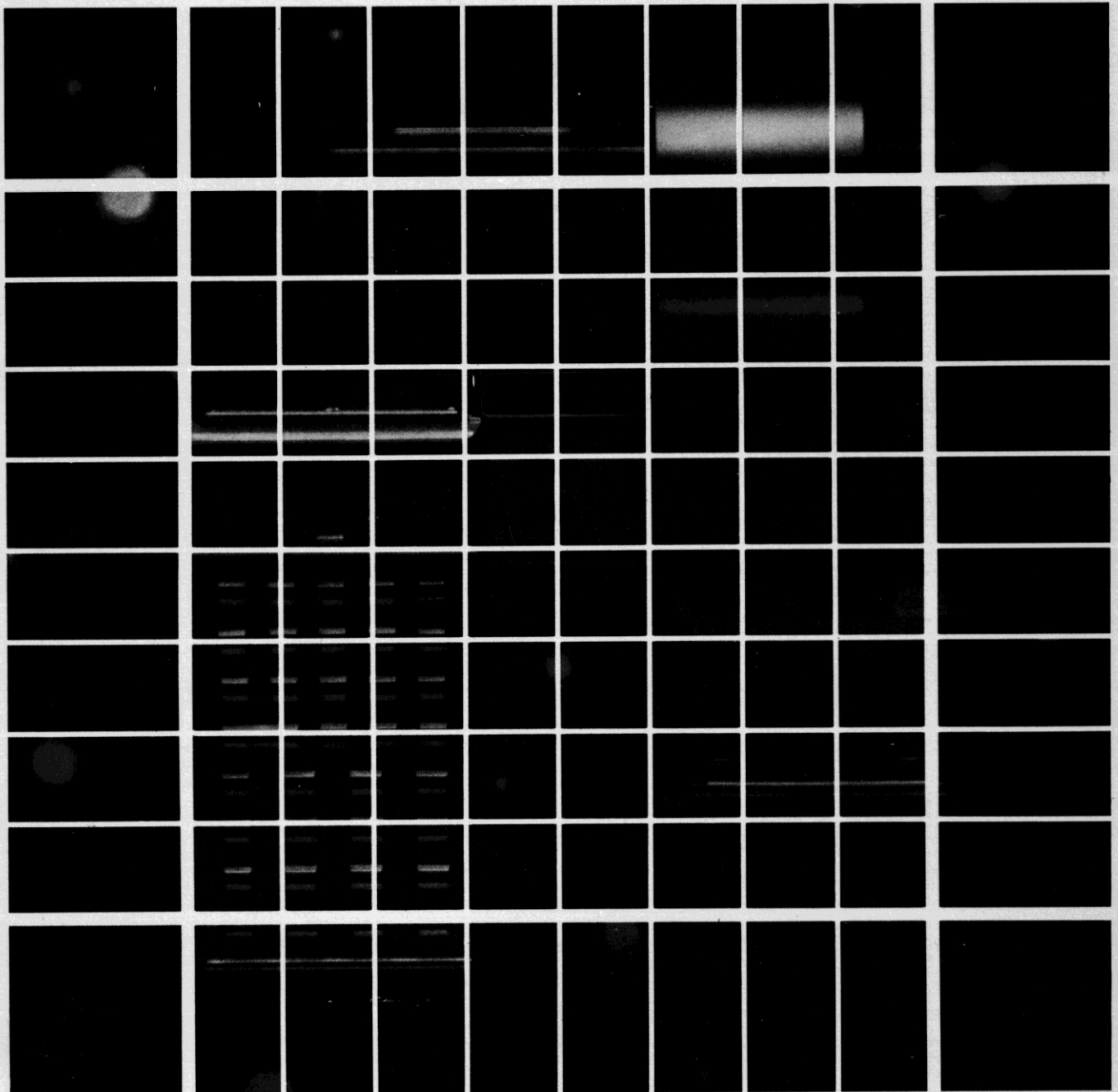


Erweitertes Funktions/Speicher-Modul

BENUTZERHANDBUCH





HP82180A Erweitertes Funktions/Speicher-Modul

Benutzerhandbuch

May 1982

82180-90003 Rev. B

Inhalt

Einleitung	5
Erweiterte Funktionen	5
Programmierbare Funktionen	5
Register- und Flagfunktionen	5
ALPHA-Registerfunktionen	5
Funktionen zum Zugriff auf den erweiterten Speicher	5
Der erweiterte Speicher	5
 Abschnitt 1: Grundlagen	 7
Kennzeichnung und Installation	7
Zulässige Konfigurationen	8
Gebrauch dieses Handbuchs	8
 Abschnitt 2: Erweiterte Funktionen	 11
Datenspeicherregister Operationen	11
Flag Operationen	11
USER Modus Operationen	13
ALPHA-String Operationen	13
Übertragen von Zeichen zwischen ALPHA- und X-Register	14
Länge eines Zeichenstrings	15
Durchsuchen von ALPHA-Strings	15
Rundumverschieben des ALPHA-Registerinhalts	16
Sonstige Operationen	16
Tastenabfrage	16
Zuordnen von Datenspeicherregistern	17
Löschen von Programmen	17
 Abschnitt 3: Erweiterter Speicher	 19
Aufbau der erweiterten Speichers	19
Filennamen und File-Pointer	20
Datenfiles	21
ASCII-Files	21
Programmfiles	23
Arbeitsfiles	23
Fileverwaltung	23
Programmfile Operationen	25
Daten- und ASCII-File Operationen	26
Datenfile Operationen	27
ASCII-File Operationen	29
Operationen mit ganzen Records	29
Operationen mit Zeichen innerhalb eines Records	30
Durchsuchen eines ASCII-Files	31
Übertragen eines ASCII-Files in den Hauptspeicher	32
Übertragen von Daten zwischen dem erweiterten Speicher und Massenspeichereinheiten	33

Abschnitt 4:	
Programmieren mit den Funktionen des Erweiterten Funktions/Speicher-Moduls	35
Anhang A: Wartung, Gewährleistung und Serviceinformation	37
Wartung des Moduls	37
Gewährleistung	37
Service	38
Benutzerberatung	40
Händler- und Produktinformation	40
Anhang B: Fehlermeldungen	43
Anhang C: Nullzeichen	47
Nullzeichen und das ALPHA-Register	47
Behandlung von Nullzeichen	47
Funktionsindex	49



Einleitung

Das Erweiterte Funktions/Speicher-Modul HP82180A beinhaltet zusätzliche Funktionen und erweiterten Speicher für Ihren Rechner HP-41. Der verfügbare Speicherbereich kann zusätzlich durch ein bzw. zwei Erweiterte Speichermodule HP82181A, deren Verwendung ebenfalls in diesem Handbuch beschrieben wird, vergrößert werden.

Erweiterte Funktionen

Die durch das Modul implementierten Funktionen lassen sich in die folgenden vier Kategorien einteilen:

Programmierbare Funktionen

Einige Standardfunktionen des Rechners, wie etwa `ASN` und `SIZE`, die normalerweise nicht programmierbar sind, haben programmierbare Äquivalente im Modul. Des weiteren enthält das Modul zusätzliche programmierbare Funktionen, die das Schreiben effizienter Programme vereinfachen und bisher nicht im Rechner verfügbar waren.

Register- und Flagfunktionen

Einige dieser Funktionen vereinfachen die Manipulation von Registerblöcken, andere steigern die Verwendbarkeit der Rechner-Flags.

ALPHA-Registerfunktionen

Diese Funktionen ermöglichen das Herausziehen numerischer Daten aus dem ALPHA-Register, das Durchsuchen des ALPHA-Registers nach vorgegebenen Zeichenketten (Strings) sowie das Konvertieren von Zeichen in die äquivalenten Dezimalwerte und umgekehrt.

Funktionen zum Zugriff auf den erweiterten Speicher

Diese Funktionen ermöglichen das Speichern und Zurückladen von Programmen und Daten in die Register des erweiterten Speichers, der durch das Erweiterte Funktions/Speicher-Modul und gegebenenfalls zusätzliche Erweiterte Speicher-Module verfügbar wird. Des weiteren können Sie mit Hilfe dieser Funktionen Textblöcke aus alphanumerischen Zeichen anlegen und beliebig korrigieren (editieren).

Der erweiterte Speicher

Das Erweiterte Funktions/Speicher-Modul selbst enthält 127 erweiterte Speicherregister*. In vieler Hinsicht ähneln diese Register den Ihnen bereits bekannten Speicherregistern des Rechners. Der wesentliche Unterschied liegt darin, daß der Rechner nicht direkt auf in diesen Registern gespeicherte Daten zugreifen kann. Vor dem eigentlichen Zugriff müssen die Daten erst in den Hauptspeicher des Rechners übertragen werden. Zur Unterscheidung wird hier für die herkömmlichen Speicherregister des HP-41 die Bezeichnung *Hauptspeicher* verwendet. Der erweiterte Speicher und der Zugriff darauf wird in den Abschnitten 1 und 3 beschrieben.

Das Erweiterte Funktions/Speicher-Modul HP82180A ermöglicht die Verwendung von ein oder zwei zusätzlichen Erweiterten Speicher-Modulen HP82181A. Jedes dieser Module vergrößert den erweiterten Speicher um 238 Register. Damit läßt sich die Speicherkapazität Ihres HP-41C oder HP-41CV um bis zu 603 Register an erweitertem Speicher erhöhen.

* Die durch das Erweiterte Funktions/Speicher-Modul HP82180A und das Erweiterte Speicher-Modul HP82181A implementierten Speicherregister unterscheiden sich von den Registern $R_{(100)}$ bis $R_{(318)}$ des Hauptspeichers, die im Bedienungshandbuch des Rechners als *erweiterte Speicherregister* bezeichnet werden, und sollten nicht mit diesen verwechselt werden.

Grundlagen

Das Erweiterte Funktions/Speicher-Modul HP82180A und das Erweiterte Speicher-Modul HP82181A können sowohl mit dem Rechner HP-41C als auch mit dem Rechner HP-41CV verwendet werden. Die in diesem Handbuch enthaltenen Erläuterungen treffen auf beide Rechnertypen zu.

VORSICHT

Schalten Sie Ihren Rechner vor dem Einsetzen oder Entfernen von Modulen immer aus. Die Nichtbeachtung dieser Vorsichtsmaßnahme kann zu Schäden am Rechner oder zu unkontrollierten Systemfunktionen führen.

Kennzeichnung und Installation

Das Erweiterte Funktions/Speicher-Modul HP82180A ist durch die Bezeichnung X-FUNCTIONS auf der Vorderseite des Moduls gekennzeichnet. Das Erweiterte Speicher-Modul HP82181A trägt die Bezeichnung X-MEMORY.

Die Installation eines Erweiterten Funktions/Speicher-Moduls ist einfach. Sie brauchen lediglich das Modul in eine der Buchsen des Rechners zu schieben, bis es leicht einrastet.



Wenn Sie das Modul wieder entfernen wollen, müssen Sie zunächst das ausziehbare Griffstück mit dem Fingernagel etwas vorziehen. Danach können Sie das Modul am Griffstück anfassen und es ganz herausziehen.



Zulässige Konfigurationen

Das Erweiterte Funktions/Speicher-Modul kann in jede der Anschlußbuchsen des Rechners eingesetzt werden. Wenn Sie zusätzlich zum Erweiterten Funktions/Speicher-Modul nur ein einziges Erweitertes Speicher-Modul verwenden wollen, so können Sie dieses in jede der verbleibenden Buchsen einsetzen.

Wenn Sie später ein zweites Erweitertes Speicher-Modul hinzufügen wollen (oder sofort zwei Erweiterte Speicher-Module verwenden wollen), so müssen diese in einer der unten gezeigten Anordnungen eingesteckt werden. Die beiden Erweiterten Speicher-Module dürfen nicht übereinander eingesetzt werden.

X MEMORY	X MEMORY

X MEMORY	X MEMORY

X MEMORY	
	X MEMORY

	X MEMORY
X MEMORY	

**Zulässige Belegungen der Einschubbuchsen
bei Verwendung von zwei Erweiterten Speicher-Modulen HP82181A.**

Wenn Sie ein Modul oder mehrere der Module entfernen, kann ein Teil oder die Gesamtheit der Daten im erweiterten Speicher verloren gehen. Es gibt eine optimale Reihenfolge zum Entfernen der Module, die den Datenverlust minimiert. 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 die Module zum gleichen Zeitpunkt eingesteckt wurden, so ist zunächst das Modul in Buchse 2 oder 4 und danach das Modul in Buchse 1 oder 3 (falls notwendig) zu entfernen.

Die Gründe für diese Prozedur werden in Abschnitt 3 erläutert.

Gebrauch dieses Handbuchs

Sobald Sie das Erweiterte Funktions/Speicher-Modul in Ihren Rechner eingesetzt haben, können Sie die dadurch implementierten Funktionen verwenden. Diese Funktionen werden in zwei Abschnitten beschrieben: Erweiterte Funktionen (Abschnitt 2) und Erweiterter Speicher (Abschnitt 3). Der Abschnitt 2 behandelt programmierbare Funktionen, Funktionen zur Manipulation von Registerinhalten und Flags sowie Funktionen zur Manipulation von Daten im ALPHA-Register. Der Abschnitt 3 behandelt die Übertragung von Daten und Programmen zwischen dem Hauptspeicher des Rechners und dem erweiterten Speicher sowie Editierungsbefehle für die Aufbereitung von ASCII Files, die im erweiterten Speicher angelegt werden können.

Der Einfachheit halber werden die durch das Erweiterte Funktions/Speicher-Modul implementierten Funktionen in diesem Handbuch durch farbige Tastensymbole – wie etwa ANUM – dargestellt. Wenn Sie eine der Funktionen ausführen wollen, können Sie entweder die Tastenfolge XEQ ALPHA *Name* ALPHA verwenden oder die Funktion mittels ASN (oder PASN) einer Taste zuweisen und dann im USER Modus diese Taste drücken.

Der Beschreibung einer jeden Funktion geht eine Zusammenfassung der für diese Funktion benötigten Informationen voran. Dadurch sehen Sie sofort, welche Vorgaben zur Ausführung der Funktion notwendig sind. Zum Beispiel:

PASN	X	Tastencode	ALPHA	Funktionsname
-------------	----------	-------------------	--------------	----------------------

Dies besagt, daß vor der Ausführung von **PASN** (über das Tastenfeld oder innerhalb eines Programms) das X-Register mit einem Tastencode und das ALPHA-Register mit einem Funktionsnamen geladen werden muß.

