PROGRAMMEINGABE

(ursprünglich 10.3, abgeändert von Rice Institute)

10.4 *)

Die Hauptfunktionen dieses Planes ist das Einlesen von Programmen (von Lochstreifen oder von Hand). Ein Programm besteht aus Befehls- und Datenwörtern. Der vorliegende Plan behandelt ausschließlich Befehle und hexadezimale Daten. Dezimale Zahlen können nicht durch 10.4 eingelesen werden.

1. Am einfachsten ist die Eingabe von Befehlen in den LGP-30, wenn sie dual verschlüsselt sind. Da aber das Programmieren in diesem System recht schwierig ist, zieht man die dezimale Schreibweise vor. Dann muß man aber für eine nachträgliche Umwandlung im Dualsystem sorgen, da die Befehlsadressen dual gespeichert werden müssen.
2. Die meisten Programme enthalten Befehle, die sich auf Stellen dieser Programme beziehen. Will man nun die Anfangsadresse eines Programmes offen lassen (das ist z. B. bei Unterprogrammen die Regel), so muß man solche Adressen umrechnen (modifizieren).
3. Manchmal ist es sehr vorteilhaft, Konstanten, die im Programm benötigt werden, von Hand ins Dualsystem zu übertragen und im Programm einzubauen. Will man dezimale Befehle und hexadezimale Daten nebeneinander eingeben, muß ein Kennzeichen für hexadezimale (oder dezimale) Wörter existieren.
4. Gelegentlich (besonders während des Testens) ist es notwendig, Daten oder Befehle eines gespeicherten Programmes zu ändern.

Der vorliegende Plan 10.4 erlaubt, die unter 1 bis 4 beschriebenen Arbeiten in einfacher Weise durchzuführen. Er benutzt

*) Modifiziertes 10,4 siehe POOL NEWS 2/65

dazu sieben Typen von Codewörtern. Sie bestehen sämtlich aus 8 Zeichen, von denen einige dezimale Ziffern sein können (z. B. dezimale Adressen). Das erste der acht Zeichen bestimmt den Typ des Codewortes. Nach der Eingabe wird es entschlüsselt und der entsprechende Teil des Programmes aufgerufen. Im folgenden werden die Codewörter und die von ihnen ausgelösten Funktionen beschrieben.

1. Befehl (0)

Die Adresse des Befehls wird dualisiert und der ganze Befehl in einer gegebenen Zelle gespeichert. Steht vor dem Operationsteil kein x, so wird die Adresse zuvor um den Inhalt des Modifiers (s. unter 4) erhöht (relative Adresse). Andernfalls unterbleibt die Modifizierung (absolute Adresse). Z. B. wird der Befehl b 2436 modifiziert, xC 6300 dagegen nicht. Das x wird nicht gespeichert.

2. Command (+)

Die letzten sieben Stellen dieses Codewortes werden von 10.4 als Befehl aufgefaßt. Dieser wird ausgeführt, wenn das folgende Wort im ACC steht. Die Befehlsadresse des Commandwortes ist dezimal und wird nicht modifiziert. Das Objekt des Commandbefehls (das folgende Wort) wird als hexadezimal angesehen. Beispiel: Das Codewort +00C6314 gefolgt von 73W08 veranlaßt den Plan, das hex. Wort 00073W08 in Sektor 14 von Spur 63 zu speichern.

3. Start fill (;)

Das Codewort gibt an, in welcher Zelle das nächste Befehls- oder Datenwort gespeichert werden soll. Die nachfolgenden Wörter werden in aufeinanderfolgenden Zellen gespeichert, bis ein Codewort ihre Folge unterbricht. Der Adreßteil des Codewortes ist dezimal und besteht aus Spur- und Zellennummer. Beispiel: Das Codewort: ;0002035 veranlaßt den Plan das nächste Wort (nicht Codewort) in die Zelle 2035 zu legen, das folgende in 2036, das dritte in 2037 usw.

4. Modifier setzen (/)

Die Adresse des Codewortes wird in einer Zelle gespeichert, die Modifier heißt. Sie wird zu den Adressen aller Befehlswörter addiert, deren Befehlsteil nicht durch ein vorgesetztes x kenntlich gemacht wurde. Der Modifier kann nur durch ein zweites Modifier-Codewort geändert werden. Die Adresse des Codewortes wird dezimal eingegeben. Gewönlich folgt der Modifier dem Start-fill und besitzt die gleiche Adresse. Beispiel: Das Codewort /0001500 veranlaßt den Plan, alle relativen Adressen der noch einzulesenden Befehlswörter um 1500 zu erhöhen. So wird aus B 2738 der Befehl B 4238, während xH 6300 unverändert bleibt.

5. Halt und Sprung (.)

Dieses Codewort bewirkt zwei Dinge. Zunächst wird die Maschine gestoppt. Nach Betätigung des Startknopfes erfolgt ein Sprung zu der Zelle, die durch die Adresse des Codewortes bestimmt ist. Das Anhalten kann durch Niederdrücken des Schalters "KEIN HALT 32" verhindert werden. Beispiel: Das Codewort .0001700 erzwingt einen Stop; nach erneutem Start erfolgt ein Sprung zum Sektor 00 von Spur 17.

