

Copyrighted © 1985 by ASCII Corporation of Japan

All rights Reserved

Edited: January 1985 by Steven M. Ting Graphic design: Mervin Fong.

The information in this document is subject to change without notice. ASCII Corp. makes no warranty with regard to the manual, including but not limited to, implied warranties of merchantability and fitness for a particular purpose. The parties above assume no responsibility for any errors which may appear in this document.

This document is not intended as "Consumer goods" under applicable federal or state law(s).

No part of this document may be copied or reproduced in any form or by any means without the prior written consent of ASC! Corporation and Qest Publishing Inc.

MSX is a registered trademark of Microsoft Corporation, Bellevue, WA. 280 is a registered trademark of Zilog, Inc.

Printed in United States

Scanned and converted to PDF by HansO, 2003

TABLE OF CONTENTS

BIOS LISTING	1 -	256
MSX BIOS CROSS REFERENCE	257 -	280
SYMBOL TABLE	281 -	285
APPENDIX A MSX USA & UK OVERLAY PATCHES BIOS CALLS		
APPENDIX B CHARACTER SET & KEYBOARD LAYOUT HOOKS & RAM ROUTINES		

(MSX ROM BASIC BIOS) Macro-80 3.44 01-Jan-85 PAGE 1 -BIOS header- BIOS calls (Basic Interpreter, Slot I/O)

1	.list
2	;
.3	;
4	; (C) Copyright by ASCII Corp., 1983
5	; Proprietary information. All rights reserved.
6	;
7	; File: BIOHDR.MAC
8	; USE: Restart calls and ROM entries table
, 9	; Written by Jey Suzuki, Rick Yamashita
10	; ASCII Corporation, Japan
10 11	;
12	; Edit: January, 1985
13	; Reason: Zilog Z80 Mnemonic version and cleanup
14	; Edited by: Steven M. Ting
15	;
16	;
17	; Labels referenced in this listing, are the absolute locations
18	; within the MSX ROM. However, "ONLY" this BIOS entry point table,
19	; and RAM variables are guaranteed to be permanent.
20	;
21	; All other locations in the ROM, will be changed without notice.
22	;
23	SUBTTL -BIOS header- BIOS calls (Basic Interpreter, Slot I/O)

		BIOS) Macro OS calls (Bas		3.44 eter,	01-Jan-85 Slot I/O)	PAGE 2
24						
25			;			
26				follow	ing RST's (RST	0 thru RST 5) are reserved for BASIC
27						nter-slot calls, and RST 7 for
28			; hard	ware i	nterrupt	
29			;		-	
30	0000	F3	BEGIN:	DI		;Fail safe
31	0001	C3 02D7		JP	CHKRAM	;Finds all connected RAM
32						and cartridges
33			;			
34			,			
35				pecial	information for	or the VDP. **
36						s the VDP hardware directly
37			; shou	ld rea	d the I/O port	address found here, to be certain
38			; the	softwa	re is compatib	le with future versions of the VDP.
39			;			
40	0004	1 BBF		DW	CGTABL	;Address of character generator table,
41						; to allow use of other character ROM.
42						1
43	0006	98		DB	98H	;Current port address for VDP Data read
44	0007	98		DB	98H	; " " " " " write
45			;			
46	0008	C3 2683		JP	SYNCHR	;Check byte following the RST 8, see
47						if equal to the byte pointed by HL
48	000B	00		DB	0	
49	000C	C3 01B6		JP	RDSLT	;Read a byte from another slot
50	000F	00		DB	0	-
51	0010	C3 2686		JP	CHRGTR	;Fetch next char from BASIC text
52	0013	00		DB	0	
53	0014	C3 01D1		JP	WRSLT	;Write a byte to another slot
54	0017	00		DB	0	

. 1

2

		BIOS) Macro-6 OS calls (Basic	3.44 3.44 Interpreter, S	01-Jan-85 lot I/O)	PAGE 2-1	3
55	0018	C3 1B45	JP	OUTDO	;Output a char to the Conso	le or printer
56	001B	00	DB	0		
57	001C	C3 0217	JP	CALSLT	;Perform Inter-slot call	
58	001F	00	DB	0		
59	0020	C3 146A	JP	DCOMPR	;Compares [HL] to [DE]	
60	0023	00	DB	0		
61	0024	C3 025E	JP	ENASLT	;Permanently enables a slot	
62	0027	00	DB	0		
63	0028	C3 2689	JP	GETYPR	;Returns the [FAC] type	
64	002B	00	DB	0	;ID Byte (1)	
65					;Format:	
66 67					; B7 B6 B5 B4 B3 B2 B1 B0	
68					,	Muna of character
69						Type of character
70						generator. 0:Japanese
71						l:International
72						2:Korea
73					-	Date format
74						0: Y-M-D 1: M-D-Y
75					; + +	2: D-M-Y
76					;	Interrupt frequency
77					;	0: 60 Hz 1: 50 Hz
78	002C	00	DB	0	;ID Byte (2)	
79					;Format:	
80					; B7 B6 B5 B4 B3 B2 B1 B0	
81					; + + + + + + + +	
82						Type of Keyboard
83 84						0:Japanese 2:French
85						1:Int 3:UK 4:DIN
					; + + + +	4:DIN
(MSX ROM		BIOS) Macro-6 DS calls (Basid	0 3.44 Interpreter,	01-Jan-85 Slot I/O)	PAGE 2-2	
86					:	Version of BASIC
87						0: Japanese
88						1: International
89	002D	00 00 00	DB	0,0,0		
90	0030	C3 0205	JP	CALLF	;Performs Far-call (i.e	., Inter-slot)
91	0033	00 00 00 00	DB	0,0,0,0,0		
92	0037	00				
93.			;			
94			; ,			
95				are used for T	/O initialization	
96			; 10110#1119	210 4004 101 1	.,	· · ·
90 97	0038	C3 0C3C	; JP	KEYINT	;Handlers for hardware	interrupt
	0038 003B	C3 049D	JP	INITIO	;Do device initializati	
	0035					
98	0032	C3 130D	חד	TNTENE		
	003E	C3 139D	JP ;	INIFNK	Reset all function key;	stext

102 103		OS calls (Vi	o-80 deo display	3.44 proces	01-Jan-85 ssor)	PAGE 3
103						
			;			
104						provides control of the
105						ode settings, and memory block
106			; move	betwee	en DRAM and VRAM	1.
107			;			
108	0041	C3 0577		JP	DISSCR	;Disables screen display
109	0044	C3 0570		JP	ENASCR	;Enables screen display
110	0047	C3 057F		JP	WRTVDP	;Write a byte to any VDP register
111	004A	C3 07D7		JP	RDVRM	;Read VRAM addressed using [HL]
112	004D	C3 07CD		JP	WRTVRM	;Write VRAM addressed using [HL]
113	0050	C3 07EC		JP	SETRD	;Sets up VDP for read
114	0053	C3 07DF		JP	SETWRT	;Sets up VDP for write
115	0056	C3 0815		JP	FILVRM	;Fills VRAM with specified data
116	0059	C3 070F		JP	LDIRMV	;Moves block of data from VRAM to memory
117	005C	C3 0744		JP	LDIRVM	; " " " " memory to VRAM
118	005F	C3 084F		JP	CHGMOD	;Change screen_mode of VDP to [SCRMOD]
119	0062	C3 07F7		JP	CHGCLR	; change Foreground, background,
120						;border, color
121	0065	00		DB	0	/
122	0000			;	v	
123			;	,		
124	0066	C3 1398	,	JP	NMI	;Handler for non-maskable interrupt
125	0000	05 1550	;	01	INTI	, manufer for non-maskable interrupt
126	0069	C3 06A8	,	JP	CLRSPR	;Init sprite data
127	006C	C3 050E		JP	INITXT	;Init VDP for 40 X 24 text mode (SCREEN 0)
128	006F	C3 0538		JP	INIT32	; " " " 32 X 24 text mode (SCREEN 1)
129	0072	C3 05D2		JP	INIGRP	; " " " High resolution mode (SCREEN :
130	0075	C3 061F		JP	INIMLT	; " " " Multi color mode (SCREEN 3)
131	0078	C3 0594		JP	SETTXT	;Sets VDP to display 40 X 24 text mode
132	007B	C3 05B4		JP	SETT32	; " " " 32 X 24 text mode
		BIOS) Macro OS calls (Vie		3.44 proces	01-Jan-85 ssor)	PAGE 3-1
133	007E	C3 0602		JP	SETGRP	;""" High-res mode
	0081	C3 0659		JP	SETMLT	, " " " Multi color mode
134	0084	C3 06E4		-TP	CALPAT	
134 135	0084	C3 06E4		JP JP	CALPAT CALATR	;Get address of sprite pattern table
134 135 136	0087	C3 06F9		JP	CALATR	;Get address of sprite pattern table
134 135 136 137	0087 008A	C3 06F9 C3 0704		JP JP	CALATR GSPSIZ	;Get address of sprite pattern table ; " " " attribute table
134 135 136 137 138	0087	C3 06F9		JP JP JP	CALATR	;Get address of sprite pattern table ; " " " attribute table ;Returns current sprite size
134 135 136 137	0087 008A	C3 06F9 C3 0704	SUBTTL	JP JP JP ;	CALATR GSPSIZ GRPPRT	;Get address of sprite pattern table ; " " " attribute table ;Returns current sprite size
134 135 136 137 138 139 140	0087 008A 008D	C3 06F9 C3 0704 C3 1510 BIOS) Macro	-80	JP JP ; -BIOS 1	CALATR GSPSIZ GRPPRT	;Get address of sprite pattern table ; " " " attribute table ;Returns current sprite size ;Print a character on the graphic screen
134 135 136 137 138 139 140	0087 008A 008D	C3 06F9 C3 0704 C3 1510 BIOS) Macro	-80	JP JP ; -BIOS 1	CALATR GSPSIZ GRPPRT header- BIOS cal	;Get address of sprite pattern table ; " " " attribute table ;Returns current sprite size ;Print a character on the graphic screen lls (Programmable Sound Generator control)
134 135 136 137 138 139 140 (MSX ROM	0087 008A 008D	C3 06F9 C3 0704 C3 1510 BIOS) Macro	-80 : grammable S	JP JP ; -BIOS 1	CALATR GSPSIZ GRPPRT header- BIOS cal	;Get address of sprite pattern table ; " " " attribute table ;Returns current sprite size ;Print a character on the graphic screen lls (Programmable Sound Generator control)
134 135 136 137 138 139 140 (MSX ROM -BIOS hea 141	0087 008A 008D	C3 06F9 C3 0704 C3 1510 BIOS) Macro	-80 grammable So ;	JP JP JP ; -BIOS 1 3.44 ound Ge	CALATR GSPSIZ GRPPRT header- BIOS cal 01-Jan-85 enerator cont	;Get address of sprite pattern table ; " " " attribute table ;Returns current sprite size ;Print a character on the graphic screen lls (Programmable Sound Generator control)
134 135 136 137 138 139 140 MSX ROM BIOS hea 141 142	0087 008A 008D	C3 06F9 C3 0704 C3 1510 BIOS) Macro	-80 grammable S ; ; Folloo	JP JP JP ; -BIOS 1 3.44 ound Ge wing er	CALATR GSPSIZ GRPPRT header- BIOS cal 01-Jan-85 enerator cont htry points are	;Get address of sprite pattern table ;""" attribute table ;Returns current sprite size ;Print a character on the graphic screen Ils (Programmable Sound Generator control) PAGE 4 used for PSG initialization,
134 135 136 137 138 139 140 (MSX ROM -BIOS hea 141 142 143 144	0087 008A 008D	C3 06F9 C3 0704 C3 1510 BIOS) Macro	-80 grammable S ; ; Follow ; read	JP JP JP ; -BIOS 1 3.44 ound Ge wing er	CALATR GSPSIZ GRPPRT header- BIOS cal 01-Jan-85 enerator cont htry points are	;Get address of sprite pattern table ; " " " attribute table ;Returns current sprite size ;Print a character on the graphic screen 11s (Programmable Sound Generator control) PAGE 4
134 135 136 137 138 139 140 BIOS hea 141 142 143 144 145	0087 008A 008D 4 BASIC H der- BIC	C3 06F9 C3 0704 C3 1510 BIOS) Macro OS calls (Pro	-80 ; ;;; ; Folloo; ; read;;	JP JP JP ; -BIOS 1 3.44 ound Ge wing er and wr	CALATR GSPSIZ GRPPRT header- BIOS cal 01-Jan-85 enerator cont ntry points are ite PSG register	;Get address of sprite pattern table ;""" attribute table ;Returns current sprite size ;Print a character on the graphic screen Ils (Programmable Sound Generator control) PAGE 4 used for PSG initialization, rs, and PLAY statement execution.
134 135 136 137 138 139 140 BIOS hea 141 142 143 144 145 146	0087 008A 008D 4 BASIC 1 der- BIC	C3 06F9 C3 0704 C3 1510 BIOS) Macro DS calls (Pro C3 04BD	-80 ; ; ; Follow ; read ; ;	JP JP JP ; -BIOS h 3.44 ound Ge wing er and wr	CALATR GSPSIZ GRPPRT header- BIOS cal 01-Jan-85 enerator cont htry points are ite PSG register GICINI	;Get address of sprite pattern table ; " " " attribute table ;Returns current sprite size ;Print a character on the graphic screen Ils (Programmable Sound Generator control) PAGE 4 used for PSG initialization, rs, and PLAY statement execution. ;Init PSG, and static data for PLAY
134 135 136 137 138 139 140 (MSX ROM BIOS hea 141 142 143 144 145 146 147	0087 008A 008D 4 BASIC 1 1der- BIC 0090 0093	C3 06F9 C3 0704 C3 1510 BIOS) Macro DS calls (Pro C3 04BD C3 1102	-80 ; grammable S ; ; Follow ; read ;	JP JP JP ; -BIOS 1 3.44 ound Ge wing er and wr JP	CALATR GSPSIZ GRPPRT header- BIOS cal 01-Jan-85 enerator cont ntry points are ite PSG register GICINI WRTPSG	;Get address of sprite pattern table ;""" attribute table ;Returns current sprite size ;Print a character on the graphic screen lls (Programmable Sound Generator control) PAGE 4 used for PSG initialization, rs, and PLAY statement execution. ;Init PSG, and static data for PLAY ;Write data to PSG
134 135 136 137 138 139 140 (MSX ROM BIOS hea 141 142 143 144 145 146 147 148	0087 008A 008D 4 BASIC H der- BIO 0090 0093 0096	C3 06F9 C3 0704 C3 1510 BIOS) Macro DS calls (Pro C3 04BD C3 1102 C3 1102	-80 grammable S grammable S ; Follor ; read ; ;	JP JP JP ; =BIOS 1 3.44 ound Ge wing er and wr JP JP	CALATR GSPSIZ GRPPRT header- BIOS cal 01-Jan-85 enerator cont htry points are ite PSG register GICINI WRTPSG RDPSG	;Get address of sprite pattern table ;""" attribute table ;Returns current sprite size ;Print a character on the graphic screen lls (Programmable Sound Generator control) PAGE 4 used for PSG initialization, rs, and PLAY statement execution. ;Init PSG, and static data for PLAY ;Write data to PSG ;Read data from PSG
134 135 136 137 138 139 140 MSX ROM BIOS hea 141 142 143 144 145 146 147	0087 008A 008D 4 BASIC 1 1der- BIC 0090 0093	C3 06F9 C3 0704 C3 1510 BIOS) Macro DS calls (Pro C3 04BD C3 1102	-80 grammable Su ; Follor ; read ;	JP JP JP ; -BIOS 1 3.44 ound Ge wing er and wr JP	CALATR GSPSIZ GRPPRT header- BIOS cal 01-Jan-85 enerator cont ntry points are ite PSG register GICINI WRTPSG	;Get address of sprite pattern table ;""" attribute table ;Returns current sprite size ;Print a character on the graphic screen lls (Programmable Sound Generator control) PAGE 4 used for PSG initialization, rs, and PLAY statement execution. ;Init PSG, and static data for PLAY ;Write data to PSG