Wenn der Rechner eine Fehlermeldung anzeigt, finden Sie eine Erläuterung der Fehlerursache in Anhang B.

Erweiterte Funktionen

Datenspeicherregister Operationen

REGMOVE

X

sss.dddnnn

Die Funktion **REGMOVE** (*register move*) kopiert den Inhalt eines Blocks von *nnn* Registern, beginnend mit Register *sss* (Quelle), in einen Block gleicher Länge, beginnend mit Register *ddd* (Ziel). Sämtliche im Zielblock zuvor vorhandenen Daten werden überschrieben.

REGSWAP

X

sss.dddnnn

Die Funktion **REGSWAP** (*register swap*) vertauscht den Inhalt eines Blocks von *nnn* Registern, beginnend mit Register *sss*, mit dem Inhalt eines Blocks gleicher Länge, beginnend mit Register *ddd*.

Wenn bei der Ausführung von **REGMOVE** oder **REGSWAP** *nnn* gleich Null ist, wird ein Register kopiert oder vertauscht.

Flag Operationen

Es ist oft wünschenswert, die Stati der Rechnerflags in einer gegebenen Systemkonfiguration zu erhalten – zum Beispiel um das ursprüngliche Anzeigeformat nach der Ausführung eines Programms wieder herstellen zu können. Die beiden folgenden Funktionen ermöglichen es Ihnen, den Status der Flags 0 bis 43 als Daten zu speichern bzw. mit Hilfe dieser Daten den Rechnerstatus dann später zu restaurieren.

RCLFLAG

Die Funktion **RCLFLAG** (*recall flags*) legt die Stati der Flags 0 bis 43 als ALPHA-Daten im X-Register ab. Sie können den Inhalt des X-Registers anschließend für eine spätere Verwendung in einem Datenspeicherregister abspeichern.

Hinweis: Der Inhalt der Anzeige nach der Ausführung von **RCLFLAG** kann nicht interpretiert werden und ist ohne Bedeutung.

STOFLAG

X

Flagstatus

STOFLAG

X

bb.ee

Y

Flagstatus

Die Funktion **STOFLAG** (*restore flags*) restauriert die Stati der Rechnerflags 0 bis 43 gemäß dem Inhalt des X-Registers. Das X-Register muß zuvor mit dem alten Flagstatus geladen worden sein. Dies kann beispielsweise durch die Ausführung von **RCLFLAG** geschehen.

Wenn Sie nur einen Teil der Flags restaurieren wollen, muß sich der Flagstatus vor der Ausführung von `STOFLAG` im Y-Register befinden, und das X-Register muß eine Zahl der Form *bb.ee*, die den Anfang und das Ende des zu restaurierenden Flagblocks darstellt, enthalten.

Beispiel: Angenommen Sie wollen ein Programm schreiben, das Ausgaben im Format `FIX0` enthält. Nach der Ausführung des Programms soll das ursprüngliche Anzeigeformat restauriert werden. Dies kann mit Hilfe der folgenden Programmzeilen geschehen.

```

01 LBL^ABC
02 RCLFLAG
03 STO 20
04 FIX 0
05 CF 29
06 .
07 .
08 .
. .
. .
. .
20 RCL 20
21 STOFLAG
22 END
```

Diese beiden Schritte übertragen den Status der Flags 0 bis 43 in das X-Register und retten den ermittelten Status in Register 20. Zu den geretteten Flags gehört Flag 29, der Zifferngruppierungsflag, sowie die Flags 36 bis 41, die Stellenanzahl- und Anzeigeformat-Flags.

Diese beiden Schritte definieren das Anzeigeformat für Ihr Programm.

Ihr Programm.

Diese Schritte laden den ursprünglichen Flagstatus in das X-Register und setzen die Flags gemäß dem geladenen Status.

X<>F

X

Status der Flags 0 bis 7

Die Funktion `X<>F` (*X exchange flags*) verwendet die Zahl im X-Register, um die Flags 0 bis 7 zu setzen. Gleichzeitig wird der alte Status dieser Flags in das X-Register übertragen.

Im X-Register wird der Flagstatus durch eine Zahl zwischen 0 und 255 dargestellt. Jedem Flag ist eine Zahl zugeordnet, die einer Potenz von 2 entspricht. Die Zahl im X-Register ist dann die Summe der allen gesetzten Flags zugeordneten Zahlen. Die den Flags zugeordneten numerischen Äquivalente sind:

Flag	0	1	2	3	4	5	6	7
Numerisches Äquivalent	1	2	4	8	16	32	64	128

Beispiel: Es sei unterstellt, daß die Flags 0, 3, 5 und 7 gesetzt sind, während die Flags 1, 2, 4 und 6 gelöscht sind. Welche Zahl wird bei der Ausführung von `X<>F` im X-Register abgelegt? Dazu sind lediglich die numerischen Äquivalente der gesetzten Flags zu addieren:

Flag	numerisches Äquivalent
0	1
3	8
5	32
7	128
	<hr/> 169

Die Zahl im X-Register.

Wenn bei der Ausführung von $\boxed{X<>F}$ das X-Register den Wert Null enthält, werden die Flags 0 bis 7 gelöscht, und das X-Register wird mit dem alten Status dieser Flags geladen.

Mit Hilfe von $\boxed{X<>F}$ können Sie zusätzliche Vielzweck-Flags erzeugen, in dem Sie Zahlen, die den Status der Flags 0 bis 7 darstellen, in Speicherregistern ablegen. Wenn Sie dann den Status eines der Zusatz-Flags abfragen wollen, brauchen Sie lediglich den Flagstatus-Code mittels \boxed{RCL} in das X-Register zu laden, $\boxed{X<>F}$ auszuführen, und anschließend mit $\boxed{FS?}$ in der üblichen Weise abfragen.

$\boxed{X<>F}$ ermöglicht die Verwendung einer großen Anzahl von Flags innerhalb von Programmen. Die Flags sind in Gruppen zu je 8 Flags eingeteilt und werden mittels $\boxed{X<>F}$ in die ersten 8 Flagpositionen übertragen. Die den Status einer Gruppe von acht Flags darstellende Zahl wird in einem Speicherregister abgelegt und kann bei Bedarf von dort abgerufen werden. In diesem Fall wird sie in das X-Register geladen und durch den Status der momentan in den ersten acht Positionen befindlichen Flags ersetzt. Anschließend kann der Status von einzelnen Flags in dieser Gruppe abgefragt oder geändert werden.

USER Modus Operationen



In analoger Weise zu \boxed{ASN} weist auch die Funktion \boxed{PASN} (*programmable assign*) einer Tastenposition eine Funktion oder ein Programm zu. Im Unterschied zu \boxed{ASN} kann \boxed{PASN} jedoch auch innerhalb eines Programms ausgeführt werden. Zur Ausführung von \boxed{PASN} müssen Sie den Tastencode derjenigen Taste, der Sie eine Funktion oder ein Programm zuweisen wollen, eingeben. Dieser Tastencode ist identisch mit dem Code, den der Rechner von sich aus anzeigt, wenn Sie mit \boxed{ASN} einer Taste eine Funktion oder ein Programm zuordnen. Eine vollständige Beschreibung der Tastencodes ist im *HP-41C/CV Bedienungs- und Programmierhandbuch* zu finden. Beachten Sie, daß umgeschaltete Tasten als negative Zahlen dargestellt werden.

Wie mit \boxed{ASN} können auch mit \boxed{PASN} den vier Tasten der ersten Reihe des Tastenfelds (Tastencodes 01 bis 04) und der Umschalttaste (Tastencode 31) keine Programme zugeordnet werden.

Wenn bei der Ausführung von \boxed{PASN} das ALPHA-Register gelöscht ist, wird eine gegebenenfalls bereits vorhandene Zuordnung der spezifizierten Taste wieder aufgehoben.



Durch die Ausführung von \boxed{CLKEYS} (*clear keys*) werden alle vorhandenen Tastenzuordnungen wieder aufgehoben.

ALPHA-String Operationen

Das Erweiterte Funktions/Speicher-Modul ermöglicht Ihnen die Übertragung von Daten zwischen dem ALPHA- und dem X-Register. Im ALPHA-Register existieren die Daten als alphanumerische oder numerische Zeichen, während im X-Register jedes alphanumerische oder numerische Zeichen durch einen numerischen Zeichencode dargestellt wird.

Innerhalb des Rechners werden alphanumerische Zeichen durch Zeichencodes dargestellt, die auf dem ASCII-Code (American Standard Code for Information Interchange) basieren. Zusätzlich zu den Ziffern und den Buchstaben des Alphabets, die dem ASCII-Standard entsprechen, gibt es einige nichtstandard Symbole, die durch eindeutige HP-41 Codes dargestellt werden. Die folgende Tabelle listet alle Symbole, die im ALPHA Modus angezeigt werden können, und die zugehörigen Zeichencodes.

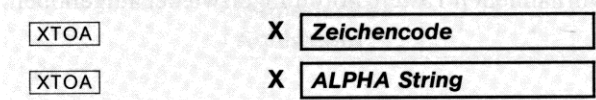
Darstellbare Zeichen und äquivalente Codes

Zeichen	Code	Zeichen	Code	Zeichen	Code
-	0	3	51	N	78
π	1	4	52	O	79
π	4	5	53	P	80
π β	5	6	54	Q	81
π	6	7	55	R	82
μ	12	8	56	S	83
μ	13	9	57	T	84
≠	29	:	58	U	85
Blank	32	;	59	V	86
!	33	<	60	W	87
"	34	=	61	X	88
#	35	>	62	Y	89
\$	36	?	63	Z	90
%	37	@	64	[91
&	38	A	65	\	92
'	39	B	66]	93
(40	C	67	^	94
)	41	D	68	_	95
*	42	E	69	τ	96
+	43	F	70	a	97
,	44	G	71	b	98
-	45	H	72	c	99
.	46	I	73	d	100
/	47	J	74	e	101
0	48	K	75	Σ β	126
1	49	L	76	τ	127
2	50	M	77		

Übertragen von Zeichen zwischen ALPHA- und X-Register

ATOX

Bei der Ausführung von **ATOX** (*ALPHA to X*) wird das linksbündige Zeichen im ALPHA-Register aus dem ALPHA-Register geschoben, und der zugehörige Zeichencode wird im X-Register abgelegt. Wenn das ALPHA-Register leer ist, wird der Wert Null in das X-Register übertragen.



Wenn das X-Register bei der Ausführung von **XTOA** (*X to ALPHA*) einen Zeichencode aus der obigen Tabelle enthält, wird das durch den Zeichencode dargestellte Zeichen an das rechte Ende des im ALPHA-Register befindlichen Strings angehängt. Bei der Ausführung von **XTOA** kann sich im X-Register jede beliebige Zahl zwischen 1 und 255 befinden; jedoch werden bei der Verwendung von Zahlen, die in der obigen Tabelle nicht als gültige Zeichencodes aufgeführt sind, nicht interpretierbare Zeichen angehängt. (Im allgemeinen werden alle zu der betreffenden Zeichenposition gehörenden Segmente der Anzeige eingeschaltet.) Wenn sich bei der Ausführung **XTOA** im X-Register die Zahl Null befindet, können nachfolgende ALPHA-Register Operationen unter Umständen fehlerhafte Ergebnisse liefern. Es wird empfohlen, in einem solchen Fall das ALPHA-Register zu löschen. (Der Anhang C enthält Informationen über Einschränkungen bei ALPHA-Operationen mit dem Nullzeichen.) Wenn das X-Register bei der Ausführung von **XTOA** einen ALPHA-Datenstring enthält, wird der gesamte String an den Inhalt des ALPHA-Registers angehängt.

Länge eines Zeichenstrings

ALENG

Die Funktion **ALENG** (*ALPHA length*) ermittelt die Anzahl der Zeichen im ALPHA-Register und legt den gefundenen Wert im X-Register ab.

Durchsuchen von ALPHA-Strings

ANUM

Die Funktion **ANUM** (*ALPHA number*) durchsucht das ALPHA-Register nach einer ALPHA-formatierten Zahl. Wenn eine Zahl gefunden wird, legt der Rechner den Wert im X-Register ab und setzt den Anwenderflag 22. Wenn keine Zahl gefunden wird, bleiben X-Register und Flag 22 unverändert.

Der durch die Ziffern im ALPHA-Register gebildete Wert muß einem der Anzeigeformate des HP-41 entsprechen. Gruppen- und Dezimaltrennzeichen werden entsprechend dem Status der Flags 28 und 29 interpretiert. Wenn das ALPHA-Register beispielsweise den String **PRICE: \$1,234.5** enthält, liefert **ANUM** in Abhängigkeit des Zustands von Flag 28 und 29 die folgenden Resultate:

Flag 28	Flag 29	Angezeigte Zahl
gesetzt	gesetzt	1,234.5000
gesetzt	gelöscht	1.0000
gelöscht	gesetzt	1,2345
gelöscht	gelöscht	1,2340

Wenn sich vor den Ziffern im ALPHA-Register ein Minuszeichen befindet, wird bei der Ausführung von **ANUM** eine negative Zahl im X-Register abgelegt.

POSA

X **Zeichencode**

POSA

X **ALPHA String**

Die Funktion **POSA** (*position in ALPHA*) durchsucht das ALPHA-Register nach dem im X-Register spezifizierten ALPHA-Zeichen oder String. Das Zeichen oder der String kann auf zwei Arten spezifiziert werden. Für ein einzelnes Zeichen können Sie den zugehörigen Zeichencode direkt eingeben, oder Sie können das aktuelle Zeichen oder die Zeichenkette mittels **ASTO** im X-Register abspeichern. Wenn das spezifizierte Zeichen oder der spezifizierte String im ALPHA-Register gefunden wird, so wird die Position des Zeichens (bzw. die Position des ersten Zeichens des Strings) in das X-Register zurückgegeben. (In Anhang C sind zusätzliche Informationen über ALPHA-Operationen mit Nullzeichen zu finden.)

Die Positionen der Zeichen werden von links nach rechts gezählt, beginnend mit Position 0. Wenn das ALPHA-Register den spezifizierten String mehr als einmal enthält, wird die Position des ersten Auftretens zurückgegeben. Wenn das ALPHA-Register den gesuchten String nicht enthält, wird die Zahl -1 zurückgegeben.

