

9. MONITOR 8 USERS GUIDE

THE MONITOR 8 SOFTWARE ALLOWS SYMBOLIC LOADING AND DUMPING OF 8008 PROGRAMS, AND ALSO OFFERS UTILITY EDITING AND MANIPULATION FACILITIES.

9.1 SYSTEM START-UP (MOD 8 HARDWARE CONFIGURATION)

- 9.1.1 ENSURE POWER OFF TO PROGRAMMER (IF ONE IS INCLUDED), TTY SET TO LOCAL.
- 9.1.2 APPLY CPU POWER.
- 9.1.3 PUSH RESET BUTTON.
- 9.1.4 TURN TTY TO "ON LINE" AND PUSH RESET AGAIN. WHEN TTY IS ON LINE AND A RESET IS EXECUTED THE TTY WILL TYPE A CRLF AND 8 DASHES FOLLOWED BY A CRLF.
(e.g. RESET BUTTON PUSHED
-----TTY RESPONSE.

9.2 SYSTEM START-UP (SIM8 -01 HARDWARE CONFIGURATION)

- 9.2.1 ENSURE POWER OFF TO PROGRAMMER, TTY SET TO LOCAL.
- 9.2.2 APPLY CPU POWER WITH RESET SWITCHES SET TO 005.
- 9.2.3 PUSH RESET BUTTON.
- 9.2.4 TURN TTY TO "ON LINE" AND PUSH RESET AGAIN. WHEN THE TTY IS ON LINE AND A RESET 005 INSTRUCTION IS EXECUTED, THE TTY WILL TYPE CRLF AND 8 DASHES, FOLLOWED BY A CRLF.
(e.g. RESET BUTTON PUSHED WITH SWITCHES SET TO 005
-----TTY RESPONSE.)
- 9.2.5 SET RESET SWITCHES TO 300.

9.3 ADDRESSING

THE MEMORY IN THE 8008 SYSTEM IS ORGANIZED INTO BANKS. EACH BANK IS 377 OCTAL (256 DECIMAL) BYTES IN LENGTH. WHEN COMMUNICATING WITH MONITOR 8 THE ADDRESSES TAKE THE FOLLOWING FORM:

$$N_5 \ N_4 \ N_3 \ N_2 \ N_1 \ N_0$$

$N_0 - N_5$ ARE OCTAL DIGITS WITH THE FOLLOWING SIGNIFICANCE:

N_5 = SPECIAL MODIFIER VALUE 0-3 POSSIBLE.

N_5 = 0 OR 1 MEMORY ACCESSED IS NORMAL ROM OR RAM.

N_5 = 2 OR 3 MEMORY ACCESSED IS THE pROM IN THE PROGRAMMING STATION, IF ONE IS ATTACHED TO THE SYSTEM.

$N_4 N_3$ = MEMORY BANK NUMBER

$N_4 N_3$ = 00 TO 07 MEMORY ACCESSED IS ROM IN SIM 08 AND MOD 8 SYSTEMS. $N_4 N_3$ = 10 TO 13 MEMORY ACCESSED IS RAM IN SIM08 AND MOD8 SYSTEMS.

$N_2 N_1 N_0$ = BYTE LOCATION WITHIN BANK, VALUE 000 to 377 POSSIBLE.

9.4 MONITOR 8 COMMAND SUMMARY

THE MONITOR 8 SYSTEM IS NOW READY TO LOAD SYMBOLIC PROGRAM INPUT OR ACCEPT ONE OF THE FOLLOWING UTILITY COMMANDS.

LOC	(SET CURRENT LOCATION POINTER)
DLP	(DISPLAY CURRENT LOCATION POINTER)
DPS	(DUMP SYMBOLIC)
LDO	(LOAD OCTAL)
DPO	(DUMP OCTAL)
LBF	(LOAD BNPF FORMAT)
DBF	(DUMP BNPF FORMAT)
EDT	(ENTER EDIT MODE)
XQT	(INITIATE PROGRAM EXECUTION)
CPY	(COPY ROUTINE)
TRN	(TRANSLATE ROUTINE)
SBP	(SET BREAK-POINT)
CBP	(CLEAR BREAK-POINT)
PRG	(PROGRAM pROM)

9.5 LOC (SET CURRENT LOCATION POINTER)

ALL DATA ENTRY AND MANIPULATION IS DONE AT THE ADDRESS INDICATED BY THE CURRENT LOCATION POINTER (CLP). THE POINTER VALUE IS STORED AND USED BY THE MONITOR SOFTWARE. AFTER EACH MACHINE INSTRUCTION IS ENTERED THE CLP IS UPDATED TO POINT AT THE NEXT AVAILABLE MEMORY LOCATION. THE TWO PSEUDO OPERATORS LOC AND DLP ALLOW THE USER TO PRESET AND DISPLAY THE CURRENT LOCATION POINTER.

WHEN LOC IS TYPED THE MACHINE RESPONDS WITH A SPACE (b). THE USER MUST THEN SPECIFY A SIX DIGIT ADDRESS (SEE ADDRESSING). AFTER THE LAST ADDRESS DIGIT HAS BEEN ENTERED, THE MACHINE RESPONDS WITH CRLF AND WAITS FOR THE NEXT COMMAND. THE MONITOR SOFTWARE USES RAM ADDRESSES 013350 - 013377 INCLUSIVE, BUT ALL OTHER ADDRESSES ARE AVAILABLE TO THE USER.

9.6 DLP (DISPLAY CURRENT LOCATION POINTER)

IF THE USER WISHES TO DISPLAY THE CLP, HE MAY TYPE IN DLP. THE MACHINE RESPONDS BY TYPING OUT THE CLP AND THEN PERFORMS A CRLF AND WAITS FOR THE NEXT INSTRUCTION.

NOTE: THE CLP IS DESTROYED BY SEVERAL OF THE MONITOR ROUTINES. WHEN THIS IS THE CASE, THE MONITOR WILL PRINT 8 DASHES ON COMPLETION OF THE REQUESTED FUNCTION. IN THESE INSTANCES, THE USER SHOULD RESPECIFY THE CLP USING THE LOC COMMAND BEFORE PROCEEDING.

9.7 SYMBOLIC PROGRAM INPUT

ONCE THE CLP HAS BEEN INITIALIZED, THE USER MAY TYPE IN HIS PROGRAM. AFTER EACH MNEMONIC INSTRUCTION HAS BEEN ENTERED, THE MACHINE WILL RESPOND WITH A CRLF OR, IF THE INSTRUCTION REQUIRES AN ARGUMENT, WITH A SPACE. ALL IMMEDIATE INSTRUCTIONS REQUIRE A 3 DIGIT OCTAL DATA BYTE. ALL JUMP AND CALL COMMANDS REQUIRE A 6 DIGIT SPLIT OCTAL ADDRESS (SEE ADDRESSING). INPUT/OUTPUT AND RESTART INSTRUCTIONS REQUIRE A 3 DIGIT OCTAL NUMBER TO SPECIFY A PORT NUMBER OR RESTART ADDRESS. AFTER THE INSTRUCTION AND THE CORRESPONDING ARGUMENT HAVE BEEN ENTERED, A CRLF WILL BE GENERATED AND THE NEXT INSTRUCTION MAY THEN BE ENTERED. AFTER EACH ENTRY, THE CLP IS AUTOMATICALLY UPDATED TO POINT TO THE NEXT AVAILABLE MEMORY LOCATION.

THERE ARE SEVERAL BIT COMBINATIONS WHICH WILL BE INTERPRETED BY THE 8008 AS A HALT COMMAND. THE FOLLOWING COMMANDS WILL BE INTERPRETED BY THE MONITOR TO GIVE RISE TO HALT COMMAND BIT COMBINATIONS.

<u>MNEMONIC</u>	<u>RESULTANT OCTAL</u>	<u>8008 INTERPRETATION</u>
HLT	000	HLT
INA	000	HLT
DCA	001	HLT
LMM	377	HLT

9.8 DPS (DUMP SYMBOLIC)

A SYMBOLIC LISTING IS GENERATED BY TYPING DPS. THE MACHINE WILL RESPOND WITH A CRLF AND A * (THIS IS THE PROMPTER INDICATING THAT THE MACHINE REQUIRES FURTHER ADDRESS INFORMATION). THE USER MUST NOW TYPE IN THE INITIAL AND FINAL ADDRESS, DEFINING THE BLOCK OF CODE TO BE DUMPED. THESE TWO ADDRESSES MUST BE ENTERED AS A 6 DIGIT SPLIT OCTAL NUMBER (SEE ADDRESSING). WHEN THE INITIAL ADDRESS HAS BEEN ENTERED, THE MACHINE RESPONDS WITH A BLANK AND AWAITS THE FINAL ADDRESS. WHEN THE FINAL ADDRESS HAS BEEN ENTERED THE MACHINE RESPONDS WITH 3 CRLF's AND COMMENCES LISTING. THE LISTING INCLUDES THE CURRENT MEMORY ADDRESS, THE OCTAL INSTRUCTION AND THE MNEMONIC. FOR A MULTI-BYTE INSTRUCTION THE LISTED ADDRESS IS THAT OF THE FIRST BYTE OF THE INSTRUCTION. ANY DATA FIELDS ASSOCIATED WITH THE INSTRUCTION (IMMEDIATE DATA), ADDRESSES, I/O PORT NUMBERS OR RESTART ADDRESSES) WILL BE PRINTED FOLLOWING THE MNEMONIC. ONE INSTRUCTION IS LISTED PER LINE WITH 62 LINES GENERATED PER PAGE. AN AUTO PAGING FEATURE SEPARATES EACH 11" PAGE BY 3 CRLF's. INVALID INSTRUCTIONS ARE DISPLAYED AS ???.

