siehe «Zeitfunktionen» und «Erweiterte Speicherfunktionen» in Anhang I.

XROM-Programme und XROM-Funktionen

Allen durch Anwendungs- oder Erweiterungsmodule bzw. Peripheriegeräte implementierten Operationen ist gemeinsam, daß sie in einem ROM abgelegt sind und dem Anwender entweder als *Programm* oder als *Funktion* zur Verfügung gestellt werden. Ein Programm kann in den Hauptspeicher des Computers kopiert und dort gelistet oder modifiziert werden. Eine Funktion dagegen kann nur ausgeführt und nicht gelistet werden. Beim Listen von Katalog 2 sind Programme durch ein «hochgestelltes T» vor dem Programmnamen gekennzeichnet:

```
SECUR 1B  <------  ROM Header (Einheitsidentifikation)
TBONDS
TSTOCK    <------  Programm
  :
TATP
JDAY      <------  Funktion
TBEP
  :
```

Anzeige von XROM-Funktionen als Programmanweisungen

Wenn eine Programmanweisung eine externe Funktion enthält, hängt die Form der Anzeige dieser Anweisung davon ab, ob das die Funktion enthaltende Modul momentan eingesetzt ist und ob die betreffende XROM-Funktion als Funktion oder Programm realisiert ist.

Die XROM-Nummer identifiziert eine XROM-Funktion über die jeweilige externe Einheit (ROM-Identifikationsnummer) und die Position der Funktion innerhalb der externen Einheit (Funktionsnummer).

Wenn ein benötigtes Modul nicht eingesetzt ist, hat der HP-41 keinerlei Informationen über die XROM Funktionen dieses Moduls – *sofern* Sie nicht eine Funktion des Moduls einer Taste zugewiesen haben (in diesem Fall ist die XROM-Nummer bekannt, da sie der Taste zugeordnet wurde). Analog dazu werden als Programmanweisungen eingegebene Funktionen einer externen Einheit *nach* dem Entfernen dieser Einheit über die jeweiligen XROM-Nummern identifiziert.

Daher gilt:

- Wenn der Computer momentan Zugriff auf eine XROM-Funktion hat, wird diese wie folgt in eine Programmzeile eingetragen:

  Label für eine externe Funktion
  XROM^{T}*Label* für ein externes Programm

  Dies gilt auch, wenn eine benutzerdefinierte Taste zur Eingabe der Programmanweisung verwendet wird.

- Die *Label* oder **XROM**^{T}*Label* Anzeige einer Programmanweisung wird durch **XROM** *Nummer*, *Nummer* ersetzt, *wenn das betreffende Modul entfernt wird.* Die XROM-Nummer wird nur so lange angezeigt, wie das jeweilige Modul nicht eingesetzt ist; d.h. die ursprüngliche Anzeige erscheint wieder, sobald das Modul neu eingesteckt wird.

  Dies ist auch der Fall, wenn eine benutzerdefinierte Taste verwendet wird und das betreffende Modul fehlt.*

- Wenn das betreffende Modul nicht eingesetzt ist und Sie eine XROM-Funktion über Alpha-Ausführung in eine Programmzeile eingeben, wird diese Programmzeile wie der Aufruf eines Programms im Hauptspeicher angezeigt:

  XEQ^{T}*Label*

  Selbst wenn das Modul nachfolgend eingesetzt wird, bleibt die Anzeige *unverändert:*

  XEQ^{T}*Label*

Anzeige einer Programmanweisung

A. Das betreffende Modul ist eingesetzt, oder eine benutzerdefinierte Taste wird verwendet:

XROM^{T}*Label* (Programm)

oder ⟵ Modul wird entfernt ⟶ / ⟵ Modul wieder eingesetzt ⟶ **XROM** *nn,mm*

Label (Funktion)

(Diese Programmanweisung belegt zwei Bytes an Speicherplatz.)

B. Das betreffende Modul ist nicht eingesetzt, und es wird keine benutzerdefinierte Taste verwendet:

XEQ^{T}*Label* ⟵ Modul wird eingesetzt ⟶ / ⟵ Modul wieder entfernt ⟶ **XEQ**^{T}*Label*

(Diese Programmanweisung belegt zwei Bytes plus ein Byte pro Zeichen im Label.)

* Sie können eine Taste nur dann mit einer externen Funktion belegen, wenn das die jeweilige Funktion enthalten Funktion in den HP-41 eingesetzt ist. Bei fehlendem Modul erscheint die Meldung **NONEXISTENT**.

Ausführungszeit. Obwohl eine bei fehlendem Modul eingegebene Anweisung **XEQ**ᵀ*Label* die gewünschte externe Funktion/Programm korrekt ausführt, sobald das betreffende Modul eingesetzt ist (obiger Fall B), ist diese Instruktion nicht vollständig äquivalent zu **XROM**ᵀ*Label* oder *Label* (obiger Fall A). Fall B ist aus dem folgenden Grund weniger effizient und benötigt mehr Zeit zur Ausführung: bei einem XROM Aufruf (einschließlich eines einfachen *Label* für eine XROM-Funktion) wird sofort der Katalog 2 nach der betreffenden XROM-Funktion durchsucht. Dagegen wird bei einem **XEQ**ᵀ*Label* Befehl *zuerst* Katalog 1 mit allen Benutzerprogrammen* durchsucht. Erst wenn das gesuchte Label nicht in Katalog 1 vorhanden ist, wird die Suche mit Katalog 2 fortgesetzt.

Speicherbedarf. Eine **XROM**ᵀ*Label* oder *Label* Instruktion (Fall A) benötigt zwei Bytes an Speicherplatz, während eine **XEQ**ᵀ*Label* Anweisung zwei Bytes plus ein Byte pro Zeichen im Label belegt.

Doppelte XROM-Nummern

Alle einsteckbaren ROM Module tragen ROM-Identifikationsnummern, von denen einige doppelt vergeben sind. *Vermeiden Sie es, ROM Module mit identischen ROM-Identifikationsnummern gleichzeitig zu verwenden.* Alle erweiterten Funktionen tragen die XROM-Identifikationsnummer 25, während allen Zeitfunktionen die XROM-Identifikationsnummer 26 zugeordnet ist. (Keine der übrigen internen Funktionen haben XROM-Nummern.)

* Dies führt zu der interessanten Fragestellung, was passiert, wenn ein Benutzerprogramm im Hauptspeicher den Namen einer XROM-Funktion oder eines XROM-Programms als globales Label hat. Da die Suche nach **XEQ**ᵀ*Label* immer in Katalog 1 beginnt, wird grundsätzlich das Benutzerprogramm und nicht die XROM-Funktion ausgeführt. Dadurch haben Sie die Möglichkeit, ein Programm von einem externen ROM Modul in den Hauptspeicher zu kopieren, es dort zu modifizieren und danach die modifizierte Version anstelle der Version im ROM Modul auszuführen (selbst wenn das Modul eingesteckt ist).

Anhang I

Vergleich HP-41CX und HP-41C/CV

Inhalt

Überblick 398
Katalogorganisation der neuen Funktionen 399
Benutzerhandbuch 399
Speicherkonfiguration 399
Katalogoperationen 400
Kataloge 1, 2 und 3 400
Neue Kataloge: 4, 5 und 6 400
Zeitfunktionen 400
Erweiterter Speicher und erweiterte Funktionen 401
Erweiterte Speicherfunktionen 401
Erweiterte Funktionen 402
Neue Terminologie 402

Wenn Sie bereits mit dem HP-41C oder dem HP-41CV gearbeitet haben, sind Sie schon mit wesentlichen Bestandteilen der Funktionsweise des HP-41CX vertraut. Dieser Anhang definiert die Unterschiede zwischen dem HP-41CX und dem HP-41C/CV.

Alle für den HP-41C/CV geschriebenen Programme (einschließlich der Programme in Einsteck-Modulen) sind auch auf dem HP-41CX uneingeschränkt verwendbar. Für den HP-41CX geschriebene Programme können jedoch nicht unbedingt auf dem HP-41C/CV ausgeführt werden.

Überblick

Der Computer HP-41CX basiert auf dem HP-41CV (der wiederum einem HP-41C mit fünffacher Speicherkapazität entspricht). Zusätzlich enthält der HP-41CX sämtliche Funktionen und Speicherregister des Time-Moduls HP 82182A und des Erweiterten Funktions/Speicher-Moduls HP 82180A sowie weitere Alarm-, Stoppuhr-, Speicher- und sonstige Dienstfunktionen. Die Funktionsweise der Kataloge unterscheidet sich geringfügig von der beim HP-41C/CV und die Standardspeicherkonfiguration ist unterschiedlich: 100 Datenregister gegenüber 273 beim HP-41CV. Im folgenden finden Sie eine detaillierte Gegenüberstellung der einzelnen Vergleichspunkte unter Angabe von Seitenreferenzen für dieses Handbuch.

Katalogorganisation der neuen Funktionen

Die HP-41CX Funktionen sind so katalogisiert, daß alle neuen Funktionen (im HP-41C/CV nicht vorhandene Funktionen) in Katalog 2, dem Katalog der externen Funktionen, gelistet werden, während Katalog 3, der Katalog der Standardfunktionen, gegenüber dem HP-41C/CV unverändert ist. Diese Aufteilung der Kataloge wurde aus Gründen der Konsistenz gegenüber früheren Produkten gewählt: der Standardfunktionssatz bleibt unverändert, die Zeitfunktionen und die erweiterten Funktionen erscheinen in Katalog 2, wie dies auch der Fall wäre, wenn diese Funktionen über das Time-Modul bzw. das Erweiterte Funktions/Speicher-Modul zur Verfügung gestellt worden wären. Daher identifiziert der Computer die Zeit- und erweiterten Funktionen anhand der XROM-Nummern (siehe Anhang H) der ursprünglichen Module. Die ROM-Identifikationsnummer für alle erweiterten Funktionen ist 25, die der Zeitfunktionen 26. (Verwenden Sie kein ROM-Einschubmodul mit einer dieser Identifikationsnummern.)

Benutzerhandbuch

Das Benutzerhandbuch für den HP-41CX wurde vollkommen neu geschrieben. Viele der Erläuterungen in der bisherigen Literatur (für den HP-41C/CV und die beiden Module) wurden verändert, überarbeitet und verdeutlicht. Die in diesem Handbuch benutzte Terminologie ist bis zu einem gewissen Grad neu, insbesondere in den Bereichen Programmierung, Speicher und Alarme. Die zur Darstellung der umgeschalteten Funktionen und Nicht-Tastenfeldfunktionen verwendeten Konventionen wurden ebenfalls geändert; eine Beschreibung finden Sie auf der Innenseite des Rückumschlags.

Speicherkonfiguration

Die unterschiedliche Speicheraufteilung der einzelnen HP-41 Computer wird aus der folgenden Tabelle ersichtlich.

Speicherkonfiguration

Typ	Hauptspeicher			Erweiterter Speicher
	gesamt	Standardkonfiguration		
		Datenspeicher	ungebunden*	
HP-41CX	319	100 (R_{00}–R_{99})	219	124
HP-41CV	319	273 (R_{00}–R_{272})	46	0
HP-41C	63	17 (R_{00}–R_{16})	46	0

* Der für Programmanweisungen, Alarme und Tastenzuordnungen benötigte Speicherplatz wird von den frei verfügbaren Registern abgezogen. Details finden Sie in Abschnitt 12.

Katalogoperationen

Beim Listen eines HP-41CX Katalogs können Sie den Listvorgang durch Drücken einer beliebigen Taste (mit Ausnahme von [R/S] und [ON]) *beschleunigen*. (Beim HP-41C/CV würde dies die Listoperation *verlangsamen*.)

Kataloge 1, 2 und 3

Katalog 1 zeigt nun die Anzahl der Bytes für jedes Programm (d.h. das Segment zwischen zwei END-Anweisungen – siehe Seite 171). Katalog 2 wurde wesentlich verändert und ist nun in Funktionsgruppen aufgeschlüsselt. Katalog 3 ist unverändert.

Neue Kataloge: 4, 5 und 6

Der HP-41CX enthält drei neue Kataloge. Im Gegensatz zu den Katalogen 1, 2 und 3 nehmen diese Kataloge – selbst bei einer Unterbrechung – genausoviel Strom wie ein ablaufendes Programm auf. Aus diesem Grund werden diese Katalogoperationen zwei Minuten nach einem Anhalten (bzw. eine Minute bei geringer Batteriespannung) automatisch beendet.

Die neuen Kataloge beginnen zu blinken, wenn Sie bei einer Listoperation mit [SST] das Ende bzw. mit [BST] den Anfang des Katalogs erreicht haben. (Beim HP-41C/CV beendet [SST] am Ende des Katalogs die Katalogfunktion.)

Katalog 4: Inhaltsverzeichnis des erweiterten Speichers (Seite 206). Sie können eine Liste aller Files im erweiterten Speicher entweder mit [CATALOG] 4 oder mit [EMDIR] anzeigen. [CATALOG] 4 ist im Gegensatz zu [EMDIR] *nicht* programmierbar. Die Funktion [EMDIR] ist im wesentlichen identisch mit der gleichnamigen Funktion im Erweiterten Funktions/Speicher-Modul. *Es gelten die folgenden Unterschiede:* Sie können die Listoperation nun mit [R/S] starten und anhalten; [SST] und [BST] ermöglichen ein schrittweises Durchlaufen des Katalogs; zur Protokollierung des Katalogs muß sich der Drucker im Trace-Modus befinden.

Katalog 5: Der Alarmkatalog (Seite 255). Mit den Funktionen [CATALOG] 5 und [ALMCAT] können Sie auf eine Liste aller im Speicher befindlichen Alarme zugreifen. [CATALOG] 5 ist im Gegensatz zu [ALMCAT] *nicht* programmierbar. [ALMCAT] entspricht im wesentlichen der gleichen Funktion im Time-Modul *mit der Ausnahme*, daß die Meldung CAT EMPTY angezeigt wird, wenn keine Alarme im Speicher vorhanden sind.

Katalog 6: Tastenzuordnungen (Seite 168). Diese Funktion erzeugt eine nach Tastencodes geordnete Liste aller Tastenbelegungen (Funktionen und globale Labels). Sie können durch Drücken von ■ [C] eine bestimmte Tastenzuordnung löschen.

Zeitfunktionen

Sie finden in Teil IV dieses Handbuchs eine Beschreibung aller Zeitfunktionen des HP-41CX: «Uhr- und Datumsfunktionen» (Abschnitt 15), «Alarmfunktionen» (Abschnitt 16) und «Stoppuhrbetrieb» (Abschnitt 17). Zusätzlich zu den im Time-Modul implementierten Zeitfunktionen enthält der HP-41CX noch weitere Alarm-Operationen, insbesondere Prozeduren zum Löschen von Alarmen, sowie eine zusätzliche Stoppuhrfunktion.

- Ein wiederholender Meldealarm kann während der Abarbeitung des Alarms durch Drücken von ■ C gelöscht werden (siehe Seite 255).
- Es stehen Ihnen drei programmierbare Alarm-Löschfunktionen zur Verfügung: CLALMA (*clear alarm by Alpha*), CLALMX (*clear alarm by X*) und CLRALMS (*clear all alarms*) (siehe Seite 258).
- Die programmierbare Funktion RCLALM (*recall alarm*) gibt die Parameter eines Alarms in die Stackregister und das Alpha-Register zurück (siehe Seite 252).
- Das minimale Wiederholungsintervall für einen Alarm wurde von 10 Sekunden auf eine Sekunde verkleinert (siehe Seite 250).
- Die programmierbare Funktion SWPT (*stopwatch and pointers*) ermöglicht das Aktivieren der Stoppuhr und das Setzen der Stoppuhrzeiger (siehe Seite 274).

Die dem Time-Modul entnommenen Funktionen werden in Katalog 2 unter –TIME 2B (*time functions, revision 2B*) gelistet. Die neuen Zeitfunktionen erscheinen in Katalog 2 unter –X TIME (*extra time functions*).