(MSX ROM BASIC BIOS) Macro-80	3.44	01-Jan-85	PAGE	5
-BIOS header- BIOS calls (Keyboar	d, CRT, and	Printer)		

			;			
			; Gene	eral IN	NPUT and PRINT ut	ilities.
			;			
009C	C3	0D6A		JP	CHSNS	;Checks status of keyboard status
009F	C3	10CB		JP	CHGET	;Return char typed, with wait
00A2	C3	08BC		JP	CHPUT	;Output character to console
00A5	C3	085D		JP	LPTOUT	; " to printer, if possible
8A00	C3	0884		JP	LPTSTT	;Checks status of line printer
00AB	C3	089D		JP	CNVCHR	;Checks for graphic header byte
						;and convert code
00AE	C3	23BF		JP	PINLIN	;Read line from keyboard to buffer
00B1	C3	23D5		JP	INLIN	;Same as above, except in case of
						;AUTFLG is set
00B4	C3	23CC		JP	QINLIN	;Print a "?", then jump to INLIN
0087	C3	046F		JP	BREAKX	;[Control-STOP] pressed??
00 BA	C3	03FB		JP	I SCNTC	;[Shift-STOP] pressed??
00BD	C3	10F9		JP	CKCNTC	;Same as ISCNTC, but used by BASIC
00C0	C3	1113		JP	BEEP	; Buzz
00C3	C3	0848		JP	CLS	;Clear screen
00C6	C3	088E		JP	POSIT	;Place cursor at Column [H], Row [L]
00C9	C3	0B26		JP	FNKSB	Display Function key, if neccessary
00CC	C3	0815		JP	ERAFNK	Stop displaying the Function keys
00CF	C3	0B2B		JP	DSPFNK	Enable Function key display
00D2				JP	TOTEXT	;Force screen to text mode
				;		
			SUBTTL	-BIOS	header- BIOS cal	lls (Game and Cassette I/O, Oueue handler)
	009F 00A2 00A8 00A8 00A8 00B1 00B4 00B7 00BA 00B0 00C0 00C3 00C6 00C9 00CCF	009P C3 00A2 C3 00A5 C3 00A6 C3 00A8 C3 00A8 C3 00A8 C3 00A8 C3 00B1 C3 00B4 C3 00B7 C3 00B8 C3 00B7 C3 00B0 C3 00C0 C3 00C2 C3 00C4 C3	009F C3 10CB 00A2 C3 08BC 00A5 C3 085D 00A8 C3 0884 00AB C3 089D 00AE C3 23BF 00B1 C3 23D5 00B4 C3 23CC 00B7 C3 046F 00B8 C3 10F9 00C0 C3 1113 00C3 C3 0848 00C6 C3 0826 00C2 C3 0826 00C2 C3 0815 00C4 C3 0828	; Gene ; 009C C3 0D6A 009F C3 10CB 00A2 C3 08BC 00A5 C3 08BC 00A8 C3 0884 00AB C3 089D 00AE C3 23BF 00B1 C3 23D5 00B4 C3 23D5 00B4 C3 03FB 00BA C3 03FB 00BA C3 03FB 00BA C3 03FB 00BD C3 10F9 00CC C3 0848 00C2 C3 0848 00C2 C3 0826 00CC C3 0825 00CC C3 0826 00CC C3 0825 00CC C3 0826	; General II ; 009C C3 0D6A JP 009F C3 10CB JP 00A2 C3 08BC JP 00A8 C3 08BC JP 00A8 C3 0884 JP 00A8 C3 089D JP 00A8 C3 089D JP 00A8 C3 23D5 JP 00B1 C3 23D5 JP 00B4 C3 23CC JP 00B4 C3 03FB JP 00B5 C3 10F9 JP 00B6 C3 113 JP 00C6 C3 0848 JP 00C6 C3 0848 JP 00C6 C3 0826 JP 00C7 C3 0826 JP 00C7 C3 0828 JP 00C7 C3 0828 JP	; General INPUT and PRINT ut ; 009C C3 0D6A JP CHSNS 009F C3 10CB JP CHGET 00A2 C3 08BC JP CHPUT 00A8 C3 08BC JP LPTOUT 00A8 C3 0884 JP LPTSTT 00A8 C3 089D JP CNVCHR 00A8 C3 23D5 JP INLIN 00B1 C3 23D5 JP INLIN 00B1 C3 23D5 JP INLIN 00B4 C3 03FB JP SERAX 00BA C3 01F9 JP CKCNTC 00BD C3 10F9 JP CKCNTC 00BD C3 10F9 JP CKCNTC 00C0 C3 1113 JP BEEP 00C3 C3 084E JP POSIT 00C6 C3 084E JP POSIT 00C7 C3 0826 JP FNKSB 00CCF C3 082B JP DSPFNK 00D2 C3 083B JP DSPFNK 00D2 C3 083B JP CKCNTC

8

9

(MSX ROM BASIC BIOS) Macro-80 3.44 01-Jan-85 -BIOS header- BIOS calls (Game and Cassette I/O, Queue hand PAGE 6

179

180				;				
181				;	Foll	owing a	are used to read	the value from Joysticks,
182				;	Grap	hic pad	d (tablet), and	Paddles.
183				;				
184	00D5	C3 1	lee			JP	GTSTCK	;Return status of joystick
185	00D8	C3 1	253			JP	GTTRIG	;Read joystick trigger button
186	00DB	C3 1	2AC			JP	GTPAD	;Returns status of graphic pad
187	00DE	C3 1	273			JP	GTPDL	;Read paddle
188						;		
189				;				
190				;	Foll	owing a	are used to acce	ss the cassette tape,
191				;	data	read/w	write, and motor	on/off
192				;				
193	00E1	C3 1	A63			JP	TAPION	;Turn on motor and read tape header
194	00E4	C3 1	ABC			JP	TAPIN	;Read tape data
195	00E7	C3 1	9E9			JP	TAPIOF	;Stops reading from tape
196	00EA	C3 1	9F1			JP	TAPOON	;Turn on motor and write tape header
197	00ED	C3 1	A19			JP	TAPOUT	;Write data to tape
198	00F0	C3 1	9DD			JP	TAPOFF	Stops writing to tape
199	00F3	C3 1	384			JP	STMOTR	;Start, stop cassette motor, or
200								flip motor (on to off, off to on)
201						;		· · · · · ·
202				;				
203				;	BASI	C queu	es	
204				;				•
205	00F6	C3 1	L4EB			JP	LFTQ	;Bytes left in queue
206	00F9	C3 1	L492			JP	PUTQ	;Send a byte to queue
207						;		
208				S	UBTTL	-BIOS	header- BIOS cal	ls (Generalized graphics)

-BIOS he		BIOS) Macro OS calls (Gen		.44 01-Jan-8	5 PAGE 7
	Judei Di	55 carrs (den	crarizea gru	pricov	
209 210					
211			; ; For BA	SIC interpreter	's GENGRP and ADVGRP modules use
212	00FC	C3 16C5	J	-	;Moves one pixel right
213	00FF	C3 16EE	J		; " " " left
214	0102	C3 175D	J	P UPC	; " " up
215	0105	C3 173C	J	P TUPC	; " " " "
216	0108	C3 172A	J		; " " down
217	010B	C3 170A		P TDOWNC	, " " " "
218	010E	C3 1599		P SCALXY	;Scales X Y cordinates
219	0111	C3 15DF		P MAPXYC	;Maps cordinates to physical address ;Get current physical address and
220 221	0114	C3 1639	J	P FETCHC	; mask pattern
222	0117	C3 1640	л	P STOREC	;Put current physical address and
223	0117				;mask pattern
224	011A	C3 1676	J	P SETATR	;Sets the color attribute byte
225	011D	C3 1647	J	P READC	;Reads attribute of current pixel
226	0120	C3 167E	J	P SETC	;Sets current pixel to specified attribution
227	0123	C3 1809		P NSETCX	;Sets pixel horizontally
228	0126	C3 18C7		P GTASPC	Returns aspect ratio
229	0129	C3 18CF		P PNTINI	;Do paint initialization
230	012C	C3 18E4		P SCANR	;Scan pixels to the right
231	012F	C3 197A		P SCANL	; " " " left
232 233			; SUBTTL -B		S calls (Misc. Entries)
235			SOBIL D	neo neo dei Di	o carro (moc. marro)
		BIOS) Macro		.44 01-Jan-8	PAGE 8
-BIOS h	eader- Bl	IOS calls (Mis	c. Entries)		
234					
235			;		
236			;		
237	0132	C3 0F3D	Ji	P CHGCAP	;Turn [CAPSLOCK] light, on/off
238	0135	C3 OF7A	. JI	P CHGSND	;Change status of 1 bit sound port
239	0138	C3 144C	J		Return output of primary slot register;
240	013B	C3 144F	J		;Write to primary slot register
241 242	013E 0141	C3 1449 C3 1452	J1 J1		;Read VDP status register
243	0141	CJ 14J2	0.	P SNSMAT	;Read a specified row in the ;keyboard matrix
244	0144	C3 148A	· J.	P PHYDIO	;Performs operation for mass storage
245					devices (such as disks)
246	0147	C3 148E	J	P FORMAT	;Initialize mass storage device
247	014A	C3 145F	J		;Are we doing device I/O
248	014D	C3 1B63	3		;Output to line printer
249	0150	C3 1470	J:		;Used by Music background tasking
250 251	0153	C3 1474 C3 0468	J. J.		, " " " " " " "
252	0159	C3 01FF	J		;Clear the keyboard buffer ;Performs far-call into BASIC
253	015C		D		RESERVED FOR EXPANSION
254			;		•••••••••••••••••••••••••••••••••••••••
255			SUBTTL -	SLOT - Slot ha	dler stuff
(MSX RC	M BASIC H	BIOS) Macro-	-80 3.	44 01-Jan-85	PAGE 9
- SLOT -	 Slot has 	andler stuff			
256					
256 257	00A8		007 80 00	11 0305	New road from DET Deet D
257	00A8 00A8		PPI.AR EQ PPI.AW EQ		A8H read from PPI Port A A8H Write to PPI Port A
258	~~~~		PPI.AW EQ	o unon	ABH Write to PPI Port A
260				rtridge located	at 0000-3FFFH must contain codes in
261		•			ntered via following addresses.
262			;		
263				RDSLT	
264				WRSLT	
265				CALSLT	
266			; 0024H	I ENASLT	,
267 268			1		
			;		RDSLT
269			;		NDOL1
269 270			•	the appropriate	slot according to the value given
269 270 271					read the content of memory from the
270			; slot.	,,	
270 271			;		
270 271 272			; Input pa	rameters:	
270 271 272 273 274 275			; A - Fx)	XSSPP	
270 271 272 273 274 275 276			,		
270 271 272 273 274 275 276 277			; [
270 271 272 273 274 275 276 277 278			;	++ primary	
270 271 272 273 274 275 276 277 278 279			; ;	++ primary ++ seconda	ry slot # (0-3)
270 271 272 273 274 275 276 277 278 279 280			; ;	++ primary ++ seconda	
270 271 272 273 274 275 276 277 278 279			; ;	++ primary ++ seconda	ry slot # (0-3) condary slot # specified
270 271 272 273 274 275 276 277 278 279 280 281			; ; ; + ;	++ primary ++ seconda l if se HL - ac	ry slot # (0-3)
270 271 272 273 274 275 276 277 278 279 280 281 282			; ;	++ primary ++ seconda l if so HL - ac d value	ry slot # (0-3) condary slot # specified