9.9 LDO (LOAD OCTAL)

TYPING LDO WILL INITIATE THE OCTAL LOAD ROUTINE. AS IN THE DUMP ROUTINE, THE MACHINE WAITS FOR TWO OCTAL ADDRESSES. IT THEN OUTPUTS A CRLF AND WILL BEGIN READING IN FROM THE KEYBOARD OR TAPE READER. EACH LINE WHICH CONTAINS DATA MUST HAVE A / SYMBOL TO THE LEFT OF THE DATA FIELD. EACH 3 DIGIT OCTAL VALUE WHICH FOLLOWS THE / IS INTERPRETED AS DATA. LEADING ZEROS MUST BE INCLUDED AND EACH VALUE MUST BE SEPARATED BY AT LEAST 1 BLANK. ANY DATA TO THE LEFT OF THE FIRST / IS IGNORED (NOTE THAT THIS IS USUALLY THE ADDRESSES GENERATED BY THE DPO ROUTINE). WHEN THE FINAL ADDRESS SPECIFIED HAS BEEN FILLED, THE ROUTINE RETURNS TO THE MONITOR.

9.13 EDT (ENTER EDIT MODE)

THE EDIT MODE IS ENTERED BY TYPING EDT. THE EDITOR RESPONDS WITH A CRLF AND TYPES THE VALUE OF THE CLP FOLLOWED BY A /. IT IS NOW READY TO ACCEPT ONE OF THE FOLLOWING COMMANDS:

- nnn - WHERE nnn IS A THREE DIGIT OCTAL VALUE TO BE LOADED INTO MEMORY.
- † - DISPLAY MEMORY VALUE
- ↑ - DECREMENT THE CURRENT LOCATION POINTER
- *AAAAAA - REDEFINE THE CURRENT LOCATION POINTER WITH THE VALUE AAAAAA
- @ - EQUIVALENT TO XQT
- R - RETURN TO THE MONITOR

IF DATA IS TO BE LOADED IT MUST IMMEDIATELY FOLLOW THE / SYMBOL. AN INVALID SYMBOL WILL CAUSE A CRLF WITH THE CLP RETYPED. THE nnn VALUE IS ASSEMBLED AS AN 8 BIT WORD AND STORED IN THE MEMORY. ATTEMPTING TO WRITE INTO A ROM ADDRESS WILL NOT BE FLAGGED, YET THE DATA WILL NOT (CANNOT) BE WRITTEN.

IF A BLANK IS ENTERED AFTER THE / THE CURRENT MEMORY LOCATION WILL BE DISPLAYED. TWO OPTIONS ARE THEN AVAILABLE:

- a) ← nnn REPLACE THE CURRENT VALUE WITH nnn.
- b) ANY OTHER SYMBOL WILL INCREMENT THE CLP.

FOLLOWING THE CLP / THE EDITOR EXAMINES THE FIRST CHARACTER INPUTTED TO DETERMINE THE COMMAND. IF DATA IS TO BE INPUT IMMEDIATELY, IT MUST BE IN THE FIRST THREE LOCATIONS FOLLOWING THE /. IF THE DATA FOLLOWS a← (USED TO REPLACE DISPLAYED DATA) THE INPUT IS RELATIVELY FORMAT FREE. THE FIRST OCTAL DIGIT WILL DEFINE THE REPLACEMENT DATA, ANY OTHER SYMBOLS MAY APPEAR BETWEEN THE ← SYMBOL AND THE DATA. THE SAME IS TRUE OF THE *AAAAAA COMMAND. FOLLOWING THE COMMAND OR DATA THE EDITOR TYPES THE NEW CLP ON THE NEXT LINE AND IS READY TO ACCEPT THE NEXT COMMAND.

9.14 XQT (INITIATE PROGRAM EXECUTION)

THE XQT COMMAND ALLOWS THE USER TO START THE EXECUTION OF HIS PROGRAM. FOLLOWING THE TYPING OF XQT THE MACHINE WILL RESPOND WITH A SPACE AND WAIT FOR THE STARTING ADDRESS OF THE PROGRAM. THE ENTIRE USER ROUTINE IS TREATED AS A SUBROUTINE WHICH IS CALLED FROM THE MONITOR. THE USER MAY RETURN TO MONITOR BY INCLUDING A RET (RETURN) AT THE END OF HIS ROUTINE.

9.15 CPY (COPY ROUTINE)

TYPING CPY WILL INITIATE A COPY OF BLOCKS OF MEMORY. LIKE THE DUMP AND LOAD ROUTINES THIS ROUTINE REQUIRES A START ADDRESS AND AN END ADDRESS (DEFINING THE BLOCK TO BE MOVED). IN ADDITION AFTER THE BLOCK END ADDRESS HAS BEEN ENTERED, THE MACHINE WILL RESPOND WITH A CRLF* AND WAIT FOR THE ENTRY OF A THIRD ADDRESS, THE NEW START ADDRESS FOR THE BLOCK TO BE COPIED. AFTER THE THIRD ADDRESS HAS BEEN ENTERED, THE ENTIRE BLOCK SPECIFIED WILL BE COPIED UNCHANGED STARTING AT THE NEW START ADDRESS. WHEN THE COPY HAS BEEN COMPLETED CONTROL RETURNS TO THE MONITOR.

9.16 TRN (TRANSLATE ROUTINE)

TYPING TRN WHEN IN THE MONITOR MODE INITIATES THE TRANSLATE. THIS ROUTINE IS INTENDED FOR USE AFTER A PROGRAM IS RUNNING IN RAM AND IT IS DESIRED TO STORE IT IN PROM WHICH WILL RESIDE IN A DIFFERENT BANK. NO MOVEMENT OF DATA OCCURS, BUT ALL JUMP AND CALL ADDRESSES WHICH ARE INTERNAL TO THE BANK WILL BE CHANGED TO REFLECT THE NEW SPECIFIED BANK. THIS ROUTINE AGAIN REQUIRES A START OF BLOCK AND AN END OF BLOCK ADDRESS, TO DEFINE THE BLOCK TO BE OPERATED ON. AFTER THE SECOND ADDRESS HAS BEEN ENTERED, THE MACHINE RESPONDS WITH A CRLF. THE MACHINE IS NOW WAITING FOR TWO THREE DIGIT OCTAL BANK NUMBERS (POSSIBLE RANGE 000 TO 077). AFTER THE FIRST BANK NUMBER HAS BEEN ENTERED (THE SOURCE BANK NUMBER), THE MACHINE RESPONDS WITH A + AND WAITS FOR THE SECOND BANK NUMBER (THE DESTINATION BANK NUMBER). AFTER THE SECOND BANK NUMBER HAS BEEN ENTERED, THE MACHINE SEARCHES THE SPECIFIED BLOCK FOR ALL CALL AND JUMP REFERENCES TO THE SOURCE BANK AND CHANGES THESE TO REFER TO THE DESTINATION BANK. WHEN THE CHANGES HAVE BEEN COMPLETED, THE MACHINE RETURNS TO THE MONITOR MODE.

9.17 SBP (SET BREAK-POINT)

BREAK-POINTS ALLOW THE TRACING OF PROGRAM FLOW DURING ITS EXECUTION. IF A RST 060 COMMAND IS ENCOUNTERED DURING PROGRAM EXECUTION THE MONITOR SOFTWARE WILL PRINT OUT THE CONTENTS OF THE CARRY FLAG, A B C H AND L REGISTERS, THE MEMORY CONTENTS ADDRESSED BY THE H AND L REGISTERS AND THEN RETURN TO THE MONITOR SOFTWARE.

THE SBP COMMAND INSERTS A RST 060 COMMAND AT THE ADDRESS SPECIFIED BY THE USER. THE ADDRESS AT WHICH THE BREAK-POINT IS INSERTED AND THE INSTRUCTION ORIGINALLY FOUND THERE IS RETAINED BY THE MONITOR. BEFORE SETTING SUBSEQUENT BREAK-POINTS, THE MONITOR WILL FIRST RESTORE THE DATA AT THE PREVIOUS BREAK-POINT LOCATION.

9.18 CBP (CLEAR BREAK-POINT)

THE CBP COMMAND WILL RESTORE THE DATA AT THE PRESENT BREAK-POINT LOCATION.

9.19 PRG (PROGRAM pROM)

THE MONITOR SOFTWARE ALSO CONTAINS THE FACILITY FOR CONTROLLING A pROM PROGRAMMING STATION IF ONE IS ATTACHED TO THE SYSTEM. THE PROGRAMMING ROUTINE IS ENTERED BY TYPING PRG. THE PROGRAMMING ROUTINE WILL ALLOW PROGRAMMING A pROM WITH DATA PRESENTLY LOCATED IN MEMORY. AN INITIAL AND FINAL ADDRESS MUST BE SPECIFIED. THE ROUTINE WILL PROGRAM THE DATA FROM SPECIFIED LOCATION TO THE CORRESPONDING WORD LOCATION WITHIN THE ROM.

e.g. 010177 LOCATION 177 OF THE pROM

e.g. THERE IS A ONE TO ONE CORRESPONDENCE BETWEEN THE ADDRESS BEING READ WITHIN A BANK AND THE ADDRESS BEING PROGRAMMED IN THE pROM.

THE PROGRAMMING ROUTINE WILL FIRST CHECK IF THE pROM DATA IS EQUAL TO THE PROGRAM DATA. IF THE BYTE PATTERNS ARE IDENTICAL THE ROUTINE PROCEEDS TO THE NEXT ADDRESS. IF THE LOCATION MUST BE PROGRAMMED, THE pROM IS HIT WITH PROGRAMMING PULSES IN TWO SECOND INTERVALS FOR A MAXIMUM OF 10 CYCLES. AFTER EACH INTERVAL, THE DATA IS RECHECKED. ONCE THE DATA IS READ AS PROGRAMMED, A FINAL 2 SECOND HIT CYCLE IS EXECUTED BEFORE PROCEEDING TO THE NEXT LOCATION. IF THE pROM FAILS TO PROGRAM AFTER 10 TRIES THE CURRENT LOCATION IS PRINTED FOLLOWED BY A ? AND THE ROUTINE RETURNS TO THE MONITOR.

CONTROL A

INCLUDED IN THE TTY INPUT ROUTINE IS A CHECK FOR THE CTRLA KEY. DEPRESSING THE CTRL BUTTON AND A KEY SIMULTANEOUSLY WILL CAUSE THE MACHINE TO IMMEDIATELY RETURN TO THE MONITOR ROUTINE, AND IS EQUIVALENT TO A MONITOR RESTART.