Erweiterter Speicher und erweiterte Funktionen

Der HP-41CX enthält erweiterte Speicherregister und erweiterte Funktionen, von denen die meisten dem Erweiterten Speicher/Funktions-Modul entnommen wurden. Im einzelnen sind dies:

- 124 Register an *erweitertem Speicher* für Ihre Programme, Daten und Textfiles (ASCII-Files).
- Funktionen zur Erzeugung und Manipulation von Files im erweiterten Speicher (*erweiterte Speicherfunktionen*).
- Funktionen zur Manipulation von Flags, Daten und Alpha-Strings (*erweiterte Funktionen*).
- Neue Abfragen für bedingte Verzweigungen (*erweiterte Funktionen*).
- Zusätzliche Dienstfunktionen (*erweiterte Funktionen*).

Die dem Erweiterten Funktions/Speicher-Modul entnommenen Funktionen des HP-41CX werden in Katalog 2 unter –EXT FCN 2C (*extended functions, revision 2C*) gelistet. Die neuen, *zusätzlichen* erweiterten Funktionen erscheinen in Katalog 2 unter –X EXT FCN (*extra extended functions*).

Erweiterte Speicherfunktionen

Abschnitt 13 («Erweiterter Speicher») und Abschnitt 14 («Der Texteditor») beschreiben den Aufbau des erweiterten Speichers und die Manipulation von Files im erweiterten Speicher mittels der erweiterten Speicherfunktionen. Der HP-41CX enthält einige zusätzliche, neue erweiterte Speicherfunktionen, die im folgenden vorgestellt werden.

Der Texteditor ist eine der wesentlichen Erweiterungen des HP-41CX gegenüber dem HP-41C/CV. Die Funktion ED (*editor*) verändert die Funktionsweise von Tastenfeld und Anzeige, so daß Sie einen Textfile aufrufen und den Inhalt eines Records während der Bearbeitung betrachten können, anstatt den Text über einzelne Operationen im Alpha-Register manipulieren zu müssen. Die Tastenfeldbelegung des Texteditors ist auf der Gehäuserückseite des HP-41CX dargestellt und beinhaltet den Alpha-Zeichensatz. (Die Rückseite des HP-41C/CV zeigt nur das Alpha-Tastenfeld.)

Zusätzlich stehen Ihnen die folgenden neuen erweiterten Speicherfunktionen zur Verfügung:

- ASROOM (*ASCII room*) und EMROOM (*extended memory*) geben die Größe des in einem Text- oder ASCII-File (siehe Seite 222) bzw. im erweiterten Speicher (siehe Seite 208) noch verfügbaren Speicherbereichs zurück.
- EMDIRX (*extended memory directory by X*) ruft den Namen und den Typ eines bestimmten Files ab und macht ihn zum momentanen File (siehe Seite 207).
- RESZFL (*resize file*) verändert die Größe eines Text- oder Datenfiles (siehe Seite 213).

Erweiterte Funktionen

Die erweiterten Funktionen des HP-41CX lassen sich in die im folgenden aufgeführten drei Kategorien einteilen. Einige dieser Funktionen sind ebenfalls im Erweiterten Funktions/Speicher-Modul vorhanden. Die dort nicht implementierten Funktionen sind hier als *zusätzliche* erweiterte Funktionen gekennzeichnet.

Funktionen zur Manipulation von Flags, Daten und Alpha-Strings. Der HP-41CX enthält Funktionen des Erweiterten Funktions/Speicher-Moduls zur Manipulation von Information in und zwischen Registern. Diese Funktionen werden in Abschnitt 12 «Der Hauptspeicher» beschrieben. Flagoperationen werden in Abschnitt 19, «Flags», behandelt. Funktionen zur Manipulation von Alpha-Daten werden in Abschnitt 21, «Alpha- und interaktive Funktionen» erläutert.

Vergleichsabfragen. Die Vergleichsabfragen sind zusätzliche erweiterte Funktionen, die den Vergleich des X-Registerinhalts mit den Inhalten anderer Register ermöglichen (siehe Abschnitt 20 «Programmverzweigungen»).

Verschiedene. Die folgenden Dienstfunktionen wurden vom Erweiterten Funktions/Speicher-Modul übernommen:

- PASN (*programmable assign*) und CLKEYS (*clear key assignments*) – siehe Abschnitt 9 «Das Tastenfeld und die Anzeige».
- SIZE? und PSIZE (*programmable size*) – siehe Abschnitt 12 «Der Hauptspeicher».
- PCLPS (*programmable clear programs*) – siehe Abschnitt 18 «Einfache Programmierung».
- GETKEY – siehe Abschnitt 21 «Alpha- und interaktive Funktionen». Diese Funktion unterbricht beim Drücken einer Taste die Ausführung eines Programms. Der Tastencode der gedrückten Taste kann dann zur Verzweigung zu einem bestimmten Unterprogramm benutzt werden.

Des weiteren enthält der HP-41CX die folgenden *zusätzlichen* Dienstfunktionen:

- ΣREG? (*statistics registers?*) – siehe Abschnitt 11 «Numerische Funktionen».
- CLRGX (*clear registers by X*) – siehe Abschnitt 12 «Der Hauptspeicher».
- GETKEYX (*get key by X*) – siehe Abschnitt 17 «Alpha- und interaktive Funktionen».

Neue Terminologie

Viele der in diesem Handbuch verwendeten Begriffe und Namen stimmen nicht mit der Terminologie überein, die in der für den HP-41 und die Module bereits vorhandenen Literatur verwendet wird. Die folgende Tabelle soll Ihnen helfen, wenn Sie mehr mit den alten Begriffen vertraut sind.

Äquivalente Begriffe

HP-41CX	HP-41C/CV	Erläuterung
Alpha-Ausführung	Ausführung über die Anzeige	
bedingter Alarm	nichtunterbrechender Steueralarm	Verdeutlichung des Unterschieds zwischen bedingten Alarmen und Steueralarmen.
Datenregister	Datenspeicherregister	
Eingabezeichen	Unterstreichungszeichen	
Flags		
Benutzer- (00–10)	allgemeine Anwenderflags	Benutzerflags werden ausschließlich vom Benutzer definiert; Steuerflags dagegen vom HP-41. Sie können jedoch im Gegensatz zu den Systemflags den Status jedes Benutzer- und Steuerflags beliebig verändern.
Steuer- (11–29)	spezielle Anwenderflags	
indirekt adressierbare Register (Register über R_{99})	erweiterte Speicherregister	Verdeutlichung des Unterschieds zu den Registern im erweiterten Speicher.
momentaner File oder Programm	Arbeitsfile oder Programm	
Steueralarm	unterbrechender Steueralarm	
Tastenfelder Alarmkatalog- Alpha- Normal- Stoppuhr- User-	Modi	Diese Betriebszustände werden durch umdefinierte Tastenfelder charakterisiert.
übergangener überfälliger Alarm	inaktiver überfälliger Alarm	
ungebundene Register	Programmregister; Programmspeicher	In diesem Teil des Speichers werden nicht nur Programme abgelegt.
GTO	■ GTO	Alle umgeschalteten Funktionen erscheinen goldfarben (siehe Innenseite Vorderumschlag).

Strichcode-Listings der Beispielprogramme

Inhalt

ALMEIN 404
ALMREL 404
ZEITEN 405
ZE 406
Σ 409

ALMEIN (aus Abschnitt 16)

Benötigte Programmregister: 11

REIHE 1: ZEILEN 1-3

REIHE 2: ZEILEN 3-7

REIHE 3: ZEILEN 7-9

REIHE 4: ZEILEN 9-11

REIHE 5: ZEILEN 11-21

REIHE 6: ZEILEN 21-27

ALMREL (aus Abschnitt 16)

Benötigte Programmregister: 10

REIHE 1: ZEILEN 1-3

ALMREL (Fortsetzung)

REIHE 2: ZEILEN 3-5

REIHE 3: ZEILEN 6-10

REIHE 4: ZEILEN 11-21

REIHE 5: ZEILEN 22-30

REIHE 6: ZEILEN 30-34

ZEITEN (aus Abschnitt 17)
Benötigte Programmregister: 14

REIHE 1: ZEILEN 1-3

REIHE 2: ZEILEN 4-6

REIHE 3: ZEILEN 6-8

REIHE 4: ZEILEN 8-15

REIHE 5: ZEILEN 15-22

REIHE 6: ZEILEN 23-28

REIHE 7: ZEILEN 28-36

REIHE 8: ZEILEN 37-40

ZE (aus Abschnitt 22)
Benötigte Programmregister: 71

REIHE 1: ZEILEN 1-6

REIHE 2: ZEILEN 6-10

REIHE 3: ZEILEN 10-15

REIHE 4: ZEILEN 15-19

REIHE 5: ZEILEN 19-20

REIHE 6: ZEILEN 20-29

REIHE 7: ZEILEN 30-37

REIHE 8: ZEILEN 38-42

REIHE 9: ZEILEN 43-46

REIHE 10: ZEILEN 46-52

REIHE 11: ZEILEN 52-54

REIHE 12: ZEILEN 54-61

REIHE 13: ZEILEN 62-63

ZE (Fortsetzung)

REIHE 14: ZEILEN 64-72

REIHE 15: ZEILEN 73-80

REIHE 16: ZEILEN 81-89

REIHE 17: ZEILEN 90-98

REIHE 18: ZEILEN 98-99

REIHE 19: ZEILEN 100-107

REIHE 20: ZEILEN 107-114

REIHE 21: ZEILEN 114-121

REIHE 22: ZEILEN 121-129

REIHE 23: ZEILEN 129-136

REIHE 24: ZEILEN 136-140

REIHE 25: ZEILEN 141-148

REIHE 26: ZEILEN 149-155

ZE (Fortsetzung)

REIHE 27: ZEILEN 155-158

REIHE 28: ZEILEN 159-166

REIHE 29: ZEILEN 166-172

REIHE 30: ZEILEN 173-180

REIHE 31: ZEILEN 180-185

REIHE 32: ZEILEN 185-188

REIHE 33: ZEILEN 188-197

REIHE 34: ZEILEN 197-207

REIHE 35: ZEILEN 208-216

REIHE 36: ZEILEN 217-222

REIHE 37: ZEILEN 222-230

REIHE 38: ZEILEN 230-232

REIHE 39: ZEILEN 232-233

Σ (aus Abschnitt 22)
Benötigte Programmregister: 102

REIHE 1: ZEILEN 1-7

REIHE 2: ZEILEN 7-10

REIHE 3: ZEILEN 11-16

REIHE 4: ZEILEN 16-23

REIHE 5: ZEILEN 23-31

REIHE 6: ZEILEN 32-36

REIHE 7: ZEILEN 36-41

REIHE 8: ZEILEN 42-51

REIHE 9: ZEILEN 51-56

REIHE 10: ZEILEN 56-65

REIHE 11: ZEILEN 65-72

REIHE 12: ZEILEN 72-80

REIHE 13: ZEILEN 80-88

Σ (Fortsetzung)

REIHE 14: ZEILEN 88-90

REIHE 15: ZEILEN 91-99

REIHE 16: ZEILEN 100-109

REIHE 17: ZEILEN 110-118

REIHE 18: ZEILEN 119-127

REIHE 19: ZEILEN 127-130

REIHE 20: ZEILEN 130-137

REIHE 21: ZEILEN 138-141

REIHE 22: ZEILEN 141-146

REIHE 23: ZEILEN 146-150

REIHE 24: ZEILEN 150-157

REIHE 25: ZEILEN 158-167

REIHE 26: ZEILEN 167-177

REIHE 27: ZEILEN 178-184

Σ (Fortsetzung)

REIHE 28: ZEILEN 184-193

REIHE 29: ZEILEN 193-200

REIHE 30: ZEILEN 201-209

REIHE 31: ZEILEN 210-216

REIHE 32: ZEILEN 216-227

REIHE 33: ZEILEN 228-234

REIHE 34: ZEILEN 234-242

REIHE 35: ZEILEN 242-250

REIHE 36: ZEILEN 250-251

REIHE 37: ZEILEN 251-256

REIHE 38: ZEILEN 257-263

REIHE 39: ZEILEN 263-270

REIHE 40: ZEILEN 271-278

REIHE 41: ZEILEN 279-283

Σ (Fortsetzung)

REIHE 42: ZEILEN 284-291

REIHE 43: ZEILEN 292-299

REIHE 44: ZEILEN 300-305

REIHE 45: ZEILEN 305-306

REIHE 46: ZEILEN 306-310

REIHE 47: ZEILEN 310-313

REIHE 48: ZEILEN 314-322

REIHE 49: ZEILEN 323-332

REIHE 50: ZEILEN 332-335

REIHE 51: ZEILEN 335-342

REIHE 52: ZEILEN 343-347

REIHE 53: ZEILEN 348-355

REIHE 54: ZEILEN 356-364

REIHE 55: ZEILEN 364-369

Funktionsverzeichnis

Funktionsverzeichnis

Inhalt

Einführung 414
Auffinden einer Funktion 414
Erläuterungen zum Funktionsverzeichnis 414
System/Format-Funktionen 416
Löschfunktionen 418
Stack- und Datenregister-Funktionen 420
Numerische Funktionen 423
Funktionen des erweiterten Speichers 425
Zeitfunktionen 429
Editierungs-Funktionen 431
Funktionen zur Steuerung der Programmausführung 432
Alpha-Funktionen 436
Interaktive Funktionen 438

Einführung

Die zehn Funktionstabellen in diesem Abschnitt beschreiben die Funktionen des HP-41CX. Die Funktionen sind nach Oberbegriffen geordnet; einige Funktionen erscheinen in mehreren Tabellen. Die meisten Tabellen enthalten die in «Erläuterungen zum Funktionsverzeichnis» angegebenen Informationen. Die Tabelle zu den Funktionen des erweiterten Speichers enthält einige Sonderinformationen; die Erläuterungen hierzu finden Sie bei der Tabelle.

Auffinden einer Funktion

- Wenn Sie eine Funktion suchen, die eine bestimmte Operation ausführt, gehen Sie die Tabelle durch, deren Überschrift der Typ der gewünschten Operation beschreibt.
- Wenn Sie den Namen einer Funktion kennen und herausfinden möchten, welche Operation damit verbunden ist, schlagen Sie im Funktionsindex auf Seite 452 nach. Der letzte Seitenverweis bezieht sich auf die entsprechende Funktionstabelle.

Erläuterungen zum Funktionsverzeichnis

Alpha-Name. Unter diesem Namen wird eine Funktion in den Katalogen 2 und 3 sowie in Programm-Listings aufgeführt. Wenn Sie eine Funktionstaste gedrückt halten, erscheint der Alpha-Name in der Anzeige. Sie müssen den Alpha-Namen verwenden, wenn sie eine Funktion einer Taste des User-Tastenfelds zuordnen wollen; eine Zuordnung ist nicht möglich, wenn Sie in dieser Spalte keine Eintragung finden.

Tastenfeld-Name. Dies ist der Funktionsname, wie Sie ihn auf dem Normal- oder Alpha-Tastenfeld finden. (Wenn das Tastensymbol in dieser Spalte goldfarben gesetzt ist, muß vor der Funktionstaste die Umschalttaste ■ gedrückt werden.) Wenn eine Funktion keine Eintragung in dieser Spalte besitzt, müssen Sie zur Ausführung die Taste [XEQ] und den Alpha-Namen benutzen oder die Funktion einer Taste des User-Tastenfelds zuordnen.

IND. Ein «I» in dieser Spalte zeigt an, daß der Parameter dieser Funktion indirekt spezifiziert werden kann. Geben Sie dazu die Funktion ein und drücken Sie dann die Taste ■; daraufhin erscheint in der Anzeige IND hinter dem Funktionsnamen. Spezifizieren Sie nun das Register, das die *Adresse* des gewünschten Registers enthält. Die indirekte Ausführung einer Funktion wird ausführlich in Abschnitt 9 erläutert.

Stack. Die Eintragungen in diese Spalte beschreiben den Einfluß einer Funktion auf den automatischen Speicherstack.

L = LAST X. Der vorhergehende Inhalt des X-Registers wird in das LAST X Register kopiert.

↓ = Der Stack wird nach unten verschoben (Stack Drop). Der Inhalt des Z-Registers wird ins Y-Register kopiert und der Inhalt des T-Registers wird ins Z-Register kopiert.

↑ = Der Stack wird angehoben (Stack Lift). Wenn der Stack Lift vorher freigegeben wurde, werden die Inhalte der X-, Y- und Z-Register jeweils um ein Register angehoben und in die Y-, Z- und T-Register kopiert. Der vorhergehende Inhalt des T-Registers geht dabei verloren.