6. Hexadezimale Wörter (,)

Dieses Codewort veranlaßt das Programm, die nächsten Wörter als hexadezimale Konstanten zu speichern (die Umrechnung des Adreßteiles unterbleibt). Die Anzahl der hex. Wörter wird durch den Sektorteil der Adresse des Codewortes angegeben (der Spurteil ist immer Null). Beispiel: Das Codewort ,0000014 veranlaßt den Plan, die nächsten 14 Wörter als hex. Wörter zu speichern. Mit einem Codewort können bis zu 63 hex. Wörter eingelesen werden.

7. Hex. fill (v)

Dieses Codewort veranlaßt die Maschine, die nächsten n Wörter zu lesen und als hex. Daten in aufeinanderfolgenden Zellen, beginnend bei m, zu speichern (m und n sind hexadezimale

Zahlen). Während des Einlesens werden die dualen Wörter addiert; aus ihnen wird eine Kontrollsumme gebildet. Nach der Eingabe aller m Wörter vergleicht die Maschine die errechnete Zahl mit der Kontrollsumme vom Lochstreifen. Durch Niederdrücken des Schalters "SPRUNG" kann man den Test verhindern. Sind die Zahlen gleich, so ist die Operation beendet; andernfalls stoppt die Maschine und druckt "error". Das Codewort hat die folgende Form:

V n n n m m m m

nnn ist die hex. Anzahl der einzulesenden Wörter und mmmm die hex. Adresse für das erste Wort. Beispiel: V1B02W00 heißt: die nächsten 448 Wörter werden eingelesen und in Zelle 4700 und folgende gespeichert. Mit einem Codewort können bis zu 2047 hex. Wörter eingegeben werden.

Allgemeines:

Linksseitige Nullen brauchen nicht mitgelocht zu werden, da der Akkumulator vor jeder Eingabe gelöscht wird. Alle anderen Nullen müssen eingegeben werden. Z. B.: / 3001700 muß vollständig gelocht werden, von 000B1700 braucht nur B1700 gelocht werden. Wenn die Maschine ein fehlerhaftes Codewort liest, führt sie einen Wagenrücklauf durch, druckt "error" und stoppt. Das letzte gelesene Wort enthält den Fehler.

Während der Eingabe führt die Maschine keinen Wagenrücklauf durch. Diese müssen im Streifen enthalten sein. Fehlen einige Rücklaufsymbole auf dem Streifen, kann ein Stopp die Folge sein. Nach Betätigung der Wagenrücklauftaste fährt die Maschine mit Einlesen ohne Fehler fort.

Zeitbedarf:

Eine Spur kann in 60-70 sec. mit Befehlen gefüllt werden. Bei Eingabe mit Hex. fill wird eine Spur in 50-60 sec. gefüllt.

Speicherbedarf:
Die Routine belegt die Spuren 00 bis 03 und benutzt keine Zwischenspeicher aus Spur 63.

Halt:
Die Maschine stoppt bei 0062, wenn ein fehlerhaftes Codewort gelesen wird oder wenn die Kontrollsummen nicht gleich sind.

Im folgenden bedeutet tt eine Spur-Nr. ss eine Sektor-Nr. und W einen der 16 LGP-30 Befehle.

Codewort	Interpretation
000Wttss	modifizierter Befehl
00xWttxx	nicht modifizierter Befehl
800Wttss	modifizierter negativer Befehl
80xWttss	nicht modifizierter negativer Befehl
+00Wttss	Nach Einlesen des nächsten Wortes in den ACC wird der Befehl Wttss ausgeführt
;000ttss	Das erste Eingabewort wird in Zelle ttss gespeichert
/000ttss	Die Adresse ttss wird gespeichert und zu allen Adressen der folgenden Befehle addiert, wenn diese nicht durch ein "x" kenntlich gemacht wurden.
.000ttss	Die Maschine stoppt, wenn Schalter "EIN HALT 32" nicht niedergedrückt ist. Bei Betätigung des Start-knopfes springt sie nach Zelle ss der Spur tt.
,00000NN	Die nächsten NN Wörter werden hexa-dezimal gespeichert, für NN gilt die Bedingung $1 \leq NN \leq 63$.

V	nnn mmmm	Die nächsten nnn Wörter werden in aufeinanderfolgenden Zellen, beginnend mit der Adresse mmmm, gespeichert. Während des Einlesens wird eine Kontrollsumme berechnet, die mit einer hex. Zahl des Lochstreifens verglichen wird. Bei gedrücktem Schalter "SPRUNG" unterbleibt das Lesen und Vergleichen der Kontrollsumme. nnn und mmmm sind hex. Zahlen mit $1 \leq nnn \leq 2047$.