RUBOUT

OCTAL DATA INPUT ROUTINES WILL ACCEPT A RUBOUT COMMAND. EACH TIME THE RUBOUT KEY IS PRESSED A ← SYMBOL IS PRINTED AND A CHARACTER IS DELETED. TYPING TWO RUBOUTS WILL DELETE TWO CHARACTERS ETC. THE RUBOUT ROUTINE FOR OCTAL VALUES WILL "BACK SPACE" ONLY TO THE BEGINNING OF THE FIELD. DATA IS REPRESENTED BY 1 FIELD (OR BYTE) WHEREAS ADDRESSES ARE REPRESENTED BY TWO BYTES (FIELDS). THE ROUTINE WILL TYPE A ← FOR EACH RUBOUT UNTIL IT REACHES THE BEGINNING OF THE FIELD WHERE IT WILL ACCEPT A RUBOUT BUT WILL NOT TYPE ANY SYMBOL AND WILL NOT CONTINUE TO BACK SPACE.

10. MONITOR 8 SOFTWARE LISTINGS

THE FOLLOWING SECTION CONTAINS A COMPLETE LISTING OF THE MONITOR 8 SOFTWARE. IN ADDITION PAGE 76 CONTAINS A LIST OF THE EIGHT RESET POINTS (RESTART 0-7) AND A LIST OF ENTRY POINTS FOR THE MONITOR 8 SUBROUTINES. TO SAVE THE TIME REQUIRED TO RECODE THIS SOFTWARE, THE COMPLETE SOFTWARE PACKAGE, 7 ROM'S, MAY BE PURCHASED FROM MICROSYSTEMS INTERNATIONAL LTD. AT A NOMINAL SURCHARGE OVER THE NORMAL COMPONENT PRICE, TO COVER THE COST OF PROGRAMMING THE ROM'S.

RESET NO.	RESET INDEX FUNCTION
RST 000	COLD START, GENERAL RESTART
RST 010	G0 TO ROM 7 (FOR USER)
RST 020	OUTPUT AN ASCII CHARACTER
RST 030	INPUT AN ASCII CHARACTER
RST 040	TEST FOR RUBOUT
RST 050	SEARCH FOR CHARACTER IN 'E' REGISTER
RST 060	BREAKPOINT EXECUTE
RST 070	TIMING LOOP

SUBROUTINE INDEX
(START ADDRESSES OF MANY OF THE ROUTINES USED HERE,
WHICH MAY BE USABLE IN OTHER SOFTWARE)

START ADDRESS	FUNCTION
000013	OUTPUT CARRIAGE RETURN AND LINE FEED
000177	TEST FOR OCTAL CHARACTER
000205	3 DIGIT OCTAL INPUT (COMPRESSED TO 1 BYTE)
000253	3 DIGIT OCTAL OUTPUT (USED TO DISPLAY 1 BYTE)
000311	ADDRESS INCREMENT (USES CLP-L0C 013377,013376)
000326	ADDRESS DECREMENT
000344	ADDRESS COMPARE (CLP,CLP-1)
000362	COMPARE AND INCREMENT (USED TO TEST FOR END OF ROUTINE)
001000	OCTAL DUMP (DP0)
001023	FETCH DATA FROM LOCATION ADDRESSED BY CLP
001047	DISPLAY DATA AT CLP
001055	DISPLAY BLANK, CLP (ADDRESS)
001073	OUTPUT CR/LF, CLP
001111	PUT DATA INTO CLP
001120	OCTAL INPUT (LD0)
001200	INPUT AN ADDRESS (2 BYTES)
001236	OCTAL EDITOR (EDT)
001336	INDIRECT JUMP
001353	CLEAR BREAKPOINT (CBP)
002000	PR0M PR0GRAMMING ROUTINE (PRG)
002110	SET UP CLP (L0C)
002115	DUMP IN BNPF FORMAT (DBF)
002201	LOAD IN BNPF FORMAT (LBF)
002257	BANK TO BANK TRANSLATE (TRN)
002347	SET BREAKPOINT (SBP)
003000	CONTROLLER ROUTINE
003131	GENERAL ERROR ROUTINE
003150	TABLE SEARCH
003244	BREAKPOINT EXECUTE
005063	REGISTER DECODE
005313	PRINT 3 ASCII BYTES

*N

```

000000/ 006 LAI 001      (RST 000)      COLD START
000002/ 125 OUT 012      IDLE TTY
000003/ 250 XRA
000004/ 127 OUT 013      IDLE PTR
000005/ 104 JMP 003000    GO TO CONTROLLER
000010/ 104 JMP 007000    (RST 010)      USERS ROUTINE
000013/ 016 LBI 215      (CR)           CR/LF ROUTINE
000015/ 025 RST 020
000016/ 016 LBI 212      (LF)
000020/ 026 LCI 375      (RST 020)      O/P ONE CHARACTER
000022/ 036 LDI 177      SET UP TIMING
000024/ 075 RST 070      1ST BIT IS LONGER
000025/ 104 JMP 000140    CONTINUED ELSEWHERE
000030/ 006 LAI 001      (RST 030)      I/P CHARACTER
000032/ 127 OUT 013      ENABLE PTR
000033/ 036 LDI 302      SET UP TIMING
000035/ 104 JMP 000075    CONTINUED
000040/ 006 LAI 177      (RST 040)      RUBOUT TEST
000042/ 271 CPB
000043/ 013 RFZ          NOT RUBOUT SO RETURN
000044/ 016 LBI 337      O/P ARROW
000046/ 025 RST 020
000047/ 007 RET          FLAG SET TO IGNORE INPUT
000050/ 035 RST 030      (RST 050)      SEARCH FOR CHAR
000051/ 301 LAB          FETCH I/P      IN REG E
000052/ 274 CPE          COMPARE
000053/ 053 RTZ          GOT CHAR
000054/ 104 JMP 000050    TRY NEXT ONE
000057/ 377 HLT          UNUSED BYTE
000060/ 104 JMP 003244    (RST 060)      XQT BRKPT
000063/ 301 LAB          I/P (CONT)
000064/ 074 CPI 001      CNTRL A I/P
000066/ 013 RFZ          NO- GO AHEAD
000067/ 005 RST 000      YES- PANIC AND RESTART
000070/ 030 IND          (RST 070)      TIMING LOOP
000071/ 110 JFZ 000070    LOOPING
000074/ 007 RET          DONE
000075/ 377 HLT          WAIT FOR I/P    I/P(CONT)
000076/ 075 RST 070      TIME 1ST BIT
000077/ 250 XRA          CLEAR A REG
000100/ 127 OUT 013      IDLE PTR FOR NOW
000101/ 125 OUT 012      START O/P
000102/ 026 LCI 370      SET UP 1 BIT DELAY
000104/ 036 LDI 171
000106/ 075 RST 070      WAIT FOR IT
000107/ 101 INP 000      GET BIT
000110/ 054 XRI 377      COMPLEMENT I/P
000112/ 125 OUT 012      ECHO TO O/P
000113/ 032 RAR          ROTATE INTO B
000114/ 301 LAB          WITH PREVIOUS
000115/ 032 RAR          BITS
000116/ 310 LBA
000117/ 020 INC          BUMP COUNTER
000120/ 110 JFZ 000104    LOOP FOR MORE BITS
000123/ 301 LAB          GOT 8 BITS NOW
000124/ 044 NDI 177      IGNORE PARITY (MSB)
000126/ 310 LBA
000127/ 036 LDI 171      O/P STOP
000131/ 075 RST 070      AND O/P IDLE STATE
000132/ 006 LAI 001
000134/ 125 OUT 012
000135/ 104 JMP 000063    TO BE CONTINUED

```

```

000140/ 020 INC                                O/P (CONT)
000141/ 110 JFZ 000022 KEEP TIMING
000144/ 250 XRA CLEAR A
000145/ 125 OUT 012 START O/P
000146/ 026 LCI 370 SET UP TIMING
000150/ 036 LDI 171
000152/ 075 RST 070 WAIT FOR NEXT BIT
000153/ 301 LAB FETCH BIT FROM B
000154/ 125 OUT 012 AND OUTPUT BIT
000155/ 032 RAR NOW SET UP THE NEXT
000156/ 310 LBA BIT, STORE IT IN B
000157/ 006 LAI 000
000161/ 032 RAR
000162/ 201 ADB
000163/ 310 LBA
000164/ 020 INC BUMP COUNT
000165/ 110 JFZ 000150 MORE TO O/P, SO LOOP
000170/ 036 LDI 171 DONE
000172/ 075 RST 070 O/P STOP AND IDLE BITS
000173/ 006 LAI 001
000175/ 125 OUT 012
000176/ 007 RET GOODBYE
000177/ 035 RST 030 FETCH CHAR TEST FOR OCTAL
000200/ 044 NDI 370 MASK 3 BITS
000202/ 074 CPI 060 IS IT 06X? (ZF TELLS ALL)
000204/ 007 RET GO AWAY
000205/ 106 CAL 000177 GET DIGIT OCTAL I/P
000210/ 110 JFZ 000205 NOT OCTAL, TRY AGAIN
000213/ 301 LAB PUT DIGIT IN A
000214/ 012 RRC ROTATE
000215/ 012 RRC ROTATE
000216/ 370 LMA STASH IT
000217/ 035 RST 030 FETCH DIGIT
000220/ 045 RST 040 TEST FOR RUBOUT
000221/ 150 JIZ 000205 TRY AGAIN
000224/ 307 LAM FETCH LAST DIGIT
000225/ 044 NDI 300 MASK UNUSED BITS
000227/ 370 LMA STORE IT
000230/ 006 LAI 007 PUT IN 3 MORE BITS
000232/ 241 NDB ROTATE INTO POSITION
000233/ 002 RLC
000234/ 002 RLC
000235/ 002 RLC
000236/ 207 ADM ADD IN THE OLD DATA
000237/ 370 LMA STORE IT
000240/ 035 RST 030 FETCH DIGIT ENTRY FOR
000241/ 045 RST 040 TEST CORRECTION DONE
000242/ 150 JIZ 000217 RUBOUT
000245/ 006 LAI 007 MASK ALL BUT 3 BITS
000247/ 241 NDB
000250/ 207 ADM ADD THE PREVIOUS BITS
000251/ 370 LMA STASH DATA
000252/ 007 RET DONE
000253/ 016 LBI 240 (BLANK) OCTAL O/P
000255/ 025 RST 020 3 DIGITS
000256/ 307 LAM FETCH BYTE
000257/ 002 RLC MOVE BITS 7 AND 8
000260/ 002 RLC TO POS 1 AND 2
000261/ 044 NDI 003 MASK THE REST
000263/ 004 ADI 260 CONVERT TO ASCII
000265/ 310 LBA SET UP FOR O/P
000266/ 025 RST 020 O/P