Rundumverschieben des ALPHA-Registerinhalts

AROT

X

Anzahl der Zeichen

Die Ausführung von AROT (ALPHA rotate) bewirkt eine Rundumverschiebung des Inhalts des ALPHA-Registers um die durch das X-Register spezifizierte Anzahl von Zeichen. Ein positiver X-Registerinhalt bedingt eine Linksverschiebung, ein negativer X-Registerinhalt eine Rechtsverschiebung.

AROT kann zusammen mit ANUM und POSA dazu benutzt werden, um Folgen von Zahlen aus dem ALPHA-Register zu isolieren.

Beispiel: Das ALPHA-Register enthalte nach einer Datenübertragung von einer Peripherieeinheit die Folge 68.2 69.88 (zwei durch ein Leerzeichen getrennte Zahlen). Sie wollen diese beiden Zahlen nacheinander aus dem ALPHA-Register herausziehen, um Sie in einem Programm zu verwenden.

Die folgende Befehlssequenz illustriert diesen Vorgang:

Tastenfolge	Anzeige	
XEQ ALPHA ANUM	XEQ ANUM_	
ALPHA	68.2000	Übertragen der ersten Zahl ins X-Register.
STO 20	68.2000	Speichern des Werts für eine spätere Verwendung.
32	32_	Eingabe des Zeichencodes für ein Leerzeichen.
XEQ ALPHA XTOA	XEQ XTOA_	
ALPHA	32.0000	Anhängen eines Leerzeichens an die Zeichenkette 69.88 im ALPHA-Register.
XEQ ALPHA POSA	XEQ POSA_	
ALPHA	4.0000	Durchsuchen des ALPHA-Registers nach dem ersten Auftreten eines Leerzeichens (das X-Register enthielt den Code 32).
XEQ ALPHA AROT	XEQ AROT_	
ALPHA	4.000	Rundumverschieben des ALPHA-Registers um vier Zeichen nach links. Das ALPHA-Register hat jetzt den Inhalt 69.88 68.2. Beachten Sie, daß sich ohne die XTOA Operation der String 69.8868.2 ergeben würde.
XEQ ALPHA ANUM	XEQ ANUM_	
ALPHA		Ablegen der Zahl 69.88 im X-Register.

Sonstige Operationen

Tastenabfrage

GETKEY

Bei der Ausführung von GETKEY (get key) wird die Programmausführung solange unterbrochen, bis eine Taste gedrückt wird oder ein Zeitintervall von ungefähr 10 Sekunden abgelaufen ist. Wenn eine Taste gedrückt wird, enthält das X-Register danach den entsprechenden Tastencode. Wenn keine Taste gedrückt wird, enthält das X-Register nach Ablauf des Zeitintervalls den Wert 0.

GETKEY reagiert auf die zuerst gedrückte Taste, daher sind keine umgeschalteten Tastenfunktionen als Antwort auf eine **GETKEY** Operation möglich. Wenn Sie die goldfarbene Umschalttaste (SHIFT) während eines **GETKEY** Warteintervalls drücken, wird der zugehörige Tastencode (31) im X-Register abgelegt.

GETKEY ermöglicht Ihnen, in Abhängigkeit von der Eingabe über das Tastenfeld zu verschiedenen Unterprogrammen zu verzweigen. Dies gilt sogar dann, wenn die gedrückte Taste keine Zifferntaste ist.

Zuordnen von Datenspeicherregistern

PSIZE X **Anzahl der Datenspeicherregister**

PSIZE und **SIZE** sind identisch mit der Ausnahme, daß die Funktion **PSIZE** auch innerhalb eines Programms ausgeführt werden kann. Dadurch kann ein Programm während der Ausführung die Register des Hauptspeichers nach den aktuellen Anforderungen neu aufteilen.

SIZE?

Bei der Ausführung von **SIZE?** wird die Anzahl der momentan als Datenspeicher zugeordneten Register im X-Register abgelegt.

SIZE? kann innerhalb eines Programms zur Verhinderung der Ausführung von **PSIZE** verwendet werden, wenn eine Neuaufteilung des Hauptspeichers nicht erforderlich ist.

Das folgende Programmsegment illustriert die Verwendung von **SIZE?** und **PSIZE** innerhalb eines Programms.

01	.	}	Ihr Programm.
02	.		
.	.		
.	.		
07	SIZE?		Bestimmen der Anzahl der momentan zugeordneten Datenspeicherregister. (Das Resultat wird im X-Register abgelegt.)
08	<i>nn</i>		Eingabe der Anzahl der von Ihrem Programm benötigten Register. Das Ergebnis der vorangegangenen Operation befindet sich nun im Y-Register.
09	X > Y?		Abfrage ob das Programm mehr Speicherregister benötigt, als momentan zugeordnet sind.
10	PSIZE		Neuaufteilung des Hauptspeichers, wenn die obige Bedingung erfüllt ist.

Löschen von Programmen

PCLPS

ALPHA **Programmname**

Die Funktion **PCLPS** (*programmable clear programs*) dient zum Löschen von einem oder mehreren im Hauptspeicher befindlichen Programmen. Beginnend mit dem im ALPHA-Register spezifizierten Programm (bzw. dem momentanen Programm, wenn das ALPHA-Register leer ist) werden alle dahinterstehenden Programme bis zum Ende des Programmspeichers gelöscht. Wenn ein ablaufendes Programm sich selbst spezifiziert (oder das ALPHA-Register löscht) und danach **PCLPS** ausführt, werden das ablaufende Programm und alle Folgeprogramme gelöscht, und die Programmausführung bricht ab.

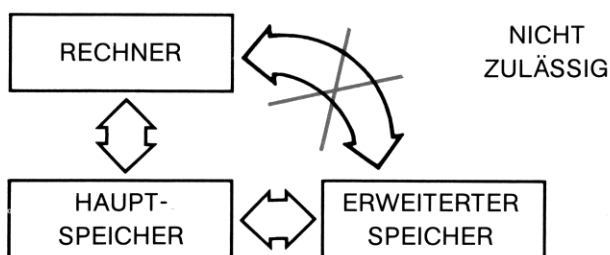
Erweiterter Speicher

Aufbau des erweiterten Speichers

Um das Erweiterte Funktions-/Speicher-Modul und die Erweiterten Speicher-Module effektiv nutzen zu können, sollten Sie den Unterschied zwischen dem Hauptspeicher Ihres Rechners und dem erweiterten Speicher, der durch diese Module verfügbar wird, verstanden haben.

Ihr Rechner selbst enthält eine gewisse Menge an Hauptspeicher. Wenn Sie einen HP-41C besitzen, können Sie die Größe des Hauptspeichers durch Verwendung von Speichererweiterungsmodulen bis auf die Kapazität eines HP-41CV steigern. Unabhängig von der Größe enthält der Hauptspeicher Programme und Daten, die für den Rechner immer direkt verfügbar sind. Sie brauchen nur **[XEQ]** zu drücken und einen Programmnamen eingeben oder **[RCL]** drücken und eine Registeradresse eingeben, wenn Sie ein Programm ausführen bzw. Daten abrufen wollen.

Der erweiterte Speicher unterscheidet sich vom Hauptspeicher des Rechners. Wenn der Rechner im erweiterten Speicher abgelegte Programme oder Daten verwenden will, müssen diese zuvor in den Hauptspeicher übertragen werden; der direkte Zugriff auf Daten oder Programme im erweiterten Speicher ist nicht möglich. Der erweiterte Speicher liefert Ihnen zusätzlichen Speicherplatz für Programme und Daten; für diese erhöhte Speicherkapazität müssen Sie jedoch die Notwendigkeit zusätzlicher Schritte zur Übertragung der Daten zwischen erweitertem Speicher und Hauptspeicher in Kauf nehmen.



Die Register im erweiterten Speicher* sind in Strukturen organisiert, die als „Files“ bezeichnet werden. Ein Programm, das Sie im Hauptspeicher erzeugt haben, kann im erweiterten Speicher als ein File bestimmten Typs abgelegt werden. Ein File besteht aus zwei Registern (dem sog. *Header*), die Informationen über den File selbst enthalten, und einem oder mehr Registern, die Daten enthalten. Auf den letzten File folgt ein weiteres Register, das zur Trennung zwischen dem belegten und dem nicht belegten Teil des erweiterten Speichers dient.

* Siehe Fußnote Seite 5.

Im erweiterten Speicher werden Files in der Reihenfolge gespeichert, in der Sie angelegt werden. Ein File ist nicht auf einen Modul beschränkt, er kann sich über mehrere Module erstrecken. In der Illustration rechts beginnt der File 5 im Erweiterten Speicher-Modul A, die letzten Register des Files liegen jedoch bereits im Erweiterten Speicher-Modul B. Aus diesem Grund ist es wichtig, daß Sie beim Entfernen von Erweiterten Speicher-Modulen die richtige Reihenfolge beachten. Wenn Sie ein Modul entfernen, wollen Sie so wenig Files wie möglich verlieren. In unserer Illustration würden Sie durch das Entfernen des Erweiterten Speicher-Moduls B nur die Files 5 und 6 verlieren. Wenn Sie jedoch das Erweiterte Speicher-Modul A herausnehmen, würde im gesamten erweiterten Speicher nur der File 1 als einziger File vollständig erhalten bleiben.

Die Richtlinien für das Entfernen von Erweiterten Speicher-Modulen in Abschnitt 1 basieren auf dem folgenden Prinzip: Wenn alle erweiterten Speicherregister im Erweiterten Funktions-/Speicher-Modul belegt sind und weitere Daten gespeichert werden sollen, prüft das Erweiterte Funktions-/Speicher-Modul das System darauf ab, ob ein oder zwei Erweiterte Speicher-Module installiert sind. Wenn nur ein Erweitertes Speicher-Modul vorhanden ist, werden die zusätzlichen Daten unabhängig von der verwendeten Anschlußbuchse in dessen Registern gespeichert. Wenn Sie später ein zweites Erweitertes Speicher-Modul einsetzen, wird dieses erst verwendet, wenn das erste Modul überläuft.

Wenn zu dem Zeitpunkt, an dem das Erweiterte Funktions-/Speicher-Modul zusätzliche Register benötigt, zwei Erweiterte Speicher-Module installiert sind, werden die Daten zuerst in der Anschlußbuchse 1 oder 3 gespeichert. Das Erweiterte Speicher-Modul in der Anschlußbuchse 2 oder 4 wird nur dann verwendet, wenn das erste Modul voll ist.

Man unterscheidet drei Arten von Files, die im erweiterten Speicher abgelegt bzw. von dort zurückgeladen werden können:

- Datenfiles
- ASCII-Files
- Programmfiles

File 1
File 2

**Erweitertes
Funktions/Speicher-
Modul**

File 2
File 3
File 4
File 5

**Erweitertes
Speicher-
Modul „A“**

File 5
File 6
Trennregister
Unbenutzte Register

**Erweitertes
Speicher-
Modul „B“**

Filenamen und File-Pointer

Wie bereits erwähnt, enthält jeder File zwei Register, die als Header bezeichnet werden. In ihnen werden vom Rechner benötigte Informationen gespeichert. Das erste Register enthält den Filenamen; dies kann eine beliebige, maximal sieben Zeichen lange Kombination aus alphanumerischen Zeichen sein. Wenn Sie versuchen, einen längeren Namen zu spezifizieren, werden die überzähligen Zeichen abgeschnitten. Wenn Sie einen kürzeren Namen spezifizieren, füllt der Rechner die verbleibenden Stellen mit Leerzeichen auf.

Das zweite Register des Headers enthält Informationen über Länge und Typ des Files sowie einen bzw. zwei *Pointer* (im Deutschen etwa „Zeiger“), die zum Zugriff auf bestimmte Elemente im File verwendet werden.

Datenfiles

Datenfiles ermöglichen Ihnen das Erhalten wichtiger Daten, wenn Sie sämtliche Register des Hauptspeichers für andere Zwecke benötigen. Bei den meisten Datenfile-Operationen wird der „Register-Pointer“ des Files benutzt. Dies ist ein ganzzahliger Wert, der auf ein bestimmtes Register innerhalb des Files deutet. In dem Beispiel rechts zeigt der Pointer auf das fünfte Register des Datenfiles.

Register	HEADER
	HEADER
000	1.900 01
001	2.700 01
002	4.600 01
003	6.300 01
004	1.090 02
005	1.720 02
006	2.810 02
007	4.530 02
008	7.340 02
009	1.187 03

Struktur eines Datenfiles

Viele Funktionen in diesem Abschnitt werden anhand von Beispielen erläutert. Um diese Beispiele nachvollziehen zu können, sollten Sie die folgende Tastenfolge eingeben. Sie legen damit einen Datenfile an und beschreiben ihn mit Daten. Die benötigten Funktionen **CRFLD** und **SAVEX** werden später innerhalb dieses Abschnitts beschrieben (Seite 24 und Seite 28).

Tastenfolge

[ALPHA] SAMPL-D
[ALPHA] 20
[XEQ] [ALPHA] CRFLD
[ALPHA]
[■] [ASN] [ALPHA]
SAVEX [ALPHA] [Σ+]

Anzeige

SAMPL-D_
20_
XEQ CRFLD_
20.0000
ASN_
ASN SAVEX 11
20.0000
20.0000
19.0000
27.0000
46.0000
63.0000
109.0000
172.0000
281.0000
453.0000
734.0000
1,187.0000
ASN_
1,187.0000

Zuordnen von **SAVEX** der Taste **Σ+**.
Anzeige des X-Registerinhalts.
Umschalten auf USER Modus.