	· Slot h	BIOS) Macro andler stuff	-80	3.44	01-Jan-85	PAGE 9-1	13
287			;	by th	is routine.		
288			;				
289	01B6		RDSLT:				
290	01.B6	CD 027E		CALL	SELPRM	;Calculate bit pattern and	mark code
291	0189	FA 01C6		JP	M, RDESLT		mask code
292	01BC	DB A8				;Expanded slot specified	
				IN	A,(PPI.AR)		
293	01 BE	57		LD	D,A	;Save current setting	
294	OlBF	Al.		AND	С	;Cancel current setting for	r target address
295	01C0	B0		OR	B.	;Add new setting	
296	01C1	CD F380		CALL	RAMLOW	;Call read primitive routing	ne (in system area)
297	01C4	7B		LD	A,E	;Return value via [Acc]	-
298	01C5	C9	•	RET			
299	01C6		RDESLT:				
300	01C6	E5		PUSH	HL	Save target address	
301	01C7	CD 02A3		CALL	SELEXP	;Select secondary slot	
302	01CA	E3		EX	(SP),HL	;Restore target address an	d save [HL]
303	01CB	C5		PUSH	BC		
304	01CC	CD 0186		CALL	RDSLT		
305	01CF	18 1B		JR	WRESED	;Restore old slot select r	egister
306			SUBTTL	-SLOT-			- ,
(MSX F	OM BASIC	BIOS) Macro	0-80	3.44	01-Jan-85	PAGE 10	14
-SLOT-	Slot ha	ndler (Write :	slot)				
307					·		
308 309			;			WRSLT	
310						HINDLI	
311			, Sele	cts the	appropriate elo	t according to the value given	
312						e to the memory in the specifi	
313			; slot		scers, and write	e co che memory in che specifi	leu
314			, 5100	•			
315							
				t parame			
316 317				FxxxSSP			
318			;			t II (0.2)	
319			,		+ primary slo		
320			1		secondary s		
321			<i>.</i>	+	i li seconu	ary slot # specified	
322			;	нт	- address of t	arget memory	
323			<u>.</u>		uuuress or c	arget memory	
324				F	- value to be	written	
325			<u>'</u>	5	varue co be	#TICCEN	
326			, ; Not	e: Inter	runts are disab	led automatically but never er	
327			;		is routine.	red adcompercarry but never en	labied
328			;	-1	io routine.		
329	0101		WRSLT:				
330	01D1	D5		PUSH	DE	;Save data to be written	
331	01D2	CD 027E		CALL	SELPRM	;Calculate bit pattern and	mask code
332	01D5	FA OlEL		JP	M, WRESLT	;Expanded slot specified	mask code
333	01D8	Dl		POP	DE	Restore data to be writter	
334	01D9	DB A8		IN	A,(PPI.AR)	,	•
335	01DB	57		LD	D,A	;Save current setting	
336	01DC	Al		AND	c	;Cancel current setting for	target address
337	01DD	BO		OR	в	;Add new setting	
						-	
		NOS) Macro- dler (Write s		3.44	01-Jan-85	PAGE 10-1	15
			,				- ···
		C3 F385	WRESLT:	JP	WRPRIM	;Call write primitive rout	ine (in system area)
338	01DE 01E1						
338 339	01E1	E3		EX	(SP).HT	Save target address out	data to be written
338 339 340	01E1 01E1	E3		EX	(SP),HL HT.	;Save target address, get .	data to be written
338 339 340 341	01E1 01E1 01E2	E5		PUSH	HL	;Save data to be written	data to be written
338 339 340 341 342	01E1 01E1 01E2 01E3	E5 CD 02A3		PUSH CALL	HL SELEXP	;Save data to be written ;Select secondary slot	
338 339 340 341 342 343	01E1 01E1 01E2 01E3 01E6	E5 CD 02A3 D1		PUSH CALL POP	HL SELEXP DE	;Save data to be written ;Select secondary slot ;Restore data to be writte	n
338 339 340 341 342 343 343	01E1 01E1 01E2 01E3 01E6 01E7	E5 CD 02A3 D1 E3		PUSH CALL POP EX	HL SELEXP DE (SP),HL	;Save data to be written ;Select secondary slot	n
338 339 340 341 342 343 344 345	01E1 01E2 01E3 01E6 01E7 01E8	E5 CD 02A3 D1 E3 C5		PUSH CALL POP EX PUSH	HL SELEXP DE (SP),HL BC	;Save data to be written ;Select secondary slot ;Restore data to be writte	n
338 339 340 341 342 343 344 345 346	01E1 01E2 01E3 01E6 01E7 01E8 01E9	E5 CD 02A3 D1 E3		PUSH CALL POP EX	HL SELEXP DE (SP),HL	;Save data to be written ;Select secondary slot ;Restore data to be writte	n
338 339 340 341 342 343 343 344 345 346 347	01E1 01E1 01E2 01E3 01E6 01E7 01E8 01E9 01EC	E5 CD 02A3 D1 E3 C5 CD 01D1	WRESED:	PUSH CALL POP EX PUSH CALL	HL SELEXP DE (SP),HL BC WRSLT	;Save data to be written ;Select secondary slot ;Restore data to be writte	n
338 339 340 341 342 343 344 345 346 347 348	01E1 01E2 01E3 01E6 01E7 01E8 01E9 01EC 01EC	E5 CD 02A3 D1 E3 C5 CD 01D1 C1	WRESED:	PUSH CALL POP EX PUSH CALL POP	HL SELEXP DE (SP),HL BC WRSLT BC	;Save data to be written ;Select secondary slot ;Restore data to be writte ;Restore target address an	n d save (HL)
338 339 340 341 342 343 344 345 346 347 348 348 349	01E1 01E2 01E3 01E6 01E7 01E8 01E9 01EC 01EC 01EC	E5 CD 02A3 D1 E3 C5 CD 01D1 C1 E3	WRESED:	PUSH CALL POP EX PUSH CALL POP EX	HL SELEXP DE (SP),HL BC WRSLT BC (SP),HL	;Save data to be written ;Select secondary slot ;Restore data to be writte ;Restore target address an ;Save target address and g	n d save [HL] et old [HL]
338 339 340 341 342 343 344 345 346 347 348	01E1 01E2 01E3 01E6 01E7 01E8 01E9 01EC 01EC	E5 CD 02A3 D1 E3 C5 CD 01D1 C1	WRESED:	PUSH CALL POP EX PUSH CALL POP	HL SELEXP DE (SP),HL BC WRSLT BC	;Save data to be written ;Select secondary slot ;Restore data to be writte ;Restore target address an ;Save target address and g ;Save value returned by RD	n d save [HL] et old [HL]
338 339 340 341 342 343 344 345 346 347 348 348 349	01E1 01E2 01E3 01E6 01E7 01E8 01E9 01EC 01EC 01EC	E5 CD 02A3 D1 E3 C5 CD 01D1 C1 E3	WRESED:	PUSH CALL POP EX PUSH CALL POP EX	HL SELEXP DE (SP),HL BC WRSLT BC (SP),HL	;Save data to be written ;Select secondary slot ;Restore data to be writte ;Restore target address an ;Save target address and g	n d save [HL] et old [HL]
338 339 340 341 342 343 344 345 346 347 348 349 350	01E1 01E2 01E3 01E6 01E7 01E8 01E9 01EC 01EC 01EC 01ED 01EE	E5 CD 02A3 D1 E3 C5 CD 01D1 C1 E3 F5	WRESED:	PUSH CALL POP EX PUSH CALL POP EX PUSH	HL SELEXP DE (SP),HL BC WRSLT BC (SP),HL AF	;Save data to be written ;Select secondary slot ;Restore data to be writte ;Restore target address an ;Save target address and g ;Save value returned by RD	n d save [HL] met old [HL] SLT
338 339 340 341 342 343 344 345 346 347 348 349 350 351	01E1 01E2 01E3 01E6 01E7 01E8 01E9 01EC 01EC 01EC 01EE 01EF	E5 CD 02A3 D1 E3 C5 CD 01D1 C1 E3 F5 78	WRESED:	PUSH CALL POP EX PUSH CALL POP EX PUSH LD	HL SELEXP DE (SP),HL BC WRSLT BC (SP),HL AF A,B	;Save data to be written ;Select secondary slot ;Restore data to be writte ;Restore target address an ;Save target address and g ;Save value returned by RD ;Get current setting	n d save [HL] met old [HL] SLT
338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353	01E1 01E1 01E2 01E3 01E6 01E7 01E8 01E9 01EC 01EC 01EC 01ED 01EE 01F0 01F2	E5 CD 02A3 D1 E3 C5 CD 01D1 C1 E3 F5 78 E6 3F B1	WRESED:	PUSH CALL POP EX PUSH CALL POP EX PUSH LD AND OR	HL SELEXP DE (SP),HL BC WRSLT BC (SP),HL AF A,B 00111111B C	<pre>;Save data to be written ;Select secondary slot ;Restore data to be writte ;Restore target address an ;Save target address and g ;Save value returned by RD ;Get current setting ;Cancel current setting fo</pre>	n d save [HL] et old [HL] SLT r OCOOOHOFFFFH
338 339 340 341 342 343 344 345 346 346 347 348 349 350 351 352 353 354	01E1 01E2 01E3 01E6 01E7 01E8 01E9 01EC 01EC 01EC 01EE 01EF 01F0 01F2 01F3	E5 CD 02A3 D1 E3 C5 CD 01D1 C1 E3 F5 78 E6 3F B1 D3 A8	WRESED:	PUSH CALL POP EX PUSH CALL POP EX PUSH LD AND OR OUT	HL SELEXP DE (SP),HL BC WRSLT BC (SP),HL AF A,B 00111111B C (PPI.AW),A	<pre>;Save data to be written ;Select secondary slot ;Restore data to be writte ;Restore target address an ;Save target address and g ;Save value returned by RD ;Get current setting ;Cancel current setting fo ;Enable 0C000H0FFFFH of</pre>	n d save [HL] et old [HL] SLT r OC000HOFFFFH target bank
338 339 340 341 342 343 344 345 346 347 348 347 348 347 350 351 351 352 353 354 355	01E1 01E2 01E3 01E6 01E7 01E8 01E9 01EC 01EC 01EC 01EC 01ED 01EP 01F9 01F9 01F3 01F5	E5 CD 02A3 D1 E3 C5 CD 01D1 C1 E3 F5 78 E6 3F B1 D3 A8 7D	WRESED:	PUSH CALL POP EX PUSH CALL POP EX PUSH LD OR OUT LD	HL SELEXP DE (SP),HL BC WRSLT BC (SP),HL AF A,B 0011111B C (PPI.AW),A A,L	<pre>;Save data to be written ;Select secondary slot ;Restore data to be writte ;Restore target address an ;Save target address and g ;Save value returned by RD ;Get current setting ;Cancel current setting fo</pre>	n d save [HL] et old [HL] SLT r OC000HOFFFFH target bank
338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 355	01E1 01E2 01E3 01E6 01E7 01E6 01E8 01E9 01EC 01ED 01EC 01ED 01EE 01F0 01F2 01F5 01F5	E5 CD 02A3 D1 E3 C5 CD 01D1 C1 E3 F5 78 E6 3F B1 D3 A8 7D 32 FFFF	WRESED:	PUSH CALL POP EX PUSH CALL POP EX PUSH LD AND OR OUT LD LD	HL SELEXP DE (SP),HL BC WRSLT BC (SP),HL AF A,B 00111111B C (PPI.AW),A A,L (OFFFFH),A	;Save data to be written ;Select secondary slot ;Restore data to be writte ;Restore target address and g;Save target address and g ;Save value returned by RD ;Get current setting ;Cancel current setting fo ;Enable 0C000H0FFFFH of ;Restore old setting of sl	n d save [HL] et old [HL] SLT r OCOOOHOFFFFH target bank ot register
338 339 340 341 342 343 344 345 346 346 347 348 349 350 351 352 353 354 355 356 356 357	01E1 01E2 01E3 01E6 01E7 01E6 01E9 01EC 01EC 01EC 01EC 01ED 01EE 01F9 01F3 01F5 01F6 01F9	E5 CD 02A3 D1 E3 C5 CD 01D1 C1 E3 F5 78 E6 3F B1 D3 A8 7D 32 FFFF 78	WRESED:	PUSH CALL POP EX PUSH CALL POP EX PUSH LD LD LD LD	HL SELEXP DE (SP),HL BC WRSLT BC (SP),HL AF A,B 0011111B C (PPI.AW),A A,L (OFFFFH),A A,B	<pre>;Save data to be written ;Select secondary slot ;Restore data to be writte ;Restore target address an ;Save target address and g ;Save value returned by RD ;Get current setting ;Cancel current setting fo ;Enable 0C000H0FFFFH of</pre>	n d save [HL] et old [HL] SLT r OCOOOHOFFFFH target bank ot register
338 339 340 341 342 343 344 345 346 347 348 347 350 351 352 353 354 355 356 355 355 356	01E1 01E2 01E3 01E6 01E7 01E8 01E9 01EC 01EC 01EC 01ED 01ED 01F0 01F0 01F3 01F5 01F5 01F9 01F4	E5 CD 02A3 D1 E3 C5 CD 01D1 C1 E3 F5 78 E6 3F B1 D3 A8 7D 32 FFFF 78 D3 A8	WRESED:	PUSH CALL POP EX PUSH CALL POP EX PUSH LD AND OR OUT LD LD LD UUT	HL SELEXP DE (SP),HL BC WRSLT BC (SP),HL AF A,B 001,1111B C (PPI.AW),A A,L (OFFFFH),A A,B (PFFFH),A	<pre>;Save data to be written ;Select secondary slot ;Restore data to be writte ;Restore target address and ;Save target address and g ;Save value returned by RD ;Get current setting ;Cancel current setting fo ;Enable 0C000H0FFFFH of ;Restore old setting of sl ;Finally restore old prima</pre>	n d save [HL] Het old [HL] ISLT or OC000HOFFFFH target bank ot register ry slot register
338 339 340 341 342 343 344 345 346 346 347 348 349 350 351 352 353 354 355 356 356 357	01E1 01E2 01E3 01E6 01E7 01E6 01E9 01EC 01EC 01EC 01EC 01ED 01EE 01F9 01F3 01F5 01F6 01F9	E5 CD 02A3 D1 E3 C5 CD 01D1 C1 E3 F5 78 E6 3F B1 D3 A8 7D 32 FFFF 78	WRESED:	PUSH CALL POP EX PUSH CALL POP EX PUSH LD LD LD LD	HL SELEXP DE (SP),HL BC WRSLT BC (SP),HL AF A,B 0011111B C (PPI.AW),A A,L (OFFFFH),A A,B	<pre>;Save data to be written ;Select secondary slot ;Restore data to be writte ;Restore data to be writte ;Restore target address and ;Save target address and g ;Save value returned by RD ;Get current setting ;Cancel current setting fo ;Enable 0C000H0FFFFH of ;Restore old setting of sl ;Finally restore old prima ;Restore value returned by</pre>	n d save [HL] Het old [HL] ISLT or OC000HOFFFFH target bank ot register ry slot register
338 339 340 341 342 343 344 345 346 346 347 348 350 351 352 353 354 355 356 357 358 359 360	01E1 01E2 01E3 01E6 01E7 01E8 01E9 01EC 01ED 01EC 01ED 01EC 01F0 01F2 01F3 01F5 01F6 01F9 01F6 01FC 01FD	E5 CD 02A3 D1 E3 C5 CD 01D1 C1 E3 F5 78 E6 3F B1 D3 A8 7D 32 FFFF 78 D3 A8 F1 D3 A8 F1	WRESED:	PUSH CALL POP EX PUSH CALL POP EX PUSH LD CAND OR OUT LD LD LD LD LD COUT POP POP	HL SELEXP DE (SP),HL BC (SP),HL AF A,B 0011111B C (PPI.AW),A A,L (0FFFFH),A A,B (PFFFH),A AF	<pre>;Save data to be written ;Select secondary slot ;Restore data to be writte ;Restore target address and ;Save target address and g ;Save value returned by RD ;Get current setting ;Cancel current setting fo ;Enable 0C000H0FFFFH of ;Restore old setting of sl ;Finally restore old prima</pre>	n d save [HL] et old [HL] SLT r OCO00HOFFFFH target bank ot register ry slot register
338 339 340 341 342 343 344 345 346 347 348 349 350 351 352 353 354 355 355 356 357 358 358	01E1 01E2 01E3 01E6 01E7 01E7 01E8 01E9 01E0 01E0 01E0 01E0 01F0 01F2 01F5 01F6 01F9 01F6 01F9	E5 CD 02A3 D1 E3 C5 CD 01D1 C1 E3 F5 78 E6 3F B1 D3 A8 7D 32 FFFF 78 D3 A8 F1	WRESED:	PUSH CALL POP EX PUSH CALL POP EX PUSH LD LD LD LD LD LD LD LD LD LD LD	HL SELEXP DE (SP),HL BC (SP),HL AF A,B 0011111B C (PPI.AW),A A,L (0FFFFH),A A,B (PFFFH),A AF	<pre>;Save data to be written ;Select secondary slot ;Restore data to be writte ;Restore data to be writte ;Restore target address and ;Save target address and g ;Save value returned by RD ;Get current setting ;Cancel current setting fo ;Enable 0C000H0FFFFH of ;Restore old setting of sl ;Finally restore old prima ;Restore value returned by</pre>	n d save [HL] et old [HL] SLT or OCOOOHOFFFFH target bank ot register ry slot register

•) Macro-8 (Write slo		3.44	01-Jan-85	PAGE	11
362								
363	01FF			CALBAS:				
364	01FF	FD	2A FCC0		LD	IY, (EXPTBL-1)		
365	0203	18	12		JR	CALSLT		
366	0205			CALLF:				
367	0205	E3			EX	(SP),HL	;Get re	turn address, save [HL]
368	0206	F5			PUSH	AF	;Save w	orking registers
369	0207	D5			PUSH	DE		
370	0208	7E			LD	A,(HL)	;Get de	stination slot
371	0209	F5			PUSH	AF		
372	020A	FD	El		POP	IY	;Move i	t to IYH
373	020C	23			INC	HL		
374	020D	5E			LD	E,(HL)	;Get de	stination address
375	020E	23			INC	HL		
376	020F	56			LD	D,(HL)		
377	0210	23			INC	HL	;Prepar	e true return address
378	0211	D5			PUSH	DE		
379	0212	DD	El		POP	IX	;Move i	t to IX
380	0214	D1			POP	DE	; Restor	e working registers
381	0215	Fl			POP	AF		
382	0216	E3			EX	(SP),HL	; Restur	e [HL], save true return address
383				SUBTTL	-SLOT-			