F = Stack Lift wird freigegeben. Der Stack wird angehoben, wenn die als nächstes ausgeführte Funktion ein «↑» in der Spalte «Stack» hat oder wenn Sie eine Zahl eingeben. (Fast alle Funktionen geben den Stack Lift frei.)

G = Stack Lift wird gesperrt. Wenn die als nächstes ausgeführte Funktion ein «↑» in der Spalte «Stack» hat oder wenn Sie eine Zahl eingeben, überschreibt die neue Zahl den vorhergehenden Inhalt des X-Registers und der Stack wird nicht angehoben. (Der Stack Lift wird nur durch die Funktionen [CLx], [ENTER↑], [Σ+] und [Σ−] gesperrt.)

N = Neutral. Der Stack Lift wird weder gesperrt noch freigegeben; der vorhergehende Zustand wird aufrechterhalten.

Flags. In dieser Spalte sind die von der Ausführung der Funktion betroffenen Flags aufgeführt.

Bytes. Die Zahl in dieser Spalte zeigt an, wieviel Bytes des Programmspeichers belegt werden, wenn die Funktion als Programmanweisung aufgezeichnet wird. Eine Funktion, die keinen Eintrag in dieser Spalte besitzt, ist nicht programmierbar.

Seite. Dies sind Verweise auf Textstellen des Handbuchs.

System/Format-Funktionen

Der durch die Ausführung der meisten dieser Funktionen gesetzte Zustand bleibt dauerhaft bestehen: Anzeigeformate, trigonometrischer Modus, Speicherzuweisungen, Zuordnungen des User-Tastenfelds usw. Die Tabelle führt auch bestimmte System-Operationen an, wie z.B. die Modustasten und die Kataloge.

Alpha-Name	Tastenfeld-Name	Beschreibung	IND	Stack	Flags	Bytes	Seite
	ALPHA	Schaltet das Alpha-Tastenfeld ein und aus.			48		155
AOFF		Schaltet das Alpha-Tastenfeld aus.		F	48	1	159
AON		Schaltet das Alpha-Tastenfeld ein.		F	48	1	159
ASN	ASN	Ordnet eine anzugebende Funktion (oder globales Label) einer zu spezifizierenden Taste des User-Tastenfelds zu.		F			166
CAT *n*	CATALOG *n*	Führt den Katalog *n* aus ($1 \leqslant n \leqslant 6$).					
		Kataloge 1, 2, 3, 6.		N			170
		Katalog 4 (EMDIR).		F		(2)	206
		Katalog 5 (ALMCAT).		F	31	(2)	255
CF *nn*	CF *nn*	Löscht Flag *nn*, $00 \leqslant nn \leqslant 29$.	I	F	*nn*	2	288
CLK12		Schaltet die Anzeige der Uhr auf das 12-Stunden-Format.		F		2	239
CLK24		Schaltet die Anzeige der Uhr auf das 24-Stunden-Format.		F		2	239
CLKEYS		Löscht alle Tastenzuordnungen des User-Tastenfelds.		F		2	168
CLKT		Schaltet die Anzeige der Uhr auf das Nur-Uhrzeit-Format.		F		2	239
CLKTD		Schaltet die Anzeige der Uhr auf das Uhrzeit/Datum-Format.		F		2	239
DEG		Wählt den Winkelmodus «Altgrad» (in Dezimalform).		F	42–43	1	186
DMY		Schaltet die Anzeige des Datums auf das Tag-Monat-Jahr-Format.		F	31	2	242
ENG *n*	ENG *n*	Schaltet auf das technische Anzeigeformat (Anzeige von $n+1$ Stellen).	I	F	36–41	2	161

System/Format-Funktionen (Fortsetzung)

Alpha-Name	Tastenfeld-Name	Beschreibung	IND	Stack	Flags	Bytes	Seite
FIX *n*	FIX *n*	Schaltet die Anzeige auf das Festkommaformat (Anzeige von *n* Dezimalstellen).	I	F	36–41	2	160
GRAD		Wählt den Winkelmodus «Neugrad».		F	42–43	1	186
MDY		Schaltet die Anzeige des Datums auf das Monat-Tag-Jahr-Format.		F	31	2	242
	ON	Schaltet den Computer ein und aus.			12–26, 44–55		155
ON		Desaktiviert die Abschaltautomatik (der Computer bleibt eingeschaltet).			44	1	155
PASN		Ordnet die im Alpha-Register spezifizierte Funktion (bzw. das spezifizierte Label) der durch den Tastencode im X-Register spezifizierten Taste des User-Tastenfelds zu.		F		2	166
	PRGM	Schaltet den Programm-Modus ein und aus.		N			155
PSIZE		Ordnet die im X-Register spezifizierte Anzahl von Hauptspeicher-Registern der Datenspeicherung zu.		F		2	199
RAD		Wählt den Winkelmodus «Bogenmaß».		F	42–43	1	186
RCLFLAG		Ruft den Status der Flags 00 bis 43 ab.		↑, F		2	296
SCI *n*	SCI *n*	Schaltet auf das wissenschaftliche Anzeigeformat (Anzeige von *n* Dezimalstellen).	I	F	36–41	2	161
SF *nn*	SF *nn*	Setzt Flag *nn*, $00 \leqslant nn \leqslant 29$.	I	F	*nn*	2	288
ΣREG *nn*		Definiert die Register R_{nn} *bis* R_{nn+5} als Statistikregister.	I	F		2	190
ΣREG?		Ruft die Adresse des ersten Registers aus dem Block der gegenwärtig definierten Statistikregister ab.		↑, F		2	190
SIZE *nnn*		Ordnet *nnn* Hauptspeicher-Register der Datenspeicherung zu.		F			199

System/Format-Funktionen (Fortsetzung)

Alpha-Name	Tastenfeld-Name	Beschreibung	IND	Stack	Flags	Bytes	Seite
SIZE?		Ruft die Anzahl der Hauptspeicher-Register ab, die gegenwärtig der Datenspeicherung zugewiesen sind.		↑, F		2	199
STOFLAG		Stellt den Status der Flags 00 bis 43 unter Verwendung der im X-Register **abgelegten Statusdaten** wieder her. Oder:		F	00–43	2	296
		Stellt den Status der Flags *bb* bis *ee* (spezifiziert **durch die Zahl *bb.ee* im X-Register)** unter Verwendung der im **Y-Register abgelegten Statusdaten** wieder her.		F	bb–ee	2	296
	USER	Schaltet das User-Tastenfeld ein und aus.		N	27		155

Löschfunktionen

Die Bedeutung der Eintragungen in dieser Tabelle entnehmen Sie bitte dem Absatz «Erläuterungen zum Funktionsverzeichnis» auf Seite 414.

Alpha-Name	Tastenfeld-Name	Beschreibung	Stack	Flags	Bytes	Seite
	←	Wenn das Eingabezeichen (_) angezeigt wird, löscht ← die zuletzt eingegebene Ziffer bzw. das zuletzt eingegebene Zeichen. Wenn die Zahlen- oder Zeicheneingabe beendet ist, löscht ← im Ausführungs-Modus das X- oder Alpha-Register und im Programm-Modus die angezeigte Programmzeile.	*			158 159 285
		Löscht eine angezeigte Meldung.	N	50		
	← gedrückt, ON, ← losgelassen	Löscht den gesamten Computer-Speicher mit Ausnahme von Uhrzeit Datum.		00–55		155
CLA	CLA	Löscht das Alpha-Register.	F		1	159

* Wenn durch Drücken von ← das X-Register gelöscht wird, bedingt dies eine Freigabe des Stack Lifts. In allen anderen Fällen verhält sich ← neutral.

Löschfunktionen (Fortsetzung)

Alpha-Name	Tastenfeld-Name	Beschreibung	Stack	Flags	Bytes	Seite
CLALMA		Löscht den ersten Alarm, dessen Meldung mit dem **Inhalt des Alpha-Registers** übereinstimmt.	F		2	258
CLALMX		Löscht den Alarm, dessen Stellung im Alarmkatalog (ALMCAT) durch **die Zahl im X-Register spezifiziert** ist.	F		2	259
CLD		Löscht eine Meldung aus der Anzeige.	F	50	1	318
CLFL		Löscht den Inhalt des **im Alpha-Register spezifizierten** Files.	F		2	213
CLKEYS		Löscht alle Tastenzuordnungen des User-Tastenfelds.	F		2	168
CLP *Label*		Löscht ein Programm aus dem Programmspeicher; benötigt ein globales Label des Programms als Parameter.	F			286
CLRALMS		Löscht alle gespeicherten Alarme.	F		2	258
CLRG		Löscht alle Datenspeicher-Register im Hauptspeicher.	F		1	202
CLRGX		Löscht jedes *ii*-te Hauptspeicher-Register zwischen R_{bbb} und R_{eee} **(spezifiziert durch die Zahl *bbb.eeeii* im X-Register).**	F		2	202
CLΣ	CLΣ	Löscht die Statistikregister.	F		1	190
CLST		Löscht den automatischen Speicherstack.	F		1	183
CLX	CL*x*	Löscht das X-Register.	G		1	159
DEL *nnn*		Löscht *nnn* Programmzeilen, beginnend mit der angezeigten Zeile.	N			286
DELCHR		Löscht die im **X-Register spezifizierte Anzahl** von Zeichen aus dem momentanen Record des momentanen Textfiles, ausgehend von der momentanen Stellung des Zeigers.	F		2	225
DELREC		Löscht den momentanen Record des momentanen Textfiles.	F		2	224
PCLPS		Löscht ein durch ein globales Label **im Alpha-Register spezifiziertes** Programm und alle nachfolgenden Programme aus dem Programmspeicher.	F		2	286
PURFL		Löscht den **im Alpha-Register spezifizierten** File.	F		2	208

Stack- und Datenregister-Funktionen

Diese Funktionen manipulieren die Stack- bzw. die Datenspeicher-Register oder benutzen ein Register als Parameter. (Funktionen, die Daten zwischen Datenfiles des erweiterten Speichers und dem Speicherstack bzw. Datenregistern des Hauptspeichers transferieren, finden Sie unter «Funktionen des erweiterten Speichers» auf Seite 425.) Die Bedeutung der Eintragungen in dieser Tabelle entnehmen Sie bitte dem Absatz «Erläuterungen zum Funktionsverzeichnis» auf Seite 414.

Alpha-Name	Tastenfeld-Name	Beschreibung	IND	Stack	Flags	Bytes	Seite
ASTO *nn*	ASTO *nn*	Kopiert die sechs links außen im Alpha-Register stehenden Zeichen nach R_{nn}.	I	F		2	200
ARCL *nn*	ARCL *nn*	Hängt den Inhalt von R_{nn} an den Inhalt des Alpha-Registers an.	I	F	28, 29, 36–41	2	200
CLRG		Löscht alle Datenspeicher-Register.		F		1	202
CLRGX		Löscht jedes *ii*-te Register zwischen R_{bbb} und R_{eee} **(spezifiziert durch die Zahl *bbb.eeeii* im X-Register).**		F		2	202
CLΣ	CLΣ	Löscht die Statistikregister.		F		1	190
CLST		Löscht den automatischen Speicherstack.		F		1	183
CLX	CL*x*	Löscht das X-Register.		G		1	159
DSE *nn*		Erniedrigt *iiiii* um *cc* und überspringt die nächste Programmzeile, wenn gilt: *iiiii-cc* $\leqslant$ *fff* (*iiiii.fffcc* in R_{nn}).	I	F		2	306
ENTER↑	ENTER↑	Kopiert die Zahl im X-Register ins Y-Register und hebt den Stack an.		↑, G		1	175
ISG *nn*	ISG *nn*	Erhöht *iiiii* um *cc* und überspringt die nächste Programmzeile, wenn gilt: *iiiii* + *cc* > *fff* (*iiiii.fffcc* in R_{nn}).	I	F		2	306
LASTX	LAST*x*	Ruft die Zahl im LAST X-Register zurück.		↑, F		1	179
PSIZE		Ordnet die **im X-Register spezifizierte Anzahl** von Hauptspeicher-Registern der Datenspeicherung zu.		F		2	199

Stack- und Datenregister-Funktionen (Fortsetzung)

Alpha-Name	Tastenfeld-Name	Beschreibung	IND	Stack	Flags	Bytes	Seite
R↑		Verschiebt den Stack zyklisch nach oben.		F		1	181
RCL *nn*	RCL *nn*	Ruft den Inhalt von R_{nn} zurück.	I	↑, F		*	200
RDN	R↓	Verschiebt den Stack zyklisch nach unten.		F		1	181
REGMOVE		Kopiert den Inhalt eines Registerblocks mit *nnn* Registern (beginnend mit R_{sss}) in einen anderen Registerblock mit *nnn* Registern (beginnend mit R_{ddd}). **Spezifiziert durch die Zahl *sss.dddnnn* im X-Register.**		F		2	201
REGSWAP		Vertauscht den Inhalt eines Registerblocks mit *nnn* Registern (beginnend mit R_{sss}) mit dem eines anderen Registerblocks mit *nnn* Registern (beginnend mit R_{ddd}). **Spezifiziert durch die Zahl *sss.dddnnn* im X-Register.**		F		2	201
Σ+	Σ+	Summationen in Statistikregistern.		L, G		1	191
Σ−	Σ−	Summationskorrektur.		L, G		1	191
ΣREG *nn*		Definiert die Register R_{nn} bis R_{nn+5} als Statistikregister.	I	F		2	190
ΣREG?		Ruft die Adresse des ersten Registers aus dem Block der gegenwärtig definierten Statistikregister ab.		↑, F		2	190
SIZE *nnn*		Ordnet *nnn* Hauptspeicher-Register der Datenspeicherung zu.		F			199
SIZE?		Ruft die Zahl der Hauptspeicher-Register ab, die gegenwärtig der Datenspeicherung zugewiesen sind.		↑, F		2	199
ST+ *nn*	STO + *nn*	Addiert die Zahl im X-Register zur Zahl in R_{nn} und speichert das Ergebnis in R_{nn}.	I	F		2	201

* Belegt 1 Byte für $00 \leqslant nn \leqslant 15$ und 2 Bytes in allen anderen Fällen.

Stack- und Datenregister-Funktionen (Fortsetzung)

Alpha-Name	Tastenfeld-Name	Beschreibung	IND	Stack	Flags	Bytes	Seite
ST- *nn*	STO − *nn*	Subtrahiert die Zahl im X-Register von der Zahl in R_{nn} und speichert das Ergebnis in R_{nn}.	I	F		2	201
ST* *nn*	STO × *nn*	Multipliziert die Zahl im X-Register mit der Zahl in R_{nn} und speichert das Ergebnis in R_{nn}.	I	F		2	201
ST/ *nn*	STO ÷ *nn*	Dividiert die Zahl in R_{nn} durch die Zahl im X-Register und speichert das Ergebnis in R_{nn}.	I	F		2	201
STO *nn*	STO *nn*	Kopiert den Inhalt des X-Registers in R_{nn}.	I	F		*	200
VIEW *nn*	VIEW *nn*	Zeigt den Inhalt von R_{nn} an.	I	F	21, 50, 55	2	319
X<> *nn*		Vertauscht den Inhalt des X-Registers mit dem Inhalt von R_{nn}.	I	F		2	201
X<>F		Vertauscht die Zahl im X-Register mit dem Status der Flags 00 bis 07.		F	00–07	2	295
X<>Y	$x \gtrless y$	Vertauscht den Inhalt des X-Registers mit dem Inhalt des Y-Registers.		F		1	181

* Belegt 1 Byte für $00 \leqslant nn \leqslant 15$ und 2 Bytes in allen anderen Fällen.

Numerische Funktionen

Alle numerischen Funktionen sind programmierbar und belegen 1 Byte des Programmspeichers. Die Bedeutung der Eintragungen in dieser Tabelle entnehmen Sie bitte dem Absatz «Erläuterungen zum Funktionsverzeichnis» auf Seite 414.