```

000267/	307	LAM	FECTH BYTE	
000270/	012	RRC	SET UP BITS 4,5,6	
000271/	012	RRC		
000272/	012	RRC		
000273/	044	NDI 007	MASK	
000275/	004	ADI 260	CONVERT	
000277/	310	LBA		
000300/	025	RST 020	Ø/P	
000301/	307	LAM	BITS 1,2,3 THIS TIME	
000302/	044	NDI 007		
000304/	004	ADI 260		
000306/	310	LBA	NØW Ø/P	
000307/	025	RST 020		
000310/	007	RET	ALL DØNE	
000311/	066	LLI 376	SET TØ CLP	ADDRESS INCR
000313/	056	LHI 013		2 BYTES
000315/	317	LBM	FETCH	
000316/	010	INB	INCR LSB	
000317/	371	LMB	STØRE	
000320/	013	RFZ	CARRY	
000321/	060	INL	YES-INCR MSB	
000322/	317	LBM	FETCH	
000323/	010	INB	INCR	
000324/	371	LMB	STØRE	
000325/	007	RET	DØNE	
000326/	066	LLI 376	SET TØ CLP	ADDRESS DECR
000330/	056	LHI 013		2 BYTES
000332/	317	LBM	FETCH	
000333/	011	DCB	DECR	
000334/	371	LMB	STØRE	
000335/	060	INL	PØINT TØ MSB	
000336/	010	INB	WAS LSB	
000337/	013	RFZ	NØ-RETURN	
000340/	317	LBM	YES-DØ MSB	
000341/	011	DCB		
000342/	371	LMB		
000343/	007	RET	DØNE	
000344/	066	LLI 377	SET TØ CLP	ADDRESS CØMP
000346/	056	LHI 013		4 BYTES
000350/	307	LAM	FETCH MSB	
000351/	061	DCL		
000352/	317	LBM	FETCH LSB	
000353/	061	DCL		
000354/	277	CPM	CØMPARE MSB	
000355/	013	RFZ	NØT EQUAL-RETURN	
000356/	301	LAB	PUT LSB INTØ A	
000357/	061	DCL		
000360/	277	CPM	CØMPARE	
000361/	007	RET	GØ AWAY (ZF=1 IF EQUAL)	
000362/	106	CAL 000344	CØMP ADDR	CØMP ADDR AND
000365/	110	JFZ 000311	INCR ADDR	INCR IF NØT =
000370/	005	RST 000	NØW RESTART	
000371/	300	LAA		UNUSED LOCATIONS
000372/	300	LAA		
000373/	300	LAA		
000374/	300	LAA		
000375/	300	LAA		
000376/	300	LAA		
000377/	300	LAA		
001000/	377	HLT	WAIT	ØCTAL DUMP (DPØ)
001001/	046	LEI 010	SET UP CHAR/LINE	
001003/	106	CAL 001073	Ø/P CLP	

```

001006/ 106 CAL 001047 0/P CLP CONTENTS
001011/ 106 CAL 000362 INCR AND COMPARE CLP
001014/ 041 DCE INCR LINE COUNT
001015/ 150 JTZ 001001 NEW LINE, PRINT ADR
001020/ 104 JMP 001006 SAME LINE, JUST LOOP
001023/ 066 LLI 377 SET TO CLP GET DATA (FROM CLP)
001025/ 056 LHI 013 EXTENDED ADDRESS
001027/ 307 LAM MS IN 'A' PUTS DATA INTO 013370
001030/ 061 DCL
001031/ 367 LLM LS IN L
001032/ 350 LHA NOW PUT MS IN H
001033/ 002 RLC ROTATE TO TEST BIT 8
001034/ 307 LAM FETCH LS
001035/ 003 RFC RETURN IF NOT EXTENDED MEMORY
001036/ 056 LHI 013 GET DATA FROM I/P PORT
001040/ 306 LAL SET TEMP STORE LOCATION
001041/ 066 LLI 370
001043/ 121 OUT 010 0/P LS
001044/ 103 INP 001 GET DATA
001045/ 370 LMA PUT INTO MEMORY
001046/ 007 RET G0!
001047/ 106 CAL 001023 GET DATA FROM CLP
001052/ 104 JMP 000253 AND PRINT IT
001055/ 066 LLI 377 SET TO CLP 0/ 'HHHLLL'
001057/ 056 LHI 013
001061/ 016 LBI 240 0/P BLANK
001063/ 025 RST 020
001064/ 106 CAL 000256 0/P MS BYTE
001067/ 061 DCL 0/P LS BYTE
001070/ 104 JMP 000256 AND RETURN TO CALL WHEN DONE
001073/ 066 LLI 377 SET TO CLP PRINT CR/LF HHHLLL
001075/ 056 LHI 013
001077/ 106 CAL 000013 0/P CR/LF
001102/ 106 CAL 001064 0/P ADR
001105/ 016 LBI 257 0/P SLASH
001107/ 025 RST 020
001110/ 007 RET DONE
001111/ 106 CAL 001023 SET TO CLP GET CLP PUT DATA THERE
001114/ 104 JMP 000205 FETCH DATA
001117/ 300 LAA NOP NOT USED
001120/ 106 CAL 000013 0/P CR/LF OCTAL INPUT (LD0)
001123/ 046 LEI 057 SEARCH FOR SLASH (/)
001125/ 055 RST 050
001126/ 035 RST 030 FETCH CHARACTER
001127/ 301 LAB
001130/ 074 CPI 015 IS IT A CR
001132/ 150 JTZ 001123 YES- WAIT FOR ANOTHER SLASH
001135/ 106 CAL 001111 NO- PUT DATA AT CLP
001140/ 106 CAL 000362 COMPARE AND INCR CLP
001143/ 104 JMP 001126 LOOP
001146/ 066 LLI 373 SET UP 'L' COPY (CPY)
001150/ 106 CAL 001205 INPUT NEW START OF BLOCK
001153/ 106 CAL 001023 SET UP H AND L
001156/ 327 LCM FETCH DATA
001157/ 066 LLI 373
001161/ 106 CAL 001025 SET H, L TO NEW ADR
001164/ 372 LMC STORE DATA
001165/ 106 CAL 000362 INCR FROM ADR
001170/ 066 LLI 372
001172/ 106 CAL 000313 INCR TO ADR
001175/ 104 JMP 001153 LOOP
001200/ 066 LLI 377 SET CLP GET ADDRESS (2 BYTES)

```

```

001202/ 106 CAL 000013 Ø/P CR/LF
001205/ 016 LBI 252 Ø/P *
001207/ 025 RST 020
001210/ 056 LHI 013 CLP PAGE
001212/ 106 CAL 000205 GET A BYTE (MS)
001215/ 061 DCL SET L FØR LS
001216/ 035 RST 030 GET NEXT BYTE
001217/ 045 RST 040 RUB-ØUT?
001220/ 110 JFZ 000213 NØ-GET THE NEW BYTE AS BEFØRE
001223/ 060 INL YES-RESTØRE L TØ MS ADR
001224/ 307 LAM FETCH MS BYTE
001225/ 044 NDI 370 MASK 3 BITS
001227/ 370 LMA STØRE
001230/ 106 CAL 000240 GET 3 NEW BITS
001233/ 104 JMP 001215 NØW- THE LS BYTE
001236/ 106 CAL 001073 CR/LF+CLP ØCTAL EDITØR (EDT)
001241/ 106 CAL 001247 PRØCESS LINE (BYTE)
001244/ 104 JMP 001236 LØØP
001247/ 035 RST 030 FETCH I/P
001250/ 301 LAB
001251/ 074 CPI 122 TEST FØR 'R'
001253/ 150 JTZ 003014 YES-THEN RETURN
001256/ 074 CPI 052 TEST FØR '*'
001260/ 066 LLI 377 SET L TØ CLP
001262/ 150 JTZ 002110 GØ TØ LØC RØUTINE
001265/ 074 CPI 100 TEST FØR '@'
001267/ 150 JTZ 003320 GØ TØ XQT
001272/ 074 CPI 136 TEST FØR 'UP ARRØW'
001274/ 150 JTZ 000326 THEN DECR CLP
001277/ 074 CPI 040 TEST FØR BLANK
001301/ 150 JTZ 001321 PRINT THIS BYTE
001304/ 106 CAL 000200 FAILED ALL TESTS, IS I/P ØCTAL?
001307/ 013 RFZ NØ- IGNØRE IT
001310/ 106 CAL 001023 YES-SET H AND L
001313/ 106 CAL 000213 GET 2 MØRE DIGITS AND STØRE THE BYTE
001316/ 104 JMP 000311 INCR CLP AND LØØP
001321/ 106 CAL 001047 FETCH AND PRINT DATA
001324/ 035 RST 030 I/P MØRE
001325/ 301 LAB TØ 'A' REG
001326/ 074 CPI 137 IS IT BACK ARRØW
001330/ 152 CTZ 000205 YES-REPLACE DATA BYTE
001333/ 104 JMP 000311 INCR CLP AND LØØP
001336/ 066 LLI 371 INDIRECT JUMP
001340/ 056 LHI 013 SET H,L TØ UNUSED RAM
001342/ 076 LMI 104 STØRE 'JMP'
001344/ 060 INL
001345/ 371 LMB LS ADR IN 'B'
001346/ 060 INL
001347/ 370 LMA MS ADR IN 'A'
001350/ 104 JMP 013371 GØ JMP IN
001353/ 066 LLI 365 CLEAR BREAKPØINT (CBP)
001355/ 056 LHI 013 3 BYTES FØR BRKPT PØINTERS
001357/ 347 LEM WHAT WAS INSTR
001360/ 060 INL
001361/ 060 INL
001362/ 106 CAL 001027 SET H AND L
001365/ 036 LDI 100 IS L= 100 (NØ BRKPT SET)
001367/ 273 CPD
001370/ 053 RTZ YES-RETURN UNTØUCHED
001371/ 374 LME NØ- CLEAR BRKPT
001372/ 066 LLI 367 REPLACE INSTR
001374/ 056 LHI 013 PUT 100 IN MS ADR LØCATION