(MSX ROM BASIC BIOS) Macro-80 3.44 01-Jan-85 PAGE 12 -SLOT-

384										
385			;							
386			;		CALSLT					
387			;	• • • • •						
388			; Performs int	Performs inter-slot call to specified address,						
389			;							
390			; Input parame	Input parameters:						
391			; IY - FxxxSS	IY - FxxxSSPP						
392			; []	11						
393			;]]]	++ primary sl	ot # (0-3)					
394			; ; ++							
395			; +	+ 1 if secondary slot # specified						
396			;							
397			; IX - addres	; IX - address to call						
398			;	;						
399			; Note: Inter	; Note: Interrupts are disabled automatically but never enabled						
400			; by th	; by this routine.						
401			; You c	an never pass a	rguments via alternate registers					
402			; of 28							
403			;							
404	0217		CALSLT:							
405	0217	D9	EXX		;Save environments					
406	0218	08	EX	AF,AF'						
407	0219	FD E5	PUSH	IY						
408	021B	Fl	POP	AF	;Get target slot information					
409	021C	DD E5	PUSH	IX						
410	021E	El	POP	HL	;Get target address					
411	021F	CD 027E	CALL	SELPRM						
412	0222	FA 022E	JP	M, CALESL	;Call expanded slot					
413	C225	DB A8	IN	A, (PPI.AR)						
414	0227	F5	PUSH	AF	;Save current value of primary slot register					

16

17

-SLOT-	M BASIC	BIOS) Macro-8	0	3.44	01-Jan-85	PAGE 12-1	18
415	0228	Al			AND	с	;Cancel current setting for target addres	~
416	0229	BO			OR	в	;Add new setting	з
417	022A	D9			EXX	2	Restore environments except PSW	
418	022B	C3	F38C		JP	CLPRIM	;Jump to primitive routine (in system are	a)
419	022E			CALESL:				
420	022E	CD	02A3		CALL	SELEXP	;Select secondary slot register	
421	0231	F5			PUSH	AF	;Move primary slot # in [IYH]	
422	0232	FD	El		POP	IY		
423	0234	E5			PUSH	HL	;Save [B,C,L] which contain information	
424	0235	C5			PUSH	BC	; for restoring slot environments	
425	0236	4 F			LD	C,A	;Move primary slot # to [BC]	
426	0237		00		LD	в,0		
427	0239	7D			LD	A,L	Re-calculate what is currently output;	
428	023A	A4			AND	н	;to expansion slot register	
429 430	023B 023C	B2	FCC5		OR	D		
431	023C	09	rccs		LD ADD	HL, SLTTBL HL, BC	;Calculate address into SLTTBL	
432	0240	77			LD	(HL),A	;Set current value output to expansion	
433					20	(slot register	
434	0241	E5			PUSH	HL	;Remember this address	
435	0242	08			EX	AF, AF'	Restore possible arguments passed	
436	0243	D9			EXX		;via registers	
437	0244	CD	0217		CALL	CALSLT	;Call by primary slot #	
438	0247	D9			EXX		Save possible values returned via	
439	0248	08			EX	AF, AF'	;registers	
440	0249	El			POP	HL.	Restore address into SLTTBL	
441	024A	C1			POP	BC	Restore information about old slots;	
442 443	024B 024C	D1 78			POP	DE		
444	024C		3F		LD	A,B	;Get current setting	
445	024D	B1	31		AND OR	00111111B C	;Cancel current setting for 0C000H0FFF	FH
(MSX ROM -SLOT-	4 BASIC	BIOS) Macro-80	ļ -	3.44	01-Jan-85	PAGE 12-2	19
446	0250	F3			DI			
447	0251	D3 .	88		OUT		Della Annone anno 1	
448	0253	7B			LD	(PPI.AW),A A,E	;Enable 0C000HOFFFFH of target bank	
449	0254		FFFF		LD	(OFFFFH),A	Restore old setting of slot register	
450	0257	78			LD	A,B	Finally restore old primary alat we stat	
	0258	D3	88		OUT	(PPI.AW),A	;Finally restore old primary slot registe	r
451								
451 452	025A	73					And change CIMMPI -1	
					LD	(HL),E	And change SLTTBL also	
452	025A	73					;And change SLTTBL also ;Restore possible returned values	
452 453	025A 025B	73 08			LD EX	(HL),E		
452 453 454 455	025A 025B 025C 025D	73 08 D9 C9) Macro-8	0	LD EX EXX	(HL),E		20
452 453 454 455 (MSX RC -SLOT- 456 457	025A 025B 025C 025D	73 08 D9 C9		;	LD EX EXX RET 3.44	(HL),E AF,AF' 01-Jan-85	;Restore possible returned values PAGE 13	20
452 453 454 455 (MSX RC - SLOT- 456 457 458	025A 025B 025C 025D	73 08 D9 C9		; ;	LD EX EXX RET 3.44	(HL),E AF,AF' 01-Jan-85	Restore possible returned values;	20
452 453 454 455 (MSX RC -SLOT- 456 457 458 459	025A 025B 025C 025D	73 08 D9 C9		;	LD EX EXX RET 3.44	(HL),E AF,AF' 01-Jan-85	;Restore possible returned values PAGE 13 ENASLT	20
452 453 454 455 (MSX RC -SLOT- 456 457 458 459 460	025A 025B 025C 025D	73 08 D9 C9		; ; ; ; Selec	LD EX EXX RET 3.44	(HL),E AF,AF' 01-Jan-85 	<pre>;Restore possible returned values PAGE 13 ENASLT : according to the value given</pre>	20
452 453 454 455 (MSX RC -SLOT- 456 457 458 459 460 461	025A 025B 025C 025D	73 08 D9 C9		; ; ; Selec ; throu	LD EX EXX RET 3.44	(HL),E AF,AF' 01-Jan-85 	;Restore possible returned values PAGE 13 ENASLT	20
452 453 454 455 (MSX RC -SLOT- 456 457 458 459 460 461 462	025A 025B 025C 025D	73 08 D9 C9		; ; ; Selec ; throw ;	LD EX EXX RET 3.44	(HL),E AF,AF' 01-Jan-85 appropriate slot sters, and perma	<pre>;Restore possible returned values PAGE 13 ENASLT : according to the value given</pre>	20
452 453 454 455 (MSX RC -SLOT- 456 457 458 459 460 461 462 463	025A 025B 025C 025D	73 08 D9 C9		; ; ; Selec ; throu ; ; Input	LD EX EXX RET 3.44	(HL),E AF,AF' 01-Jan-85 appropriate slot sters, and perma	<pre>;Restore possible returned values PAGE 13 ENASLT : according to the value given</pre>	20
452 453 454 455 (MSX RC - SLOT- 456 457 458 459 460 461 462	025A 025B 025C 025D	73 08 D9 C9		; ; ; Selec ; throu ; ; Input ;	LD EX EXX RET 3.44 ets the gh reginst parameters	(HL),E AF,AF' 01-Jan-85 appropriate slot sters, and perma	<pre>;Restore possible returned values PAGE 13 ENASLT : according to the value given</pre>	20
452 453 454 455 (MSX RC -SLOT- 456 457 458 459 460 461 462 463 464	025A 025B 025C 025D	73 08 D9 C9		; ; ; Selec; ; throu ; ; Input ; ; A -	LD EX EXX RET 3.44 ets the igh regi parame FxxxSSE	(HL),E AF,AF' 01-Jan-85 appropriate slot sters, and perma sters: op	<pre>;Restore possible returned values PAGE 13 ENASLT : according to the value given</pre>	20
452 453 454 455 (MSX RC -SLOT- 456 457 458 459 460 461 462 463 464 465	025A 025B 025C 025D	73 08 D9 C9		; ; ; Selec ; throu ; ; Input ; ; A - ;	LD EX EXX RET 3.44 :ts the igh regin parame FxxxSSE 	(HL),E AF,AF' 01-Jan-85 appropriate slot sters, and perma sters:	Restore possible returned values PAGE 13 ENASLT c according to the value given unently enables the slot.	20
452 453 454 455 (MSX RC -SLOT- 456 457 458 459 460 461 462 463 464 465	025A 025B 025C 025D	73 08 D9 C9		; ; ; Selec; ; throu ; ; Input ; ; A -	LD EX EXX RET 3.44 	(HL),E AF,AF' 01-Jan-85 appropriate slot sters, and perma eters: 	<pre>;Restore possible returned values PAGE 13 ENASLT</pre>	20
452 453 454 455 (MSX RC -SLOT- 456 457 458 459 460 461 462 463 464 465 466 465	025A 025B 025C 025D	73 08 D9 C9		; ;; ; Selec ; throu ; Input ; A - ; ;	LD EX EXX RET 3.44 	(HL),E AF,AF' 01-Jan-85 appropriate slot sters; and perma eters: 	<pre>;Restore possible returned values PAGE 13 PAGE 13 content of the value given mently enables the slot. t # (0-3) ot # (0-3)</pre>	20
452 453 454 455 (MSX RC -SLOT- 456 457 458 459 460 461 462 463 464 465 466 467 468	025A 025B 025C 025D	73 08 D9 C9		; ; ; Selec; ; throu; ; Input ; A - ;	LD EX EXX RET 3.44 	(HL),E AF,AF' 01-Jan-85 appropriate slot sters; and perma eters: 	<pre>;Restore possible returned values PAGE 13 ENASLT</pre>	20
452 453 454 455 (MSX RC -SLOT- 456 457 458 459 460 461 462 463 464 465 466 465 466 465	025A 025B 025C 025D	73 08 D9 C9		;; ; Selec; ; throu; ; Input; ; A - ; ;	LD EX EXX RET 3.44 ts the lgh regi parame FxxxSSE + ++	(HL),E AF,AF' 01-Jan-85 appropriate slot sters; and perma eters: 	<pre>;Restore possible returned values PAGE 13 PAGE 13 content of the value given amently enables the slot. t # (0-3) ot # (0-3) ury slot # specified</pre>	20
452 453 454 455 (MSX RC -SLOT- 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470	025A 025B 025C 025D	73 08 D9 C9		;; ; Selec; ; throu; ; Input; ; A - ; ;	LD EX EXX RET 3.44 ts the lgh regi parame FxxxSSE + ++	(HL),E AF,AF' 01-Jan-85 appropriate slot sters, and perma eters: pp primary slot secondary slot	<pre>;Restore possible returned values PAGE 13 PAGE 13 content of the value given amently enables the slot. t # (0-3) ot # (0-3) ury slot # specified</pre>	20
452 453 454 455 (MSX RC -SLOT- 456 457 458 459 460 461 462 463 464 465 466 465 466 467 468 469 470 471 473	025A 025B 025C 025D	73 08 D9 C9		; ;; ; Selec; ; throu ; Input ; ; A - ; ; ; ; ; HL -	LD EX EXX RET 3.44 	<pre>(HL),E AF,AF' 01-Jan-85 appropriate slot sters, and perma sters: p + primary slot secondary slot l if secondary s of target memory</pre>	<pre>;Restore possible returned values PAGE 13 PAGE 13 content of the value given amently enables the slot. t # (0-3) ot # (0-3) ury slot # specified</pre>	20
452 453 454 455 (MSX RC -SLOT- 456 457 458 459 460 461 462 463 464 465 466 466 466 466 466 466 467 468 469 470 471 472 473 474	025A 025B 025C 025D	73 08 D9 C9		; ;; ; Selec; ; throu ; Input ; ; A - ; ; ; ; ; HL -	LD EX EXX RET 3.44 cts the logh regins parameter FxxxSSF ++ +	<pre>(HL),E AF,AF' 01-Jan-85 appropriate slot sters, and perma sters: p + primary slot secondary slot l if secondary s of target memory</pre>	<pre>;Restore possible returned values PAGE 13 ENASLT caccording to the value given mently enables the slot. : # (0-3) ot # (0-3) ury slot # specified :y</pre>	20
452 453 454 455 (MSX RC -SLOT- 456 457 458 459 460 461 462 463 464 465 466 465 466 467 468 469 470 471 472 473 475	025A 025B 025C 025D 025D 025D	73 08 D9 C9 C9 C BIOS		; ; ; Selec; ; throu ; Input ; ; A - ; ; ; HL - ; ; Note ;	LD EX EXX RET 3.44 	<pre>(HL),E AF,AF' 01-Jan-85 appropriate slot sters, and perma sters: pp + primary slot 1 if secondary slot 1 if secondary slot of target memoi rupts are disable</pre>	<pre>;Restore possible returned values PAGE 13 ENASLT caccording to the value given mently enables the slot. : # (0-3) ot # (0-3) ury slot # specified :y</pre>	20
452 453 454 455 (MSX RC -SLOT- 456 457 458 459 460 461 462 463 464 465 466 465 466 467 468 469 470 471 472 473 474 475 476	025A 025C 025C 025D DM BASIC	73 08 D9 C9 C9 C9) Macro-8	; ; Selec; ; throu; ; Input; ; A - ; ; ; ; ; ; HL - ; ; Note;	LD EX EXX RET 3.44 	<pre>(HL),E AF,AF' 01-Jan-85 appropriate slot sters, and perma sters: pp + primary slot l if secondary sl l if secondary sl condary slot secondary slot l if secondary slot condary slot stars are disable is routine.</pre>	<pre>;Restore possible returned values PAGE 13 ENASLT caccording to the value given mently enables the slot. : # (0-3) ot # (0-3) ury slot # specified :y</pre>	20
452 453 454 455 (MSX RC -SLOT- 456 457 458 459 460 461 462 461 462 463 464 465 466 467 468 466 467 468 469 470 471 472 473 474 475 477	025A 025D 025C 025D M BASIC	73 08 D9 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9 C9) Macro-8	; ; ; Selec; ; throu ; Input ; ; A - ; ; ; HL - ; ; Note ;	LD EX EXX RET 3.44 	(HL),E AF,AF' 01-Jan-85 appropriate slot sters, and perma eters: primary slot secondary slot l if secondary slot a of target memor trupts are disable sis routine. SELPRM	<pre>;Restore possible returned values PAGE 13 PAGE 13 contemposation of the value given anonally enables the slot. contemposation of the value given anona</pre>	20
452 453 454 455 (MSX RC -SLOT- 456 457 458 459 460 461 462 463 464 465 466 465 466 467 468 469 470 471 472 473 474 475 476 477	025A 025B 025C 025D 04 BASIC 025D 00 BASIC	73 08 D9 C9 C BIOS C BIOS) Macro-8 027E 026B	; ; ; Selec; ; throu ; Input ; ; A - ; ; ; HL - ; ; Note ;	LD EX EXX RET 3.44 	(HL),E AF,AF' 01-Jan-85 appropriate slot sters, and perme eters: primary slot secondary sl l if secondary sl l if secondary sl secondary sl 	<pre>;Restore possible returned values PAGE 13 PAGE 13ENASLT c according to the value given mently enables the slot. c # (0-3) ot # (0-3) mry slot # specified ry Led automatically but never enabled</pre>	20
452 453 454 455 (MSX RC -SLOT- 456 457 458 459 460 461 462 463 464 465 466 467 468 466 467 468 469 470 471 473 474 473 474 475 476 477 478 479	025A 025C 025C 025D DM BASIC	73 08 D9 C9 C9 E BIOS C9 C0 C0 C0 FA DB) Macro-8	; ; ; Selec; ; throu ; Input ; ; A - ; ; ; HL - ; ; Note ;	LD EX EX RET 3.44 	<pre>(HL),E AF,AF' 01-Jan-85 appropriate slot sters, and perma sters: pp ! primary slot l if secondary sl l if secondary sl l if secondary sl sters are disable this routine. SELPRM M,ENESLT A,(PPI.AR)</pre>	<pre>;Restore possible returned values PAGE 13 PAGE 13 ENASLT : # (0-3) : # (0-3) : # (0-3) ury slot # specified :y Led automatically but never enabled ;Calculate bit pattern and mask code ;Expanded slot specified</pre>	
452 453 454 455 (MSX RC -SLOT- 456 457 458 459 460 461 462 463 464 465 466 467 468 466 467 468 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480	025A 025B 025C 025D 0M BASIC 025D 025D 025D 025E 025E 025E 025E 0261 0264	73 08 D9 C9 C9 C9 C0 C0 FA DB) Macro-8 027E 026B A8	; ; ; Selec; ; throu ; Input ; ; A - ; ; ; HL - ; ; Note ;	LD EX EXX RET 3.44 	<pre>(HL),E AF,AF' 01-Jan-85 appropriate slot sters, and perma eters:</pre>	<pre>;Restore possible returned values PAGE 13 PAGE 13 ENASLT</pre>	
452 453 454 455 (MSX RC -SLOT- 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481	025A 025B 025C 025D 0M BASIC 025E 025E 025E 025E 0261 0264 0267	73 08 D9 C9 C9 C BIOS C DIOS C D FA D8 A D B B0) Macro-8 027E 026B A8	; ; ; Selec; ; throu ; Input ; ; A - ; ; ; HL - ; ; Note ;	LD EX EXX RET 3.44 	<pre>(HL),E AF,AF' 01-Jan-85 appropriate slot sters, and perma eters:</pre>	<pre>;Restore possible returned values PAGE 13 PAGE 13 ENASLT : # (0-3) : # (0-3) : # (0-3) ury slot # specified :y Led automatically but never enabled ;Calculate bit pattern and mask code ;Expanded slot specified</pre>	
452 453 454 455 (MSX RC -SLOT- 456 457 458 459 460 461 462 463 464 465 466 467 468 465 466 467 470 471 472 473 474 475 476 477 475 476 477 478 479 480 481 482	025A 025D 025C 025D 0M BASIC 025E 025E 025E 025E 025E 0261 0264 0266 0268	73 08 D9 C9 C9 C9 C9 C9 C9 C9 C0 C0 C0 C0 C0 C0 C0 C0 C0 C0 C0 C0 C0) Macro-8 027E 026B A8 A8	; ; ; Selec; ; throu ; Input ; ; A - ; ; ; HL - ; ; Note ;	LD EX EX RET 3.44 	<pre>(HL),E AF,AF' 01-Jan-85 appropriate slot sters, and perma eters:</pre>	<pre>;Restore possible returned values PAGE 13 PAGE 13 ENASLT</pre>	
452 453 454 455 (MSX RC -SLOT- 456 457 458 459 460 461 462 463 464 465 466 467 468 466 467 468 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483	025A 025B 025C 025D 08 BASIC 025D 00 BASIC 025E 025E 0261 0266 0267 0268 026A	73 08 D9 C9 C9 C9 C0 C0 FA DB D4 D4 D4 D4 D4 D4 D4 D4 D4 D4 D4 C9 C2 C2 C2 C2 C2 C2 C2 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3 C3) Macro-8 027E 026B A8 A8	; ; Selec; ; throu ; Input ; A - ; ; ; HL - ; ; Note ; ;	LD EX EXX RET 3.44 	<pre>(HL),E AF,AF' 01-Jan-85 appropriate slot sters, and perma eters:</pre>	<pre>;Restore possible returned values PAGE 13 PAGE 13 ENASLT</pre>	
452 453 454 455 (MSX RC -SLOT- 456 457 458 459 460 461 462 463 464 465 466 467 468 465 466 467 470 471 472 473 474 475 476 477 475 476 477 478 479 480 481 482	025A 025D 025C 025D 0M BASIC 025E 025E 025E 025E 025E 0261 0264 0266 0268	73 08 09 09 09 09 0 0 0 0 0 0 0 0 0 0 0 0) Macro-8 027E 026B A8 A8	; ; ; Selec; ; throu ; Input ; ; A - ; ; ; HL - ; ; Note ;	LD EX EXX RET 3.44 	<pre>(HL),E AF,AF' 01-Jan-85 appropriate slot sters, and perma sters: pp i + primary slot secondary slot </pre>	<pre>;Restore possible returned values PAGE 13 PAGE 14 second ing to the value given mently enables the slot. pace of the value given second is the slot. pace of the value given pace of the value gi</pre>	
452 453 454 455 (MSX RC -SLOT- 456 457 458 459 460 461 462 463 464 465 466 467 468 469 470 471 472 473 474 475 476 477 478 479 480 481 482 483 484	025A 025B 025C 025D 04 BASIC 025D 04 BASIC 025E 025E 0261 0264 0268 0268 0268	73 08 D9 C9 C9 C9 C9 C9 C9 C0 C0 C0 FA B8 A1 B00 D3 C9 E5) Macro-8 027E 026B A8 A8	; ; Selec; ; throu ; Input ; A - ; ; ; HL - ; ; Note ; ;	LD EX EXX RET 3.44 	<pre>(HL),E AF,AF' 01-Jan-85 appropriate slot sters, and perma eters:</pre>	<pre>;Restore possible returned values PAGE 13 PAGE 13 ENASLT</pre>	