Alpha-Name	Tastenfeld-Name	Beschreibung	Stack	Seite
+	+	$y+x$.	L, ↓, F	188
-	-	$y-x$.	L, ↓, F	188
*	×	$y \times x$.	L, ↓, F	188
/	÷	y/x.	L, ↓, F	188
1/X	1/x	Kehrwert.	L, F	185
10↑X	10^x	Dekadische Exponentialfunktion.	L, F	187
ABS		$\|x\|$ (Betrag).	L, F	186
ACOS	COS^{-1}	Arcuscosinus.	L, F	186
ASIN	SIN^{-1}	Arcussinus.	L, F	186
ATAN	TAN^{-1}	Arcustangens.	L, F	186
CHS	CHS	Vorzeichenwechsel.	F	185
COS	COS	Cosinus.	L, F	186
D-R		Umwandlung von dezimalen Altgrad in Radiant (Bogenmaß).	L, F	187
DEC		Umwandlung von Oktal in Dezimal.	L, F	187
E↑X	e^x	Natürliche Exponentialfunktion.	L, F	187
E↑X-1		Natürliche Exponentialfunktion für Argumente nahe 0.	L, F	187
FACT		$x!$ (Fakultät).	L, F	185
FRC		Dezimalanteil einer Zahl.	L, F	186
HMS		Umwandlung von Dezimalstunden in Stunden, Minuten, Sekunden.	L, F	187
HMS+		Addition von Stunden, Minuten, Sekunden.	L, ↓, F	188
HMS-		Subtraktion von Stunden, Minuten, Sekunden.	L, ↓, F	188
HR		Umwandlung von Stunden, Minuten, Sekunden in Dezimalstunden.	L, F	187
INT		Ganzzahliger Anteil.	L, F	186
LN	LN	Natürlicher Logarithmus.	L, F	187
LN1+X		Natürlicher Logarithmus für Argumente nahe 1.	L, F	187

Numerische Funktionen (Fortsetzung)

Alpha-Name	Tastenfeld-Name	Beschreibung	Stack	Seite
LOG	LOG	Dekadischer Logarithmus.	L, F	187
MEAN		Mittelwert der aufsummierten *x*- und *y*-Werte.	L, F	192
MOD		Rest bei Division.	L, ↓, F	190
OCT		Umwandlung von Dezimal in Oktal.	L, F	187
P-R	P➡R	Umwandlung von Polarkoordinaten in kartesische Koordinaten.	L, F	189
%	%	*x* Prozent von *y*.	L, F	188
%CH		Prozentualer Unterschied zwischen *y* und *x*.	L, F	188
PI	π	Pi (3,141592654).	↑, F	159
R-D		Umwandlung vom Bogenmaß (Rad) in dezimale Altgrad.	L, F	187
R-P	R➡P	Umwandlung von kartesischen Koordinaten in Polarkoordinaten.	L, F	189
RND		Rundung.	L, F	186
SDEV		Standardabweichung der aufsummierten *x*- und *y*-Werte.	L, F	192
Σ+	Σ+	Summationen in Statistikregistern.	L, G	191
Σ−	Σ−	Summationskorrektur.	L, G	191
SIN	SIN	Sinus.	L, F	186
SIGN		Vorzeichen von *x*.	L, F	186
SQRT	$\sqrt{x}$	Quadratwurzel.	L, F	185
TAN	TAN	Tangens.	L, F	186
X↑2	x^2	Quadrat.	L, F	185
Y↑X	y^x	*y* hoch *x* (Potenz).	L, ↓, F	189

Funktionen des erweiterten Speichers

Alle Funktionen des erweiterten Speichers sind programmierbar und belegen 2 Bytes des Programmspeichers. Die Bedeutung der Eintragungen in dieser Tabelle entnehmen Sie bitte dem Absatz «Erläuterungen zum Funktionsverzeichnis» auf Seite 414. Die in der Tabelle enthaltenen Sonderinformationen werden nachfolgend erläutert.

Filetyp. Funktionen mit einem Eintrag in dieser Spalte wirken nur auf den angegebenen Filetyp.

P = Programmfile.

D = Datenfile.

A = ASCII-File (Textfile).

Filename. Funktionen mit einem Eintrag in dieser Spalte wirken nur auf einen File, der in der angegebenen Weise spezifiziert wird.

Ja = Sie müssen den Namen des gewünschten Files in das Alpha-Register eingeben.

OK = Sie können den Namen des gewünschten Files in das Alpha-Register eingeben oder Sie können das Alpha-Register löschen und damit den momentanen File spezifizieren.

Nein = Die Funktion wirkt nur auf den momentanen File.

Zeiger. Funktionen mit einem Eintrag in dieser Spalte wirken gemäß dem momentanen Wert des angegebenen Zeigers.

RRR = Register-Zeiger (für Datenfiles).

rrr.ccc = Record/Zeichen-Zeiger (für Textfiles).

rrr. = Record-Zeiger allein (für Textfiles).

Alpha-Name	File-typ	Beschreibung	File-name	Zeiger	Stack	Flags	Seite
APPCHR	A	Hängt den Inhalt des Alpha-Registers an das Ende des momentanen Records an.	Nein	*rrr.*	F		224
APPREC	A	Hängt den Inhalt des Alpha-Registers als neuen Record an das Ende des momentanen Files an.	Nein		F		222
ARCLREC	A	Hängt den momentanen Record (ab der Position des Zeigers) an den Inhalt des Alpha-Registers an.	Nein	*rrr.ccc*	F	17	226
ASROOM	A	Ruft die Anzahl der im momentanen File zur Verfügung stehenden Bytes ab.	Nein		↑, F		222

Funktionen des erweiterten Speichers (Fortsetzung)

Alpha-Name	Filetyp	Beschreibung	Filename	Zeiger	Stack	Flags	Seite
CLFL	A, D	Löscht den Inhalt des **im Alpha-Register spezifizierten** Files.	Ja		F		213
CRFLAS	A	Erzeugt einen Textfile **(Name im Alpha-Register spezifiziert)** mit der im **X-Register spezifizierten Anzahl** von Registern.	Ja		F		212
CRFLD	D	Erzeugt einen Datenfile **(Name im Alpha-Register spezifiziert)** mit der im **X-Register spezifizierten Anzahl** von Registern.	Ja		F		211
DELCHR	A	Löscht die im **X-Register spezifizierte Anzahl** von Zeichen aus einem Record. Die Löschoperation beginnt bei der Zeigerposition.	Nein	*rrr.ccc*	F		225
DELREC	A	Löscht den momentanen Record.	Nein	*rrr.*	F		224
ED	A	Schaltet Texteditor-Tastenfeld und Anzeige ein.	OK	*rrr.ccc*	F	26, 28, 48	228
EMDIR		Listet den Katalog des erweiterten Speichers. Drücken Sie R/S und dann ←, um die Katalogausgabe zu beenden und den angezeigten File zum momentanen File zu machen.			F		206
		Wenn die Auflistung zu Ende geführt wird, zeigt EMDIR die Anzahl der verfügbaren Register des erweiterten Speichers an.			↑, F		
EMDIRX		Findet den *n*-ten von EMDIR gelisteten File **(*n* im X-Register spezifiziert);** legt den Filenamen im Alpha-Register und den Filetyp im X-Register ab.			L, F		207
EMROOM		Ruft die Anzahl der für einen neuen File verfügbaren Register ab.			↑, F		208
FLSIZE	A, D, P	Ruft die Registeranzahl des Files ab.	OK		↑, F		208
GETAS	A	Kopiert den **im Alpha-Register spezifizierten** File aus dem Massenspeicher in den erweiterten Speicher.	Ja		F		227

Funktionen des erweiterten Speichers (Fortsetzung)

Alpha-Name	File-typ	Beschreibung	File-name	Zeiger	Stack	Flags	Seite
GETP	P	Ersetzt das letzte Programm im Programmspeicher durch den **im Alpha-Register spezifizierten** File.	Ja		F	27	209
GETR	D	Kopiert die Register des angegebenen Files in den Hauptspeicher.	OK		F		217
GETREC	A	Kopiert einen Record (ab der Zeigerposition) in das Alpha-Register.	Nein	*rrr.ccc*	F	17	226
GETRX	D	Kopiert die Fileregister (ab der Zeigerposition) in R_{bbb} bis R_{eee} **(spezifiziert durch *bbb.eee* im X-Register).**	Nein	*RRR*	F		218
GETSUB	P	Kopiert den **im Alpha-Register spezifizierten** File an das Ende des Programmspeichers.	Ja		F	27	209
GETX	D	Kopiert das momentane Register in das X-Register.	Nein	*RRR*	↑, F		220
INSCHR	A	Fügt den Inhalt des Alpha-Registers ein (beginnend bei der Zeigerposition).	Nein	*rrr.ccc*	F		224
INSREC	A	Fügt den Inhalt des Alpha-Registers als neuen Record an der Zeigerposition ein.	Nein	*rrr.*	F		222
POSFL	A	Durchsucht den File nach dem **im Alpha-Register abgelegten** String und legt, wenn der String gefunden wird, den Zeigerwert im X-Register ab (–1, wenn der String nicht gefunden wird).	Nein	*rrr.ccc*	↑, F		226
PURFL	A, D, P	Löscht den **im Alpha-Register spezifizierten** File.	Ja		F		208
RCLPT	A, D	Ruft die Zeigerposition im momentanen File ab.	Nein	*rrr.ccc* *RRR*	↑, F		216
	P	Ruft die Anzahl der vom momentanen Programmfile belegten Bytes ab.	Nein		↑, F		211

Funktionen des erweiterten Speichers (Fortsetzung)

Alpha-Name	File-typ	Beschreibung	File-name	Zeiger	Stack	Flags	Seite
RCLPTA	A, D	Ruft die Zeigerposition ab.	OK	*rrr.ccc RRR*	↑, F		216
	P	Ruft die Anzahl der Bytes des spezifizierten Programms ab.	OK		↑, F		211
RESZFL	A, D	Vergrößert oder verkleinert den momentanen File auf die **im X-Register spezifizierte Anzahl** von Registern.	Nein		F		213
SAVEAS	A	Kopiert den **im Alpha-Register spezifizierten** File des erweiterten Speichers in den **im Alpha-Register spezifizierten** Massenspeicherfile. (Filenamen im Alpha-Register durch ein Komma getrennt.)	Ja		F		227
SAVEP	P	Kopiert das **im Alpha-Register spezifizierte** Programm in den **im Alpha-Register spezifizierten** File. (Programm- und Filename im Alpha-Register durch ein Komma getrennt.)	Ja		F		208
SAVER	D	Kopiert alle Register des Hauptspeichers in den angegebenen File.	OK		F		217
SAVERX	D	Kopiert den Inhalt der Hauptspeicher-Register R_{bbb} bis R_{eee} **(spezifiziert durch *bbb.eee* im X-Register)** in den angegebenen File. Die Operation beginnt bei dem momentanen Fileregister.	Nein	*RRR*	F		218
SAVEX	D	Kopiert die Zahl im X-Register in das momentane Fileregister.	Nein	*RRR*	F		220
SEEKPT	A, D	Stellt den Zeiger des momentanen Files auf die im X-Register spezifizierte Position.	Nein		F		215
SEEKPTA	A, D	Stellt den Zeiger auf die im X-Register spezifizierte Position.	OK		F		215

Zeitfunktionen

Die Bedeutung der Eintragungen in dieser Tabelle entnehmen Sie bitte dem Absatz «Erläuterungen zum Funktionsverzeichnis» auf Seite 414.

Alpha-Name	Beschreibung	Stack	Flags	Bytes	Seite
ADATE	Hängt die Zahl im X-Register im momentanen Datumformat an den Inhalt des Alpha-Registers an.	F	31, 36–39	2	243
ALMCAT	Listet alle Alarme in chronologischer Reihenfolge auf. Kann durch die Sequenz CATALOG 5 auch über das Tastenfeld ausgeführt werden.	F	31	2	255
ALMNOV	Aktiviert den am weitesten zurückliegenden überfälligen Steueralarm oder bedingten Alarm.	F		2	261
ATIME	Hängt die Zahl im X-Register im momentanen Zeitformat an den Inhalt des Alpha-Registers an.	F	36–39	2	240
ATIME24	Hängt die Zahl im X-Register im CLK24 Format an den Inhalt des Alpha-Registers an.	F	36–39	2	241
CLALMA	Löscht den ersten Alarm, dessen Meldung mit dem Inhalt des Alpha-Registers übereinstimmt.	F		2	258
CLALMX	Löscht den Alarm, dessen Stellung im Alarmkatalog (ALMCAT) durch **die Zahl im X-Register spezifiziert** ist.	F		2	259
CLK12	Schaltet die Anzeige der Uhr auf das 12-Stunden-Format.	F		2	239
CLK24	Schaltet die Anzeige der Uhr auf das 24-Stunden-Format.	F		2	239
CLKT	Schaltet die Anzeige der Uhr auf das Nur-Uhrzeit-Format.	F		2	239
CLKTD	Schaltet die Anzeige der Uhr auf das Uhrzeit/Datum-Format.	F		2	239
CLOCK	Zeigt die Uhr an.	F	31	2	238
	Kann auch durch die Tastenfolge ■ ON ausgeführt werden; dabei wird der Computer aus- und eingeschaltet und dann die Uhr angezeigt.	F	12–26, 44–55		
CLRALMS	Löscht alle Alarme.	F		2	258
CORRECT	Stellt die Uhrzeit und justiert den Kompensationsfaktor.	F		2	238
DATE	Zeigt das Datum an und legt eine das aktuelle Datum repräsentierende Zahl im X-Register ab.	↑, F	31	2	242
DATE+	Berechnet das neue Datum aus dem im X-Register spezifizierten Datum und der im Y-Register spezifizierten Anzahl von Tagen. Wenn die Anzahl von Tagen positiv ist, liegt das neue Datum später; wenn die Anzahl negativ ist, liegt das neue Datum früher.	L, ↓, F	31	2	244

Zeitfunktionen (Fortsetzung)

Alpha-Name	Beschreibung	Stack	Flags	Bytes	Seite
DDAYS	Berechnet die Differenz in Tagen zwischen dem Datum im X-Register und dem Datum im Y-Register. Die Differenz ist positiv, wenn das Datum im Y-Register früher liegt und negativ, wenn das Datum im Y-Register später liegt.	L, ↓, F	31	2	244
DMY	Schaltet die Anzeige des Datums auf das Tag-Monat-Jahr-Format.	F	31	2	242
DOW	Ersetzt die im X-Register abgelegte, das Datum darstellende Zahl durch eine den Wochentag repräsentierende Zahl. (0 = Sonntag, 6 = Samstag.)	L, F	31	2	244
HMS	Wandelt die Zahl im X-Register von einem Dezimalstunden-Format in ein Stunden-Minuten-Sekunden-Format um.	L, F		1	187
HMS+	Addiert die Zahl im X-Register zu der Zahl im Y-Register (im Stunden-Minuten-Sekunden-Format).	L, ↓, F		1	188
HMS−	Subtrahiert die Zahl im X-Register von der Zahl im Y-Register (im Stunden-Minuten-Sekunden-Format).	L, ↓, F		1	188
HR	Wandelt die Zahl im X-Register von einem Stunden-Minuten-Sekunden-Format in ein Dezimalstunden-Format um.	L, F		1	187
MDY	Schaltet die Anzeige des Datums auf das Monat-Tag-Format.	F	31	2	242
RCLAF	Ruft den Kompensationsfaktor der Uhr ab.	↑, F		2	376
RCLALM	Ruft die XYZALM Parameter des an *n*-ter Stelle im Alarmkatalog ALMCAT stehenden Alarms ab (***n* im X-Register spezifiziert**).	L, ↑, F *	31	2	252
RCLSW	Ruft die Stoppuhrzeit ab.	↑, F		2	273
RUNSW	Startet die Stoppuhr.	F		2	273
SETAF	Stellt den Kompensationsfaktor der Uhr ein.	F		2	376
SETDATE	Stellt das Datum auf den **im X-Register spezifizierten Wert.**	F	31	2	242
SETIME	Stellt die Uhrzeit auf den **im X-Register spezifizierten Wert.**	F		2	237
SETSW	Stellt die Stoppuhr auf die **im X-Register spezifizierte Startzeit.**	F		2	273
STOPSW	Hält die laufende Stoppuhr an.	F		2	273
SW	Schaltet Stoppuhr-Tastenfeld und Anzeige ein.	F	26	2	266

* Kopiert den Inhalt des Y-Registers in das T-Register, unabhängig davon, ob der Stack Lift freigegeben ist.