```

```

001376/ 373 LMD
001377/ 007 RET          GØ AWAY
002000/ 106 CAL 002050 COMPARE DATA      PRØM PRØGRAMMING
002003/ 110 JFZ 002014 PRØGRAM IF NEEDED  RØUTINE (PRG)
002006/ 106 CAL 000362 INCR ADR
002011/ 104 JMP 002000 LØØP
002014/ 046 LEI 013    ALLØW 10. TRY5
002016/ 106 CAL 002060 GØ PRØGRAM
002021/ 150 JTZ 002037 RØM AND RAM THE SAME, THEN DØNE
002024/ 041 DCE          NØ-TRY AGAIN
002025/ 110 JFZ 002016 IF ALLØWED
002030/ 106 CAL 001055 TØØ MANY TRY5 =BAD RØM
002033/ 016 LBI 277     SØ Ø/P-CLP,?
002035/ 025 RST 020
002036/ 005 RST 000     GØ AWAY AND SULK
002037/ 106 CAL 002060 DATA ØK-MAKE SURE!!!
002042/ 016 LBI 002     Ø/P NULL TØ ITY
002044/ 025 RST 020     KEEP EVERYØNE AWAKE
002045/ 104 JMP 002006 DØ NEXT BYTE
002050/ 106 CAL 001023 SET UP H,L
002053/ 306 LAL          TELL RØM LS BYTE
002054/ 121 ØUT 010
002055/ 103 INP 001     DATA FRØM RØM
002056/ 277 CPM          IS RAM DIFFERENT?
002057/ 007 RET          GØ BACK AND THINK (ZF WILL TELL)
002060/ 307 LAM          FETCH DATA
002061/ 054 XRI 377     CØMPLEMENT
002063/ 123 ØUT 011     Ø/P TØ PRØM
002064/ 006 LAI 002     ENABLE PRØGRAMMER
002066/ 127 ØUT 013
002067/ 026 LCI 000     TIME FØR ABØUT
002071/ 036 LDI 177     2 SECØNDS
002073/ 075 RST 070     WITH NESTED LØØPS
002074/ 020 INC
002075/ 110 JFZ 002071
002100/ 250 XRA          TURN ØFF PRØGRAMMER
002101/ 127 ØUT 013
002102/ 036 LDI 000     WAIT FØR THINGS TØ SETTLE
002104/ 075 RST 070
002105/ 103 INP 001     GET RØM DATA
002106/ 277 CPM          DID IT PRØGRAM?
002107/ 007 RET          GØ THINK
002110/ 066 LLI 377     ADR ØF CLP      SET CLP (LØC)
002112/ 104 JMP 003143 I/P ADR, RET HØME
002115/ 000 HLT          WAIT          BNPF DUMP (DBF)
002116/ 066 LLI 371     SCRATCH LØCATION
002120/ 076 LMI 005     Ø/P 5 BYTES PER LINE
002122/ 016 LBI 240     NØW, Ø/P A BLANK
002124/ 025 RST 020
002125/ 106 CAL 001023 GET DATA
002130/ 360 LLA          SAVE IT IN L
002131/ 016 LBI 302     Ø/P 'B'
002133/ 025 RST 020
002134/ 046 LEI 010     8 BITS PER BYTE (I THINK)
002136/ 306 LAL          RØTATE DATA IN 'L'
002137/ 002 RLC          PUT NEXT BIT IN CARRY
002140/ 360 LLA
002141/ 016 LBI 316     SET 'B' TØ 'N'
002143/ 100 JFC 002150 IF BIT IS 0, JUMP
002146/ 016 LBI 320     BIT= 1 CHANGE TØ 'P'
002150/ 025 RST 020     Ø/P WHATEVER IT IS
002151/ 041 DCE          ØNE MØRE BIT DØNE

```

```

002152/ 110 JFZ 002136 LOOP IF MORE
002155/ 016 LBI 306 DONE BYTE,0/P 'F'
002157/ 025 RST 020
002160/ 106 CAL 000362 INCR,C0MP CLP
002163/ 066 LLI 371 SET UP 'L' AGAIN
002165/ 317 LBM ONE MORE BYTE 0/P
002166/ 011 DCB
002167/ 371 LMB
002170/ 110 JFZ 002122 MORE ON THIS LINE
002173/ 106 CAL 000013 NEW LINE (CR/LF)
002176/ 104 JMP 002116 KEEP GOING
002201/ 300 LAA NOP BNPf LOAD (LBF)
002202/ 046 LEI 102 WAIT FOR A 'B'
002204/ 055 RST 050
002205/ 046 LEI 370 NOW 8 BITS EXPECTED
002207/ 106 CAL 001023 SET H,L
002212/ 076 LMI 000 CLEAR SOME RAM
002214/ 035 RST 030 FETCH CHARACTER
002215/ 301 LAB INTO 'A'
002216/ 074 CPI 116 IS IT 'N'
002220/ 150 JIZ 002232 YES- STASH IT
002223/ 054 XRI 377 NO -COMPLEMENT
002225/ 074 CPI 257 IS IT 'P'
002227/ 110 JFZ 002030 NO-ERROR
002232/ 032 RAR YES- PUT BIT IN CARRY
002233/ 307 LAM GET PREVIOUS BITS
002234/ 022 RAL ROTATE IN NEW BIT
002235/ 370 LMA STASH IT
002236/ 040 INE COUNT YOUR BITS
002237/ 110 JFZ 002214 NOT DONE,LOOP
002242/ 035 RST 030 YES-ONE MORE CHECK
002243/ 301 LAB
002244/ 074 CPI 106 LAST CHARACTER MUST BE AN 'F'
002246/ 110 JFZ 002030 NO-PANIC
002251/ 106 CAL 000362 YES-INCR CLP,CHECK IF DONE
002254/ 104 JMP 002202 LOOP IF YOU GET HERE
002257/ 066 LLI 373 BANK TO BANK TRANSLATE(TRN)
002261/ 106 CAL 000205 FETCH OLD BANK NO.
002264/ 016 LBI 337 0/P BACK ARROW
002266/ 025 RST 020
002267/ 061 DCL FETCH NEW BANK NO.
002270/ 106 CAL 000205
002273/ 106 CAL 001023 GET DATA (INSTR)
002276/ 347 LEM
002277/ 106 CAL 006320 IS IT 1,2 OR 3 BYTE INSTR
002302/ 340 LEA 'A' HAS POINTER (0=1BYTE)
002303/ 106 CAL 000362 INCR CLP (1=2BYTE)
002306/ 304 LAE ROTATE POINTER (3=3BYTE)
002307/ 012 RRC
002310/ 140 JTC 002302 LOOP FOR MORE
002313/ 074 CPI 140 WAS IT A 3 BYTE INSTR (JMP OR CAL)
002315/ 110 JFZ 002273 NO-GO TO NEXT BYTE
002320/ 106 CAL 001023 YES-SET UP H,L
002323/ 061 DCL TO LAST BYTE OF JMP OR CAL
002324/ 307 LAM FETCH BYTE
002325/ 056 LHI 013 WAS IT OUR MAGIC NO.?
002327/ 066 LLI 373
002331/ 277 CPM
002332/ 110 JFZ 002273 NO-GO AWAY
002335/ 061 DCL YES-GET THE NEW ONE
002336/ 347 LEM
002337/ 106 CAL 001023 SET UP H,L