[USER]
19 [Σ+]
27 [Σ+]
46 [Σ+]
63 [Σ+]
109 [Σ+]
172 [Σ+]
281 [Σ+]
453 [Σ+]
734 [Σ+]
1187 [Σ+]

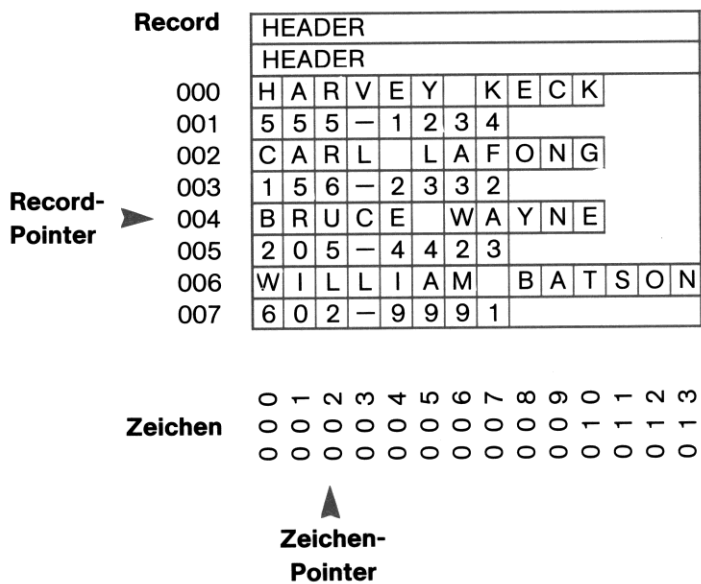
[■] [ASN] [ALPHA]
[ALPHA] [Σ+]

Aufheben der **SAVEX** Zuordnung der **Σ+** Taste.

Die obigen Daten sind nun in den ersten 10 Registern des Files SAMPL-D gespeichert; die restlichen Register des Files sind mit Null besetzt.

ASCII-Files

ASCII-Files dienen zur Speicherung von Textblöcken aus alphanumerischen Zeichen und können nach bestimmten Zeichenfolgen durchsucht sowie korrigiert (editiert) werden. Um Einschränkungen, die durch Verwendung des Begriffs Register in Bezug auf diese Files entstehen würden, zu vermeiden, sind ASCII-Files in Records (Datensätze) eingeteilt, die wiederum aus den Zeichen bestehen, wie die folgende Illustration zeigt. Jeder Record in einem ASCII-File kann bis zu 254 Zeichen an Information enthalten. Wie Sie der Illustration entnehmen können, verfügt jeder ASCII-File über zwei Pointer – einen Record-Pointer und einen Zeichen-Pointer.



In der obigen Illustration deutet der Record-Pointer auf den fünften Record und der Zeichen-Pointer auf das dritte Zeichen. Als Kombination deuten sie auf das **U** in **BRUCE WAYNE**. Sie können einem ASCII-File mit dem Inhalt der obigen Illustration in Ihrem Rechner erzeugen, in dem Sie die folgende Tastenfolge eingeben. Die dabei verwendeten Funktionen `CRFLAS` und `APPREC` werden später in diesem Abschnitt beschrieben (Seite 24 und 29).

Tastensfolge

ALPHA SMPL-AS

ALPHA 20

XEQ ALPHA CRFLAS

ALPHA

■ ASN ALPHA

APPREC ALPHA Σ+

ALPHA HARVEY KECK

ALPHA Σ+

ALPHA 555-1234

ALPHA Σ+

ALPHA CARL LAFONG

ALPHA Σ+

ALPHA 156-2332

ALPHA Σ+

ALPHA BRUCE WAYNE

ALPHA Σ+

ALPHA 205-4423

ALPHA Σ+

ALPHA WILLIAM BATSON

ALPHA Σ+

ALPHA 602-9991

ALPHA Σ+

■ ASN ALPHA

ALPHA Σ+

Anzeige

XEQ SMPL-AS_

20_

XEQ CRFLAS_

20.0000

ASN _

ASN APPREC 11

20.0000

HARVEY KECK_

20.0000

555-1234_

20.0000

CARL LAFONG_

20.0000

156-2332_

20.0000

BRUCE WAYNE_

20.0000

205-4423_

20.0000

WILLIAM BATSON_

20.0000

602-9991_

20.0000

ASN _

20.0000

Programmfiles

Programmfiles dienen zur Speicherung von Programmen im erweiterten Speicher. Sie können mehr Platz im Hauptspeicher für andere Zwecke freihalten, wenn Sie die meisten Ihrer Programme im erweiterten Speicher ablegen und bei Bedarf von dort abrufen.

Die nächste Tastenfolge enthält ein Programm, das den Inhalt des ASCII-Files SMPL-AS liest und anzeigt. Die letzten drei Schritte der Tastenfolge bewirken die Speicherung des zuvor eingegebenen Programms als Programmfile im erweiterten Speicher. Die Funktionen `SEEKPTA`, `GETREC` und `SAVEP` werden später in diesem Abschnitt beschrieben (Seite 25, 26 und 32).

Tastensequenz

```

PRGM █ GTO . .
█ LBL ALPHA
SAMPL-P ALPHA
ALPHA SMPL-AS ALPHA
0
XEQ ALPHA SEEKPTA ALPHA
█ LBL 01
XEQ ALPHA GETREC ALPHA
ALPHA █ AVIEW ALPHA
█ GTO 01
PRGM
ALPHA SAMPL-P ALPHA
XEQ ALPHA SAVEP
ALPHA
  
```

Anzeige

```

00 REG nn
01 LBLTSAMPL-P
02TSMPL-AS
03 0_
04 SEEKPTA
05 LBL 01
06 GETREC
07 AVIEW
08 GTO 01
20.0000
20.0000
XEQ SAVEP_
20.0000
  
```

Arbeitsfiles

Einige der erweiterten Speicherfunktionen erfordern, daß vor ihrer Ausführung ein Filename in das ALPHA-Register eingegeben wird. Nach der Ausführung ist der Rechner dann auf den betreffenden File positioniert*. Dies bedeutet, daß dieser File zu Ihrem *Arbeitsfile* geworden ist, genauso wie durch die Ausführung von `GTO` unter Angabe eines Programmnamens das spezifizierte Programm im Hauptspeicher zu Ihrem *Arbeitsprogramm* wird. Einige der im folgenden beschriebenen Funktionen wirken nur auf Arbeitsfiles.

Fileverwaltung

EMDIR

Die Ausführung von `EMDIR` (*extended memory directory*) bewirkt die Anzeige eines Verzeichnisses aller Files im erweiterten Speicher. Das Verzeichnis kann auch ausgedruckt werden. Für jeden File erscheint links der Filename gefolgt vom Filetyp (der durch die Buchstaben D, A oder P angedeutet wird) und der Anzahl der von dem File belegten Register. Nach der Anzeige des Verzeichnisses wird die Anzahl der für das Speichern von Files noch verfügbaren erweiterten Speicherregister im X-Register abgelegt.

Beispiel: Wenn Sie die bisher in diesem Abschnitt beschriebenen Files angelegt haben, wird durch die Ausführung von `EMDIR` das folgende Verzeichnis gelistet:

*Eine Ausnahme bildet die Funktion `PURFL` (*purge file*), die den angegebenen File aus dem Speicher löscht.

Filename Filetyp
 Anzahl der belegten Register

SAMPL-D D020
SMPL-AS A020
SAMPL-P P005*
549.0000

Anzahl der noch verfügbaren Register (bei Verwendung von zwei Erweiterten Speicher-Modulen).

Wenn der erweiterte Speicher keine Files enthält, wird die Meldung **DIR EMPTY** angezeigt und im X-Register die Anzahl der für das Speichern von Files zur Verfügung stehenden Register abgelegt.

Sie können das Anzeigen (oder Drucken) des Verzeichnisses unterbrechen, in dem Sie eine beliebige Taste mit Ausnahme von **R/S** oder **ON** drücken; die Ausgabe des Verzeichnisses wird fortgesetzt, sobald Sie die Taste wieder loslassen. Wenn Sie **R/S** oder **ON** drücken, wird die Ausgabe des Verzeichnisses beendet. Wenn Sie die Ausgabe des Verzeichnisses abbrechen, während ein Filename angezeigt wird, so wird der betreffende File zum Arbeitsfile (sofern er ein Daten- oder ASCII-File ist).

CRFLD	X	Anzahl der Register	ALPHA	Name eines Datenfiles
CRFLAS	X	Anzahl der Register	ALPHA	Name eines ASCII-Files

Die Funktionen **CRFLD** (*create file-data*) und **CRFLAS** (*create file-ASCII*) dienen zum Anlegen eines Daten- bzw. ASCII-Files. Bei der Ausführung von **CRFLD** muß im X-Register die Anzahl der zu speichernden Datenelemente spezifiziert werden. Dabei brauchen die beiden zusätzlichen Register für den Header nicht angegeben zu werden; der Rechner berücksichtigt diese Register automatisch.

Wenn Sie bei der Anlage eines ASCII-Files die Anzahl der für den File vorgesehenen Zeichen und Records genau kennen, können Sie die Anzahl der für den File erforderlichen Register wie folgt berechnen:

1. Addieren Sie die Anzahl der Zeichen und die Anzahl der Records.
2. Erhöhen Sie das Ergebnis um 1.
3. Dividieren Sie das Ergebnis durch 7 und runden Sie es auf einen ganzzahligen Wert.

Diese Zahl stellt dann die Anzahl der benötigten Register dar.

In der Regel werden Sie bei der Anlage eines ASCII-Files jedoch noch nicht genau wissen, wieviele Zeichen und Records in dem File abgelegt werden sollen. In diesem Fall können Sie als Faustregel verwenden: Schätzen Sie die ungefähre Anzahl der Zeichen, erhöhen Sie den Schätzwert um 20 Prozent, und teilen Sie das Ergebnis durch 7. Das Resultat stellt dann eine Schätzung für die Anzahl der benötigten Register dar.

Beim Anlegen eines Files mit **CRFLD** oder **CRFLAS** wird der spezifizierte File zu Ihrem Arbeitsfile.

PURFL	ALPHA	Filename
--------------	--------------	-----------------

Durch die Ausführung der Funktion **PURFL** (*purge file*) wird der angegebene File aus dem erweiterten Speicher gelöscht.

* Durch das Packen des Speichers und/oder Änderungen der momentanen Speicherkonfiguration kann die Anzahl der von einem Programmfile belegten Register leicht variieren.

CLFL**ALPHA** Filename

Bei der Ausführung von **CLFL** (*clear file*) bleibt der angegebene File erhalten; nur die im File gespeicherten Daten werden gelöscht. (Der angegebene File darf kein Programmfile sein). Bei Datenfiles besetzt **CLFL** alle Register mit Null; bei ASCII-Files wird die Anzahl der Records auf Null gesetzt. Der angegebene File wird zum Arbeitsfile.

FSIZE**ALPHA** Filename

Die Funktion **FSIZE** (*file size*) berechnet die Anzahl der von dem angegebenen File belegten Register und legt das Ergebnis im X-Register ab. Der angegebene File wird zum Arbeitsfile. (Wenn das ALPHA-Register bei der Ausführung von **FSIZE** leer ist, wird die Größe des Arbeitsfiles zurückgegeben.)

Programmfile Operationen

SAVEP**ALPHA** Programmname, Filename**SAVEP****ALPHA** Programmname**SAVEP****ALPHA** , Filename

Bei der Ausführung von **SAVEP** (*save program*) wird das angegebene Programm vom Hauptspeicher in einen File mit dem spezifizierten Namen im erweiterten Speicher kopiert. Wenn im ALPHA-Register nur der Programmname spezifiziert ist, wird das Programm unter diesem Namen gespeichert (d.h. der Programmname wird zum Filenamen). Wenn nur ein Komma und ein Filename angegeben sind, wird dasjenige Programm, auf das der Rechner momentan im Hauptspeicher positioniert ist, unter dem spezifizierten Filenamen gerettet.

Wenn bei der Ausführung von **SAVEP** ein Programmfile mit dem spezifizierten Namen bereits existiert, wird der alte Fileinhalt mit dem neuen Programm überschrieben.

GETP**ALPHA** Name eines Programmfiles

Die Funktion **GETP** (*get program*) ersetzt das letzte im Hauptspeicher befindliche Programm durch das in dem angegebenen File gespeicherte Programm*. Bei der Ausführung von **GETP** über das Tastenfeld wird der Rechner auf die erste Zeile des neuen Programms positioniert. Wenn **GETP** von einem ablaufenden Programm ausgeführt wird, hängt die Positionierung des Rechners davon ab, ob das ablaufende Programm das letzte Programm im Hauptspeicher ist. Wenn das ablaufende Programm nicht das letzte Programm ist, wird die Ausführung normal fortgesetzt. Ist dagegen das ablaufende Programm das letzte Programm im Hauptspeicher, so wird es durch das Programm in dem spezifizierten File ersetzt und die Ausführung mit der ersten Anweisung des neu geladenen Programms fortgesetzt.

GETSUB**ALPHA** Name eines Programmfiles

Die Funktion **GETSUB** (*get subroutine*) kopiert das in dem angegebenen File gespeicherte Programm hinter das letzte Programm im Hauptspeicher. Der Rechner wird nicht auf das neue Programm positioniert.

Eventuell mit dem spezifizierten Programm aufgezeichnete Tastenzuordnungen werden aktiv, sofern **GETP** oder **GETSUB** im USER Modus ausgeführt wird.

* Wenn Sie vor der Ausführung von **GETP** die Tastenfolge ■ **GTO** . . drücken erzeugt der Rechner einen leeren Programmbereich am Ende des Programmspeichers. Bei einer anschließenden Ausführung von **GETP** ersetzt das kopierte Programm das „Leer“-Programm – alle übrigen Programme bleiben erhalten.

Beispiel: Löschen Sie das Programm SAMPL-P im Hauptspeicher und laden Sie es anschließend vom erweiterten Speicher zurück.

Tastenfolge

XEQ ALPHA CLP ALPHA
ALPHA SAMPL-P
ALPHA

ALPHA SAMPL-P
ALPHA
XEQ ALPHA GETSUB
ALPHA
GTO ALPHA SAMPL-P
ALPHA
PRGM
PRGM

Anzeige

CLP _
CLP SAMPL-P_
572.0000

SAMPL-P_
572.000
XEQ GETSUB_
572.0000
GTO SAMPL-P_
572.0000
01 LBL^T SAMPL-P
572.0000

Löschen des Programms SAMPL-P.

(Bei der Anzeige wird das Resultat des letzten Beispiels unterstellt.)
Spezifikation des Programmfiles SAMPL-P.

Kopieren von SAMPL-P an das Ende des Programmspeichers.
Positionieren des Rechners auf SAMPL-P.

Erste Zeile des neu geladenen Programms.