-SLOT-	M BASIC E	BIOS) Macro	-80 3.44	01-Jan-85	PAGE 13-1
487	0268	4.8			
	026F	4F	LD	C,A	;Move primary slot # to [BC]
488	0270	06 00	LD	в,0	
489	0272	7 D	LD	A,L	Re-calculate what is currently output
490	0273	A4	AND	н	;to expansion slot register
491	0274	B2	OR	D	
492	0275	21 FCC5	LD	HL, SLTTBL	Calculate address into SLTTBL
493	0278	09	ADD	HL, BC	• • • • • • • • • • • • • • • • • • • •
494	0279	77	LD		. Cat muse in the submit to surrent to
495	0273		LD	(HL),A	;Set current value output to expansion
					slot register
496	027A	El	POP	HL	;Restore target address
.497	027B	79	LD	A,C	;Restore primary slot # to [Acc]
498	027C	18 EO	JR	ENASLT	;Enable by primary slot register
(MSX -SLOT-		C BIOS) Macr	co-80 3.44	01-Jan-85	PAGE 14
499					
500	027E		SELPRM:		
501	027E	F3			
			DI		
502	027F	F5	PUSH	AF	;Save slot address
503	0280	7C .	LD	А,Н	;Extract upper 2 bits
504	0281	07	RLCA		
505	0282	07	RLCA		
506.	0283	E6 03		000000115	
			AND	00000011B	
507	0285	5F	LD	E,A	
508	0286	3E C0	LD	A,OCOH	· ;Format mask pat. correspond to address
509	0288		SLPRM1:		
510	0288	07	RLCA		
511	0289	07	RLCA		
512	028A	10			
			DEC	Е	
513	028B		JP	P, SLPRM1	
514	028E	5F	LD	E,A	;Save mask pattern
515					; 00000011 0000-3FFF
516					; 00001100 4000-7FFF
517					
518					; 00110000 8000-BFFF
	0000	25			; 11000000 C000-FFFF
519	028F	2F	CPL		
520	02.90	4 F	LD	C,A	;Save mask pattern
521					; 11111100 0000-3FFF
522					
523					; 11110011 4000-7FFF
524					; 11001111 8000-BFFF
					; 00111111 C000-FFFF
525	0291	Fl	POP	AF	;Restore slot address
526	0292	F5	PUSH	AF	
527	0293	E6 03	AND	00000011B	;Extract primary slot #
528	0295	3C	INC	A	,
529	0296	47	LD	B,A	
529				27.1	
529	0200				
(MSX RO	DM BASIC	BĮOS) Macro	-80 3.44	01-Jan-85	PAGE 14-1
(MSX RO -SLOT- 530	DM BASIC	BĮOS) Macro 3E AB	LD	01-Jan-85 A,10101011B	PAGE 14-1 ;Convert slot # to proper bit pattern
(MSX RO -SLOT- 530 531	0297 0299	3E AB	LD SLPRM2:	A,10101011B	
(MSX RO -SLOT- 530	DM BASIC		LD		
(MSX RO -SLOT- 530 531	0297 0299	3E AB	LD SLPRM2:	A,10101011B	
(MSX RO -SLOT- 530 531 532	0297 0299 0299 0299 0298	3E AB C6 55 10 FC	LD SLPRM2: ADD	A,10101011B A,01010101B SLPRM2	;Convert slot # to proper bit pattern
(MSX RO -SLOT- 530 531 532 533 534	0297 0299 0299 0299	3E AB C6 55	LD SLPRM2: ADD DJNZ	A,10101011B A,01010101B	;Convert slot # to proper bit pattern ;Save bit pattern for primary slot #
(MSX RO -SLOT- 530 531 532 533 534 535	0297 0299 0299 0299 0298	3E AB C6 55 10 FC	LD SLPRM2: ADD DJNZ	A,10101011B A,01010101B SLPRM2	;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 0000000 slot #0
(MSX RO -SLOT- 530 531 532 533 534 535 536	0297 0299 0299 0299 0298	3E AB C6 55 10 FC	LD SLPRM2: ADD DJNZ	A,10101011B A,01010101B SLPRM2	;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 0000000 slot #0 ; 01010101 slot #1
(MSX RO -SLOT- 530 531 532 533 534 535 536 537	0297 0299 0299 0299 0298	3E AB C6 55 10 FC	LD SLPRM2: ADD DJNZ	A,10101011B A,01010101B SLPRM2	<pre>;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 0000000 slot #0 ; 01010101 slot #1 ; 10101010 slot #2</pre>
(MSX RO -SLOT- 530 531 532 533 534 535 536	0297 0299 0299 0299 0298	3E AB C6 55 10 FC	LD SLPRM2: ADD DJNZ	A,10101011B A,01010101B SLPRM2	<pre>;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 00000000 slot #0 ; 01010101 slot #1</pre>
(MSX RO -SLOT- 530 531 532 533 534 535 536 537	0297 0299 0299 0299 0298	3E AB C6 55 10 FC	LD SLPRM2: ADD DJNZ	A,10101011B A,01010101B SLPRM2	;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 00000000 slot #0 ; 01010101 slot #1 ; 10101010 slot #2 ; 11111111 slot #3
(MSX RO -SLOT- 530 531 532 533 534 535 536 537 538 538 539	0297 0299 0299 0298 029D 029D	3E AB C6 55 10 FC 57 A3	LD SLPRM2: ADD DJNZ LD	A,10101011B A,01010101B SLPRM2 D,A	<pre>;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 0000000 slot #0 ; 01010101 slot #1 ; 10101010 slot #2 ; 1111111 slot #3 ;Extract significant bits</pre>
(MSX RO -SLOT- 530 531 532 533 534 535 536 537 538 539 539 540	0M BASIC 0297 0299 0299 0298 029D 029D 029E 029F	3E AB C6 55 10 FC 57 A3 47	LD SLPRM2: ADD DJNZ LD AND LD	A,10101011B A,01010101B SLPRM2 D,A E B,A	<pre>;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 0000000 slot #0 ; 0101010 slot #1 ; 10101010 slot #1 ; 1111111 slot #3 ;Extract significant bits ;Set it to [B]</pre>
(MSX RO -SLOT- 530 531 532 533 534 535 536 537 538 539 540 541	0297 0299 0299 0298 0299 029B 029D 029E 029F 029F 029F 02A0	3E AB C6 55 10 FC 57 A3 47 F1	LD SLPRM2: ADD JJNZ LD AND LD POP	A,10101011B A,01010101B SLPRM2 D,A 'E B,A AF	<pre>;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 0000000 slot #0 ; 0101010 slot #1 ; 1010100 slot #2 ; 1111111 slot #3 ;Extract significant bits ;Set it to [B] ;Expanded slot specified?</pre>
(MSX RO -SLOT- 530 531 532 533 534 535 536 537 538 537 538 539 540 541 542	0297 0299 0299 0298 0298 0290 0290 0290 0297 0297 0297 0297 0297	3E AB C6 55 10 FC 57 A3 47 F1 A7	LD SLPRM2: ADD DJN2 LD AND LD POP AND	A,10101011B A,01010101B SLPRM2 D,A ČE B,A	<pre>;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 0000000 slot #0 ; 01010101 slot #1 ; 10101010 slot #2 ; 11111111 slot #3 ;Extract significant bits ;Set it to [B]</pre>
(MSX RO -SLOT- 530 531 532 533 534 535 536 537 538 539 540 541 542 543	0M BASIC 0297 0299 0299 0290 029D 029D 029D 029E 029F 029F 02A0 02A1 02A2	3E AB C6 55 10 FC 57 A3 47 F1	LD SLPRM2: ADD DJNZ LD LD POP AND RET	A,10101011B A,01010101B SLPRM2 D,A 'E B,A AF	<pre>;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 0000000 slot #0 ; 0101010 slot #1 ; 1010100 slot #2 ; 1111111 slot #3 ;Extract significant bits ;Set it to [B] ;Expanded slot specified?</pre>
(MSX RO -SLOT- 530 531 532 533 534 535 536 537 538 537 538 539 540 541 542	0297 0299 0299 0298 0298 0290 0290 0290 0297 0297 0297 0297 0297	3E AB C6 55 10 FC 57 A3 47 F1 A7	LD SLPRM2: ADD DJN2 LD AND LD POP AND	A,10101011B A,01010101B SLPRM2 D,A 'E B,A AF	<pre>;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 0000000 slot #0 ; 0101010 slot #1 ; 1010100 slot #2 ; 1111111 slot #3 ;Extract significant bits ;Set it to [B] ;Expanded slot specified?</pre>
(MSX RO -SLOT- 530 531 532 533 534 535 536 537 538 537 538 537 538 537 538 537 538 537 538 537 538 537 540 541 542 543 544	04 BASIC 0297 0299 0299 0298 0290 0290 0290 0290 0290 0290 0290 0291 0292 0292 0292 0292 0292 0293 0292 0293 0295 025 025 025 025 025 025 025 02	3E AB C6 55 10 FC 57 A3 47 F1 A7 C9	LD SLPRM2: ADD JJNZ LD POP AND RET SELEXP:	A,10101011B A,01010101B SLPRM2 D,A 'E B,A AF A	<pre>;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 00000000 slot #0 ; 01010101 slot #1 ; 101010010 slot #2 ; 11111111 slot #3 ;Extract significant bits ;Set it to [B] ;Expanded slot specified? ;Set sign flag if so</pre>
(MSX RO -SLOT- 530 531 532 533 534 535 536 537 538 538 539 540 541 542 543 544 545	0297 0299 0299 0298 0290 0290 0290 0290 0290	3E AB C6 55 10 FC 57 A3 47 F1 A7 C9 F5	LD SLPRM2: ADD DJN2 LD AND LD POP AND RET SELEXP: PUSH	A,10101011B A,01010101B SLPRM2 D,A E B,A AF A AF A	<pre>;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 00000000 slot #0 ; 01010101 slot #1 ; 10101010 slot #2 ; 11111111 slot #3 ;Extract significant bits ;Set it to [B] ;Expanded slot specified? ;Set sign flag if so ;Save target slot</pre>
(MSX RO -SLOT- 530 531 532 533 534 535 536 537 538 537 538 537 538 539 540 541 542 543 544 545 545	0M BASIC 0297 0299 0299 0290 0290 0290 0290 0290 0290 0241 0242 0243 0244	3E AB C6 55 10 FC 57 A3 47 F1 A7 C9 F5 7A	LD SLPRM2: ADD DJNZ LD POP AND RET SELEXP: PUSH LD	A,10101011B A,01010101B SLPRM2 D,A B,A AF AF A,D	<pre>;Convert slot # to proper bit pattern ; Save bit pattern for primary slot # ; 0000000 slot #0 ; 0100101 slot #1 ; 10101010 slot #1 ; 10101010 slot #2 ; 11111111 slot #3 ;Extract significant bits ;Set it to [B] ;Expanded slot specified? ;Set sign flag if so ;Save target slot ;Get bit pattern for primary slot</pre>
(MSX RO -SLOT- 530 531 532 533 534 535 536 537 538 537 538 537 538 537 538 537 538 537 538 537 534 541 542 543 544 545 546 547	04 BASIC 0297 0299 0299 029B 029D 029D 029D 029D 029D 024D 02A0 02A1 02A3 02A3 02A3 02A3 02A5	3E AB C6 55 10 FC 57 A3 47 F1 A7 C9 F5 7A E6 C0	LD SLPRM2: ADD JJN2 LD POP AND RET SELEXP: PUSH LD AND AND	A,10101011B A,01010101B SLPRM2 D,A 'E B,A AF A AF A,D 11000000B	<pre>;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 0000000 slot #0 ; 0101010 slot #1 ; 1010100 slot #2 ; 11111111 slot #3 ;Extract significant bits ;Save tit to [B] ;Expanded slot specified? ;Save target slot ;Gave target slot ;Gave target slot ;Extract slot # for 0C000HOFFFH</pre>
(MSX RO -SLOT- 530 531 532 533 534 535 536 537 538 539 540 541 542 541 542 541 544 545 546 547 548	04 BASIC 2 0297 0299 0299 0298 029D 029D 029D 029D 029D 029D 0240 02A1 02A2 02A3 02A3 02A3 02A4 02A5 02A7	3E AB C6 55 10 FC 57 A3 47 F1 A7 C9 F5 7A E6 C0 4F	LD SLPRM2: ADD JJN2 LD AND SELEXP: PUSH LD AND LD AND LD	A,10101011B A,01010101B SLPRM2 D,A 'E B,A AF A AF A,D 110000000B C,A	<pre>;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 00000000 slot #0 ; 0101010 slot #1 ; 1010100 slot #2 ; 1111111 slot #3 ;Extract significant bits ;Set it to [B] ;Expanded slot specified? ;Set sign flag if so ;Save target slot ;Get bit pattern for primary slot ;Extract slot # for 0C000H0FFFFH ;Save it</pre>
(MSX RO -SLOT- 530 531 532 533 534 535 536 537 538 537 538 537 538 537 538 537 538 537 538 537 534 541 542 543 544 545 546 547	04 BASIC 0297 0299 0299 029B 029D 029D 029D 029D 029D 024D 02A0 02A1 02A3 02A3 02A3 02A3 02A5	3E AB C6 55 10 FC 57 A3 47 F1 A7 C9 F5 7A E6 C0	LD SLPRM2: ADD JJN2 LD POP AND RET SELEXP: PUSH LD AND AND	A,10101011B A,01010101B SLPRM2 D,A 'E B,A AF A AF A,D 11000000B	<pre>;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 00000000 slot #0 ; 0101010 slot #1 ; 1010100 slot #2 ; 1111111 slot #3 ;Extract significant bits ;Set it to [B] ;Expanded slot specified? ;Set sign flag if so ;Save target slot ;Get bit pattern for primary slot ;Extract slot # for 0C000H0FFFFH ;Save it</pre>
(MSX RO -SLOT- 530 531 532 533 534 535 536 537 538 539 540 541 542 543 541 542 543 544 545 546 547 548 549	04 BASIC 0297 0299 0299 0299 0290 0290 0290 0290 0281 0282 0281 0283 0283 0284 0285 0285 0285 0285 0285	3E AB C6 55 10 FC 57 A3 47 F1 A7 C9 F5 7A E6 C0 4F F1	LD SLPRM2: ADD DJNZ LD POP AND SELEXP: PUSH LD AND LD POP	A,10101011B A,01010101B SLPRM2 D,A B,A AF A,D 11000000B C,A AF	<pre>;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 0000000 slot #0 ; 0101010 slot #1 ; 10101010 slot #1 ; 1011111 slot #3 ;Extract significant bits ;Set it to [B] ;Expanded slot specified? ;Set sign flag if so ;Save target slot ;Get bit pattern for primary slot ;Extract slot # for 0C000H0FFFFH ;Save it ;Restore target slot</pre>
(MSX RC -SLOT- 530 531 532 533 534 535 536 537 538 537 538 537 538 537 540 541 542 543 544 543 544 543 544 545 546 547 548 549 550	04 BASIC 0297 0299 0298 0290 029D 029D 029D 029D 029D 029D 024D 024D 0243 02A3 02A3 02A3 02A3 02A5 02A7 02A9 02A3 02	3E AB C6 55 10 FC 57 A3 47 F1 A7 C9 F5 7A E6 C0 4F F1 F5	LD SLPRM2: ADD DJNZ LD POP AND RET SELEXP: PUSH LD AND LD POP PUSH	A,10101011B A,01010101B SLPRM2 D,A B,A AF A, AF A,D 11000000B C,A AF AF AF	<pre>;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 0000000 slot #0 ; 0101010 slot #1 ; 1010100 slot #2 ; 11111111 slot #3 ;Extract significant bits ;Save it to [B] ;Expanded slot specified? ;Save target slot ;Gave target slot ;Gave target slot ;Extract slot # for 0C000HOFFFFH ;Save it ;Restore target slot ;Save target slot ;Save target slot</pre>