```



```

002342/ 061 DCL      (LS-1) OF COURSE
002343/ 374 LME      REPLACE MS BYTE
002344/ 104 JMP 002273 NOW- WE ARE REALLY DONE
002347/ 106 CAL 001353 CLEAR OLD      SET BREAKPOINT (SBP)
002352/ 106 CAL 001200 FETCH ADR OF NEW BRKPT
002355/ 106 CAL 001023 SET UP H,L TO CLP
002360/ 326 LCL      SAVE H,L
002361/ 335 LDH
002362/ 076 LMI 065   SET RST 060 INTO LOCATION
002364/ 056 LHI 013   SAVE THE OLD INSTR
002366/ 066 LLI 365
002370/ 370 LMA      IT WAS LEFT IN 'A'
002371/ 060 INL
002372/ 372 LMC      'L' (LS ADR)
002373/ 060 INL
002374/ 373 LMD      'H' (MS ADR)
002375/ 104 JMP 003000 GO HOME TO MOMMY
003000/ 106 CAL 000013 CR/LF      MONITOR AND CONTROLLER
003003/ 046 LEI 010   SET UP LOOP
003005/ 016 LBI 255   CHARACTER IS '-'
003007/ 025 RST 020   GO PRINT '-----'
003010/ 041 DCE      COUNT
003011/ 110 JFZ 003007 LOOP
003014/ 146 CAL 000013 NEW LINE
003017/ 146 CAL 003067 FETCH INPUT
003022/ 100 JFC 003017 LOOP IF NOT 'A'-'Z'
003025/ 146 CAL 003100 NOW GET TWO MORE CHARACTERS
003030/ 146 CAL 003150 FIND IT IN THE TABLE
003033/ 150 JTZ 006000 NOT FOUND!! GO TO LDS
003036/ 066 LLI 373      EXEC ROUTINE
003040/ 056 LHI 013
003042/ 370 LMA      STORE ADDRESS (MS)
003043/ 061 DCL      FROM 5 BYTE TABLE
003044/ 371 LMB      STORE ADDRESS (LS)
003045/ 044 NDI 200   FROM 5 BYTE TABLE
003047/ 112 CFZ 003137 GO FETCH INITIAL AND FINAL ADDRESS
003052/ 112 CFZ 000013 IF MS=1XXXXXXX, START WITH CR/LF
003055/ 066 LLI 371   JMP IN 371
003057/ 076 LMI 104   YES IT IS AN INDIRECT JMP
003061/ 106 CAL 013371 SO GO
003064/ 144 JMP 003014 AND CONTINUE WHEN DONE
003067/ 035 RST 030   GET CHAR      CHAR TEST
003070/ 006 LAI 100   TEST FOR LT 'A'
003072/ 271 CPB
003073/ 301 LAB
003074/ 003 RFC      PASS IF GT OR EQ 'A'
003075/ 074 CPI 133   TEST IF GT 'Z'
003077/ 047 RET      GO, CARRY TELLS ALL
003100/ 046 LEI 002      SYM INPUT
003102/ 066 LLI 350   CHAR IN 013350
003104/ 056 LHI 013
003106/ 370 LMA      STORE
003107/ 146 CAL 003067 TEST NEXT CHAR
003112/ 100 JFC 003131 NOT 'A'-'Z' !! ERROR!
003115/ 060 INL      SET UP NEXT ONE
003116/ 041 DCE      COUNT
003117/ 110 JFZ 003106 NOT DONE, LOOP
003122/ 370 LMA      DONE- STORE LAST CHAR
003123/ 340 LEA      NOW SET UP REG, 3 GOES IN 'E'
003124/ 061 DCL
003125/ 337 LDM      2 IN 'D'
003126/ 061 DCL

```

```

003127/ 327 LCM          AND 1 IN 'C'
003130/ 007 RET          ALL DONE
003131/ 016 LBI 277      ERRORS COME HERE
003133/ 025 RST 020      PRINT '?'
003134/ 144 JMP 003014    GO GET ANOTHER INPUT
003137/ 106 CAL 001200    CR/LF AND '*' INITIAL AND FINAL ADR
003142/ 061 DCL          GOT FIRST ADR
003143/ 016 LBI 240      PRINT BLANK
003145/ 104 JMP 001207    GET FINAL ADR, GO BACK HOME
003150/ 016 LBI 022      SEARCH TABLE
003152/ 066 LLI 021      5 BYTE TABLE
003154/ 056 LHI 004
003156/ 307 LAM          NOW GET 1ST CHAR
003157/ 060 INL          READY FOR NEXT CHAR
003160/ 272 CPC          COMPARE TABLE AND I/P
003161/ 110 JFZ 003204    JMP IF NOT EQUAL
003164/ 307 LAM          GET 2ND
003165/ 060 INL          READY FOR 3RD
003166/ 273 CPD          COMPARE
003167/ 110 JFZ 003205    JMP IF NOT THE SAME
003172/ 307 LAM          NOW FOR THE 3RD
003173/ 060 INL          AND PREPARE FOR DATA
003174/ 274 CPE          COMPARE AS BEFORE
003175/ 110 JFZ 003206    AND JUMP IF NOT NICE
003200/ 317 LBM          GET 'GO TO' ADDRESS
003201/ 060 INL
003202/ 307 LAM          2 BYTES OF IT
003203/ 007 RET          AND RETURN
003204/ 060 INL          LOOK AT NEXT SYMBOL
003205/ 060 INL          IN THE TABLE
003206/ 060 INL
003207/ 060 INL
003210/ 011 DCB          COUNT OUR TRYs
003211/ 053 RTZ          END OF TABLE
003212/ 104 JMP 003156    MORE TO CHECK
003215/ 056 LHI 004      3BYTE TABLE SEARCH
003217/ 021 DCC          'C' IS COUNTER
003220/ 053 RTZ          RETURN WHEN DONE TABLE
003221/ 307 LAM          NOW LOOK AT THE FIRST ENTRY
003222/ 060 INL          COMPARE WITH DATA
003223/ 273 CPD          JMP IF NOT LIKED
003224/ 110 JFZ 003237    2ND ENTRY AS ABOVE
003227/ 307 LAM
003230/ 274 CPE
003231/ 110 JFZ 003237    AND JUMP, MAYBE
003234/ 060 INL          FETCH DATA FROM TABLE
003235/ 307 LAM
003236/ 007 RET          RETURN TO LDS ROUTINE
003237/ 060 INL          NEXT ENTRY
003240/ 060 INL          E00P AND TRY AGAIN
003241/ 104 JMP 003217    LOOP AND TRY AGAIN
003244/ 345 LEH          BRKPT EXECUTE
003245/ 336 LDL          SAVE L,H LOSING D,E
003246/ 066 LLI 364      SAVE REGISTERS A-E
003250/ 056 LHI 013      IN RAM (LOC 013364 TO 013360)
003252/ 374 LME
003253/ 061 DCL
003254/ 373 LMD
003255/ 061 DCL
003256/ 372 LMC
003257/ 061 DCL
003260/ 371 LMB

```

```

003261/ 061 DCL
003262/ 370 LMA
003263/ 006 LAI 030 NOW DISPLAY CARRY FLAG
003265/ 022 RAL ROTATE IN CARRY AND CONVERT TO
003266/ 340 LEA ASCII
003267/ 016 LBI 240 O/P BLANK
003271/ 025 RST 020
003272/ 314 LBE O/P CARRY FLAG
003273/ 025 RST 020
003274/ 046 LEI 005 SET UP COUNT TO PRINT REGISTERS
003276/ 066 LLI 360 START OF SAVED REGISTERS
003300/ 106 CAL 000253 PRINT BYTE AS OCTAL
003303/ 060 INL NEXT REGISTER
003304/ 041 DCE COUNT
003305/ 110 JFZ 003300 LOOP TILL DONE
003310/ 061 DCL GO BACK ONE REG
003311/ 106 CAL 001027 AND GET DATA AT H,L LOCATION
003314/ 106 CAL 000253 AND PRINT IT
003317/ 005 RST 000 NOW WERE DONE, GO HOME
003320/ 066 LLI 373 XQT ROUTINE
003322/ 106 CAL 003143 LOAD ADDRESS
003325/ 104 JMP 003052 EXEC WILL SEND US THERE

```

R0M NUMBER 3 CONTINUED WITH DPS ROUTINES

R0M 4 CONTAINS THE SYMBOL TABLES,
AS FOLLOWS:

1. 5 BYTE TABLE

THE 5 BYTE TABLE OCCUPIES POSITIONS 004021 TO 004157 INCLUSIVE AND CONTAINS ALL MONITOR COMMANDS PLUS THE MACHINE COMMANDS HLT, INP, OUT, RST, AND THE SPECIAL SYMBOL ???, INDICATING A NO FIND CONDITION ON OUTPUT. THE INPUT ROUTINE DOES NOT USE THIS SYMBOL. THE FORMAT IS THUS:

ASCII	X	Q	T	(DATA FIELD)
OCTAL	130	121	124	320 003
ADDRESS	021	022	023	024 025

WHEN A FIND IS MADE DURING A SEARCH, THE DATA FIELD IS MOVED TO REGISTERS A AND B, AND AN INDIRECT JUMP MADE TO THAT ADDRESS, IF THE MS HALF OF THE ADDRESS IS A 2XX, THE EXEC WILL LOOK FOR TWO ADDRESSES BEFORE GOING TO THE ROUTINE.

DURING A SYMBOLIC DUMP, THE LAST 5 SYMBOLS ARE USED FOR THE APPROPRIATE MACHINE COMMANDS, AND ARE STORED AS OUTPUT.

ADR	SYMB	LS	MS (ADDRESS OF ROUTINE)
004021/	XQT	320	003
004026/	EDT	236	001
004033/	LD0	120	201
004040/	LBF	201	202
004045/	DP0	000	201
004052/	DBF	115	202
004057/	DPS	000	205
004064/	CPY	146	201
004071/	TRN	257	202
004076/	SBP	347	002
004103/	CBP	353	001
004110/	PRG	000	202
004115/	L0C	110	002

```

004122/ DLP 055 001
004127/ HLT 046 006
004134/ RST 270 006
004141/ INP 270 006
004146/ OUT 270 006
004153/ ???

```

2. 3 BYTE TABLE

THIS TABLE CONTAINS TWO BYTES OF ASCII CODE AND ONE DATA BYTE, WHICH IS A MASKED PORTION OF THE INSTRUCTION. THE FORMAT IS:

ASCII	N	D	(DATA)
0CTAL	116	104	244
LOCATION	252	253	254

THE TABLE OCCUPIES LOCATIONS 004156 TO 004273, AND IS USED IN TWO WAYS. THE LDS ROUTINE COMPARES THE TWO ASCII CHARACTERS TO THE INPUT CHARACTERS, AND RETURNS THE DATA IN THE A REGISTER IF A FIND IS MADE. FOR THE DPS ROUTINE, THE PARTIAL WORD (DATA) IS TESTED, AND THE H AND L REGISTERS ARE USED TO RETRIEVE THE ASCII AS NEEDED.

3 BYTE TABLE:

LOCATION	ASCII	DATA	
004156	LC	002	
004161	RC	012	
004164	AL	022	
004167	AR	032	
004172	MP	104	(JMP)
004176	AL	106	(CAL)
004203	ET	007	(RET)
004206	TC	040	
004211	FC	000	
004214	TZ	050	
004217	FZ	010	
004222	TS	060	
004225	FS	020	
004230	TP	070	
004233	FP	030	
004236	AD	204	
004241	AC	214	
004244	SU	224	
004247	SB	234	
004252	ND	244	
004255	XR	254	
004260	OR	264	
004263	CP	274	
004266	IN	000	
004271	DC	001	

3. 4 BYTE TABLE

THE 4 BYTE TABLE OCCUPIES POSITIONS 004274 TO 004377, AND IS USED BY THE DPS ROUTINE.