Daten- und ASCII-File Operationen

SEEKPTA

X

rrr

SEEKPTA

X

rrr.ccc

ALPHA

Name eines Datenfiles

ALPHA

Name eines ASCII-Files

Die Funktion **SEEKPTA** (*seek pointer by ALPHA*) macht den angegebenen File zum Arbeitsfile und positioniert den oder die Pointer nach Maßgabe des X-Registerinhalts. Bei einem Datenfile gibt der ganzzahlige Anteil der Zahl im X-Register dasjenige Register an, auf das der Pointer zu positionieren ist. Bei einem ASCII-File bestimmt der ganzzahlige Anteil der Zahl im X-Register die Position des Record-Pointers, und die ersten drei Ziffern des dezimalen Anteils bestimmen die Position des Zeichen-Pointers.

Beispiel: Machen Sie den ASCII-File SMPL-AS zu Ihrem Arbeitsfile und positionieren Sie den Record- und Zeichen-Pointer neu.

Tastenfolge

ALPHA SMPL-AS
ALPHA 6.013

XEQ ALPHA SEEKPTA
ALPHA

Anzeige

SMPL-AS_
6.013_

XEQ SEEKPTA_
6.0130

Spezifikation des Filenamens.
Spezifikation des Records
(006) und des Zeichens (013).

Definition von SMPL-AS als Arbeitsfile
und Positionieren der Pointer auf das „N“ in
„WILLIAM BATSON“.

SEEKPT

X

rrr

SEEKPT

X

rrr.ccc

Die Funktion **SEEKPT** (*seek pointers*) ist identisch mit **SEEKPTA** mit der Ausnahme, daß sie sich stets auf den Arbeitsfile bezieht. Die Ausführung von **SEEKPTA** bei leerem ALPHA-Register hat die gleiche Wirkung wie

SEEKPT

RCLPTA**ALPHA** **Filename**

Die Funktion **RCLPTA** (*recall pointer by ALPHA*) macht den spezifizierten File zum Arbeitsfile und legt die Position des Pointers bzw. der Pointer im X-Register ab. Die Position des Register-Pointers eines Datenfiles wird als ganzzahliger Wert zurückgegeben. Die Positionen des Record-Pointers und des Zeichen-Pointers eines ASCII-Files werden in der Form *rrr.ccc* zurückgegeben, wobei *rrr* der Wert des Record-Pointers und *ccc* der Wert des Zeichen-Pointers ist.

RCLPT

Die Funktion **RCLPT** (*recall pointers*) legt den Wert bzw. die Werte des oder der Pointer des Arbeitsfiles im X-Register ab. Bei leerem ALPHA-Register hat **RCLPTA** die gleiche Wirkung wie **RCLPT**.

Beispiel: Fragen Sie die Werte der Pointer des ASCII-Files aus dem letzten Beispiel ab:

Tastenfolge**Anzeige**

0	0_	Löschen der Anzeige.
XEQ ALPHA RCLPT	XEQ RCLPT_	
ALPHA	6.0130	Abrufen der Pointer-Werte des Arbeitsfiles, die im letzten Beispiel gesetzt wurden.

Wenn Sie bei der Ausführung von **RCLPTA** einen Programmfile spezifizieren, wird die Anzahl der Bytes des Programms im X-Register abgelegt.

Datenfile Operationen

SAVER**ALPHA** **Name eines Datenfiles**

Die Funktion **SAVER** (*save registers*) kopiert alle Datenspeicherregister des Hauptspeichers in den spezifizierten Datenfile (oder in den Arbeitsfile, wenn das ALPHA-Register leer ist). Das erste Register des Hauptspeichers wird in Register 000 des Datenfiles kopiert, das zweite in Register 001, usw. Am Ende des Kopiervorgangs deutet der Register-Pointer des Datenfiles entweder auf das nächste verfügbare Register oder auf das Ende des Datenfiles.

SAVERX**X** **bbb.eee**

Die Funktion **SAVERX** (*save registers by X*) kopiert einen Registerblock vom Hauptspeicher in den Arbeitsfile. Der Arbeitsfile muß ein Datenfile sein. Der zu rettende Registerblock im Hauptspeicher wird durch die Zahl im X-Register definiert. Diese Zahl muß die Form *bbb.eee* haben, wobei *bbb* die Nummer (oder Adresse) des ersten Registers und *eee* die Nummer des letzten Registers angibt. Die Register werden in den Datenfile kopiert; dabei wird mit dem Register (im Datenfile) begonnen, auf das der Register-Pointer momentan zeigt. Nach der Beendigung einer **SAVERX** Operation zeigt der Register-Pointer entweder auf das nächste verfügbare Register oder auf das Ende des Files. **SAVERX** wird nicht ausgeführt, wenn der Arbeitsfile für den zu kopierenden Registerblock zu klein ist.

SAVEX

Die Funktion **SAVEX** (*save X-register*) überträgt den Inhalt des X-Registers in dasjenige Register des Arbeitsfiles, auf das der Register-Pointer momentan deutet. Nach der Ausführung von **SAVEX** deutet der Pointer auf das nächste Register im Datenfile.

Die Tastenfolge auf Seite 21 zum Anlegen und Beschreiben des Files SAMPL-D illustriert auch die Verwendung von **SAVEX**.

GETR**ALPHA** **Name eines Datenfiles**

Die Funktion **GETR** (*get registers*) lädt den Inhalt des spezifizierten Files in den Hauptspeicher zurück. Der Inhalt von Register 000 des angegebenen Files wird in Hauptspeicherregister 000 übertragen, der Inhalt von Register 001 des Datenfiles in das entsprechende Hauptspeicherregister, usw. Die Ausführung von **GETR** bricht ab, wenn das letzte Datenregister im Hauptspeicher oder das Ende des Datenfiles im erweiterten Speicher erreicht ist. Bei leerem ALPHA-Register kopiert **GETR** den Inhalt des Arbeitsfiles in den Hauptspeicher, sofern der Arbeitsfile ein Datenfile ist.

GETRX**X** **bbb.eee**

Die Funktion **GETRX** (*get registers by X*) kopiert Daten aus dem Arbeitsfile in einen Registerblock im Hauptspeicher. Der Registerblock beginnt mit Register *bbb* und endet mit Register *eee*. Die Übertragung der Daten aus dem Arbeitsfile beginnt mit dem durch den Register-Pointer spezifizierten Register. Die Ausführung von **GETRX** bricht ab, wenn sämtliche der spezifizierten Register besetzt worden sind oder das Ende des Arbeitsfiles erreicht ist.

GETX

Die Funktion **GETX** (*get to X-register*) kopiert den Inhalt desjenigen Arbeitsfileregisters, auf das der Register-Pointer momentan positioniert ist, in das X-Register und positioniert den Pointer auf das nächste Register.

Beispiel: Laden Sie den Inhalt eines Registers im File SAMPL-D in den Hauptspeicher zurück.

Tastenfolge

8
ALPHA SAMPL-D
ALPHA
XEQ **ALPHA** SEEKPTA
ALPHA
XEQ **ALPHA** GETX
ALPHA

Anzeige

8_
SAMPL-D_
8.000
XEQ SEEKPTA_
8.0000
XEQ GETX_
734.0000

Pointer soll auf Register 008 positioniert werden.
Spezifizieren des Filenamens.

Definieren von SAMPL-D als Arbeitsfile und
Positionieren des Pointers auf Register 008.

Inhalt von Register 008 des Arbeitsfiles.

ASCII-File Operationen

Operationen mit ganzen Records

APPREC

ALPHA Text

Die Funktion **APPREC** (*append record*) hängt den Inhalt des ALPHA-Register als neuen Record an den Arbeitsfile an. Die Tastenfolge auf Seite 22 zum Beschreiben des Files SMPL-AS illustriert auch die Verwendung von

APPREC

DELREC

Die Funktion **DELREC** (*delete record*) löscht denjenigen Record aus dem Arbeitsfile, auf den der Record-Pointer momentan positioniert ist. **DELREC** setzt den Zeichen-Pointer auf Null, verändert jedoch nicht den Record-Pointer.

Beispiel: Löschen Sie einen Eintrag auf dem File SMPL-AS.

Tastenfolge

Anzeige

2	2_	Pointer sollen auf Zeichen 000 in Record 002 gesetzt werden.
ALPHA SMPL-AS	SMPL-AS_	Spezifizieren des Filenamens.
ALPHA	2.0000	
XEQ ALPHA SEEKPTA	XEQ SEEKPTA_	Definieren von SMPL-AS als Arbeitsfile und Positionieren der Pointer.
ALPHA	2.0000	
XEQ ALPHA DELREC	XEQ DELREC_	Löschen des Records „CARL LAFONG“.
ALPHA	2.0000	Record 003 ist nun 002.
XEQ ALPHA DELREC	XEQ DELREC_	Löschen des Records „156-2332“; die Folge-Records verschieben sich nach oben. „BRUCE WAYNE“ ist jetzt Record 002.
ALPHA	2.0000	

INSREC

ALPHA Text

Die Funktion **INSREC** (*insert record*) fügt vor dem Record, auf den der Record-Pointer momentan positioniert, den Inhalt des ALPHA-Registers als neuen Record ein. **INSREC** setzt den Zeichen-Pointer auf das letzte Zeichen des eingefügten Records aber es verändert nicht den Record-Pointer.

Beispiel: Fügen Sie im File SMPL-AS einen Eintrag vor „BRUCE WAYNE“ ein. (Es wird unterstellt, daß im vorangegangenen Beispiel SMPL-AS als Arbeitsfile definiert wurde und die Pointer auf Record 003 und Zeichen 000 positioniert sind.)

Tastenfolge

Anzeige

ALPHA 702-1133	702-1133_	Telefon-Nr.
ALPHA	2.0000	
XEQ ALPHA INSREC	XEQ INSREC_	Die Telefon-Nr. wird in Record 002 eingefügt;
ALPHA	2.0000	die Daten in den anderen Records werden nach unten verschoben. Der Zeichen-Pointer steht jetzt auf dem Ende der Telefon-Nr., die jetzt Record 002 ist.

Tastenfolge

ALPHA
 ALPHA BILL BAILEY
 ALPHA
 XEQ ALPHA INSREC

Anzeige

2.0000
 BILL BAILEY_
 2.0000
 XEQ INSREC_

Name.

Einfügen des Namens vor der Telefon-Nr.
 Telefon-Nr. und Daten in den anderen Records
 werden nach unten verschoben.

ALPHA
 XEQ ALPHA SAMPL-P
 ALPHA

2.0000
 XEQ SAMPL-P_
 HARVEY KECK
 555-1234
 BILL BAILEY
 702-1133
 BRUCE WAYNE
 205-4423
 WILLIAM BATSON
 602-9991
 END OF FL

Starten des Programms zum Listen des Files.

Operationen mit Zeichen innerhalb eines Records

APPCHR

ALPHA Text

Die Funktion APPCHR (*append characters*) hängt den Inhalt des ALPHA-Registers an denjenigen Record im Arbeitsfile an, auf den der Record-Pointer momentan positioniert ist. APPCHR setzt den Zeichen-Pointer auf das letzte angehängte Zeichen aber verändert nicht den Record-Pointer.

Beispiel: Modifizieren Sie einen Record in SMPL-AS, in dem Sie den Zusatz „JR.“ an Record 000 „HARVEY KECK“ anhängen. Es wird unterstellt, daß SMPL-AS weiterhin der Arbeitsfile ist.

Tastenfolge

0
 XEQ ALPHA SEEKPT
 ALPHA
 ALPHA SPACE JR.
 ALPHA
 XEQ ALPHA APPCHR
 ALPHA
 0
 XEQ ALPHA SEEKPT
 ALPHA
 XEQ ALPHA GETREC
 ALPHA
 ALPHA
 ALPHA

Anzeige

0_
 XEQ SEEKPT_
 0.0000
 JR._
 0.0000
 XEQ APPCHR_
 0.0000
 0_
 XEQ SEEKPT_
 0.0000
 XEQ GETREC_
 0.0000
 HARVEY KECK JR.
 0.0000

Spezifizieren des Records 000.
 Positionieren der Pointer.

Eingabe des anzuhängenden Texts.

Anhängen der Zeichen an den Record.

Spezifizieren des Zeichen 000 von Record 000.

Positionieren der Pointer.
 Lesen von Record 000.

DELCHR

X Anzahl der Zeichen

Die Funktion DELCHR (*delete characters*) löscht die im X-Register spezifizierte Anzahl von Zeichen aus dem Arbeitsfile. Die Löschung beginnt mit dem Zeichen im Arbeitsfile, auf das der Zeichen-Pointer momentan positioniert ist. DELCHR löscht nicht Zeichen nach dem Ende des Records und verändert nicht den Record-Pointer.

Beispiel: Modifizieren Sie einen Record in SMPL-AS, in dem Sie „HARVEY KECK JR.“ in „H KECK JR.“ abändern.

Tastenfolge	Anzeige	
<div>• 001</div>	.001_	Pointer sollen auf Zeichen 001 in Record 000 positioniert werden.
<div>XEQ ALPHA SEEKPT</div> <div>ALPHA</div>	XEQ SEEKPT_ 0.0010	Positionieren der Pointer.
<div>5</div>	5_	Anzahl der zu löschenden Zeichen.
<div>XEQ ALPHA DELCHR</div> <div>ALPHA</div>	XEQ DELCHR_ 5.0000	Löschender Zeichen.
<div>0</div>	0_	Pointer sollen auf Zeichen 000 in Record 000 positioniert werden.
<div>XEQ ALPHA SEEKPT</div> <div>ALPHA</div>	XEQ SEEKPT_ 0.0000	Positionieren der Pointer.
<div>XEQ ALPHA GETREC</div> <div>ALPHA</div>	XEQ GETREC_ 0.0000	Lesen des momentanen Records.
<div>ALPHA</div>	H KECK JR.	Der geänderte Record.
<div>ALPHA</div>	0.0000	

INSCHR

ALPHA

Text

Die Funktion **INSCHR** (*insert characters*) trägt den Inhalt des ALPHA-Registers in den Arbeitsfile ein, und zwar vor demjenigen Zeichen, auf das der Zeichen-Pointer momentan positioniert ist. **INSCHR** setzt den Zeichen-Pointer auf das letzte eingefügte Zeichen.

Beispiel: Modifizieren Sie einen Record in SMPL-AS, in dem Sie einen Punkt hinter das „H“ in „H KECK JR.“ einfügen.