Zeitfunktionen (Fortsetzung)

Alpha-Name	Beschreibung	Stack	Flags	Bytes	Seite
SWPT	Stellt die Speicher und Rückruf-Zeiger der Stoppuhr auf die gewünschte Position ein **(spezifiziert durch *sss.rrr* im X-Register)**. Schaltet Stoppuhr-Tastenfeld und Anzeige ein; legt die momentane Zeigerpositionen im X-Register ab, wenn das Stoppuhr-Tastenfeld ausgeschaltet wird.	L, F	26	2	274
T+X	Korrigiert die Uhrzeit unter Verwendung der im **X-Register spezifizierten Zeitänderung.**	F		2	238
TIME	Legt eine die aktuelle Uhrzeit repräsentierende Zahl im X-Register ab.	↑, F		2	240
XYZALM	Stellt einen Alarm ein, dessen Parameter folgendermaßen zu spezifizieren sind: **Zeit im X-Register, Datum im Y-Register, Wiederholungsintervall im Z-Register, Meldung oder globales Label im Alpha-Register.**	F	31	2	250

Editierungs-Funktionen

Diese nicht programmierbaren Funktionen werden im Programm-Modus ausgeführt und dienen der Eingabe und Modifikation von Programmen. Die Editierungs-Funktionen erfordern wie die Modustasten keine Umschaltung in den Ausführungs-Modus zur Funktionsausführung. Die Bedeutung der Eintragungen in dieser Tabelle entnehmen Sie bitte dem Absatz «Erläuterungen zum Funktionsverzeichnis» auf Seite 414.

Alpha-Name	Tastenfeld-Name	Beschreibung	Flags	Seite
	←	← löscht die zuletzt eingegebene Ziffer bzw. das zuletzt eingegebene Zeichen, wenn das Eingabezeichen (_) angezeigt wird. Bei fehlendem Eingabezeichen löscht ← die angezeigte Programmzeile.		285
ASN	ASN	Ordnet eine zu spezifizierende Funktion (oder ein globales Label) einer ebenfalls anzugebenden Taste des User-Tastenfelds zu.		166
BST	BST	Zeigt die vorhergehende Programmzeile an.		284
CAT *n*	CATALOG *n*	Startet die Katalogausgabe ($1 \leqslant n \leqslant 6$).		170
CLP *Label*		Löscht das Programm mit dem spezifizierten globalen Label aus dem Programmspeicher.		286
COPY *Label*		Kopiert das ROM-Programm mit dem spezifizierten globalen Label in den Programmspeicher.		281

Editierungs-Funktionen (Fortsetzung)

Alpha-Name	Tastenfeld-Name	Beschreibung	Flags	Seite
DEL *nn*		Löscht *nnn* Programmzeilen, beginnend mit der angezeigten Zeile.		286
	GTO ·	Springt zu einer spezifizierten Programmzeile bzw. zu einem globalen Label.		283
	GTO · ·	Springt an das Ende des Programmspeichers, packt den Programmspeicher und erzeugt ein Null-Programm.		281
ON		Desaktiviert die Abschaltautomatik (der Computer bleibt eingeschaltet).	44	155
PACK		Packt den Programmspeicher.		198
SIZE *nnn*		Ordnet *nnn* Hauptspeicher-Register der Datenspeicherung zu.		199
SST	SST	Zeigt die nächste Programmzeile an.		284

Funktionen zur Steuerung der Programmausführung

Die folgenden Funktionen dienen dazu, die Programmausführung anzuhalten oder Programmzeilen in anderer Reihenfolge als aufgezeichnet auszuführen. Die Bedeutung der Eintragungen in dieser Tabelle entnehmen Sie bitte dem Absatz «Erläuterungen zum Funktionsverzeichnis» auf Seite 414.

Alpha-Name	Tastenfeld-Name	Beschreibung	IND	Stack	Flags	Bytes	Seite
AVIEW	AVIEW	Zeigt den Inhalt des Alpha-Registers an; stoppt die Programmausführung, wenn Flag 21 gesetzt und Flag 55 gelöscht ist.		F	21, 50, 55	1	318
CLOCK		Stoppt die Programmausführung und zeigt die Uhr an.		F	31	2	238
DSE *nn*		Erniedrigt *iiiii* um *cc* und überspringt die nächste Programmzeile, wenn gilt: *iiiii-cc* $\leqslant$ *fff* (*iiiii.fffcc* in R_{nn}).	I	F		2	306
END		Markiert das Programmende.		F		3	301
FC? *nn*		Fragt Flag *nn* ab und überspringt die nächste Programmzeile, wenn Flag *nn* gesetzt ist ($00 \leqslant nn \leqslant 55$).	I	F	*nn*	2	304

Funktionen zur Steuerung der Programmausführung (Fortsetzung)

Alpha-Name	Tastenfeld-Name	Beschreibung	IND	Stack	Flags	Bytes	Seite
FC?C *nn*		Fragt Flag *nn* ab und löscht den Flag nach der Abfrage; überspringt die nächste Programmzeile, wenn Flag *nn* gesetzt ist ($00 \leqslant nn \leqslant 29$).	I	F	*nn*	2	304
FS? *nn*	FS? *nn*	Fragt Flag *nn* ab und überspringt die nächste Programmezeile, wenn Flag *nn* gelöscht ist ($00 \leqslant nn \leqslant 55$).	I	F	*nn*	2	304
FS?C *nn*		Fragt Flag *nn* ab und löscht den Flag nach der Abfrage; überringt die nächste Programmzeile, wenn Flag *nn* gelöscht ist ($00 \leqslant nn \leqslant 29$).	I	F	*nn*	2	304
GETP		Ersetzt das letzte Programm im Speicher durch den im Alpha-Register spezifizierten Programmfile. Wenn das letzte Programm das neue aufruft (und dadurch ersetzt wird), beginnt die Ausführung mit der ersten Zeile des neuen Programms.		F	27	2	209
GTO *Label*	GTO *Label*	Setzt die Programmausführung mit dem spezifizierten globalen, numerischen oder lokalen Alpha-Label fort.	I	F		*	300
ISG *nn*	ISG *nn*	Erhöht *iiiii* um *cc* und überspringt die nächste Programmzeile, wenn gilt: *iiiii* + *cc* > *fff* (*iiiii.fffcc* in R_{nn}).	I	F		2	306
LBL	LBL	Globales, numerisches oder lokales Alpha-Label.		F		**	299
OFF		Schaltet den Computer aus.		N	11–26 44–55	1	292
PROMPT		Zeigt den Inhalt des Alpha-Registers an und stoppt die Programmausführung.		F	50	1	314

* Belegt 2 Bytes, wenn der Parameter indirekt spezifiziert wird oder wenn $00 \leqslant nn \leqslant 14$; belegt $2 + m$ Bytes, wenn der Parameter ein globales Label mit *m* Zeichen ist; belegt 3 Bytes in allen anderen Fällen.

** Belegt 1 Byte, wenn $00 \leqslant nn \leqslant 14$; belegt $4 + m$ Bytes, wenn der Parameter ein globales Label mit *m* Zeichen ist; belegt 2 Bytes in allen anderen Fällen.

Funktionen zur Steuerung der Programmausführung (Fortsetzung)

Alpha-Name	Tastenfeld-Name	Beschreibung	IND	Stack	Flags	Bytes	Seite
RTN	RTN	Gibt die Programmausführung an die Zeile nach der XEQ Anweisung, mit der das Unterprogramm aufgerufen wurde, zurück.		F		1	301
STOP	R/S	Stoppt die Programmausführung.		F		1	302
VIEW *nn*	VIEW *nn*	Zeigt den Inhalt von R_{nn} an und stoppt die Programmausführung, wenn Flag 21 gesetzt und Flag 55 gelöscht ist.	I	F	21, 50, 55	2	319
X=0?	x=0?	Überspringt die nächste Anweisung, wenn die Zahl im X-Register ungleich 0 ist.		F		1	304
X≠0?		Überspringt die nächste Anweisung, wenn die Zahl im X-Register gleich 0 ist.		F		1	304
X<0?		Überspringt die nächste Anweisung, wenn die Zahl im X-Register größer oder gleich 0 ist.		F		1	304
X<=0?		Überspringt die nächste Anweisung, wenn die Zahl im X-Register größer 0 ist.		F		1	304
X>0?		Überspringt die nächste Anweisung, wenn die Zahl im X-Register kleiner oder gleich 0 ist.		F		1	304
X=Y?	x=y?	Überspringt die nächste Anweisung, wenn der Inhalt des X-Registers ungleich dem Inhalt des Y-Registers ist.		F		1	304
X≠Y?		Überspringt die nächste Anweisung, wenn der Inhalt des X-Registers gleich dem Inhalt des Y-Registers ist.		F		1	304
X<Y?		Überspringt die nächste Anweisung, wenn die Zahl im X-Register größer oder gleich der Zahl im Y-Register ist.		F		1	304
X<=Y?	$x \leq y$?	Überspringt die nächste Anweisung, wenn die Zahl im X-Register größer als die Zahl im Y-Register ist.		F		1	304
X>Y?	$x > y$?	Überspringt die nächste Anweisung, wenn die Zahl im X-Register kleiner oder gleich der Zahl im Y-Register ist.		F		1	304

Funktionen zur Steuerung der Programmausführung (Fortsetzung)

Alpha-Name	Tastenfeld-Name	Beschreibung	IND	Stack	Flags	Bytes	Seite
X=NN?		Überspringt die nächste Anweisung, wenn der Inhalt des X-Registers ungleich dem Inhalt von R_{nn} (***nn* im Y-Register spezifiziert**) ist.		F		2	305
X≠NN?		Überspringt die nächste Anweisung, wenn der Inhalt des X-Registers gleich dem Inhalt von R_{nn} (***nn* im Y-Register spezifiziert**) ist.		F		2	305
X<NN?		Überspringt die nächste Anweisung, wenn der Inhalt des X-Registers größer oder gleich dem Inhalt von R_{nn} (***nn* im Y-Register spezifiziert**) ist.		F		2	305
X<=NN?		Überspringt die nächste Anweisung, wenn der Inhalt des X-Registers größer als der Inhalt von R_{nn} (***nn* im Y-Register spezifiziert**) ist.		F		2	305
X>NN?		Überspringt die nächste Anweisung, wenn der Inhalt des X-Registers kleiner oder gleich von R_{nn} (***nn* im Y-Register spezifiziert**) ist.		F		2	305
X>=NN?		Überspringt die nächste Anweisung, wenn der Inhalt des X-Registers kleiner als der Inhalt von R_{nn} (***nn* im Y-Register spezifiziert**) ist.		F		2	305
XEQ *Label*	XEQ *Label*	Ruft das spezifizierte globale, numerische oder lokale Alpha-Label als Unterprogramm auf. (Wenn Sie eine Funktion spezifizieren, schlagen Sie bitte den Tabelleneintrag für die betreffende Funktion nach.)	I	F		*	301

* Belegt 2 Bytes, wenn das Label indirekt spezifiziert wird; belegt 3 Bytes, wenn ein lokales Label spezifiziert wird; belegt $2+m$ Bytes, wenn ein globales Label mit m Zeichen spezifiziert wird.

Alpha-Funktionen

Mit den Alpha-Funktionen können Daten im Alpha-Register abgelegt und wieder abgerufen werden. Im Alpha-Register abgelegte Daten können mit Hilfe dieser Funktionen modifiziert werden. Funktionen, die die Spezifikation eines Filenamens im Alpha-Register erfordern, werden in der Tabelle nicht berücksichtigt. Die Bedeutung der Eintragungen in dieser Tabelle entnehmen Sie bitte dem Absatz «Erläuterungen zum Funktionsverzeichnis» auf Seite 414.

Alpha-Name	Tastenfeld-Name	Beschreibung	IND	Stack	Flags	Bytes	Seite
	[⊢]	Hängt die nachfolgend einzugebenden Zeichen an den Inhalt des Alpha-Registers an.		N			159
[ADATE]		Hängt die Zahl im X-Register im momentanen Datumsformat an den Inhalt des Alpha-Register an.		F	31, 36–39	2	243
[ALENG]		Ruft die Anzahl der im Alpha-Register abgelegten Zeichen ab.		↑, F		2	313
[ANUM]		Ruft den ersten im Alpha-Register abgelegten Ziffernstring ab.		↑, F	22, 28, 29	2	311
[AOFF]		Schaltet das Alpha-Tastenfeld aus.		F	48	1	159
[AON]		Schaltet das Alpha-Tastenfeld ein.		F	48	1	159
[ARCL] *nn*	[ARCL] *nn*	Hängt den Inhalt von R_{nn} an den Inhalt des Alpha-Registers an.	I	F	28, 29, 36–41	2	200
[ARCLREC]		Hängt einen Record (ab der Position des Zeigers) an den Inhalt des Alpha-Registers an.		F	17	2	226
[AROT]		Rundumverschiebung des Inhalts des Alpha-Registers um *n* Stellen (***n* im X-Register spezifiziert**). Verschiebung nach links für positives *n*, nach rechts für negatives *n*.		F		2	313
[ASHF]		Schiebt die sechs links außen stehenden Zeichen aus dem Alpha-Register.		F		1	200
[ASTO] *nn*	[ASTO] nn	Kopiert die sechs links außen im Alpha-Register stehenden Zeichen nach R_{nn}.	I	F		2	200
[ATIME]		Hängt die Zahl im X-Register im momentanen Zeitformat an den Inhalt des Alpha-Registers an.		F	36–39	2	240

Alpha-Funktionen (Fortsetzung)

Alpha-Name	Tastenfeld-Name	Beschreibung	IND	Stack	Flags	Bytes	Seite
ATIME24		Hängt die Zahl im X-Register im CLK24 Format an den Inhalt des Alpha-Registers an.		F	36–39	2	241
ATOX		Schiebt das links außen stehende Zeichen aus dem Alpha-Register und legt den zugehörigen Zeichencode im X-Register ab.		↑, F		2	311
AVIEW	AVIEW	Zeigt den Inhalt des Alpha-Registers an.		F	21, 50, 55	1	318
CLA	CLA	Löscht das Alpha-Register.		F		1	159
GETREC		Kopiert einen Record (ab der Position des Zeigers) in das Alpha-Register.		F	17	2	226
POSA		Durchsucht das Alpha-Register nach dem oder den **im X-Registers spezifizierten** Zeichen. Legt die Position des ersten gefundenen Zeichens im X-Register ab (–1, wenn die gesuchten Zeichen nicht gefunden werden).		L, F		2	312
PROMPT		Zeigt den Inhalt des Alpha-Registers an und stoppt die Programmausführung.		F	21, 50, 55	1	314
XTOA		Hängt das durch den Zeichencode im X-Register bestimmte Zeichen an den Inhalt des Alpha-Registers an.		F		2	309

Interaktive Funktionen

Die Bedeutung der Eintragungen in dieser Tabelle entnehmen Sie bitte dem Absatz «Erläuterungen zum Funktionsverzeichnis» auf Seite 414.

Alpha-Name	Tastenfeld-Name	Beschreibung	IND	Stack	Flags	Bytes	Seite
ADV		Papiervorschub (bei angeschlossenem Drucker).		F	21, 55	1	368
BEEP	BEEP	Erzeugt vier Tonsignale.		F	26	1	319
GETKEY		Wartet bis zu 10 Sekunden darauf, daß eine Taste gedrückt wird. Legt den Tastencode im X-Register ab (0, wenn keine Taste gedrückt wird).		↑, F		2	317
GETKEYX		Wartet bis zu *SS.s* Sekunden darauf, daß eine Taste gedrückt wird (Zeitspanne **spezifiziert durch *SS.s* im X-Register**). Legt den Tastencode im Y-Register und den ASCII-Code im X-Register ab.		L, ↑, E *	28, 48	2	317
PROMPT		Zeigt den Inhalt des Alpha-Registers an und stoppt die Programmausführung.		F	50	1	314
PSE		Unterbricht die Programmausführung für ungefähr eine Sekunde.		F		1	315
TONE *n*		Erzeugt ein Tonsignal, $0 \leqslant n \leqslant 9$.	I	F	26	2	319

* Kopiert den Inhalt des Y-Registers in das Z-Register und den Inhalt des Z-Registers in das T-Register. Dies geschieht unabhängig davon, ob der Stack Lift freigegeben ist oder nicht.