(MSX RO -SLOT- 530 531 532 533 534 535 536 537 538 537 538 539 541 542 541 544 545 544 545 544 545 544 547 548 549 550 551	04 BASIC 0297 0299 0299 029B 029D 029D 029D 029D 029D 024D 02A1 02A2 02A3 02A3 02A3 02A5 02A7 02A8 02A	3E AB C6 55 10 FC 57 A3 47 F1 A7 C9 F5 7A C9 F5 7A E6 C0 4F F1 F5 57	LD SLPRM2: ADD JJNZ LD POP AND RET SELEXP: PUSH LD AND LD POP PUSH LD	A,10101011B A,01010101B SLPRM2 D,A E B,A AF A,D 11000000B C,A AF AF AF AF AF AF AF AF AF AF	<pre>;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 0000000 slot #0 ; 01010101 slot #1 ; 10101010 slot #2 ; 11111111 slot #3 ;Extract significant bits ;Set it to [B] ;Expanded slot specified? ;Set sign flag if so ;Save target slot ;Get bit pattern for primary slot ;Extract slot # for 0C000H0FFFFH ;Save it ;Restore target slot</pre>
(MSX RO -SLOT- 530 531 532 533 534 535 536 537 538 537 538 537 538 539 540 541 542 543 544 543 544 545 545 545 545 550	0M BASIC 0297 0299 0299 0299 0290 0290 0290 0290 0290 0201 0281 0283 0283 0284 0285 0284 0285 0284 0286 0289 0288 0299 0288 0290	3E AB C6 55 10 FC 57 A3 47 F1 A7 C9 F5 7A C9 F5 7A E6 C0 4F F1 F5 57 DD A8	LD SLPRM2: ADD DJNZ LD POP AND RET SELEXP: PUSH LD AND LD POP PUSH	A,10101011B A,01010101B SLPRM2 D,A B,A AF A, AF A,D 11000000B C,A AF AF AF	<pre>;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 0000000 slot #0 ; 0101010 slot #1 ; 1010100 slot #2 ; 11111111 slot #3 ;Extract significant bits ;Save it to [B] ;Expanded slot specified? ;Save target slot ;Gave target slot ;Gave target slot ;Extract slot # for 0C000HOFFFFH ;Save it ;Restore target slot ;Save target slot ;Save target slot</pre>
(MSX RO -SLOT- 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 545 544 545 544 545 544 547 548 549 550 551	04 BASIC 0297 0299 0299 029B 029D 029D 029D 029D 029D 024D 02A1 02A2 02A3 02A3 02A3 02A5 02A7 02A8 02A	3E AB C6 55 10 FC 57 A3 47 F1 A7 C9 F5 7A C9 F5 7A E6 C0 4F F1 F5 57	LD SLPRM2: ADD JJNZ LD POP AND RET SELEXP: PUSH LD AND LD POP PUSH LD	A,10101011B A,01010101B SLPRM2 D,A E B,A AF A,D 11000000B C,A AF AF AF AF AF AF AF AF AF AF	<pre>;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 0000000 slot #0 ; 0101010 slot #1 ; 1010100 slot #2 ; 11111111 slot #3 ;Extract significant bits ;Save it to [B] ;Expanded slot specified? ;Save target slot ;Gave target slot ;Gave target slot ;Extract slot # for 0C000HOFFFFH ;Save it ;Restore target slot ;Save target slot ;Save target slot</pre>
(MSX RO -SLOT- 530 531 532 533 534 535 536 537 538 537 538 537 538 537 538 537 541 541 543 544 543 544 543 544 545 546 547 548 549 550 551 552 553	04 BASIC 0297 0299 0298 0290 0240 0243 0243 0245 0240 02	3E AB C6 55 10 FC 57 A3 47 F1 A7 F1 A7 C9 F5 7A E6 C0 4F F1 F5 57 DB A8 47	LD SLPRM2: ADD DJNZ LD AND RET SELEXP: PUSH LD AND LD POP PUSH LD IN LD IN LD	A,10101011B A,01010101B SLPRM2 D,A E B,A AF A,D 11000000B C,A AF AF AF AF AF AF AF AF AF AF AF AF AF	<pre>;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 0000000 slot #0 ; 0101010 slot #1 ; 10101010 slot #2 ; 11111111 slot #3 ;Extract significant bits ;Set it to [B] ;Expanded slot specified? ;Set sign flag if so ;Save target slot ;Get bit pattern for primary slot ;Extract slot # for 0C000H0FFFFH ;Save it ;Restore target slot ;Save target slot ;Save target slot ;Save target slot ;Load [D] with specified slot address ;Save current setting</pre>
(MSX RO -SLOT- 530 531 532 533 534 535 536 537 538 539 541 542 541 544 545 544 545 544 545 544 545 547 548 547 548 549 550 551 552 553 554	0M BASIC 0297 0299 0299 0298 029D 029D 029D 029D 029D 024D 02A1 02A2 02A3 02A3 02A3 02A3 02A3 02A5 02A7 02A8 02A9 02AB 02AB 02AB 02AB 02AB 02AB 02AB	3E AB C6 55 10 FC 57 A3 47 F1 A7 C9 F5 7A E6 C0 4F F1 F5 57 DB A8 47 E6 3F	LD SLPRM2: ADD JJNZ LD POP AND RET SELEXP: PUSH LD AND LD POP PUSH LD IN LD AND LD POP PUSH LD AND AND	A,10101011B A,01010101B SLPRM2 D,A E B,A AF A,D 11000000B C,A AF AF AF D,A A,(PPI.AR) B,A 00111111B	<pre>;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 0000000 slot #0 ; 0101010 slot #1 ; 10101010 slot #2 ; 11111111 slot #3 ;Extract significant bits ;Set it to [B] ;Expanded slot specified? ;Set sign flag if so ;Save target slot ;Get bit pattern for primary slot ;Extract slot # for 0C000H0FFFFH ;Save it ;Restore target slot ;Save target slot ;Save target slot ;Save target slot ;Load [D] with specified slot address ;Save current setting</pre>
(MSX RO -SLOT- 530 531 532 533 534 535 536 537 538 537 538 539 540 541 542 543 544 543 544 545 545 546 547 545 551 552 553 554 555	0M BASIC 0297 0299 0299 0299 0290 0290 0290 0290 0201 0202 0201 0202 0201 0202 0201 0202 0201 0202 00	3E AB C6 55 10 FC 57 A3 47 F1 A7 C9 F5 7A E6 C0 4F F1 F5 57 DB A8 47 E6 3F B1	LD SLPRM2: ADD DJNZ LD ND POP AND DJNZ LD POP PUSH LD AND LD POP PUSH LD IN LD AND DO POP PUSH LD IN IN SELEXP:	A,10101011B A,01010101B SLPRM2 D,A B,A AF A,D 11000000B C,A AF AF AF AF AF AF AF AF AF AF AF AF AF	<pre>;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 0000000 slot #0 ; 01010101 slot #1 ; 1010101 slot #2 ; 1111111 slot #3 ;Extract significant bits ;Set it to [B] ;Expanded slot specified? ;Set sign flag if so ;Save target slot ;Get bit pattern for primary slot ;Extract slot # for 0C000HOFFFH ;Save it ;Restore target slot ;Save target slot ;Load [D] with specified slot address ;Save current setting ;Cancel current setting for 0C000HOFF</pre>
(MSX RO -SLOT- 530 531 532 533 534 535 536 537 538 537 538 537 538 537 538 537 541 541 543 544 543 544 543 544 545 554 550 551 552 553 555 555 555	0M BASIC 0297 0299 0298 0290 0200 0000 0000 00000 0000 0000 0000 0000 0	3E AB C6 55 10 FC 57 A3 47 F1 A7 F1 A7 C9 F5 7A E6 C0 4F F1 F5 57 DB A8 47 E6 3F B1 D3 A8	LD SLPRM2: ADD JJNZ LD AND RET SELEXP: PUSH LD AND LD POP PUSH LD IN LD IN LD NC SUSH LD IN SUSH LD IN SUSH LD IN SUSH LD IN SUSH LD IN SUSH LD IN SUSH LD IN SUSH LD IN SUSH LD IN SUSH LD IN SUSH LD IN SUSH LD IN SUSH LD IN SUSH SUSH IN IN SUSH IN SUS SUSH IN SUSH SUS SUS SUS SUS SUS IN SUS SUS SUS SUS IN SUS	A,10101011B A,01010101B SLPRM2 D,A E B,A AF A,D 11000000B C,A AF AF AF AF D,A A,(PPI.AR) B,A 00111111B C (PPI.AW),A	<pre>;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 0000000 slot #0 ; 0101010 slot #1 ; 10101010 slot #2 ; 11111111 slot #3 ;Extract significant bits ;Set it to [B] ;Expanded slot specified? ;Set sign flag if so ;Save target slot ;Get bit pattern for primary slot ;Extract slot # for 0C000H0FFFFH ;Save it ;Restore target slot ;Save target slot ;Load [D] with specified slot address ;Save current setting ;Cancel current setting for 0C000H0FF ;Enable 0C000H0FFFFH or target bank</pre>
(MSX RO -SLOT- 530 531 532 533 534 535 537 538 537 538 539 540 541 542 543 544 545 545 546 547 545 546 547 548 549 550 551 552 553 554 555	0M BASIC 0297 0299 0299 0299 0290 0290 0290 0290 0201 0202 0201 0202 0201 0202 0201 0202 0201 0202 00	3E AB C6 55 10 FC 57 A3 47 F1 A7 C9 F5 7A E6 C0 4F F1 F5 57 DB A8 47 E6 3F B1	LD SLPRM2: ADD DJNZ LD ND POP AND DJNZ LD POP PUSH LD AND LD POP PUSH LD IN LD AND DO POP PUSH LD IN IN SELEXP:	A,10101011B A,01010101B SLPRM2 D,A B,A AF A,D 11000000B C,A AF AF AF AF AF AF AF AF AF AF AF AF AF	<pre>;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 0000000 slot #0 ; 0101010 slot #1 ; 10101010 slot #2 ; 11111111 slot #3 ;Extract significant bits ;Set it to [B] ;Expanded slot specified? ;Set sign flag if so ;Save target slot ;Get bit pattern for primary slot ;Extract slot # for 0C000H0FFFFH ;Save it ;Restore target slot ;Save target slot ;Load [D] with specified slot address ;Save current setting ;Cancel current setting for 0C000H0FF ;Enable 0C000H0FFFFH or target bank</pre>
(MSX RO -SLOT- 530 531 532 533 534 535 536 537 538 539 541 542 543 544 545 543 544 545 543 544 545 548 547 551 552 553 554 555 556 557	0M BASIC 0297 0299 0299 0298 029D 029D 029D 029D 029D 024D 024D 02A1 02A3 02A3 02A3 02A3 02A3 02A3 02A3 02A5 02A7 02A9 02A9 02A9 02A9 02A9 02A3 02A3 02A3 02A3 02A4 02A5 02A9 02A9 02A3 02A3 02A3 02A3 02A4 02A5 02A9 02A9 02A3 02A5 02A9 02A9 02A3 02A3 02A3 02A5 02A9 02A5 02A9 02A3 02A3 02A5 02A9 02A9 02A3 02A3 02A3 02A5 02A9 02A9 02A3 02A5 02A9 02A9 02A9 02A3 02A3 02A3 02A5 02A9 02A9 02A9 02A3 02A5 02A9 02A9 02A9 02A3 02A3 02A3 02A4 02A5 02A9 02A9 02A9 02A9 02A9 02A9 02A9 02A3 02A3 02A3 02A3 02A4 02A5 02A9 02A8 02B8 02	3E AB C6 55 10 FC 57 A3 47 F1 A7 C9 F5 7A E6 C0 4F F1 F5 57 DB A8 47 E6 3F B1 D3 A8 7A	LD SLPRM2: ADD JJNZ LD POP AND RET SELEXP: PUSH LD AND LD POP PUSH LD IN LD AND COR OR OUT	A,10101011B A,01010101B SLPRM2 D,A E B,A AF A,D 11000000B C,A AF AF AF AF D,A A,(PPI.AR) B,A 00111111B C (PPI.AW),A	<pre>;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 0000000 slot #0 ; 01010101 slot #1 ; 1010101 slot #2 ; 1111111 slot #3 ;Extract significant bits ;Set it to [B] ;Expanded slot specified? ;Set sign flag if so ;Save target slot ;Get bit pattern for primary slot ;Extract slot # for 0C000HOFFFFH ;Save it ;Restore target slot ;Save target slot ;Load [D] with specified slot address ;Save current setting ;Cancel current setting for 0C000HOFF</pre>
(MSX RO -SLOT- 530 531 532 533 534 535 536 537 538 539 540 541 542 543 544 543 544 545 545 546 547 545 551 552 553 555 555 556 557 558	0M BASIC 0297 0299 0299 0299 0290 0290 0290 0290 0201 0202 0201 0202 0201 0202 0201 0202 0203 0203 0204 0205 0204 0205 00	3E AB C6 55 10 FC 57 A3 47 F1 A7 F1 A7 C9 F5 7A E6 C0 4F F1 F5 57 DB A8 47 E6 3F B1 D3 A8 7A OF	LD SLPRM2: ADD DJNZ LD ND POP AND DJNZ LD POP PUSH LD AND LD POP PUSH LD IN LD AND DO POP PUSH LD IN IN SELEXP:	A,10101011B A,01010101B SLPRM2 D,A E B,A AF A,D 11000000B C,A AF AF AF AF D,A A,(PPI.AR) B,A 00111111B C (PPI.AW),A	<pre>;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 0000000 slot #0 ; 0101010 slot #1 ; 10101010 slot #2 ; 11111111 slot #3 ;Extract significant bits ;Set it to [B] ;Expanded slot specified? ;Set sign flag if so ;Save target slot ;Get bit pattern for primary slot ;Extract slot # for 0C000H0FFFFH ;Save it ;Restore target slot ;Save target slot ;Load [D] with specified slot address ;Save current setting ;Cancel current setting for 0C000H0FF ;Enable 0C000H0FFFFH or target bank</pre>
(MSX RO -SLOT- 530 531 532 533 534 535 536 537 541 542 543 544 545 544 545 544 545 544 545 547 548 547 548 547 551 552 553 554 555 556 557	0M BASIC 0297 0299 0299 0298 029D 029D 029D 029D 029D 024D 024D 02A1 02A3 02A3 02A3 02A3 02A3 02A3 02A3 02A5 02A7 02A9 02A9 02A9 02A9 02A9 02A3 02A3 02A3 02A3 02A4 02A5 02A9 02A9 02A3 02A3 02A3 02A3 02A4 02A5 02A9 02A9 02A3 02A5 02A9 02A9 02A3 02A3 02A3 02A5 02A9 02A5 02A9 02A3 02A3 02A5 02A9 02A9 02A3 02A3 02A3 02A5 02A9 02A9 02A3 02A5 02A9 02A9 02A9 02A3 02A3 02A3 02A5 02A9 02A9 02A9 02A3 02A5 02A9 02A9 02A9 02A3 02A3 02A3 02A4 02A5 02A9 02A9 02A9 02A9 02A9 02A9 02A9 02A3 02A3 02A3 02A3 02A4 02A5 02A9 02A8 02B8 02	3E AB C6 55 10 FC 57 A3 47 F1 A7 C9 F5 7A E6 C0 4F F1 F5 57 DB A8 47 E6 3F B1 D3 A8 7A	LD SLPRM2: ADD JJNZ LD POP AND RET SELEXP: PUSH LD AND LD POP PUSH LD IN LD AND COR OR OUT	A,10101011B A,01010101B SLPRM2 D,A E B,A AF A,D 11000000B C,A AF AF AF AF D,A A,(PPI.AR) B,A 00111111B C (PPI.AW),A	<pre>;Convert slot # to proper bit pattern ;Save bit pattern for primary slot # ; 0000000 slot #0 ; 0101010 slot #1 ; 10101010 slot #2 ; 11111111 slot #3 ;Extract significant bits ;Set it to [B] ;Expanded slot specified? ;Set sign flag if so ;Save target slot ;Get bit pattern for primary slot ;Extract slot # for 0C000H0FFFFH ;Save it ;Restore target slot ;Save target slot ;Load [D] with specified slot address ;Save current setting ;Cancel current setting for 0C000H0FF ;Enable 0C000H0FFFFH or target bank</pre>