Tastenfolge	Anzeige	
<div>• 001</div>	.001_	Pointer sollen auf Zeichen 001 in Record 000 positioniert werden.
<div>XEQ ALPHA SEEKPT</div> <div>ALPHA</div>	XEQ SEEKPT_ 0.0010	Positionieren der Pointer.
<div>ALPHA</div> <div>ALPHA</div>	._	Positionieren der Pointer.
<div>ALPHA</div>	0.0010	Eingabe des Punktes.
<div>XEQ ALPHA INSCHR</div> <div>ALPHA</div>	XEQ INSCHR_ 0.0010	Einfügen der Zeichen.
<div>0</div>	0_	Pointer sollen auf Zeichen 000 in Record 000 positioniert werden.
<div>XEQ ALPHA SEEKPT</div>	XEQ SEEKPT_ 0.0000	Positionieren der Pointer auf dem Anfang des Records.
<div>ALPHA</div>	0.0000	
<div>XEQ ALPHA GETREC</div> <div>ALPHA</div>	XEQ GETREC_ 0.0000	Lesen des momentanen Records.
<div>ALPHA</div>	H. KECK JR.	Der Punkt ist eingefügt.
<div>ALPHA</div>	0.0000	

Durchsuchen eines ASCII-FILES

<div>POSFL</div>	ALPHA <div>Text</div>
------------------	-----------------------

Die Funktion **POSFL** (*position in file*) durchsucht, beginnend mit der momentanen Pointerposition, den Arbeitsfile nach einem Textstring, der mit dem Inhalt des ALPHA-Registers übereinstimmt. Wenn der gesuchte

String gefunden wird, werden die Pointer auf das erste Zeichen des gefundenen Texts positioniert. Die neuen Pointer-Werte werden im X-Register abgelegt. Wenn der gesuchte Text nicht gefunden wird, bleiben die Pointer unverändert, und im X-Register wird die Zahl -1 abgelegt.

Beispiel: Durchsuchen Sie den File SMPL-AS nach dem Textstring „BATSON“.

Tastenfolge	Anzeige	
0	0_	Pointer sollen auf Zeichen 000 in Record 000 positioniert werden.
<div>XEQ ALPHA SEEKPT</div>	XEQ SEEKPT_	Positionieren der Pointer auf den Fileanfang.
<div>ALPHA</div>	0.0000	
<div>ALPHA BATSON</div>	BATSON_	Eingabe des gesuchten Textstrings.
<div>ALPHA</div>	0.0000	
<div>XEQ ALPHA POSFL</div>	XEQ POSFL_	
<div>ALPHA</div>	6.0080	Der gesuchte String beginnt in Zeichen 008 von Record 006.

Übertragen von ASCII-Files in den Hauptspeicher

GETREC

Die Funktion `GETREC` (*get record*) löscht das ALPHA-Register und überträgt danach bis zu 24 Zeichen aus dem Arbeitsfile in das ALPHA-Register. Die Übertragung der Zeichen beginnt an der momentanen Pointer-Position.

Nach der Übertragung eines Zeichens deutet der Pointer auf das Zeichen, das auf das letzte kopierte Zeichen folgt. Wenn weniger als 24 Zeichen zwischen Pointer und dem Ende des Records vorhanden sind, wird die Übertragung am Ende des Records abgebrochen. Bei der Ausführung von `GETREC` wird Flag 17 gesetzt, wenn das Ende des Records nicht erreicht wird. Dies ist besonders dann nützlich, wenn Sie einen Drucker über HP-IL (die Hewlett-Packard Interface Loop) an den Rechner angeschlossen haben, mit einer Zeilenlänge größer als 24 Zeichen. Wenn auf jede `GETREC` Anweisung der HP-IL Befehl `OUTA` folgt, wird der Inhalt des ALPHA-Registers auf den Drucker ausgegeben. Diese Datensequenz wird nicht mit den Steuerzeichen Wagenrücklauf (CR) und Zeilenvorschub (LF) abgeschlossen, sofern Flag 17 gesetzt ist. Dies ermöglicht das Ausdrucken des Inhalts eines vollständigen Records, mit mehr als 24 Zeichen in einer einzigen Zeile.

ARCLREC

Die Funktion `ARCLREC` (*ALPHA recall record*) hängt einen Record oder einen Teil eines Records an den Inhalt des ALPHA-Registers an. Die Übertragung bricht ab, wenn das ALPHA-Register voll ist. Bei der Ausführung von `ARCLREC` wird Flag 17 gesetzt, wenn das Ende des Records nicht erreicht wird. Bei Erreichen des Endes des Records wird Flag 17 wieder gelöscht. `ARCLREC` setzt den Zeichen-Pointer auf das Zeichen, das auf das letzte kopierte Zeichen folgt.

Beispiel: Hängen Sie einen Record aus dem File SMPL-AS an den Inhalt des ALPHA-Registers an.

Tastenfolge	Anzeige	
0	0_	Pointer sollen auf Zeichen 000 in Record 000 positioniert werden.
<div>XEQ ALPHA SEEKPT</div>	XEQ SEEKPT_	Positionieren der Pointer.
<div>ALPHA</div>	0.0000	
<div>XEQ ALPHA GETREC</div>	XEQ GETREC_	Das ALPHA-Register wird gelöscht, der Inhalt des momentanen Records im ALPHA-Register abgelegt und der Record-Pointer auf den nächsten Record gesetzt.
<div>ALPHA</div>	0.0000	

Tastenfolge

[ALPHA] [] [APPEND] [SPACE]

[ALPHA]

[XEQ] [ALPHA] ARCLREC

[ALPHA]

[ALPHA]

[ALPHA]

Anzeige

H. KECK JR. _

0.0000

XEQ ARCLREC _

0.0000

H. KECK JR.

555-1234

0.0000

Anhängen des momentanen Records an den Inhalt des ALPHA-Registers.

Übertragen von Daten zwischen dem erweiterten Speicher und Massenspeicher-Einheiten

Sie können im erweiterten Speicher befindliche ASCII-Files auf Massenspeicher-Einheiten, beispielsweise über das HP-IL Modul HP82160A auf eine Bandkassette in einem Digitalkassettenlaufwerk HP82161A, übertragen. Die Kassette stellt ein permanentes Speichermedium dar, von dem Files bei Bedarf abgerufen werden können.

[SAVEAS]

ALPHA *Filename im erweiterten Speicher, Filename im Massenspeicher*

[SAVEAS]

ALPHA *Filename im erweiterten Speicher*

Die Funktion [SAVEAS] (*save ASCII*) kopiert den im erweiterten Speicher befindlichen, angegebenen ASCII-File in den spezifizierten Zielfile auf dem Massenspeichermedium. Wenn Sie das Komma und den Namen des Zielfiles weglassen und nur den Namen des Quellfiles in das ALPHA-Register eingeben, wird der File im erweiterten Speicher in einen File gleichen Namens auf dem Massenspeichermedium kopiert (sofern ein solcher File existiert). Es ist zu beachten, daß [SAVEAS] keinen File auf dem Massenspeichermedium anlegt; dies muß vor Ausführung von [SAVEAS] mittels der HP-IL Funktion [CREATE] erfolgen.

[GETAS]

ALPHA *Filename im Massenspeicher, Filename im erweiterten Speicher*

[GETAS]

ALPHA *Filename im Massenspeicher*

Die Funktion [GETAS] (*get ASCII*) kopiert den angegebenen ASCII-File auf dem Massenspeichermedium in den spezifizierten Zielfile im erweiterten Speicher. Wenn Sie das Komma und den Namen des Zielfiles weglassen und nur den Namen des Quellfiles in das ALPHA-Register eingeben, wird der File auf dem Massenspeichermedium in einen File gleichen Namens im erweiterten Speicher kopiert (sofern ein solcher File existiert). Vor der Ausführung von [GETAS] muß der File im erweiterten Speicher mittels [CRFLAS] angelegt werden.

Die Ausführung von [SAVEAS] und [GETAS] bricht ab, wenn das Ende des Quellfiles oder des Zielfiles erreicht wird. Wenn der Zielfile zu klein ist, wird eine Fehlermeldung angezeigt; die bis dahin übertragenen Daten bleiben jedoch erhalten.

Programmieren mit den Funktionen des Erweiterten Funktions/Speicher-Moduls

Alle durch das Erweiterte Funktions/Speicher-Modul HP82180A implementierten Funktionen können eingegeben werden, sobald das Modul in den Rechner eingesteckt ist. Bei eingestecktem Modul zeigt der Rechner Programmzeilen, die erweiterte Funktionen enthalten, wie gewöhnliche Programmzeilen an.

Wenn das Modul später entfernt wird, werden diese Programmzeilen als XROM Funktionen gefolgt von zwei Identifikationsnummern angezeigt bzw. ausgedruckt. Die erste Identifikationsnummer ist 25 und deutet an, daß die betreffende Funktion im Erweiterten Funktions/Speicher-Modul vorhanden ist. Die zweite Identifikationsnummer dient zur eigentlichen Identifikation der einzelnen Funktionen. Die XROM Identifikationsnummern der Funktionen des Erweiterten Funktions/Speicher-Moduls sind im folgenden aufgeführt:

Funktion	XROM Nummer	Funktion	XROM Nummer	Funktion	XROM Nummer
ALENG	XROM 25,01	GETKEY	XROM 25,17	RCLPT	XROM 25,33
ANUM	XROM 25,02	GETP	XROM 25,18	RCLPTA	XROM 25,34
APPCHR	XROM 25,03	GETR	XROM 25,19	REGMOVE	XROM 25,35
APPREC	XROM 25,04	GETREC	XROM 25,20	REGSWAP	XROM 25,36
ARCLREC	XROM 25,05	GETRX	XROM 25,21	SAVEAS	XROM 25,37
AROT	XROM 25,06	GETSUB	XROM 25,22	SAVEP	XROM 25,38
ATOX	XROM 25,07	GETX	XROM 25,23	SAVER	XROM 25,39
CLFL	XROM 25,08	INSCHR	XROM 25,24	SAVERX	XROM 25,40
CLKEYS	XROM 25,09	INSREC	XROM 25,25	SAVEX	XROM 25,41
CRFLAS	XROM 25,10	PASN	XROM 25,26	SEEKPT	XROM 25,42
CRFLD	XROM 25,11	PCLPS	XROM 25,27	SEEKPTA	XROM 25,43
DELCHR	XROM 25,12	POSA	XROM 25,28	SIZE?	XROM 25,44
DELREC	XROM 25,13	POSFL	XROM 25,29	STOFLAG	XROM 25,45
EMDIR	XROM 25,14	PSIZE	XROM 25,30	X<>F	XROM 25,46
FLSIZE	XROM 25,15	PURFL	XROM 25,31	XTOA	XROM 25,47
GETAS	XROM 25,16	RCLFLAG	XROM 25,32		

Wenn Sie Programmzeilen, die erweiterte Funktionen enthalten, eingeben, und das Modul nicht eingesteckt ist, werden diese Funktionen in der Form XEQ *Funktionsname* aufgezeichnet und angezeigt. Die Programmausführung wird durch derartige Programmzeilen verlangsamt, da der Rechner bei der Abarbeitung dieser Zeile zuerst nach einem Programm mit dem angegebenen Namen bzw. nach einer Programmzeile mit einer entsprechenden Marke sucht.

Wartung, Gewährleistung und Serviceinformation

Wartung des Moduls

VORSICHT

Schalten Sie den Rechner immer aus, bevor Sie ein Modul oder Peripheriegeräte anschließen oder entfernen. Die Nichtbeachtung dieser Vorsichtsmaßnahme kann zu Schäden am Rechner oder zu Störungen im Systembetrieb führen.

- Halten Sie die Kontaktfläche des Moduls frei von störenden Einflüssen. Sollten die Kontakte verschmutzen, so ist die Schmutzschicht vorsichtig von der Kontaktfläche abzubürsten oder wegzublasen. Verwenden Sie keine Flüssigkeiten zum Reinigen der Kontakte.
- Lagern Sie das Modul an einem sauberen, trockenen Platz.
- Schalten Sie den Rechner immer aus, bevor Sie Module oder Peripheriegeräte anschließen oder entfernen.
- Beachten Sie die folgenden Temperaturspezifikationen:

Betriebstemperatur: 0° bis 45 °C

Lagertemperatur: -40° bis 75 °C

Gewährleistung

Hewlett-Packard gewährleistet, daß das Modul frei von Material- und Verarbeitungsfehlern ist, und verpflichtet sich, etwaige fehlerhafte Teile kostenlos instandzusetzen oder auszutauschen, wenn das Produkt – direkt oder über einen autorisierten Hewlett-Packard Vertragshändler – an eine Hewlett-Packard Serviceniederlassung eingeschickt wird. Die Gewährleistungsfrist beträgt 12 Monate ab Verkaufsdatum.

Weitergehende Ansprüche, insbesondere auf Ersatz von Folgeschäden, können nicht geltend gemacht werden. Schäden, die auf unsachgemäße Veränderungen des Produkts durch Dritte zurückzuführen sind, werden von dieser Gewährleistung nicht umfaßt. Die Gewährleistung erstreckt sich nicht auf die im Modul gespeicherte Software.

Die Gewährleistung gilt nur bei Vorlage der vollständige ausgefüllten Service-Karte in Verbindung mit entweder

- a) dem von einem autorisierten HP-Vertragshändler ausgestellten und unterschriebenen Original-Kaufbeleg aus dem das Kaufdatum und die Seriennummer (sofern vorhanden) des Produkts hervorgehen.

oder

- b) der Originalrechnung von Hewlett-Packard.

Die Ansprüche des Käufers aus dem Kaufvertrag bleiben von dieser Regelung unberührt.

Änderungsverpflichtung

Die Produkte von Hewlett-Packard werden auf der Basis der zum Zeitpunkt der Herstellung gegebenen technischen Spezifikationen verkauft. Hewlett-Packard übernimmt keine Verpflichtung zur nachträglichen Anpassung oder Modifikation einmal verkaufter Produkte.

Gewährleistungsinformation

Wenn Sie Fragen zu dieser Gewährleistungserklärung haben, nehmen Sie bitte Kontakt mit einem autorisierten Hewlett-Packard Händler oder einer Hewlett-Packard Verkaufs- oder Serviceniederlassung auf. Sollte dies nicht möglich sein, wenden Sie sich bitte an:

- In Europa:

Hewlett-Packard S.A.
7, rue du Bois-du-Lan
P.O. Box
CH-1217 Meyrin 2
Genf
Schweiz

Hinweis: Bitte senden Sie an diese Adresse keine Geräte zur Reparatur.