THE FORMAT IS:

	MASK	DATA	ADDRESS	DATA FIELD
	361	101	161	144
L0C	310	311	312	313

THE MASK CHARACTER IS USED TO MASK (AND) DON'T CARE BITS IN THE INPUT BYTE, THE REMAINING BITS ARE COMPARED TO THE DATA IN THE

NEXT FIELD TO DECODE AN INSTRUCTION. IF A FIND IS MADE THE ADDRESS IS USED FOR AN INDIRECT JUMP (TO 005AAA). THE LAST ENTRY IS AN UNCONDITIONAL FIND WHICH OUTPUTS THE ERROR SYMBOL ???.

THE DATA FIELD COLUMN IS USED FOR VARIOUS PURPOSES BY THE CALLED ROUTINES.

4 BYTE TABLE

LOCATION	MASK	DATA	ADDRESS	DATA FIELD
004274	377	377	155	132
004300	376	000	155	132
004304	376	070	155	156
004310	361	101	161	144
004314	347	002	251	037
004320	307	006	262	352
004324	307	005	161	137
004330	307	004	125	111
004334	307	001	142	273
004340	307	000	142	270
004344	303	003	215	202
004350	303	102	215	176
004354	301	101	161	151
004360	303	100	215	172
004364	300	300	272	000
004370	300	200	120	000
004374	000	000	155	156

SYMBOLIC ROUTINES

NOTE: THESE ROUTINES COVER PART OF ROM 3,4 AND ALL OF ROMS 5,6

```

003330/ 106 CAL 005352 GET 3 BYTES   DPS OUTPUT
003333/ 106 CAL 005104 LOAD THEM INTO REGISTERS
003336/ 106 CAL 005313 OUTPUT THEM
003341/ 347 LEM          LOAD E WITH DATA
003342/ 106 CAL 006320 DECODE LENGTH
003345/ 012 RRC          1 BYTE INSTR?
003346/ 100 JFC 005363 YES-GO TO LINE CHECK
003351/ 340 LEA          SAVE LENGTH BITS
003352/ 106 CAL 000362 INCR ADR
003355/ 106 CAL 001023 GET DATA
003360/ 041 DCE          3 BYTES MAYBE?
003361/ 160 JTS 003372 SIGN FLAG =1 IF S0
003364/ 106 CAL 000253 O/P IMMEDIATE DATA
003367/ 104 JMP 005363 AND GO TO LINE CHECK
003372/ 327 LCM          YES ITS 3 BYTE! GET LS ADR
003373/ 106 CAL 000362 INC CLP
003376/ 312 LBC          MOVE ADR TO B
003377/ 300 LAA          NOP (UNUSED BYTE)
004000/ 106 CAL 001023 GET DATA (MS ADR BYTE)
004003/ 327 LCM          SAVE IN C
004004/ 106 CAL 005104 LOAD 3 BYTES
004007/ 060 INL          SET UP DATA POINTERS
004010/ 060 INL
004011/ 106 CAL 001061 OUTPUT THIS ADDRESS
004014/ 104 JMP 005363 AND GO ON TO LINE CHECK
005000/ 016 LBI 012
005002/ 025 RST 020      PRINT 3 LF'S
005003/ 025 RST 020
005004/ 025 RST 020
005005/ 046 LEI 076      SET UP LINES/PAGE

```

SYMBOLIC DUMP (DPS)

```

005007/ 066 LLI 353      AND STØRE NUMBER AT 013353
005011/ 056 LHI 013
005013/ 374 LME
005014/ 106 CAL 001073 GET CLP AND PRINT IT
005017/ 106 CAL 001047 GET DATA AND PRINT IT
005022/ 347 LEM          SAVE DATA IN 'E'
005023/ 337 LDM          AND IN 'D'
005024/ 106 CAL 005063 ASSUME BITS 3-5 ARE A DESTINATION REG.
005027/ 061 DCL          STØRE IT IN 013351,013352
005030/ 370 LMA
005031/ 066 LLI 274      SET UP START ØF 4 BYTE TABLE
005033/ 056 LHI 004
005035/ 303 LAD          GET MASK FRØM TABLE
005036/ 247 NDM          AND MASK DØNT CARE BITS
005037/ 060 INL          NØW CHECK THE REST
005040/ 277 CPM          WITH THE TABLE
005041/ 110 JFZ 005055 JUMP IF NØ FIND
005044/ 006 LAI 005      LØAD MS BYTE ØF ADR
005046/ 060 INL
005047/ 317 LBM          LØAD LS BYTE ØF ADR
005050/ 060 INL
005051/ 327 LCM          LØAD C WITH DATA FRØM TABLE
005052/ 104 JMP 001336 AND DØ AN INDIRECT JUMP TØ RØUTINE
005055/ 060 INL          (NØ FIND) INCR L TØ
005056/ 060 INL          NEXT TABLE ENTRY
005057/ 060 INL
005060/ 104 JMP 005035 GØ LØØP
005063/ 303 LAD          GET DATA          REGISTER DECØDE
005064/ 012 RRC
005065/ 012 RRC          LØØK AT BITS 3-5
005066/ 012 RRC
005067/ 044 NDI 007      MASK THE REST
005071/ 004 ADI 370      AND ADD START ØF TABLE
005073/ 360 LLA          TABLE ADR TØ 'L'DR
005074/ 056 LHI 006      MS ADR ØF TABLE
005076/ 307 LAM          GET REGISTER
005077/ 066 LLI 352
005101/ 104 JMP 005114 DØNE
005104/ 066 LLI 352      SET DP          3 BYTE LØAD
005106/ 056 LHI 013
005110/ 372 LMC          SAVE C
005111/ 061 DCL
005112/ 371 LMB          SAVE B
005113/ 061 DCL          L=350 NØW
005114/ 056 LHI 013      ENTRY FØR 1 BYTE LØAD
005116/ 370 LMA          AND SAVE A
005117/ 007 RET          GØ AWAY SØMWHERE
005120/ 303 LAD          ACC GRØUP RØUTINE
005121/ 106 CAL 005067 DECØDE SØURCE REG
005124/ 320 LCA          AND PUT IN C
005125/ 303 LAD          ENTRY FØR IMMEDIATE
005126/ 044 NDI 070      MASK ØUT SØURCE PART
005130/ 004 ADI 204      (SPECIALLY FØR 'I' INSTR)
005132/ 066 LLI 240      START ØF ACC IN 3 BYTE TABLE
005134/ 106 CAL 005336 GØ FIND DATA IN TABLE
005137/ 104 JMP 003333 GØ PRINT IT
005142/ 106 CAL 005063          INX,DCX RØUTINE
005145/ 362 LLC          SET UP ADR FØR 3 BYTE TABLE
005146/ 320 LCA          SAVE 'A' FØR NØW
005147/ 303 LAD          GET BINARY DATA
005150/ 044 NDI 001      MASK ALL BUT LS BIT
005152/ 104 JMP 005134 SEACH TABLE, GØ HØME

```

```

005155/ 362 LLC          GET ADR FOR 5 BYTE TABLE
005156/ 104 JMP 003330 G0 TO OUTPUT
005161/ 362 LLC          TABLE ADR      INP/OUT/RST
005162/ 343 LED
005163/ 106 CAL 005305 0/P SYMBOL
005166/ 307 LAM          FETCH DATA
005167/ 044 NDI 300      CHECK BITS 6-7
005171/ 307 LAM          AND RESTORE DATA
005172/ 150 JTZ 005210 JMP IF OOOXXXXX (RST)
005175/ 044 NDI 076      MASK TO OOOXXXXXO
005177/ 012 RRC          SET UP I/O PORT NO.
005200/ 066 LLI 352      PUT THE NUMBER AWAY
005202/ 056 LHI 013      FOR NOW
005204/ 370 LMA
005205/ 104 JMP 003364 G0 TO OUTPUT
005210/ 044 NDI 070      MASK DATA TO OOOXXX000 (RST NO.)
005212/ 104 JMP 005200 G0 OUTPUT IT
005215/ 362 LLC          JMP/CAL/RET GROUP
005216/ 056 LHI 004      SET UP FOR TABLE
005220/ 347 LEM          FETCH J,C,OR R FROM TABLE
005221/ 303 LAD          RESTORE BINARY
005222/ 044 NDI 307      MASK AND CHECK IF UNCONDITIONAL
005224/ 060 INL          TRANSFER
005225/ 060 INL
005226/ 060 INL
005227/ 277 CPM
005230/ 150 JTZ 003330 YES-G0 TO OUTPUT
005233/ 066 LLI 210      NO-LOOK UP CONDITION
005235/ 303 LAD          IN 3 BYTE TABLE
005236/ 044 NDI 070      MASK ALL BUT CONDITION
005240/ 106 CAL 005336 G0 SEARCH
005243/ 321 LCB          CHAR 3
005244/ 310 LBA          CHAR 2
005245/ 304 LAE          CHAR 1
005246/ 104 JMP 003333 G0 OUTPUT
005251/ 303 LAD          GET DATA      ROTATE GROUP
005252/ 242 NDC          MASK AS PER TABLE
005253/ 066 LLI 160      ROT IN 3 BYTE TABLE
005255/ 046 LEI 122      LOAD E WITH 'R' AND PRETEND
005257/ 104 JMP 005240 ITS A TRANSFER
005262/ 362 LLC          SET UP ADR      LOAD IMMEDIATE
005263/ 056 LHI 013
005265/ 076 LMI 111      LOAD 3RD CHAR AS 'I' (SOURCE REG.)
005267/ 104 JMP 005276 AND TREAT AS ORDINARY LOAD
005272/ 303 LAD          GET DATA      LOAD (REG TO REG)
005273/ 106 CAL 005067 GET SOURCE REG
005276/ 061 DCL
005277/ 061 DCL
005300/ 076 LMI 114      LOAD 'L' AS 1ST CHAR
005302/ 104 JMP 003336 G0 TO OUTPUT
005305/ 106 CAL 005352 3 BYTE TRANSFER
005310/ 106 CAL 005104 3 BYTE LOAD
005313/ 016 LBI 240      PRINT 3 BYTES (ROUTINE)
005315/ 025 RST 020      0/P TWO BLANKS
005316/ 025 RST 020
005317/ 066 LLI 350      ADR OF FIRST CHAR
005321/ 056 LHI 013
005323/ 317 LBM          FETCH IT
005324/ 025 RST 020      PRINT
005325/ 060 INL          NEXT CHAR
005326/ 317 LBM
005327/ 025 RST 020      PRINT