J1 - 10.4

00000263

```
0000  c0143 p0044 i0000 t0132 s0047 t0034
0006  s0049 t0134 s0144 t0162 u0011 s0047
0012  t0113 s0049 t0209 s0144 t0153 u0018
0018  s0047 t0248 h0039 e0100 n0137 s0002
0024  c0203 c0263 b0048 e0039 u0031 z0001
0030  ,027kwkwj a0203 u0054 z1000 r0063 u0050
0036  s0101 t0102 u0105 z4000 z0000 r0063
0042  u0050 s0101 t0103 a0217 u0105 ,10000000
0048  z6363 ,10000000 n0029 e0030 h0110 u0057
0054  y0226 h0225 u0222 e0129 m0130 a0110
0060  h0039 u0158 ,00000002 u0155 ,07ww0000 ,00900000
0102  a0145 e0246 a0033 c1400 b0149 a0029
0108  u0126 ,0wwwww00 ,00004000 ,70000000 u0062 r0063
0114  u0050 u0126 t0134 s0260 u0205 y0105
0120  y0149 s0143 t0124 u0000 c0203 u0147
0126  y0105 y0149 u0000 ,0003j3j0 ,k0000000 ,f0000000
0132  a0111 t0041 p1606 z0000 p3701 z0004
0138  c0110 u0213 c0225 ,800g0000 u0252 z0000
0144  ,10000000 ,00900000 c0143 p0026 i0000 c1400
0150  h0029 a0105 u0119 r0063 u0050 y0157
0156  z3200 u1000 e0109 m0131 a0039 u0063
0162  r0063 u0050 c0203 p0044 i0000 ,wwwwwwwj
0204  u0000 t0000 u0134 ,000wwwwj z0000 r0063
0210  u0050 y0033 u0000 p1356 ,40000000 p1358
0216  y0063 ,80000000 p3540 ,40000000 p1356 u0111
0222  c0258 p0002 i0000 z0000 b0000 t0256
0228  s0214 t0261 s0259 t0261 u0233 a0062
0234  a0263 s0214 t0240 s0259 t0240 u0241
0240  a0219 c0263 s0207 a0258 t0140 u0054
0246  ,800w3wwj z0000 r0063 u0050 a0105 u0146
0252  p0031 i0000 s0263 u0116 a0214 u0229
0258  z0000 ,3wwwwwwq ,00000002 a0219 u0234 z0000
```

POOL CLASSIFICATION

PROGRAM TITLE	PROTECTED INPUT AND DEC. MEMORY PRINTOUT ROUTINES
PROGRAM NUMBER	J1-204-3
POOL CLASSIFICATION	J1
POOL NUMBER	204

DESCRIPTION:

LGP-30

STATUS	U.R.	GEN.	SPL.	REJ.	LOAN

A protected version of P.I.R. 10.4 and D.M.P.O. 21.0; bootstrap has checksum to insure correct read-in. Can be used as a subroutine to compute checksums, to binarize BCD addresses, as a decimal address printout routine. The fill counter can be set in the normal manner or can be set to any address relative to the modifier counter.

SUPPLEMENTARY ROUTINES AND/OR SYSTEMS REQUIRED: None

STORAGE REQUIREMENTS: 0000-0763: (protected)
Track 63: working storage

PROG. BY: Arthur I. Larky
FOR:
ADDRESS: Lehigh University
Bethlehem, Pennsylvania

DATE: 3/22/65

TAPE FILE	DATE	LOC.	DATE	LOC.
MOD. DEC.	3-22			
AB. DEC.				
HEX				
OTHER				

DOCUMENTATION FILE	DATE	LOC.	DATE	LOC.
FLOW CHART	3-22			
CODING	3-22			
OPER. INST.	3-22			

POOL CLASSIFICATION

POOL CLASSIFICATION

PROGRAM TITLE: MODIFIED PROGRAM INPUT ROUTINE (10.4)

DESCRIPTION: LGP-30

STATUS	U.R.	GEN.	SPL.	REJ.	LOAN

This routine serves the same functions as program 10.4 but with protection against operating mistakes:

a) Will not accept instruction words or hex words after a (,) code unless preceded by a start fill.
b) Modification to the start fill and stop and transfer codes.
c) Additional error stops.

SUPPLEMENTARY ROUTINES AND/OR SYSTEMS REQUIRED: none

STORAGE REQUIREMENTS: All of tracks 00, 01, and 02

PROG. BY: Mr. Ming-Kuei Hu
FOR: Syracuse University
ADDRESS: Syracuse 10, New York

DATE: April 20, 1964

PROGRAM NUMBER	
POOL CLASSIFICATION	J1-
POOL NUMBER	318-3

TAPE FILE	DATE	LOC.	DATE	LOC.
MOD. DEC.				
AB. DEC.	4/20/64			
HEX				
OTHER				

DOCUMENTATION FILE	DATE	LOC.	DATE	LOC.
FLOW CHART				
CODING	4/20/64			
OPER. INST.				

POOL CLASSIFICATION

J1 - 318 - 3