Sach- und Funktionsindex

Sachindex

Fettgesetzte Seitenzahlen stehen für erste Referenzen; Seitenzahlen Normaldruck verweisen auf zweite Referenzen. *Kursiv* gesetzte Seiten beziehen sich auf Band I; nicht kursiv gesetzte Seitenzahlen weisen auf Band II hin.

A

Abschaltautomatik, Desaktivieren der, 155
Addition, ***51***
Adressierung, *37*, 162
  indirekte **162**
Alarm
  Aktivierung, **253**
  Anzeige, **253**
  Arten, 247, **249**
  Bedingung, **361**
  Beispiel, ***69***
  Datum, ***68***, **250**
  Ebenen, **363**
  Einstellung, Programm zur, **261**, 263
  Erinnerung, **260**
  Katalog. *Siehe* Katalog 5
  Katalog-Tastenfeld, **256–257**
  Katalog-Tastenfeld, ***71–73***
  Meldung, ***68***, **247**, **251**
  Nummer, 252, 255, **258**
  Signal, **253**
  Wiederholungsintervall, ***68***, **250**
  Zeit, ***68***, **251**
Alarme,
    Anhalten ***69***, **258**
  automatische Aktivierung von überfälligen, **260**
  Bestätigen, ***69***, 247, 248, 253, **254**
  Einstellen, **250–251**
  gleichzeitige, **255**
  Löschen, ***69***, **258**, 259, 261
  Melde-, **247**
  nicht aktivierte, 260
  Speicheranforderungen, **198**
  überfällige. *Siehe* Überfällige Alarme
  übergangene, **260**
  Zurückrufen, **252**
ALMEIN, Programm, **261**
ALMREL, Programm, 263
ALPHA, **15**, 230
Alpha-Anzeige, ***27***, **161**
  Verschieben der, ***27***
  von Nullzeichen, **366**
Alpha-Ausführung, ***44–45***, **282**
Alpha-Eingabe, **159**
Alpha-Labels, **169**
Alpha-Namen, **44**
  bei der Programmeingabe, **282**
Alpha-Parameterspezifikation, **162**
Alpha-Register, ***27***, 158, 206, 222, 226, 312
  Anhängen an das, 226
  Anhängen eines Datums, **243**
  Anhängen eines Zeitwerts, **240–241**
  Anzeige in einem Programm, ***94***
  Anzeige, **318**
  Kapazität, *27*, **159**
  Kopieren in das, 226
  Kopieren nach X, **309**
  Löschen, **160**
  Manipulieren von Daten im, 308
  Meldealarme und das, ***68***
  Nullzeichen im, **366**
  Suche nach einem String, **312–313**
  Suche nach Ziffern, **309**, **311**
  und Alarme, 250, **251**
  Verschieben, 200, **313**
  Zurückrufen, ***96***
Alpha-Strings, ***26***, *86*, *117*, **159–160**, 197, 213, 309, 366
  Alarm, 258
  aus Ziffern, **311**
  Berechnung der Länge von, **313**
  Eingabe, *27*
  in Programmen, ***93***, **282**
  Manipulation, 402
  mit Nullen, **37**
  Suche nach, **226**
  Vergleichen von, **305**
Alpha-Tastenfeld, *14*, *15*, ***24***, 155–156, **157**, 159, 230
  Flag, **291**
  im Ausführungsmodus, **159**
  im Programm-Modus, **159**
Alpha-Zeichen
  anzeigbare, **309**
  Eingabe, ***26***, *27*
  in Programmen, ***94***
  Kopieren, **200**
  Löschen, ***26***
  Sonder-, **309**
  Umwandlung in Zahlen, **309**, **311**
Alpha-Zeichenstring, **311**
Alpha-Ziffern, ***26***, **200**, 309
  Berechnungen mit, **200**
  Suche nach, 309, **311**
Alphanumerischer Zeichensatz, *14*, *24*, 232
Anhängen von Zeichen, 159, 367
Anwendungssoftware, **393**

Anzeige. *Siehe auch* Löschen; Meldung; Parameterfunktion, Anzeige einer; Programm; Verschieben der Anzeige
einer Meldung, **161**
einer Tastenbelegung, **169**
Formate, ***31***
Formatflags, **291**
Löschen der, ***19***, *20*
Standard-, **161**
von Nullzeichen, **366**
von Zeichen, **310**
von Zifferntrennzeichen, **290**
Applikationsmodule, Programme, **107–108**, 281
Arcuscosinus, **54**
Arcussinus, ***54***
Arcustangens, ***54***
Arithmetik, *16*, *21*, ***50–51***, **188**. *Siehe auch* Berechnungen; Nicht kommutative Operationen
in Datenspeicher-Register, ***40–42***, **201**
mit Vektoren, ***59***
mit Zeitwerten, ***65***, **188**
ASCII-Files, *113*. *Siehe auch* Textfiles
ASCII-Zeichen, **310**
Aufheben von Funktionen, **169**
Aus/Uhr Bedingung, **361**
Ausführen von Funktionen, ***17***, *44–45*
Ausführung. *Siehe* Momentanes Programm; Programm; Unterprogramm
Ausführungsmodus, *15*, *83*, 155, 282
Automatischer Speicherstack. *Siehe* Stack

B

Basisumwandlung, **187**
BAT, ***34***, **160**, 230, **383**
Batterien, **381**, 382
Einsetzen, **384**
empfohlene, **384**
Lebensdauer, **381**
Bedingte Abfrage, *104*
Bedingte Funktionen, **303**
erweiterte, 402
für Schleifen, **306**
Bedingter Alarm, 251, **248**, 260, 261
Beenden der Eingabe, *17*, ***18***
Belegen von Tasten mit Funktionen, ***46***, 156, **166**
Berechnungen, *16*, *21*, **175**, **176**. *Siehe auch* Konstanten, Berechnungen mit
im Stack, **180**, **182**
mit Datumswerten, ***66***
mit verschachtelten Ausdrücken, 176
mit Zeitwerten, ***64***
nicht kommutative, ***22***, **176**, 180, 181
Betrag, **186**
Bogenmaß, ***53***, **186**
Modus, ***53***, **186**
Umwandlung, ***53***, **187**
Bytebelegung
durch Labels, 299
durch Programmverzweigungen, **300**
durch Programmzeilen, **197**
durch Unterprogramme, **303**
Bytes
als Alpha-Zeichen, **309**
als Flagstatus, **294**
als Zahlen, **294**, **309**
im Alpha-Register, 309
in einem Programm, **211**
Null, 309
verfügbar in Textfile, **222**
zur Kennzeichnung des File-Endes, **212**
Bytezähler, *98*

C

Cosinus, ***54***
Countdown Timer, **274**
Cursor, ***118***, **229**, 230, **232**
Cursorkontrolle, ***120***, **232**

D

Daten. *Siehe auch* Datenfiles
Ausgabe, ***92***, *95*, ***96***
Austausch zwischen Registern und einem File, **218**
Eingabe, ***92***
Eingabeflags, **290**
Fileregister, 213
Filezeiger, 220. *Siehe auch* Filezeiger; Registerzeiger
Manipulation, 402
Speicherregister. *Siehe* Register
Datenfiles, ***123–124***, 205, 215, 216
Erzeugen von, **211**
Kopieren in, ***123***, **217**, **218**, **220**
Löschen von, **213**
Namen von, 211
Zurückrufen aus, ***123–124***, **218**, **219**, **220**
Datumsformat, **242**
Formatflag, **291**
Datumswerte
Addition, ***66***, **244**
Einstellen, ***62***, **242**
gültige, **242**, 243, **245**
Zeitraum zwischen zwei, ***67***, **244**
Zurückrufen, ***66***, **242**
DEG, ***53***, **186**
Dezimal-Oktal Umwandlung, **187**
Dezimalgrad, ***53***, *65*, **187**
Dezimalpunkt, 161

Dezimaltrennzeichen, ***53***, 160, **161**, 290
Differenz gestoppter Zeiten, ***78–79***, 270
  Anzeige der, **272**
  Beispiel, ***79***
  Delta-Stoppmodus, **270**, 271, **272**
  Speichern und Zurückrufen, **272**
Division, ***51***
Drift, Zeit, **375**
Drucker
  Anwesenheitsflag, **292**
  Ausgaben mit Uhrzeit und Datum, **369**
  Papiervorschub, **368**
  Steuerflag, **289**
  und Programmablauf, **368**

E

Ein- und Ausschalten, *14*, 155
Einfügen von Programmzeilen, ***102***, **286**
Einfügungsmodus (Texteditor), ***121***, 230, **232**
Eingabeaufforderungen, ***95***, **314**
Eingabezeichen, ***18***, ***30***, *158*
Eingebettete Nullen, 311, **366**
END Anweisung, ***89***, *98*, **281**
  Sprung zu einer, 284
Ersetzungsmodus (Texteditor), **121**, 230, **232**
Erstfunktionen, *14*, ***15***, *16*
Erweiterte Funktionen, 399
Erweiterte Funktions- und Speichermodule, 399
  Katalog, **394**. *Siehe auch* Katalog 2
Erweiterte Speichermodule, **370–373**
  Einsetzen und Entfernen, **371**
Erweiterter Speicher
  Aufbau, **373**
  Files, **205**. *Siehe auch* File(s)
  Funktionen, Vergleich HP-41CX und HP-41C/CV, **401–402**
  verfügbare Register, **206–208**, 211
  Verzeichnis, ***125***. *Siehe auch* Katalog 4
Exponenten, **159**, 161
  in Programm- und Druckerlistings, *19*, *32*
  Verwenden von, ***18***
Exponentialfunktion, **187**
  dekadische, ***51***
  natürliche, ***51***
Externe Funktionen
  Ausführungsdauer, **397**
  Katalog. *Siehe* Katalog 2
  Programmspeicher, **397**
  und Programmzeilen, **396**
Externe ROMs (XROMS), **394**

F

Fakultät, ***51***, **185**
Fehler
  Anzeigen, ***34***, **171**
  Bedingungen, **171**
  bei numerischen Funktionen, **185**
  bei Verwendung des Texteditors, 233
  File-, ***127***
  Flags zum Ignorieren von, **290**, 306
  Korrektur in Berechnungen, **179**
  Meldungen, **171**, **354**
  Meldungen, Löschen von, *20*
  Program-, ***98***
  Zeit-, **245**
Fehlerkorrektur, *16*
  in Berechnungen, **179**
  in der Anzeige, 159
Fehlersuche, ***91***
Festkomma-Anzeigeformat, ***31***, **160**
File. *Siehe auch* Files
  Fehler, ***127***
  Größe, **206**
  Größe, Bestimmung, **208**
  Größe, Veränderung, **213**
  Header. *Siehe* Header
  Katalog. *Siehe* Katalog 4
  Name, *116*, **205**, 206
  Speicher. *Siehe* Speicher, Files im
  Typ, Bestimmung, **207**
  Zeiger, ***115***, *126*, **206**, 213, **214**, 217
  Zeiger, Auffinden, 216
  Zeiger, Positionieren, **215**
  Zeiger, Verstellen, ***115***
Files, 205–206. *Siehe auch* Datenfiles; File; Momentaner File; Programmfiles; Textfiles
  Ablegen im Massenspeicher, **227**
  Ändern der Größe, **213**
  Änderung der Registerzuweisung für, **213**
  Durchsuchen nach Alpha-Strings, **226**
  Erzeugen von, **211**
  Löschen von, **208**, **213**
  Register in, **206**, **208**
  Speicheranforderungen von, 205, **212**
  Spezifizieren von, **206**
  Typen, *113*, **205**
  Zurückrufen aus dem Massenspeicher, **227**
  Zuweisen von Registern für, **212**
FLÄCHE, Programm, ***86***, *91*, *93*, *96*, *99*
Flag zur automatischen Ausführung, **289**
Flag-Indikatoren, **289**
Flag-Manipulation, 402
Flagabfrage, **303**, **304**

Flags
Abfrage und Löschen, **304**
Abfrage, 288, 298, **304**
Benutzer-, 288, **289**
Benutzer-, Speichern des Status, **295**
Benutzer-, Vergrößern der Anzahl, **295**
Setzen und Löschen, ***35***, **288**
Status in der Voreinstellung, **293**
Steuer-, 288, **289**
System-, **291**
zur Programmsteuerung, 288
Flagstatus
dargestellt als Byte, **294**
Speicherung des, **292**, **295**, **296**
Umwandlung in eine Zahl, **292**, **295**
Wiederherstellen des, **292**, **296**
Zurückrufen des. *Siehe* Flagstatus, Speicherung
Formate
Anzeige, ***31***, **160**, 291
Datum, ***62***, **242**, 291
Uhr, ***61***, **239**
Winkel, ***53***, **186**
Formeln
Mittelwert, **192**
Standardabweichung, **192**
Fragen, technische, **390**
Führende Nullen, **336**
Funktionsanzeige, ***48***, **169**
Funktionsprüfung, **385**

G

Ganggenauigkeit, Uhr, 238, 374, **378**
Ganzzahliger Anteil einer Zahl, **186**
Gebrochener Anteil einer Zahl, **186**
GEHEIM, File, ***114***, *116*, *121*
Gewährleistung, **386**
Service, **389**
Gleichzeitige Alarme, 360
Globale Labels, *86*, *87*, ***88***, 248, 282, **299**
Anzeige aller, ***98***, **284**
automatische Belegung des User-Tastenfelds, **211**
doppelte, 284
Einfügen, **284**
fehlende, ***100***, **284**
Sprung auf zugeordnete, **285**
Sprung auf, ***100***, 283, 284, **285**
Suche nach, **299**, **303**
und Alarme, 251
Verzweigung, 299
Grad
Minuten-Sekunden, ***53***, *65*, **187**
Modus, ***53***, **186**
Umwandlung, ***53***, **187**
GRAD, ***53***, 160, **186**

H

Handbuch, Aufbau, ***9***
Hauptspeicher, *36*, **194**
Alarme im, 196, **198**
Datenregister, **199**
Programme im, **196**, **197**
Tastenbelegungen im, 196, **198**
Voreinstellung, **196**
Zuweisung, **194–196**, **199**, 281
Header, Textfile, ***113***, **205**, 208, 212, 214
Hochgestelltes T. *Siehe* **T**
Hornerschema, **177**
HP-IL (Hewlett-Packard Interface-Loop), **394**

I

Indikatoren, ***34***, **160**
Indirekte Adressierung, **162**, **200**
Indirekte Parameter, Funktionen, **164**
Initialisieren von Programmen, ***108***

K

Kartesische Koordinaten, ***54***, **189**
Katalog 1, ***98–99***, *100*, 171, 196, **284**
Durchsuchen, 299, **303**
Katalog 2, 171, 248, **394**, 399
Durchsuchen, 299, **303**
erweiterte Funktionen in, **401**
und Alarme, 251
Zeitfunktionen in, **401**
Katalog 3, 171, 399
Durchsuchen, **303**
Katalog 4, ***125***, 171, **206**, 400
Katalog 5, ***71–73***, 171, 196, **255**, 400
schrittweise Ausführung, ***71***
Katalog 6, ***48***, **168**, 196, 400
Kataloge, **170**
Operation der, 400
Stromverbrauch, **170**, **400**
Vergleich HP-41CX und HP-41C/CV, **400**
Kehrwert, ***51***, **185**
Kennung eines leeren Records, ***118***, *121*, **229**
Kompatibilität, HP-41CX und HP-41C/CV, **398**
Kompensationsfaktor, **374–377**
Abrufen, **376**
Einstellen, **376**
Formel, **377**
Konfigurationen erweiterter Speichermodule, **370**
Konstante Faktoren, **180**
Konstanten, Berechnungen mit, ***24***, **177–178**, **180**
Kontrolltasten (Texteditor), ***120***
Koordinatenumwandlung, ***54***, **189**

Kopieren von Programmen in Applikationsmodulen, **107–108**, **281**
KREIS, Programm, ***96–97***, *99*
Kubikfunktion, **183**