```

```

005330/ 060 INL          ONCE MORE NOW
005331/ 317 LBM
005332/ 025 RST 020
005333/ 104 JMP 001023 GO GET MORE DATA
005336/ 056 LHI 004          3 BYTE TABLE SEARCH
005340/ 277 CPM          COMPARE
005341/ 150 JIZ 005356 EXIT IF FOUND
005344/ 060 INL          NEXT ENTRY
005345/ 060 INL
005346/ 060 INL
005347/ 104 JMP 005340 LOOP
005352/ 056 LHI 004          3 BYTE TRANSFER
005354/ 061 DCL
005357/ 317 LBM          2ND CHAR
005360/ 061 DCL
005361/ 307 LAM          1ST CHAR
005362/ 007 RET          GO!
005363/ 106 CAL 000362 INCR CLP          LINE CHECK
005366/ 066 LLI 353
005370/ 347 LEM          FETCH LINE COUNT
005371/ 041 DCE          UPDATE IT
005372/ 150 JIZ 005000 END OF PAGE 0/P 3 LF'S
005375/ 104 JMP 005013 OK GO ON TO NEXT LINE
006000/ 302 LAC          GET 1ST CHAR SYMBOLIC LOAD (LDS)
006001/ 074 CPI 114          IS IT AN 'L'
006003/ 110 JFZ 006122 NO-TEST FOR 'R'
006006/ 016 LBI 306          PARTIAL WORD LOAD INSTRUCTION
006010/ 303 LAD          LOOK AT 2ND CHAR
006011/ 146 CAL 006345 ENCODE AS DESTINATION REG
006014/ 002 RLC
006015/ 002 RLC
006016/ 002 RLC
006017/ 201 ADB          STASH WITH PARTIAL WORD
006020/ 310 LBA          IN 'B'
006021/ 006 LAI 111          IS 3RD CHAR AN 'I' ?
006023/ 274 CPE
006024/ 150 JIZ 006043 YES- GO TO IMMEDIATE ROUTINE
006027/ 304 LAE          NO-ENCODE SOURCE REGISTER AS ABOVE
006030/ 146 CAL 006345
006033/ 320 LCA
006034/ 301 LAB          GET DUMMY WORD
006035/ 044 NDI 370          DISCARD BITS 0-3 (A=3X6)
006037/ 202 ADC          AND PUT IN THE REAL ONE
006040/ 144 JMP 006046 NOW GO CLEAN UP
006043/ 006 LAI 077          IMMEDIATE LOAD
006045/ 241 NDB          MASK TO 00XXXXXX
006046/ 340 LEA          A HAS INSTR FINISH ROUTINE
006047/ 146 CAL 001023 GET CLP
006052/ 374 LME          PUT INSTR THERE
006053/ 300 LAA          NOP (NOT USED)
006054/ 146 CAL 006320 DECODE LENGTH
006057/ 146 CAL 000311 INCR CLP
006062/ 012 RRC          CHECK LENGTH BITS
006063/ 100 JFC 003014 LEAVE US WHEN NO MORE BITS IN CARRY
006066/ 340 LEA          NOT DONE-SAVE THE BITS
006067/ 016 LBI 240          PRINT A BLANK
006071/ 025 RST 020
006072/ 041 DCE          IS IT A 3BYTE INSTR?
006073/ 160 JIS 006105 SIGN FLAG TELLS ALL (SF=1 FOR 3 BYTE INSTR)
006076/ 106 CAL 001111 GET DATA AND INPUT

```



```

006101/ 250 XRA          CLEAR A
006102/ 104 JMP 006057 AND LOOP
006105/ 106 CAL 000311 INCR CLP          3BYTE (MUST WANT AN ADR)
006110/ 106 CAL 001023 GET MORE DATA
006113/ 106 CAL 001212 AND STORE TWO BYTES (CLP,CLP-1)
006116/ 104 JMP 006101 GO BACK TO LOOP
006121/ 377 HLT          UNUSED HALT(1)
006122/ 074 CPI 122          TEST FOR 1ST CHAR = 'R'
006124/ 110 JFZ 006146 NO- KEEP LOOKING
006127/ 026 LCI 005          IS IT A ROTATE?
006131/ 066 LLI 156
006133/ 106 CAL 003215 SEARCH 3 BYTE TABLE
006136/ 066 LLI 202
006140/ 150 JTZ 006234 IF NO FIND, TEST FOR RETURN
006143/ 144 JMP 006046 GO FINISH UP
006146/ 314 LBE          ACC GROUP
006147/ 343 LED          PUT CHARACTERS AWAY
006150/ 332 LDC
006151/ 026 LCI 013          SET UP TABLE SEARCH
006153/ 066 LLI 236          (ACC GROUP, IN(R), DC(R))
006155/ 106 CAL 003215 SEARCH TABLE
006160/ 150 JTZ 006214 NO FIND, KEEP LOOKING
006163/ 320 LCA          GET 1ST CHAR
006164/ 044 NDI 200          CHECK FOR IMMEDIATE INSTR
006166/ 302 LAC          RESTORE CHAR
006167/ 341 LEB
006170/ 110 JFZ 006020 GO AWAY IF IMMEDIATE INSTR
006173/ 301 LAB          TEST THE 3RD CHAR
006174/ 106 CAL 006345 ENCODE AS A REGISTER
006177/ 002 RLC
006200/ 002 RLC
006201/ 002 RLC
006202/ 202 ADC          ADD TO PARTIAL WORD
006203/ 104 JMP 006046 FINISH UP
006206/ 106 CAL 001023 6 BYTES NOT USED (1)
006211/ 104 JMP 003143 (111)
006214/ 323 LCD          TRANSFER GROUP (JMP,CAL,RET)
006215/ 334 LDE          MUSICAL REGISTERS
006216/ 341 LEB
006217/ 066 LLI 172          START OF TABLE (JMP)
006221/ 302 LAC
006222/ 277 CPM          TRY 1ST CHAR
006223/ 150 JTZ 006234 JUMP IF FIND
006226/ 066 LLI 176          TRY 'CAL'
006230/ 277 CPM
006231/ 110 JFZ 003131 NO (1) MUST BE AN ERROR
006234/ 060 INL          IS IT UNCONDITIONAL?
006235/ 026 LCI 002
006237/ 146 CAL 003215 THEN GO TO FINISH
006242/ 110 JFZ 006046 GET PART WORD
006245/ 061 DCL          AS MUCH AS WE CAN
006246/ 307 LAM          BLANK OUT SOME
006247/ 044 NDI 303
006251/ 310 LBA          AND LOOK UP THE CONDITION
006252/ 066 LLI 206
006254/ 026 LCI 011
006256/ 146 CAL 003215
006261/ 150 JTZ 003131 NO-FIND ERROR(1)
006264/ 201 ADB          ADD IN CONDITION BITS
006265/ 144 JMP 006046 FINISH IT
006270/ 343 LED          INP/OUT/RST
006271/ 016 LBI 240          ENTER AS MONITOR ROUTINE

```

```

006273/ 025 RST 020 PRINT A BLANK
006274/ 146 CAL 000205 INPUT THE OCTAL ARGUMENT
006277/ 300 LAA NOP (AGAIN -REALLY TØM !!)
006300/ 304 LAE GET 2ND CHAR
006301/ 074 CPI 123 IS IT AN 'S'
006303/ 307 LAM GET THE OCTAL ARGUMENT
006304/ 046 LEI 005 ASSUME IT'S RST
006306/ 150 JTZ 006314 AND SKIP AHEAD IF IT IS
006311/ 002 RLC MUST BE INP/ØUT-ROTATE ARGUMENT
006312/ 002 RLC MUST BE INP/ØUT-ROTATE ARGUMENT
006320/ 250 XRA INSTRUCTION LENGTH TEST
006321/ 310 LBA CLEAR REGISTERS
006322/ 304 LAE GET DATA
006323/ 044 NDI 305
006325/ 074 CPI 004 IS IT IMMEDIATE?
006327/ 150 JTZ 006342 YES-BEGØNE
006332/ 044 NDI 301
006334/ 074 CPI 100 IS IT A TRANSFER?
006336/ 301 LAB CLEAR A
006337/ 013 RFZ PASS IF 3 BYTE (JMP,CAL GRØUP)
006340/ 010 INB NØW SET UP B
006341/ 010 INB
006342/ 010 INB CØME HERE IF 2 BYTE
006343/ 301 LAB SØ NØW A IS 001 ØR 003
006344/ 007 RET GØ HØME AND TELL ABOUT IT
006345/ 066 LLI 370 REGISTER DECØDE
006347/ 056 LHI 006 LØØK AT TABLE
006351/ 277 CPM TEST
006352/ 110 JFZ 006361 NØ FIND -LØØP
006355/ 306 LAL A FIND! GET THE ADDRESS
006356/ 044 NDI 007 MASK 00000XXX
006360/ 007 RET AND RETURN WITH A NUMBER
006361/ 060 INL NEXT VALUE
006362/ 110 JFZ 006351 NØT ZERO GØ LØØP
006365/ 104 JMP 003131 NØT IN TABLE- ITS AN ERRØR FØLKS

```

REGISTER LØØK UP TABLE

LØCATION	REGISTER	BINARY	ASCII
006370	A	0	101
006371	B	1	102
006372	C	2	103
006373	D	3	104
006374	E	4	105
006375	H	5	110
006376	L	6	114
006377	MEMØRY(M)	7	115

1110 LINES TALLIED AT EOF

*

TIMEØUT