(MSX ROM -SLOT-	BASIC	BIOS) Macro-80)	3.44	01-Jan-85	PAGE	14-2	24
561	02B8	57			LD	D,A			
562	02 B9		AB		LD	A,10101011B	·Conve	rt secondary slot # to proper	
563	02BB			SLEXP1:	20		COnve	rt secondary slot # to proper	
564	02 BB	C6	55		ADD	A,01010101B	•hit n	attern	
565	02BD	15			DEC	D	,DIC P	accern	
566	02BE	F2	02BB		JP	P, SLEXP1	;	00000000 slot #0	
567					••	C / OBDALL I	;	01010101 slot #1	
568							;	10101010 slot #1	
569							;	11111111 slot #3	
570	02C1	A3			AND	Е	•	bit pattern to be added	
571	02C2	57			LD	D,A	; Save		
572	02C3	7B			LD	A,E		bit pattern to strip off old value	
573	02C4	2F			CPL		June	bie pacterin to strip off ofd value	
574	02C5	67			LD	H,A	;Save	thie	
575	02C6	3A	FFFF		LD	A, (OFFFFH)		expanded slot register	
576	02C9	2F			CPL		, neu u	expanded blot register	
577	02CA	6 F			LD	L,A	:Save	current setting	
578	02CB	A4			AND	Н		off old bits	
579	02CC	В2			OR	D		et new bits	
580	02CD	32	FFFF		LD	(OFFFFH),A		econdary slot register	
581	02D0	78			LD	А,В	1000 0	condury afor register	
582	02D1	D3	A8		OUT	(PPI.AW),A	:Resto	ere original primary port	
583	02D3	Fl			POP	AF		pre target slot	
584	02D4	E6	03		AND	00000011B		read from primary slot	
585	02D6	C9			RET			Primary broc	
586				SUBTTL	- MSXIC) - I/O Module			
		BIOS) Macro-80		3.44	01-Jan-85	PAGE		
- MSX10 -	1/0 M						FNGL	15	25
- MSXIO - 587 588	1/0 M						FAGE		23
587	1/0 H					,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	FAGE		23
587 588	1/0 M				,,,,,,		FRG		
587 588 589	1/0 H			,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	,,,,,,	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	FRG		
587 588 589 590	1/0 M			; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	Port d	<pre>;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;</pre>	FROL		
587 588 589 590 591	1/0 M			; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	Port d	,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,,	FROL		
587 588 589 590 591 592	1/0 H			;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	Port d	<pre>iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii</pre>			
587 588 590 591 592 593 594 595				;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	Port d	<pre>iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii</pre>			
587 588 590 591 592 593 594 595 596	0098			;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	Port d IIIIIII VDF EQU	iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii		Read/write data VDP	
587 588 590 591 592 593 594 595 596 597	0098 0099			;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	Port d IIIIIII VDF EQU EQU	<pre> iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii</pre>	lon	Read/write data VDP write command to VDP	
587 588 589 590 591 592 593 594 595 596 597 598	0098			;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	Port d IIIIIII VDF EQU EQU	iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii	Lon ;98H	Read/write data VDP	
587 588 590 591 592 593 594 595 596 597 597 598 599	0098 0099 0099			; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	Port d HIHHH VDF EQU EQU EQU	<pre> iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii</pre>	Lon ;98H ;99H	Read/write data VDP write command to VDP	
587 588 590 591 592 593 594 595 596 597 598 599 600	0098 0099			; ; ; ; ; ; VDP.DRW VDP.CW VDP.SR	Port d HIHHH VDF EQU EQU EQU	<pre> iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii</pre>	ion ;98H ;99H ;99H ;In te	Read/write data VDP write command to VDP read status from VDP xt mode, foreground and background	
587 588 590 591 592 593 594 595 596 597 598 599 600 601	0098 0099 0099			; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	Port d HIHHH VDF EQU EQU EQU	<pre> iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii</pre>	ion ;98H ;99H ;99H ;In te	Read/write data VDP write command to VDP read status from VDP	
587 588 590 591 592 593 594 595 596 597 598 599 600 601 602	0098 0099 0099			;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	Port d IIIIIII VDF EQU EQU EQU EQU	efinition ; ; ; ; ; ; address definit. 10011000B 10011001B 10011001B 10011001B	ion ;98H ;99H ;99H ;In te ;0ther	Read/write data VDP write command to VDP read status from VDP xt mode, foreground and background	
587 588 590 591 592 593 594 595 596 597 598 599 600 601 602 603	0098 0099 0099			;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	Port d IIIIIII VDF EQU EQU EQU EQU	<pre> iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii</pre>	ion ;98H ;99H ;99H ;In te ;0ther	Read/write data VDP write command to VDP read status from VDP xt mode, foreground and background	
587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604	0098 0099 0099 0007			;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	Port d TTTTTT VDF EQU EQU EQU EQU EQU	efinition ; efinition ; innoise definit loolloolb loolloolb loolloolb 7 3 address definit	ion ;98H ;99H ;99H ;In te ;Othern	Read/write data VDP write command to VDP read status from VDP xt mode, foreground and background wise background color	
587 588 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605	0098 0099 0099 0007 0007			;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	Port d TTTTTT VDF EQU EQU EQU EQU EQU PSC EQU	efinition ; ; address definit 10011000B 10011001B 10011001B 7 5 address definit 10100000B	ion ;98H ;99H ;99H ;1n te ;0thern ion ;A0H	Read/write data VDP write command to VDP read status from VDP xt mode, foreground and background wise background color latch address for PSG	
587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606	0098 0099 0099 0007 0007			;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	Port d Port d VDF EQU EQU EQU EQU PSC EQU EQU	<pre></pre>	ion ;98H ;99H ;99H ;In te ;Othern ion ;AOH ;AIH	Read/write data VDP write command to VDP read status from VDP xt mode, foreground and background wise background color latch address for PSG write data to PSG	
587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 605 606	0098 0099 0099 0007 0007			; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ; ;	Port d Port d VDF EQU EQU EQU EQU PSC EQU EQU	efinition ; ; address definit 10011000B 10011001B 10011001B 7 5 address definit 10100000B	ion ;98H ;99H ;99H ;1n te ;0thern ion ;A0H	Read/write data VDP write command to VDP read status from VDP xt mode, foreground and background wise background color latch address for PSG	
587 588 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 605 606	0098 0099 0099 0007 000A0 00A1 00A2			;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	Port d Port d Port d EQU EQU EQU EQU EQU EQU EQU EQU EQU EQU	<pre>efinition ; efinition ; iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii</pre>	ion ;98H ;99H ;99H ;0ther ion ;A0H ;A1H ;A2H	Read/write data VDP write command to VDP read status from VDP xt mode, foreground and background wise background color latch address for PSG write data to PSG read data from PSG	
587 588 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608	0098 0099 0099 0007 00007 00000 0001 0002			;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	Port d Port d Port d EQU EQU EQU EQU EQU EQU EQU EQU EQU EQU	<pre>####################################</pre>	ion ;98H ;99H ;99H ;In te ;0thern ion ;A0H ;A1H ;A2H ;Port	Read/write data VDP write command to VDP read status from VDP xt mode, foreground and background wise background color latch address for PSG write data to PSG read data from PSG A of PSG	
587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 604 605 606 607 608 609 610	0098 0099 0099 0007 000A0 00A1 00A2			;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	Port d Port d Port d EQU EQU EQU EQU EQU EQU EQU EQU EQU EQU	<pre>efinition ; efinition ; iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii</pre>	ion ;98H ;99H ;99H ;In te ;0thern ion ;A0H ;A1H ;A2H ;Port	Read/write data VDP write command to VDP read status from VDP xt mode, foreground and background wise background color latch address for PSG write data to PSG read data from PSG	
587 588 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 605 606 607 608 609 610 611	0098 0099 0099 0007 00007 00000 0001 0002			;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	Port d Port d Port d EQU EQU EQU EQU EQU EQU EQU EQU EQU EQU	<pre>efinition ; efinition ; address definit 10011000B 10011001B 10011001B 7 5 address definit 1010000B 1010000B 1010001B 1010001B 1010001B 1010001B 1010001B 1010001B 10100010B 14 15</pre>	ion ;98H ;99H ;0ther ;0ther ;0ther ;0ther ;0ther ;0ther ;0ther ;99H ;0ther ;0ther ;0ther ;0ther ;0ther ;00H ;01H ;01H ;01H ;02H ;03H ;03H ;03H ;03H ;03H ;03H ;03H ;03	Read/write data VDP write command to VDP read status from VDP xt mode, foreground and background wise background color latch address for PSG write data to PSG read data from PSG A of PSG	
587 588 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612	0098 0099 0099 0007 00007 00000 0001 0002			;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	Port d Port d Port d EQU EQU EQU EQU EQU EQU EQU EQU EQU EQU	<pre>####################################</pre>	ion ;98H ;99H ;0ther ;0ther ;0ther ;0ther ;0ther ;0ther ;0ther ;99H ;0ther ;0ther ;0ther ;0ther ;0ther ;00H ;01H ;01H ;01H ;02H ;03H ;03H ;03H ;03H ;03H ;03H ;03H ;03	Read/write data VDP write command to VDP read status from VDP xt mode, foreground and background wise background color latch address for PSG write data to PSG read data from PSG A of PSG	
587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 607 608 609 611 612 613	0098 0099 0007 0007 0000 0001 0002 000E 000F			;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	Port d Port d Port d EQU EQU EQU EQU EQU EQU EQU EQU EQU EQU	<pre> if inition if in</pre>	ion ;98H ;99H ;1n te ;0thern ion ;A0H ;A1H ;A2H ;Port	Read/write data VDP write command to VDP read status from VDP xt mode, foreground and background wise background color latch address for PSG write data to PSG read data from PSG A of PSG B of PSG	
587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 609 610 611 612 613 614	0098 0099 0007 0007 00007 0001 0002 000E 000F			;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	Port d Fort d Fort d Fort d Fort EQU EQU EQU EQU EQU EQU EQU EQU	<pre> iiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiiii</pre>	ion ;98H ;99H ;99H ;In te ;Other ion ;A0H ;A1H ;A2H ;Port ion ;A8H	Read/write data VDP write command to VDP read status from VDP xt mode, foreground and background wise background color latch address for PSG write data to PSG read data from PSG A of PSG B of PSG read from PPI Port A	
587 588 589 590 591 592 593 594 595 596 597 598 599 600 601 602 603 604 605 606 607 608 607 608 609 611 612 613	0098 0099 0007 0007 0000 0001 0002 000E 000F			;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	Port d Port d Port d EQU EQU EQU EQU EQU EQU EQU EQU EQU EQU	<pre> if inition if in</pre>	ion ;98H ;99H ;1n te ;0thern ion ;A0H ;A1H ;A2H ;Port	Read/write data VDP write command to VDP read status from VDP xt mode, foreground and background wise background color latch address for PSG write data to PSG read data from PSG A of PSG B of PSG	