L

Labels, **299–300**. *Siehe auch* Globale Labels; Lokale Alpha-Labels; Lokale Labels; Numerische Labels
  Bytebelegung durch, 299
  Suche nach, 287, **299**, **300**, 301, **303**
Laden von Programmen. *Siehe* Programme, Eingabe
Länge eines Alpha-Strings, **313**
LAST X Register, ***23***, 175, **179**, 180, 181, 185
Logarithmus
  dekadischer, ***51***, **187**
  natürlicher, ***51***, **187**
Lokale Alpha-Labels, ***88***, **169**, **300**
Lokale Labels, *87*, ***88***, **299**, **300**
  Suche nach, **300**, 301
Löschen
  der Alpha-Anzeige, ***26***
  der Anzeige, ***19***, *20*, *93*, **159**, 160, 319
  der Statistikregister, ***56***, **190**
  des Alpha-Registers, **160**
  des Speichers, Auswirkungen, **382**
  des Stacks, **183**
  von Alarmen, **258**, 259, 261
  von Datenregistern, ***38***, **202**
  von Flags, **288**
  von gestoppten Zeiten, ***75***, **267**
  von Programmen, ***102–103***, **286–287**
  von Programmzeilen, ***100–101***, **285**
  von Tastenbelegungen des User-Tastenfelds, ***47***, **168**
  von Zeichen nach einer Anhängeoperation, **367**
Löschen eines Files, **208**

M

Massenspeicher
  Kopieren von Files aus dem, **227**
  Speichern von Files im, **227**
Meldealarme, **247**, 251
Meldung
  Anzeige, *30*, ***34***, **161**
  Erzeugung, *95*
  Flag, **292**
  in Programmen, ***94***, 282, 308, **314**, **318**, 319
  Löschen, ***94***
Mittelwert, ***58***, **192**
Module, fehlende, **395**
Modulo, **190**
Modustasten, *14*, 155
Momentane Programmzeile, ***85***, *100*, 282
  Anzeige, ***92***, **284**
  Ausführung, ***91***
Momentaner File, ***115***, **206**, 208
  Änderung, ***115***, ***126***, **207**
Momentanes Programm, ***85***
  Löschen, 287
Multiplikation, ***51***

N

Nachlaufende Nullen, 366
Negative Zahlen, ***18***, 159
Neugrad-Modus, ***53***, **186**
Nicht auf dem Tastenfeld vorhandene Funktionen, ***44***, 156
Nicht kommutative Operationen, ***22***, **176**, 180, 181
Nicht programmierbare Funktionen, **283**
Normal-Tastenfeld, *14*, *15*, 155, **156**
NULL, ***48***, **169**
Null-Bytes in einem Programm, **197**, 198
Null-Programm, **281**
Null-Zeichen, 311
  Anzeige von, **366**
  im Alpha-Register, 309, **366**
  in einem String, **366**, **367**
  in Filenamen, **367**
  Löschen von, **367**
  und anhängte Zeichen, **367**
Numerische Anzeigen, ***31***, **160**
Numerische Funktionen, Fehler bei, **185**
  Numerische Labels, ***88***, **299**, 317
  Kurzform, **299**
  Langform, **299**
  Sprung auf, **285**
  Verzweigung nach, **299**
Numerische Parameterspezifikation, **162**
  Spezialtasten, **165–166**
Numerisches Tastenfeld (Texteditor), ***118***, 230, **232**

O

Oktal-Dezimal Umwandlung, **187**
Operations-Modus, *15*
Overflow, ***24***, ***42***, *56*

P

Packen des Speichers, ***85***, *89*, 196, **198**, 281
Parameterfunktion, ***20***, *30*, *38*
  Anzeige einer, *31*
Parameterspezifikation, ***30***, 158, **162**, 200
  indirekte, **162**
  Spezialtasten zur, **165–166**
Peripheriegeräte, Beschreibung, **392**
Peripheriegeräte-Steuerflag, **289**

Permanentes **.END**, ***86***, *89*, *98*, **196**, 281
Permanentspeicher, *14*, ***28–29***, **155**
  Löschen des, ***29***, **382**
Pi, ***19***, **159**
Polarkoordinaten, ***54***, **189**
Polynome, Berechnung, **177**
Position
  eines Strings im Alpha-Register, **312–313**
  eines Strings in einem Textfile, **226**
  im Programmspeicher, Änderung, **283–285**
Potenzfunktion, *51*, ***52***, **189**
PRGM, ***83***, *90*, 155, **282**, **316**
Produktinformationen, **391**
Programm. *Siehe auch* Momentanes Programm; Programmausführung; Programme; Programmfiles; Programmzeile
  Anhalten, ***93***
  Anzeige der Ergebnisse, **369**
  Ausführung, ***90***, **209–210**, **282**
  Ausgabe, ***92***, *95*, ***96***
  automatische Ausführung, 289
  Begrenzungen, *87*
  Dateneingabe, ***92***
  Editierung, *91*, ***98***, **285–286**
  Eingabe, ***89***, **280**
  Eingabeaufforderungen, ***95***
  Ergebnisanzeige, ***93***
  Fehler, ***98***
  Fehlersuche, ***91***
  Fortsetzung nach Anhalten, ***93***
  Größe, Bestimmung der, **211**
  in einem Modul, **395**
  Katalog. *Siehe* Katalog 1
  Kommunikation mit dem Benutzer, **317**
  Kompatibilität mit dem HP-41C/CV, **398**
  Kopieren in einen File, ***124***
  Kopieren von Applikationsmodulen, ***107–108***, **281**
  Korrektur, *91*, ***98***
  Löschen, ***102–103***, **286–287**
  Meldungen, ***94***, 282, 308, **314**, **318**, 319
  Modus, *15*, *83*, ***87***, 285
  Name, *86*
  Name, fehlender, ***100***, **284**
  schrittweise Anzeige, ***91***, ***100***, **284**
  Speicher. *Siehe* Speicher, Programme im
  Speichern in einem File, ***124***, **208**
  Speicherung, ***89***, **208**
  Sprung an den Anfang eines, **285**
  Sprung zu einem, ***100***, **283–285**
  Textfile-Operationen in einem, **222**
  Unterbrechung, *92*, ***93***, 319
  Verwenden des Texteditors in einem, **233**
  Verzweigungen, 298
  Zeiger, ***85***
  Zeiger, Positionieren, ***100***, **283–285**
Programm-Modus, 155
Programmausführung
  automatische, **282**
  Indikator für, *90*, 161, **282**
  Rückkehr aus einem Unterprogramm, **301**
  schrittweise, ***91***, **282**
  über das User-Tastenfeld, **282**
  und Drucker, **368**
  Unterbrechen der, 368
  wiederholte, ***90***
Programme
  Anzeige aller, ***98***, **284**
  Löschen, ***102–103***, **286–287**
Programmfiles, ***124–125***, 205, **208–211**. *Siehe auch* File
  Abrufen von, ***124***, **209**
  Erzeugung, ***124***, **208**
  Namen, 208
  Speichern von, ***124***, **208**
  Zurückrufen von, ***124***, **209**
Programmverzweigungen, *88*
  Funktionen zur Schleifensteuerung, **306**
  in Schleifen, **305**
  Speicheranforderungen, **300**
  Überspringen einer Zeile, 298, **304**
  zu einem Label, 298, 301
Programmzeile, ***84***, *85*, **197**, **282**. *Siehe auch* Momentane Programmzeile
  Einfügen einer, ***102***, **286**
  Löschen einer, ***100–101***, **285**
  Nummer, **282**
  Speicherbelegung durch eine, **197**
  Sprung zu einer, ***100***, 283
  Überspringen einer, 298, **304**
Prozentfunktion, *51*, ***52***, **188–189**
Prozentualer Unterschied, *51*, ***52***, **188**

Q

QUAD Programm, ***105***
Quadratfunktion, ***51***, **185**
Quadratische Gleichung, Programmbeispiel, ***103ff***
Quadratwurzel, ***51***, **185**

R

RAD, ***53***, 160, **186**
Record, ***113–114***, ***121***, 212, **214**, 222, 229
  Anhängen, **222**
  Einfügen (Texteditor), ***121***, **232**
  Einfügen, **223**
  Löschen (Texteditor), ***121***, **233**
  Löschen, **224**
  maximale Länge, **230**, 233
  Nummer, **229**
  Sprung zu einem (Texteditor), **233**
  Zurückrufen, **226**

Record/Zeichen-Zeiger, *113–114*, ***115***, 213–216, **214**, 228–230
Register. *Siehe auch* Stackregister; LAST X Register; Alpha-Register
Adresse, Spezifikation, **200**
Arithmetik, ***40–42***, **201**
Austausch mit Datenfile, **218**
Austausch von Inhalten, ***39–40***, **201**
Blöcke, Kopieren des Inhalts, **201**
Blöcke, Vertauschen des Inhalts, **201**
Daten, Löschen, ***38***, **202**
Datenspeicher-, ***36***, **194**
File, 212, **214**
in einem File, **206**, **208**, **213**
Inhalt, Anzeige, ***39***, **319**
Kopieren in einen File, **220**
Spezifikation, 164
Statistik-. *Siehe* Statistikregister
über R_{99}, **199**
ungebundene, ***36***, **194**, 196
verfügbar für Daten, **199**
verfügbar für Files, *125*, **206–208**, 211
verfügbar für Programme, *89*, **281**
Zeiger, 213, 215, 216
Zugriff auf Blöcke, **218**, **219**
Zurückrufen aus einem, **220**
Zurückrufen aus einem File, ***123–124***, **218**, **219**
Zuweisung, **199**, 281
Zuweisung, Änderung, **199**
Zuweisung, Überprüfung, **199**
Reparatur, Versand zur, **390**
Reparatur-Service, **386**, **388**
Rest, **190**
ROM (Read Only Memory), 281
ROM-Module, **393**, 397
Rückrufmodus, **272**
Runden einer Zahl, **186**

S

Schaltkreisbeispiel, ***55***
Schleifensteuerung, 298, **305**, **306**
Zahl zur, **306**
Schrittweise Programmanzeige, ***91***, ***100***, **284**
Schrittweise Programmausführung, ***91***, **282**
Setzen von Flags, **288**
SHIFT, ***16***, 156, 230
Aufheben, *16*, **156**
Sinus, ***54***
Software-Module, **393**
Spannungsabfall, **383**
Spannungsabfall-Flag, **292**
Speicher
Aufteilung, 399
erweiterter, verfügbar, *125*
Files im, *113*
Haupt-, *36*, **194**
Programme im, *84*, ***85***, ***99***
Stack. *Siehe* Stack
Textfiles im, ***114***
verfügbar für Files, *125*
verfügbar für Programme, *89*, **281**
Vergleich HP-41CX und HP-41C/CV, **399**
Zuweisung, *36*
Speichern
von Alpha-Zeichen, **200**
von Zahlen, ***37***, *123*, **182**, **200**, 217, 218, 220
Speichern von Daten in einem File, ***123***, **217**, **218**, **220**
Stack, ***20***, *21*, **174–175**, 185. *Siehe auch* Unterprogramm-Rücksprungstack
Auffüllen, **177**
Berechnungen im, **180**
Drop, **175**
Lift, **175**
Lift, Freigabe, **176**
Lift, neutral, **176**
Lift, Sperren, **176**
Löschen, **183**
Operationen mit numerischen Funktionen, **184**
Register, *20*, *21*, **175**, **180**
Register, Adressierung, 166
Registerarithmetik, **182**
Registeraustausch, **181**, 182
zyklisches Verschieben, **181**
Standardabweichung, ***58***, **192**
Standardfunktionen, Katalog der. *Siehe* Katalog 3
Statistikregister, ***55***, **190**
Löschen, ***56***, **190**
Überlauf, ***56–57***, **192**
Überprüfen der Zuweisung, ***56***, **190**
Zuweisung, ***56***, **190**
Statistische Daten
Korrektur, ***57***, **191**
Summation, ***55–56***, **191**
Statusmeldungen, **354**
Steueralarme, **248**, **251**, 259, 260, 261
während ablaufender Programme, **248**
Stoppuhr
als Timer, **274**
Anzeige, **267**, 270, 271
Beispiele, *77–78*
Differenz gestoppter Zeiten, *78–79*
Fehler, ***80***
Genauigkeit, **378**
Modi, ***76***
Programm, **275**
Programmierung der, **273**
Register, **270**, 272
Register-Zeiger, *77*, ***79***, *80*
Speicheranforderungen, ***78***
Starten und Anhalten, ***75***, **268**, 273

Tastenfeld, *75*, **266**, **268**, **269**
Tastenfeld, Aktivierung, **268**
Zeiger, **270**, 272
Zeiger, Anzeige, **270**
Zeiger, Begrenzungen, 271
Zeiger, Positionierung, **274**
Zeiger, Veränderung, **270**
Zeit, Einstellen, **273**
Zeit, Zurückrufen, **273**
Zeiten, Löschen, *75*, **267**
Zurücksetzen, *75*, **267**
Strings. *Siehe* Alpha-Strings
Stromunterbrechung, Auswirkungen, **382**
Subtraktion, ***51***
Summation von Daten, ***55–56***, **191**
Korrektur, ***57***, **191**

T

T, *93*, ***118***, *121*, 159, 171, **229**, **282**, 395
T-Register, *20*, **175**, 179, 180
Tangens, ***54***
Tastenbelegung, ***46***, 156, **166**
Grenezen der, **166**
Tastencodes, *46*, **166**
Tastendruck
Darstellung, ***16***
Genauigkeit, **375**
Reaktion eines Programms auf, **317**
Tastenfeld
Funktion, ***44***
Konventionen, ***15***, **156**
Modus, **362**
Tastenfelder des HP-41CX, 158
Technische Unterstützung, **390**
Technisches Anzeigeformat, ***33***, **161**
Temperaturspezifikationen, **391**
Terminologie, Vergleich HP-41CX und HP-41C/CV, **403**
Text
Editieren, ***113ff***
Löschen, **225**
Speichern, **224**
Zurückrufen, **226**
Text-Interpunktion, **229**
Texteditor, ***117ff***
Anzeige, ***118***, **229**, **230**
Ausschalten, ***117***, **230**
Automatische Abschaltung, **230**
Einschalten, ***117***, **230**
Indikatoren, ***117***, **230**
Tastenfeld, ***119–120***, **231**
Textfile-Records. *Siehe* Records
Textfile-Zeichen, ***113***, *114*, ***120***, 212, **229**
Anhängen, **224**
Einfügen, **224**
Hinzufügen (Texteditor), ***120***, **232**
Löschen (Texteditor), ***120***, **232**
Löschen, **225**
Sonder-, **229**
Suche nach, **226**
Textfile-Zeiger. *Siehe* File-Zeiger; Record/Zeichen-Zeiger
Textfiles, ***113ff***, 205, 215, 216, 222, **228**
Ändern der Größe, ***116***
Erzeugen, ***116***, **212**
Löschen, ***117***, **213**
Namen, 212
Speichern im Massenspeicher, **227**
Zurückrufen aus dem Massenspeicher, **227**
Zurückrufen aus, **226**
Tonsignale, ***94***, 260, **319**
Trigonometrische, ***53–54***
Trigonometrischer Modus, ***53***, **186**, 291
Flags, **291**

U

Überfällige Alarme, **260–261**
Aktivierung von bedingten, **261**
Ausführung von, 361
automatische Aktivierung, **260**, **361–363**
Computer-Modus und, **362**
Löschen, 261
übergangene, 360
Überlauf. *Siehe* Overflow
Uhr
Anzeige, ***61***, **238**
bei niedriger Batteriespannung, **383**
Einstellen, ***64***, **238**
Format, **239**
Ganggenauigkeit, **238**, 374, **378**
Nachstellen, **238**
Zeiten, ***63***, 237
Uhrzeit. *Siehe* Zeit
Umgekehrte Eingabe, ***17***
Umgeschaltete Funktionen, *14–16*, ***15***, 156
Umkehrfunktionen, 179
Umschalttaste, *15*, 156
Umwandlung
der Basis, **187**
von Koordinaten, ***54***, **189**
von Winkeln, ***53***, **187**
von Zeitwerten, ***65***, **187**
Underflow, ***24***, ***42***
Ungebundene Register. *Siehe* Register

Unterprogramm, **301–302**
  Aufruf eines Programms als, **209–210**
  Aufrufen, **301**, 317
  Beenden, **301**
  Bytebelegung, **303**
  Rücksprung aus einem, **301–302**
  Rücksprungstack, **302**
  und Tastencodes, **317**
UPN (Umgekehrte Polnische Notation), ***16***, 174
USER, ***47***, 155
User-Funktionen, ***46–47***
  Anzeige, ***48***
  Aufheben, ***47***
  Ausführung, ***47***
  Katalog. *Siehe Katalog 6*
  Zuordnung, ***46***, 256
User-Tastenfeld, *14*, *15*, ***46–47***, *88*, 155, **156**, 166, **198**
  Aufheben von Zuordnungen, **168**, 171
  automatische Zuordnung von globalen Labels, **211**
  Flag, **290**
  Katalog. *Siehe* Katalog 6
  Prioritäten, **169**
  Zuordnungen, **166**, **168**