- In den Vereinigten Staaten:

Hewlett-Packard
1000 N.E. Circle Blvd.
Corvallis, OR 97330
Telefon: (503) 758-1010

- In allen anderen Ländern:

Hewlett-Packard Intercontinental
3495 Deer Creek Rd.
Palo Alto, California 94304
U.S.A.
Telefon: (415) 857-1501

Hinweis: Bitte senden Sie an diese Adresse keine Geräte zur Reparatur.

Service

Hewlett-Packard unterhält Serviceniederlassungen in vielen Ländern der Welt. 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.

Normalerweise erfolgt die Reparatur und der Rückversand von Hewlett-Packard Produkten innerhalb von 5 Werktagen. In Abhängigkeit von der Auslastung der Serviceniederlassung kann diese Zeitspanne im Einzelfall überschritten werden.

Service-Zentrale in den Vereinigten Staaten

In den Vereinigten Staaten erreichen Sie die Service-Zentrale von Hewlett-Packard für Taschenrechner- und Kompaktcomputer-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
Woluwe-laan
B-1200 BRÜSSEL
Telefon (2) 762 32 00

GROSSBRITANNIEN

HEWLETT-PACKARD LTD.
King Street Lane
GB-WINNERSH, WOKINGHAM
Berkshire RG11 5AR
Telefon (734) 784774

OSTEUROPA

Wenden Sie sich bitte an die
Service-Zentrale in Österreich.

DÄNEMARK

HEWLETT-PACKARD A/ S
Datavej 52
DK-3460 BIRKERØD (Kopenhagen)
Telefon (02) 81 66 40

ITALIEN

HEWLETT-PACKARD ITALIANA S.P.A.
Casella postale 3645 (Milano)
Via G. Di Vittorio, 9
I-20063 CERNUSCO SUL NAVIGLIO (Mailand)
Telefon (2) 90 36 91

SCHWEDEN

HEWLETT-PACKARD SVERIGE AB
Box 19
Skalholtsgatan 9, Kista
S-16393 SPANGA/STOCKHOLM

DEUTSCHLAND

HEWLETT-PACKARD GmbH
Vertriebszentrale
Berner Straße 117
Postfach 560140
D-6000 FRANKFURT 56
Telefon (0611) 50041

NIEDERLANDE

HEWLETT-PACKARD NEDERLAND B.V.
Van Heuven Goedhartiaan 121
1181 KK AMSTELVEEN (Amsterdam)
P.O. Box 667
Telefon (020) 472021

SCHWEIZ

HEWLETT-PACKARD (SCHWEIZ) AG
Allmend 2
CH-8967 WIDEN
Telefon (057) 501 11

FINNLAND

HEWLETT-PACKARD OY
Revontulentie 7
SF-02100 ESPOO 10 (Helsinki)
Telefon (90) 455 02 11

NORWEGEN

HEWLETT-PACKARD NORGE A/ S
P.O. Box 34
Osterndalen 18
N-1345 OESTERAAS (Oslo)
Telefon (2) 17 11 80

SPANIEN

HEWLETT-PACKARD ESPAÑOLA S.A.
Calle Jerez 3
E-MADRID 16
Telefon (1) 458 2600

FRANKREICH

HEWLETT-PACKARD FRANCE
Division Informatique Personnelle
S.A.V. Calculateurs de poche
F-91947 Les Ulis Cedex
Telefon (1) 907 78 25

ÖSTERREICH

HEWLETT-PACKARD GmbH
Wagramerstraße-Lieblgasse
A-1220 WIEN
Telefon (0222) 23 65 11

Internationale Serviceinformation

Nicht jede Hewlett-Packard Serviceniederlassung bietet Service für alle Hewlett-Packard Produkte. Wenn Sie jedoch Ihr Gerät bei einem autorisierten HP-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 Reparaturmöglichkeiten. 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 erhalten.

Sämtliche der durch den Versand entstehenden Kosten gehen zu Ihren Lasten.

Benutzerberatung

Sollten beim Einsatz Ihres Geräts in bestimmten Anwendungsfällen Fragen auftauchen, so rufen Sie einfach unsere Kundenberatung an (siehe Verzeichnis der Niederlassungen) oder schreiben direkt an:

Hewlett-Packard GmbH
Vertriebszentrale
Bernerstrasse 117
Postfach 560140
D-6000 Frankfurt/M. 56

Viele unserer Benutzer senden uns Programmteile oder vollständige Programme, die auch für andere HP-Anwender interessant sein können. Es ist ein Grundsatz von Hewlett-Packard, derartige Informationen nur dann zu verwenden, wenn sie unverbindlich und ohne jegliche Verpflichtungen überlassen werden. Da außerdem keine Hinweise und Anregungen entgegengenommen werden, die vertraulich behandelt werden sollen, muß jeder Einsendung die folgende Erklärung beigelegt sein:

„Für die hiermit der Firma Hewlett-Packard unverbindlich überlassenen Informationen beanspruche ich weder Vertraulichkeit, noch sind damit irgendwelche Verpflichtungen verbunden. Mir ist bekannt, daß durch die Weitergabe dieser Informationen kein ausdrücklich oder stillschweigend vereinbartes Rechtsverhältnis entsteht. Hewlett-Packard kann ohne Entschädigung für mich von diesen Informationen urheberrechtlichen Gebrauch machen, sie verteilen, veröffentlichen oder über sie in irgendeiner Weise ganz oder teilweise verfügen. Unterschrift.“

Händler- und Produktinformation

Informationen betreffend des Händlernetzes, der Produkte und Preise erhalten Sie in der Bundesrepublik Deutschland über die Telefonnummer 0611/50041.

Fehlermeldungen

Im folgenden ist eine Liste der Fehlermeldungen, die von den Funktionen des Erweiterten Funktions/Speicher-Moduls erzeugt werden können, aufgeführt. Bei Auftreten eines Fehlers wird die betreffende Funktion in der Regel nicht ausgeführt. Auf die Ausnahmefälle wird jeweils hingewiesen.

Anzeige	Funktionen	Bedeutung	
CHKSUM ERR	GETP	Auf einen Teil des Programmfiles kann nicht zugegriffen werden.	
	GETSUB		
DATA ERROR	AROT	Die Zahl im X-Register ist größer als 255.	
	POSA		
	XTOA	Die Zahl im X-Register ist größer als 999.	
	X<>F		
	PSIZE		
	SEEKPT		
	SEEKPTA		
	CRFLAS		Die Zahl im X-Register ist 0. (Der anzulegende File würde 0 Register enthalten.)
	CRFLD		
	STOFLAG	Die Daten im X-Register (oder Y-Register, wenn das X-Register einen Flagblock der Form <i>bb.ee</i> definiert) entsprechen nicht dem vom RCLFLAG erzeugten Format.	
DUP FL	CRFLAS	Im erweiterten Speicher existiert bereits ein File gleichen Namens. Der bereits vorhandene File wird zum Arbeitsfile. Für ASCII-, Daten und Programmfiles kann kein gemeinsamer Name verwendet werden. Hinweis: Verwenden Sie einen Zusatzbuchstaben, um zusammengehörende Files zu kennzeichnen (SAMPL-D, SAMPL-P).	
	CRFLD		
END OF FL	APPCHR	Durch die Ausführung der Operation würde auf die Register hinter dem definierten Ende des Files zugegriffen werden. Bei SAVER und SEEKPTA wird der angegebene File zum Arbeitsfile, der Fileinhalt und die Pointerwerte werden jedoch nicht verändert.	
	APPREC		
	ARCLREC		
	DELCHR		
	DELREC		
	GETREC		
	GETRX		
	GETX		
	INSCHR		
	INSREC		
	SAVER		
	SAVERX		
	SAVEX		
	SEEKPT		
	SEEKPTA		
	GETAS		Die Übertragung des Files wurde nicht vollständig abgeschlossen, da das Ende des Zielfiles vor dem Ende des Quellfiles erreicht wurde. Der Zielfile enthält den bis zum Abbruch übertragenen Teil des Quellfiles.
	SAVEAS		

Anzeige	Funktionen	Bedeutung
END OF REC	SEEKPT	Die Operation würde den Zeichen-Pointer hinter das Ende des momentanen Records positionieren.
	SEEKPTA	
FL NOT FOUND	CLFL	Der im ALPHA-Register spezifizierte Filename ist im erweiterten Speicher nicht vorhanden, oder der erweiterte Speicher enthält bei leerem ALPHA-Register keinen Arbeitsfile.
	FLSIZE	
	GETAS	
	GETP	
	GETR	
	GETSUB	
	PURFL	
	RCLPTA	
	SAVEAS	
	SAVER	
	SEEKPTA	
	APPCHR	
	APPREC	
	ARCLREC	
	DELCHR	
	DELREC	
	GETREC	
	GETRX	
	GETX	
	INSCHR	
	INSREC	
	POSFL	
	RCLPT	
	SAVERX	
	SAVEX	
	SEEKPT	
	APPCHR	Der erweiterte Speicher enthält keinen Arbeitsfile. Hinweis: Dieser Fehler tritt auf, wenn der erweiterte Speicher überhaupt keine Files enthält oder zuvor ein File gelöscht wurde.
	APPREC	
	ARCLREC	
	DELCHR	
	DELREC	
	GETREC	
	GETRX	
	GETX	
	INSCHR	
	INSREC	
	POSFL	
	RCLPT	
	SAVERX	
	SAVEX	
	SEEKPT	
FL TYPE ERR	APPCHR	Der Typ des Arbeitsfiles oder des im ALPHA-Register spezifizierten Files entspricht nicht dem von der Funktion benötigten Typ. Diese Meldung würde erzeugt werden, wenn Sie beispielsweise bei der Ausführung von CLFL einen Programmfile spezifiziert hätten.
	APPREC	
	ARCLREC	
	CLFL	
	DELCHR	
	DELREC	
	GETAS	
	GETP	
	GETR	
	GETREC	
	GETRX	
	GETSUB	
	GETX	
	INSCHR	
	INSREC	
	POSFL	
	SAVEAS	
	SAVER	
	SAVERX	
	SAVEX	
	SEEKPT	
	SEEKPTA	

Anzeige	Funktionen	Bedeutung
KEYCODE ERR	PASN	Die Zahl im X-Register entspricht keiner zuweisbaren Taste (z.B. der Umschalttaste, Tastencode 31).
NAME ERR	CLFL	Das ALPHA-Register ist leer.
	CRFLAS	
	CRFLD	
	GETAS	
	GETP	
	GETSUB	
	PURFL	
	SAVEAS	
	SAVEP	
	PCLPS	Das spezifizierte Programm ist im Hauptspeicher nicht vorhanden.
	SAVEP	
NO DRIVE	GETAS	Der Rechner enthält kein HP-IL Modul oder die Interfaceschleife enthält keine Massenspeichereinheit.
	SAVEAS	
NO ROOM	CRFLAS	Der im erweiterten Speicher verfügbare Platz ist zu klein, um einen File der durch das X-Register spezifizierten Größe anzulegen.
	CRFLD	
	SAVEP	Der im erweiterten Speicher verfügbare Platz ist zu klein, um das spezifizierte Programm zu speichern.
	GETP	Der Hauptspeicher ist zu klein, um das spezifizierte Programm aufzunehmen.
	GETSUB	
	PSIZE	(Nur bei Ausführung innerhalb eines Programms.) Der Hauptspeicher ist für die gewünschte Speicheraufteilung zu klein.
NONEXISTENT	Alle Funktionen	Das Erweiterte Funktions/Speicher-Modul ist nicht eingesteckt oder defekt.
	GETRX	Eines oder mehrere der durch die Zahl im X-Register spezifizierten Register existieren nicht im Hauptspeicher.
	REGMOVE	
	REGSWAP	
	SAVERX	
	STOFLAG	Einer oder mehrere der durch die Zahl im X-Register spezifizierten Flags liegen nicht im Bereich 0 bis 43.
PACKING TRY AGAIN	GETP	(Nur bei Ausführung über das Tastenfeld). Der Programmspeicher innerhalb des Hauptspeichers ist zu klein.
	GETSUB	
	PSIZE	(Nur bei Ausführung über das Tastenfeld.) Der Hauptspeicher ist zu klein.
	PASN	Der Hauptspeicher ist zu klein.
REC TOO LONG	APPCHR	Bei Ausführung der Funktion würde der resultierende Record länger als 254 Zeichen sein.
	INSCHR	
ROM	SAVEP	Das spezifizierte Programm befindet sich in einem ROM (Read-Only-Memory, Festwertspeicher).
	PCLPS	

Nullzeichen

Nullzeichen und das ALPHA-Register

Das Nullzeichen des Rechners hat den Zeichencode 0 und wird durch das Symbol “ $\bar{}$ ” dargestellt*. Normalerweise zeigt der Rechner keine Nullzeichen an; Sie können jedoch unter gewissen Bedingungen mit dem Erweiterten Funktions-/Speicher-Modul Nullzeichen in ALPHA-Datenstrings einfügen. (Dadurch ist es insbesondere möglich, in Datenstrings eingebettete Nullzeichen an HP-IL Einheiten zu senden.)

Wenn Sie sich den Inhalt des ALPHA-Registers anzeigen lassen, werden sämtliche rechts des ersten Nicht-Nullzeichens auftretenden Nullzeichen angezeigt.

Behandlung von Nullzeichen

Da der Rechner Nullzeichen im ALPHA-Register speziell interpretiert, gibt es einige Operationen, deren Funktionsweise bei Auftreten von Nullzeichen in einem Datenstring eingeschränkt ist.