V

Variable, Funktionen einer, *17*, ***50***, **184**
Variable, Funktionen zweier, ***51***, **187**
VEKTOR, Programm, ***109ff***
Vektorrechnung, ***59***
  Beispiel, *109*
Vergleich von Alpha-Daten, **304**, **305**
Vergleich von X
  mit indirektem Y, **305**
  mit Null, **304**
  mit Y, **304**
Vergleichsfunktionen, **303**, **304**
Verschieben der Anzeige, ***28***, 162
Vertauschen von *x* und *y*, 181
Verzeichnis
  der Alarme. *Siehe* Katalog 5
  der externen Funktionen. *Siehe* Katalog 2
  der Files. *Siehe* Katalog 4
  der Programme. *Siehe* Katalog 1
  der Standardfunktionen. *Siehe* Katalog 3
  der Zuordnungen des User-Tastenfelds. *Siehe* Katalog 6
  des erweiterten Speichers. *Siehe* Katalog 4
Vorzeichen einer Zahl, **186**
Vorzeichenwechsel, ***18***, **159**, 185

W

Wiederholende Alarme, **255**, 258, 259
Winkelmodus, ***53***, **186**, 291
  Flags, **291**
Winkelumwandlung, ***53***, **187**
Wissenschaftliches Anzeigeformat, ***32***, 161
Wochentag, ***67***, **244**
Wurzelberechnung, **190**

X

X-Register, *20*, ***30***, *40*, 158–159, **175**, 179, 180
  Austausch des Inhalts, **201**
  Speichern aus dem, 182
  Vertauschen mit Flag-Statusdaten, **292**, **295**
  Vertauschen mit Y, ***39***
  Zurückrufen in das, 179, 182
XROM
  Funktionen, **394**, 395
  Nummer, **394**, 395, 399
  Nummer, doppelte, **397**
  Nummer, und Programmzeilen, **396**
  Programme, **395**

Y

Y-Register, *20*, **175**, 179, 180
  Vertauschen mit X, ***39***

Z

Z-Register, *20*, **175**, 179, 180
Zahl, Umwandlung in ein Alpha-Zeichen, **309**
Zahlen, **159**
  Eingabe von, *17*, ***18***, **158**, **175**, **176**
Zeichen. *Siehe auch* Alpha-Zeichen; Textfile-Zeichen
  Codes, **310**, **317**
  Eingabe. *Siehe* Alpha-Zeicheneingabe
  Löschen, *20*
  Text, 212
  Zeiger. *Siehe* Record/Zeichen-Zeiger
Zeiger. *Siehe* File-Zeiger, Programm-Zeiger, Record/Zeichen-Zeiger
Zeilennummer, Spezifikation, 166
Zeit
  Addition und Subtraktion, ***65***, **188**
  Anzeige, **238**
  Drift, **375**
  Einstellung, ***63***, **237**
  Fehler, **245**
  Funktionen, 399
  Funktionen, Vergleich HP-41CX und HP-41C/CV, **400–401**
  Genauigkeit, **238**, 374
  Konvention, ***61***, **239**

Korrektur, **238**
Module, 399
Nachstellen, ***64***, **238**
Umwandlung, **187**
Werte, 237
Zurückrufen, ***64***, **240**
Zeiten, erlaubte, **245**
ZEITEN, Programm, **276**
Zeitfunktionen, Katalog der, **394**
Zeitnahmen, ***76–77***, 270
Ausdruck, **275**
Fehler, **273**
negative (Delta-Stoppmodus), **273**
Speicherung, **270**
Zurückrufen von, **270**, **272**
Zeitraum zwischen zwei Terminen, ***67***, **244**
Ziffern. *Siehe auch* Alpha-Ziffern
Eingabetasten, ***18***, **159**
Gruppierung, ***35***, **161**
Löschen, *20*
Trennung, ***35***, **161**
Zifferntrennzeichen, ***35***, **161**, 290
Zurückrufen
von Alarmen, **252**
von Alpha-Zeichen, **200**
von Zahlen, ***37***, *123*, **182**, **200**, 218, 219, 220
Zwischenergebnisse, *20*
Zwischenstatistiken, **191**
Zyklisches Verschieben des Stacks, **181**

Funktionsindex

Jede Funktion wird unter ihrem Alpha-Namen aufgeführt (in einem blauen Tastenrahmen). Wenn die Funktion auf dem Tastenfeld vorhanden ist, erscheint das zugehörige Tastensymbol (schwarz oder goldfarben) in Klammern hinter dem Alpha-Namen. (Diese Konventionen werden auf der Innenseite des Vorderumschlags erläutert.) Wenn die Zeile mit dem Funktionsnamen eingerückt ist, kann die Funktion nur über das Tastenfeld ausgeführt werden; d.h. die Funktion ist nicht programmierbar.

Für jede Funktion finden Sie bis zu drei Seitenverweise. Die erste Seitenzahl (*kursiv* gesetzt) bezieht sich auf Band 1; die zweite Seitenzahl verweist auf Band 2. Die dritte, **fettgesetzte** Seitenzahl weist auf das Funktionsverzeichnis in Band 2 hin, in dem alle Funktionen zusammengefaßt sind.

Funktion	Seiten
←	*19*, 158, **418**
├	*27*, 159, **436**
+ (+)	*51*, 188, **423**
− (−)	*51*, 188, **423**
* (×)	*51*, 188, **423**
/ (÷)	*51*, 188, **423**
1/X (1/x)	*51*, 185, **423**
10↑X (10^x)	*51*, 187, **423**
ABS	186, **423**
ACOS (COS⁻¹)	*54*, 186, **423**
ADATE	243, **429**
ADV	368, **438**
ALENG	313, **436**
ALMCAT	*71*, 255, **429**
ALMNOW	261, **429**
ALPHA	*24*, 155, **416**
ANUM	311, **436**
AOFF	159, **436**
AON	159, **436**
APPCHR	224, **425**
APPREC	222, **425**
ARCL (ARCL) *nn*	*96*, 200, **436**
ARCLREC	226, **425**
AROT	313, **436**
ASHF	200, **436**
ASIN (SIN⁻¹)	*54*, 186, **423**
ASN (ASN) *Name, Taste*	*46*, 166, **416**
ASROOM	222, **425**
ASTO (ASTO) *nn*	200, **436**
ATAN (TAN⁻¹)	*54*, 186, **423**
ATIME	240, **429**
ATIME24	241, **429**
ATOX	311, **437**
AVIEW (AVIEW)	*94*, 318, **437**
BEEP (BEEP)	*94*, 319, **438**
BST (BST)	*91*, 284, **431**
CAT (CATALOG) *n*	170, **416**
CF (CF) *nn*	*35*, 288, **416**
CHS (CHS)	*18*, 185, **423**
CLA (CLA)	*26*, 159, **437**
CLALMA	258, **429**
CLALMX	259, **429**
CLD	*94*, 318, **419**
CLFL	*117*, 213, **426**
CLK12	*61*, 239, **429**
CLK24	*61*, 239, **429**
CLKEYS	*47*, 168, **416**
CLKT	*61*, 239, **429**
CLKTD	*61*, 239, **429**
CLOCK	*61*, 238, **429**
CLP	*102*, 286, **431**
CLRALMS	*69*, 258, **429**
CLRG	*38*, 202, **420**
CLRGX	*38*, 202, **420**
CLΣ (CLΣ)	*56*, 190, **420**
CLST	183, **420**
CLX (CLx)	*19*, 159, **420**
COPY	*107*, 281, **431**
CORRECT	238, **429**
COS (COS)	*54*, 186, **423**
CRFLAS	*116*, 212, **426**
CRFLD	*123*, 211, **426**
D-R	*53*, 187, **423**
DATE	*66*, 242, **429**
DATE+	*66*, 244, **429**
DDAYS	*67*, 244, **430**
DEC	187, **423**
DEG	*53*, 186, **416**
DEL	*101*, 286, **432**
DELCHR	225, **426**
DELREC	224, **426**
DMY	*62*, 242, **430**
DOW	*67*, 244, **430**
DSE *nn*	306, **432**
ED	*117*, 228, **426**
EEX	*18*, 159,
EMDIR	*125*, 206, **426**
EMDIRX	207, **426**
EMROOM	208, **426**
END	*89*, 301, **432**
ENG (ENG) *n*	*33*, 161, **416**
ENTER↑ (ENTER↑)	*17*, 175, **420**
E↑X (e^x)	*51*, 187, **423**
E↑X−1	187, **423**
FACT	*51*, 185, **423**
FC? *nn*	304, **432**
FC?C *nn*	304, **433**
FIX (FIX) *n*	*31*, 160, **417**
FLSIZE	208, **426**
FRC	186, **423**
FS? (FS?) *nn*	304, **433**
FS?C *nn*	304, **433**
GETAS	227, **426**

Funktion	Seiten
GETKEY	317, **438**
GETKEYX	317, **438**
GETP	*124*, 209, **427**
GETR	*123*, 217, **427**
GETREC	226, **427**
GETRX	218, **427**
GETSUB	209, **427**
GETX	*124*, 220, **427**
GRAD	*53*, 186, **417**
GTO (GTO) *Label*	*100*, 300, **433**
GTO . *nnn oder Label*	283, **432**
GTO . .	*89*, 281, **432**
HMS	*53*, 187, **430**
HMS+	*53*, 188, **430**
HMS−	*53*, 188, **430**
HR	*53*, 187, **430**
INSCHR	224, **427**
INSREC	222, **427**
INT	186, **423**
ISG (ISG) *nn*	306, **433**
LASTX (LASTx)	*23*, 179, **420**
LBL (LBL) *Label*	*88*, 299, **433**
LN (LN)	*51*, 187, **423**
LN1+X	187, **423**
LOG (LOG)	*51*, 187, **424**
MDY	*62*, 242, **430**
MEAN	*58*, 192, **424**
MOD	190, **424**
OCT	187, **424**
OFF	292, **433**
ON	155, **417**
ON	*14*, 155, **417**
P−R (P→R)	*54*, 189, **424**
PACK	198, **432**
PASN	166, **417**
PCLPS	*103*, 286, **419**
% (%)	*51*, 188, **424**
%CH	*51*, 188, **424**
PI (π)	*19*, 159, **424**
POSA	312, **437**
POSFL	226, **427**
PRGM	*87*, 155, **417**
PROMPT	*95*, 314, **438**
PSE	*93*, 315, **438**

Funktion	Seiten
PSIZE	199, **417**
PURFL	*117*, 208, **427**
R↑	181, **421**
R−D	*53*, 187, **424**
R−P (R→P)	*54*, 189, **424**
R/S	*93*, 166, **434**
RAD	*53*, 186, **417**
RCL (RCL) *nn*	*37*, 200, **421**
RCLAF	376, **430**
RCLALM	252, **430**
RCLFLAG	296, **417**
RCLPT	216, **427**
RCLPTA	216, **428**
RCLSW	273, **430**
RDN (R↓)	181, **421**
REGMOVE	201, **421**
REGSWAP	201, **421**
RESZFL	*116*, 213, **428**
RND	186, **424**
RTN (RTN)	*90*, 301, **434**
RUNSW	273, **430**
SAVEAS	227, **428**
SAVEP	*124*, 208, **428**
SAVER	*123*, 217, **428**
SAVERX	218, **428**
SAVEX	*123*, 220, **428**
SCI (SCI) *n*	*32*, 161, **417**
SDEV	*58*, 192, **424**
SEEKPT	215, **428**
SEEKPTA	*115*, 215, **428**
SETAF	376, **430**
SETDATE	*62*, 242, **430**
SETIME	*63*, 237, **430**
SETSW	273, **430**
SF (SF) *nn*	*35*, 288, **417**
Σ+ (Σ+)	*55*, 191, **424**
Σ− (Σ−)	*57*, 191, **424**
ΣREG *nn*	*56*, 190, **417**
ΣREG?	*56*, 190, **417**
SIN (SIN)	*54*, 186, **424**
SIGN	186, **424**
SIZE *nnn*	199, **417**
SIZE?	199, **418**
SQRT (√x)	*51*, 185, **424**

Funktion	Seiten
SST (SST)	*91*, 284, **432**
ST+ (STO +) *nn*	*40*, 201, **421**
ST− (STO −) *nn*	*40*, 201, **422**
ST* (STO ×) *nn*	*40*, 201, **422**
ST/ (STO +) *nn*	*40*, 201, **422**
STO (STO) *nn*	*37*, 200, **422**
STOFLAG	296, **418**
STOP (R/S)	*93*, 302, **434**
STOPSW	273, **430**
SW	*75*, 266, **430**
SWPT	274, **431**
T+X	*64*, 238, **431**
TAN (TAN)	*54*, 186, **424**
TIME	*64*, 240, **431**
TONE *n*	319, **438**
USER	*47*, 155, **418**
VIEW (VIEW) *nn*	*39*, 319, **422**
X↑2 (x²)	*51*, 185, **424**
X = 0? (x = 0?)	304, **434**
X ≠ 0?	304, **434**
X < 0?	304, **434**
X <= 0?	304, **434**
X > 0?	304, **434**
X = Y? (x = y?)	304, **434**
X ≠ Y?	304, **434**
X < Y?	304, **434**
X <= Y? (x ≤ y?)	304, **434**
X > Y? (x > y?)	304, **434**
X = NN?	305, **435**
X ≠ NN?	305, **435**
X < NN?	305, **435**
X <= NN?	305, **435**
X > NN?	305, **435**
X >= NN?	305, **435**
X<> *nn*	*40*, 201, **422**
X<>F	295, **422**
X<>Y (x ⇄ y)	*39*, 181, **422**
XEQ (XEQ) *Label*	*45*, 301, **435**
XTOA	309, **437**
XYZALM	*67*, 250, **431**
Y↑X (y^x)	*51*, 189, **424**

VERKAUFSNIEDERLASSUNGEN

Hewlett-Packard GmbH:
6000 Frankfurt 56, Bernerstraße 117, Tel. (0611) 50 04-1
7030 Böblingen, Herrenberger Straße 110, Tel. (07031) 14-0
1000 Berlin 30, Keithstraße 2–4, Tel. (030) 24 90 86
4000 Düsseldorf 11, Emanuel-Leutze-Straße 1, Tel. (0211) 59 71-1
2000 Hamburg 60, Kapstadtring 5, Tel. (040) 6 38 04-1
3000 Hannover 91, Heidering 37–39, Tel. (0511) 57 06-0
6800 Mannheim, Roßlauer Weg 2–4, Tel. (0621) 70 05-0
7910 Neu-Ulm, Messerschmittstraße 7, Tel. (0731) 70 24-1
8500 Nürnberg 10, Neumeyer Straße 90, Tel. (0911) 52 20 83-87
8028 Taufkirchen, Eschenstraße 5, Tel. (089) 61 17-1
7517 Waldbronn 2, Hewlett-Packard-Straße, Tel. (07243) 602-1

Hewlett-Packard (Schweiz) AG:
CH-8967 Widen, Allmend 2, Tel. (057) 31 21 11
CH-4058 Basel, Clarastraße 12, Tel. (061) 33 59 20
CH-1217 Meyrin 2, rue du Bois-du-Lan 7, Tel. (022) 83 11 11

Hewlett-Packard Ges.m.b.H. für Österreich/sozialistische Staaten
A-1222 Wien, Lieblgasse 1, Tel. (0222) 23 65 11-0
A-8052 Graz, Grottenhofstraße 94, Tel. (0316) 215 66

SERVICENIEDERLASSUNGEN

Hewlett-Packard GmbH:
6000 Frankfurt, Bernerstraße 117, Tel. (0611) 5004-1

Hewlett-Packard (Schweiz) AG:
CH-8967 Widen, Allmend 2, Tel. (057) 31 21 11

Hewlett-Packard Ges.m.b.H. für Österreich/sozialistische Staaten:
A-1222 Wien, Lieblgasse 1, Tel. (0222) 23 65 11-0

HEWLETT
PACKARD

00041-90494 - German - 12.83-W-5K - IPC

Printed in Singapore