- Im ALPHA-Register vorhandene Nullzeichen erscheinen in der Anzeige als “ $\bar{}$ ” und werden von den Druckern HP82143A und HP82162A als “♦” ausgedruckt.
- Wenn bei der Ausführung der Funktion **APPEND** (siehe Rückseite des Rechners) das letzte Zeichen im ALPHA-Register ein Nullzeichen ist, erscheint die ALPHA-Anzeige leer; der Inhalt des ALPHA-Registers bleibt davon jedoch unberührt. Dies hat zur Folge, daß die nach der Ausführung von **APPEND** eingegebenen Zeichen an den bereits vorhandenen ALPHA-Datenstring angehängt werden. Die ALPHA-Anzeige kann mittels der Funktion **AVIEW** oder durch Aus- und anschließendes Wiedereinschalten des ALPHA-Modus wieder restauriert werden.
- Wenn Sie einen Nullzeichen enthaltenden ALPHA-String in einem Datenregister speichern, werden die Nullzeichen bei der Ausführung der Funktion **VIEW** auf dieses Register nicht mit angezeigt. Beim Ausdrucken des Inhalts dieses Registers werden nur die links des ersten Nullzeichens stehenden Zeichen ausgedruckt; die restlichen Zeichen werden vom Drucker ignoriert. (Im Datenregister bleiben jedoch sämtliche Zeichen einschließlich der Nullzeichen erhalten. Insbesondere wird der Datenstring wieder vollständig im ALPHA-Register abgelegt, wenn der Inhalt des Datenregisters mit **ARCL** zurückgerufen wird.)
- Wenn bei der Rundumverschiebung eines Strings, der ein oder mehrere Nullzeichen enthält, ein Nullzeichen zum ersten Zeichen des Strings wird, gehen dieses und alle direkt darauf folgenden Nullzeichen verloren.
- Wenn das letzte Zeichen im ALPHA-Register ein Nullzeichen ist und der Rechner sich im APPEND-Modus (d.h. im Zustand nach Ausführung der Funktion **APPEND**); das Eingabeaufforderungssymbol “_” erscheint rechts des Nullzeichens) befindet, bewirkt das Drücken der Taste \rightarrow ein Löschen des gesamten ALPHA-Registers.
- Wenn bei der Ausführung von **POSA** der ALPHA-String im X-Register ein Nullzeichen enthält, durchsucht der Rechner das ALPHA-Register nur nach dem Teil des Strings im X-Register, der sich links des ersten Nullzeichens befindet.
- Enthält das ALPHA-Register einen Filenamen, so werden eventuell im Filenamen vorkommende Nullzeichen ignoriert.

* Das Nullzeichen mit dem Zeichencode 0 ist nicht mit der Meldung **NULL** zu verwechseln, die erscheint, wenn eine der Funktionstasten des Rechners länger als eine halbe Sekunde gedrückt wird.

Funktionsindex

Funktion	Beschreibung	ALPHA-Register	X-Register
ALENG (Seite 15)	Berechnet die Länge des Strings im ALPHA-Register. Das Ergebnis wird im X-Register abgelegt.		
ANUM (Seite 15)	Durchsucht das ALPHA-Register nach einer ALPHA-formatierten Zahl. Der ermittelte Wert wird im X-Register abgelegt.		
APPCHR (Seite 30)	Hängt den Inhalt des ALPHA-Registers an das Ende des momentanen Records im Arbeitsfile (im erweiterten Speicher) an.	Quellstring	
APPREC (Seite 29)	Hängt den Inhalt des ALPHA-Registers als neuen Record an das Ende des momentanen Arbeitsfiles im erweiterten Speicher an.	Quellstring	
ARCLREC (Seite 32)	Hängt den momentanen Record (oder eines Teils davon) im Arbeitsfile (im erweiterten Speicher) an den Inhalt des ALPHA-Registers an. Die Übertragung beginnt an der momentanen Pointerposition und hört auf, wenn entweder das ALPHA-Register voll ist, oder der Record vollständig übertragen wurde.		
AROT (Seite 16)	Rundumverschiebung des Inhalts des ALPHA-Registers.		Anzahl der Zeichen, um die verschoben werden soll (pos. Wert – Linksverschiebung; neg. Wert – Rechtsverschiebung).
ATOX (Seite 14)	Berechnet den numerischen Zeichencode des linksbündigen Zeichens im ALPHA-Register. Das Ergebnis wird im X-Register abgelegt und das Zeichen aus dem ALPHA-Register gelöscht.		
CLFL (Seite 25)	Löscht den Inhalt eines Daten- oder ASCII-Files.	Filename	
CLKEYS (Seite 13)	Löscht alle Tastenzuordnungen.		
CRFLAS (Seite 24)	Legt einen ASCII-File im erweiterten Speicher an.	Filename	Anzahl der Register.
CRFLD (Seite 24)	Legt einen Datenfile im erweiterten Speicher an.	Filename	Anzahl der Register.
DELCHR (Seite 30)	Löscht Zeichen im Arbeitsfile; die Löschung beginnt an der momentanen Pointerposition.		Anzahl der zu löschenden Zeichen.

Funktion	Beschreibung	ALPHA-Register	X-Register
DELREC (Seite 29)	Löscht den durch den Record-Pointer bestimmten Record im Arbeitsfile.		
EMDIR (Seite 23)	Ausgabe eines Verzeichnisses der Files im erweiterten Speicher.		
FLSIZE (Seite 25)	Berechnet die Anzahl der Register des spezifizierten Files. Das Ergebnis wird im X-Register abgelegt.	Filename	
GETAS (Seite 33)	Kopiert den spezifizierten ASCII-File von einem Massenspeicher in den erweiterten Speicher.	Name eines Files im Massenspeicher, Name eines Files im erweiterten Speicher	
GETKEY (Seite 16)	Hält die Programmausführung an, bis eine Taste gedrückt wird oder eine Zeitspanne von ungefähr 10 Sekunden abgelaufen ist. Das X-Register wird mit dem Tastencode der gedrückten Taste geladen, bzw. mit 0, wenn keine Taste gedrückt wird.		
GETP (Seite 25)	Ersetzt das letzte Programm im Hauptspeicher durch den Inhalt des spezifizierten Programmfiles.	Name eines Programmfiles	
GETR (Seite 28)	Kopiert den spezifizierten Datenfile in die Register des Hauptspeichers (beginnend mit Register 00).	Name eines Datenfiles	
GETREC (Seite 32)	Löscht das ALPHA-Register und überträgt einen Record (oder einen Teil davon) vom Arbeitsfile in das ALPHA-Register. Die Übertragung beginnt an der momentanen Pointerposition und endet, wenn 24 Zeichen kopiert sind oder das Ende des Records erreicht ist.		
GETRX (Seite 28)	Kopiert alle oder einen Teil der Register des Arbeitsfiles (eines Datenfiles im erweiterten Speicher) in die spezifizierten Register des Hauptspeichers. (Die Übertragung der Daten beginnt an der momentanen Pointerposition im Arbeitsfile.)		<i>bbb.eee</i> (erstes und letztes Register des Hauptspeichers, in das Daten übertragen werden sollen).
GETSUB (Seite 25)	Kopiert das spezifizierte Programm vom erweiterten Speicher an das Ende des Programmbereichs im Hauptspeicher.	Name eines Programmfiles.	
GETX (Seite 28)	Kopiert das momentane Register im Arbeitsfile in das X-Register.		
INSCHR (Seite 31)	Fügt die Zeichen im ALPHA-Register vor dem momentanen Zeichen in den ASCII-Arbeitsfile ein.	Einzufügende Zeichen	

Funktion	Beschreibung	ALPHA-Register	X-Register
INSREC (Seite 29)	Fügt die Zeichen im ALPHA-Register vor dem momentanen Record in den ASCII-Arbeitsfile ein.	Einzufügende Zeichen	
PASN (Seite 13)	Programmierbare ASN Funktion.	Funktion oder Programmname	Tastencode
PCLPS (Seite 17)	Löscht das spezifizierte Programm und alle dahinterliegenden Programme aus dem Hauptspeicher.	Programmname	
POSA (Seite 15)	Durchsucht das ALPHA-Register nach dem oder den Zeichen im X-Register und legt die Position des ersten gefundenen Zeichens im X-Register ab (−1 wenn das ALPHA-Register das oder die gesuchten Zeichen nicht enthält).		ALPHA-Teilstring oder Zeichencode
POSFL (Seite 31)	Durchsucht den ASCII-Arbeitsfile nach einem String, der mit dem Inhalt des ALPHA-Registers übereinstimmt. Wenn ein derartiger String gefunden wird, wird die Record- und Zeichennummer des ersten Zeichens (des Strings) im X-Register abgelegt. Wenn der gesuchte String nicht gefunden wird, enthält das X-Register den Wert −1.	Suchstring	
PSIZE (Seite 17)	Programmierbare SIZE Funktion.		Anzahl der benötigten Datenspeicherregister.
PURFL (Seite 24)	Löscht den spezifizierten File aus dem erweiterten Speicher.	Filename	
RCLFLAG (Seite 11)	Speichert den Status der Flags 00 bis 43 als Daten im X-Register.		
RCLPT (Seite 27)	Lädt das X-Register mit den Pointerpositionen des Arbeitsfiles (im Format <i>rrr</i> bei Datenfiles und <i>rrr.ccc</i> bei ASCII-Files) bzw. mit der Anzahl der Bytes des Programms, wenn der Arbeitsfile ein Programmfile ist.		
RCLPTA (Seite 27)	Lädt das X-Register mit den Pointerpositionen des spezifizierten Daten- oder ASCII-Files (bzw. der Anzahl der Bytes des Programms in dem spezifizierten Programmfile). Die Pointerpositionen werden bei Datenfiles im Format <i>rrr</i> und bei ASCII-Files im Format <i>rrr.ccc</i> abgelegt. Der spezifizierte File wird zum Arbeitsfile.	Filename	
REGMOVE (Seite 11)	Kopiert einen Block von <i>nnn</i> Hauptspeicherregistern beginnend mit Register <i>sss</i> in einen Block gleicher Länge ab Register <i>ddd</i> .		<i>sss.dddnnn</i>

Funktion	Beschreibung	ALPHA-Register	X-Register
REGSWAP (Seite 11)	Vertauscht <i>nnn</i> Register ab Register <i>sss</i> mit <i>nnn</i> Registern ab Register <i>ddd</i> .		<i>sss.dddnnn</i>
SAVEAS (Seite 33)	Kopiert einen ASCII-File vom erweiterten Speicher in einen Massenspeicher.	Name eines Files im erweiterten Speicher, Name eines Files im Massenspeicher.	
SAVEP (Seite 25)	Kopiert das spezifizierte Programm vom Hauptspeicher in einen Programmfile im erweiterten Speicher.	Programmname, Filename	
SAVER (Seite 27)	Kopiert alle Register des Hauptspeichers in den spezifizierten Datenfile.	Filename	
SAVERX (Seite 27)	Kopiert einen durch den Inhalt des X-Registers definierten Registerblock in den Daten-Arbeitsfile.		<i>bbb.eee</i> (erstes und letztes Register des zu rettenden Registerblocks).
SAVEX (Seite 28)	Kopiert den Inhalt des X-Registers in das durch die momentane Pointerposition bestimmte Register des Arbeitsfiles (eines Datenfiles).		
SEEKPT (Seite 26)	Positioniert den oder die Pointer des Arbeitsfiles auf den durch die Zahl im X-Register bestimmten Record (und das Zeichen).		<i>rrr</i> (bei Datenfiles) oder <i>rrr.ccc</i> (bei ASCII-Files).
SEEKPTA (Seite 26)	Positioniert den oder die Pointer des spezifizierten Files auf den durch die Zahl im X-Register bestimmten Record (und das Zeichen).	Filename	<i>rrr</i> (bei Datenfiles) oder <i>rrr.ccc</i> (bei ASCII-Files).
SIZE? (Seite 17)	Berechnet die Anzahl der Datenspeicherregister im Hauptspeicher und legt das Ergebnis im X-Register ab.		
STOFLAG (Seite 11)	Lädt den Status der Flags 0 bis 43 (oder eines Blocks innerhalb dieser Gruppe) unter Verwendung der durch die Ausführung von RCLFLAG gewonnenen Daten zurück.		Flagstatus oder <i>bb.ee</i> (erster und letzter Flag d. Blocks. Das Y-Register enthält dann den Flag-Status).
X<>F (Seite 12)	Vertauscht den Inhalt des X-Registers und den Status der Flags 0 bis 7.		Codezahl (0 bis 255)
XTOA (Seite 14)	Hängt das durch den Zeichencode im X-Register bestimmte Zeichen an den Inhalt des ALPHA-Registers an.		Zeichencode (0 bis 255)

VERKAUFS-NIEDERLASSUNGEN:

Hewlett-Packard GmbH:

6000 Frankfurt 56, Bernerstraße 117, Postfach 560140, Tel. (0611) 50 04-1
7030 Böblingen, Herrenbergerstraße 110, Tel. (07031) 667-1
4000 Düsseldorf 11, Emanuel-Leutze-Straße 1 (Seestern), Tel. (0211) 5971-1
2000 Hamburg 60, Kapstadtring 5, Tel. (040) 6 38 04-1
8028 Taufkirchen, Eschenstraße 5, Tel. (089) 61 17-1
3000 Hannover 91, Am Großmarkt 6, Tel. (0511) 46 60 01
8500 Nürnberg, Neumeyerstraße 90, Tel. (0911) 52 20 83/ 87
1000 Berlin 30, Keithstraße 2-4, Tel. (030) 24 90 86
6800 Mannheim, Roßlauer-Weg 2-4, Tel. (0621) 70050
7910 Neu-Ulm, Messerschmittstraße 7, Tel. (0731) 70241
7517 Waldbronn 2, Hewlett-Packard Strasse, Tel. (07243) 602-1

Hewlett-Packard (Schweiz) AG:

Allmend 2, CH-8967 Widen, Tel. (057) 5 01 11
Château Bloc 19, CH-1219 Le Lignon-Genève, Tel. (022) 96 03 22

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

Wagramerstraße-Lieblgasse, A-1220 Wien, Tel. (0222) 35 16 21

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

7, rue du Bois-du-Lan, Postfach, CH-1217 Meyrin 2-Genf, Schweiz

SERVICE-NIEDERLASSUNGEN:

Hewlett-Packard GmbH:

6000 Frankfurt 56, Bernerstraße 117, Postfach 560140, Tel. (0611) 50 04-1

Hewlett-Packard (Schweiz) AG:

Allmend 2, CH-8967 Widen, Tel. (057) 5 01 11

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

Wagramerstraße-Lieblgasse, A-1220 Wien, Tel. (0222) 35 16 21



**HEWLETT
PACKARD**

Scan Copyright ©
The Museum of HP Calculators
www.hpmuseum.org

Original content used with permission.

Thank you for supporting the Museum of HP
Calculators by purchasing this Scan!

Please to not make copies of this scan or
make it available on file sharing services.