- MSXIO	M BASIC BIOS) - I/O Module	Macro-80	3.44	01-Jan-85	PAGE	15-1	
618	00AA	PPI.CW	FOU	10101010B	; AAH	write to PPI Port C	
619	00AB	PPI.CM		10101011B	; ABH	write to PPI command register	
620		;	520	TOTOTOTID	1 4 6 1	write to PPI command register	
621		;	Pri	nter port defir	nition		
622		,		acer port dern	in cron		
623	0091	LPT.DW	EQU	10010001B	;Data	port	
624	0090	LPT.SB	EQU	10010000B			
625	0090	LPT.ST		10010000B		e output er status	
626	0000	;	ЕQU	TOOTOOOB	; Print	er status	
627		,	Ton	t mode (40*24)			
628		;	Tex	tt mode (40*24)		SCREEN 0	
629		,		TXTNAM, TXTCO	CD.		
630		;		INTERNET, INTO	GP		
631		;	Tov	t mode (graphic	aa 1)	SCREEN 1	
632		i	167	te moue (graphit		SCREEN 1	
633		,		M333334 M333	or		
634				T32NAM, T32CC	OL, TSZCGP,	T32ATR, T32PAT	
635		;		es mode		0000000	
636			nir	es mode		SCREEN 2	
637		:		CDDNAM CDDG		(1) 1) (1) (1) (1) (1) (1) (1) (1) (1) (
638		;		GRPNAM, GRPCC	OL, GRPCGP,	GRPATR, GRPPAT	
639		1	M-1	ti-color mode			
640		;	mui	ci-color mode		SCREEN 3	
641		;		MT 000000 107 0000			
642		;		MLTNAM, MLTCO	GP, MLTATR,	MLTPAT	
643		;	Cor	een size			
644		•	SCI	een size			
645		į		I THE DUL COMO			
646		;		LINLEN, CRTC	NT,LINL32,	LINLAU	
647		;	Buch		_		
648		;	EXC.	ernal constants	8		
	M BASIC BIOS) - I/O Module	Macro-80	3.44	01-Jan-85	PAGE	15-2	
- MSXIO ·	M BASIC BIOS) - I/O Module		3.44				
- MSXIO - 649		;	3.44	01-Jan-85 CGTABL		15-2 ter generator table	
- MSXIO - 649 650		;		CGTABL	Charac		
- MSXIO - 649 650 651		; ;			Charac		
- MSXIO - 649 650 651 652		;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;		CGTABL ernal variables	Charac	ter generator table	
- MSXIO - 649 650 651 652 653		; ; ; ;		CGTABL ernal variables FORCLR	Charac s Foregr	ter generator table ound color	
- MSXIO - 649 650 651 652 653 654		; ; ; ;		CGTABL ernal variables FORCLR BAKCLR	Charac s Foregr Backgr	ter generator table ound color ound color	
- MSXIO - 649 650 651 652 653 654 655		; ; ; ;		CGTABL ernal variables FORCLR BAKCLR BDRCLR	Charac s Foregr Backgr Border	ound color ound color ound color color for PAINT	
- MSXIO - 649 650 651 652 653 654 655 656		; ; ; ;		CGTABL ernal variables FORCLR BAKCLR	Charac s Foregr Backgr Border Curren	ound color ound color color for PAINT t screen mode	
- MSXIO - 649 650 651 652 653 654 655 656 655		; ; ; ;		CGTABL ernal variables FORCLR BAKCLR BDRCLR	Charac s Foregr Backgr Border Curren ;	ound color ound color color for PAINT t screen mode 0 - 40*24 text	
- MSXIO - 649 650 651 652 653 654 655 655 657 658		; ; ; ;		CGTABL ernal variables FORCLR BAKCLR BDRCLR	Charac s Foregr Backgr Border Curren ;	ound color ound color color for PAINT t screen mode 0 - 40*24 text 1 - 32*24 text	
- MSXIO - 649 650 651 652 653 654 655 656 657 658 659		; ; ; ;		CGTABL ernal variables FORCLR BAKCLR BDRCLR	Charac s For egr Back gr Bor der Curr en ; ; ;	ound color ound color color for PAINT t screen mode 0 - 40*24 text 1 - 32*24 text 2 - hiresolution graphics	
- MSXIO - 649 650 651 652 653 654 655 656 657 658 659 660		; ; ; ; ;		CGTABL ernal variables FORCLR BAKCLR BDRCLR SCRMOD	Charac s Foregr Backgr Border Curren ;	ound color ound color color for PAINT t screen mode 0 - 40*24 text 1 - 32*24 text	
- MSXIO 649 650 651 652 653 654 655 656 657 658 657 658 659 660 661		;;		CGTABL ernal variables FORCLR BAKCLR BORCLR SCRMOD OLDSCR	Charac Foregr Backgr Border Curren ; ; ;	ound color ound color color for PAINT t screen mode 0 - 40*24 text 1 - 32*24 text 2 - hiresolution graphics 3 - Multicolor graphics	
- MSXIO - 649 650 651 652 653 654 655 656 657 658 659 660 661 662		;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;		CGTABL PORCLR BARCLR BORCLR SCRMOD OLDSCR NAMBAS	Charac s Foregr Backgr Border Curren ; ; ; ; ; Base c	ound color ound color color for PAINT t screen mode 0 - 40*24 text 1 - 32*24 text 2 - hiresolution graphics 3 - Multicolor graphics	
- MSXIO - 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663		;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;		CGTABL ernal variables FORCLR BAKCLR BDRCLR SCRMOD OLDSCR NAMBAS CGPBAS	Charac Foregr Backgr Border Curren ; ; ; ; Base c Base c	ound color ound color color for PAINT t screen mode 0 - 40*24 text 1 - 32*24 text 2 - hiresolution graphics 3 - Multicolor graphics of current name table f current cgen table	
- MSXIO - 649 650 651 652 653 654 655 655 656 657 658 657 658 659 660 661 662 663 664		;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;		CGTABL ernal variables FORCLR BAKCLR BDRCLR SCRMOD OLDSCR NAMBAS CGPBAS PATBAS	Charac Foregr Backgr Border Curren ; ; ; Base c Base c Base c	ound color ound color color for PAINT t screen mode 0 - 40*24 text 1 - 32*24 text 2 - hiresolution graphics 3 - Multicolor graphics 6f current name table of current sprite pattern table	
- MSXIO - 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665		;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;		CGTABL FORCLR BAKCLR BORCLR SCRMOD OLDSCR NAMBAS CGPBAS PATBAS ATRBAS	Charac s Foregr Backgr Border Curren ; ; ; Base c Base c Base c Base c	ound color ound color color for PAINT t screen mode 0 - 40*24 text 1 - 32*24 text 2 - hiresolution graphics 3 - Multicolor graphics of current name table of current cgen table of current sprite pattern table of current sprite attribute tabl	Le
- MSXIO 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666		;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;		CGTABL ernal variables FORCLR BAKCLR BDRCLR SCRMOD OLDSCR NAMBAS CGPBAS PATBAS ATRBAS JIFFY	Charac Foregr Backgr Border ; ; ; Base c Base c Base c Base c Jiffy	ound color ound color color for PAINT t screen mode 0 - 40*24 text 1 - 32*24 text 2 - hiresolution graphics 3 - Multicolor graphics of current name table of current sprite pattern table of current sprite pattern table of current sprite attribute tabl count	Le
- MSXIO 649 650 651 652 653 655 655 655 655 657 660 661 662 663 664 665 665 666 667		;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;		CGTABL ernal variables FORCLR BAKCLR BDRCLR SCRMOD OLDSCR NAMBAS CGPBAS PATBAS ATRBAS JIFFY CLIKSW	Charac s Foregr Backgr Border Currer ; ; ; Base c Base c Base c Base c Jiffy Click	ound color ound color color for PAINT t screen mode 0 - 40*24 text 1 - 32*24 text 2 - hiresolution graphics 3 - Multicolor graphics of current name table of current sprite pattern table of current sprite attribute tabl count switch	
- MSXIO 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 665 666 667 668		;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;		CGTABL FORCLR BAKCLR BORCLR SCRMOD OLDSCR NAMBAS CGPBAS PATBAS ATRBAS JIFFY CLIKSW CLIKFL	Charac s Foregr Backgr Backgr Curren ; ; ; Base c Base c Base c Base c Jiffy Click Click	ound color ound color color for PAINT t screen mode 0 - 40*24 text 1 - 32*24 text 2 - hiresolution graphics 3 - Multicolor graphics of current name table of current cgen table of current sprite pattern table of current sprite attribute tabl count switch flag to suppress multiple key of	
- MSXIO 649 650 651 652 653 654 655 656 659 660 661 662 663 664 665 666 667 668 669		;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;		CGTABL ernal variables FORCLR BARCLR BDRCLR SCRMOD OLDSCR NAMBAS CGPBAS PATBAS JIFFY CLIKSW CLIKFL RGOSAV	Charac s Foregr Backgr Border ; ; ; Base c Base c Base c Base c Jiffy Click Click VDP re	ound color ound color color for PAINT t screen mode 0 - 40*24 text 1 - 32*24 text 2 - hiresolution graphics 3 - Multicolor graphics of current name table of current sprite pattern table of current sprite pattern table for urrent sprite attribute tabl count switch flag to suppress multiple key of gister #0 save area	
- MSXIO 649 650 651 652 653 655 655 655 655 655 657 658 659 660 661 662 663 664 663 664 665 666 667 668 669 670		;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;		CGTABL ernal variables FORCLR BAKCLR BDRCLR SCRMOD OLDSCR NAMBAS CGPBAS PATBAS ATRBAS JIFFY CLIKSW CLIKFL RGOSAV RGISAV	Charac Foregr Backgr Border ; ; ; Base c Base c Base c Base c Jiffy Click Click VDP re	ound color ound color color for PAINT t screen mode 0 - 40*24 text 1 - 32*24 text 2 - hiresolution graphics 3 - Multicolor graphics of current name table of current sprite pattern table of current sprite pattern table of current sprite attribute tabl count switch flag to suppress multiple key of egister #0 save area gister #1 save area	
- MSXIO 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 671		;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;		CGTABL FORCLR BAKCLR BORCLR SCRMOD OLDSCR NAMBAS CGPBAS PATBAS ATRBAS JIFFY CLIKSW CLIKFL RGOSAV RGISAV STATFL	Characo s Foregr Backgr Backgr Curren ; ; Base c Base c Base c Base c Jiffy Click Click VDP re VDP st	ound color ound color color for PAINT t screen mode 0 - 40*24 text 1 - 32*24 text 2 - hiresolution graphics 3 - Multicolor graphics of current name table of current sprite pattern table of current sprite pattern table f current sprite attribute tabl count switch flag to suppress multiple key of gister #0 save area actus register	
- MSXIO 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 669 670 672		;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;		CGTABL ernal variables FORCLR BAKCLR BDRCLR SCRMOD OLDSCR NAMBAS CGPBAS PATBAS ATRBAS JIFFY CLIKSW CLIKFL RGOSAV RGISAV	Characo s Foregr Backgr Backgr Curren ; ; Base c Base c Base c Base c Jiffy Click Click VDP re VDP st	ound color ound color color for PAINT t screen mode 0 - 40*24 text 1 - 32*24 text 2 - hiresolution graphics 3 - Multicolor graphics of current name table of current sprite pattern table of current sprite pattern table of current sprite attribute tabl count switch flag to suppress multiple key of egister #0 save area gister #1 save area	
- MSXIO 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 663 664 665 666 665 666 667 668 669 670 671 672 673		;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	Ext	CGTABL ernal variables FORCLR BAKCLR BAKCLR SCRMOD OLDSCR NAMBAS CGPBAS PATBAS ATRBAS JIFFY CLIKSW CLIKFL RGOSAV RGISAV STATFL PATWRK	Charac s Foregr Backgr Curren ; ; ; Base c Base c Base c Base c Jiffy Click VDP re VDP re VDP st Work a	ound color ound color color for PAINT t screen mode 0 - 40*24 text 1 - 32*24 text 2 - hiresolution graphics 3 - Multicolor graphics of current name table of current sprite pattern table of current sprite pattern table f current sprite attribute tabl count switch flag to suppress multiple key of gister #0 save area actus register	
- MSXIO 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 668 666 667 668 667 671 672 673 674		;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	Ext	CGTABL FORCLR BAKCLR BORCLR SCRMOD OLDSCR NAMBAS CGPBAS PATBAS ATRBAS JIFFY CLIKSW CLIKFL RGOSAV RGISAV STATFL	Charac s Foregr Backgr Curren ; ; ; Base c Base c Base c Base c Jiffy Click VDP re VDP re VDP st Work a	ound color ound color color for PAINT t screen mode 0 - 40*24 text 1 - 32*24 text 2 - hiresolution graphics 3 - Multicolor graphics of current name table of current sprite pattern table of current sprite pattern table f current sprite attribute tabl count switch flag to suppress multiple key of gister #0 save area actus register	
- MSXIO 649 650 651 652 653 654 655 656 657 658 669 661 662 663 664 665 666 667 668 669 671 672 673 674 675		;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	Ext	CGTABL ernal variables FORCLR BARCLR BDRCLR SCRMOD OLDSCR NAMBAS CGPBAS PATBAS ATRBAS JIFFY CLIKSW CLIKFL RG0SAV RG1SAV RG1SAV STATFL PATWRK	Charac s Foregr Backgr Curren ; ; ; Base c Base c Base c Base c Jiffy Click VDP re VDP re VDP st Work a	ound color ound color color for PAINT t screen mode 0 - 40*24 text 1 - 32*24 text 2 - hiresolution graphics 3 - Multicolor graphics of current name table of current sprite pattern table of current sprite pattern table f current sprite attribute tabl count switch flag to suppress multiple key of gister #0 save area actus register	
- MSXIO 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 666 667 666 667 668 669 670 671 672 673 675 676		;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	Ext	CGTABL ernal variables FORCLR BAKCLR BAKCLR SCRMOD OLDSCR NAMBAS CGPBAS PATBAS CGPBAS PATBAS ATRBAS JIFFY CLIKSW CLIKFL RGGSAV RGISAV STATFL PATWRK ternal routines GETQ	Charac s Foregr Backgr Curren ; ; ; Base c Base c Base c Base c Jiffy Click VDP re VDP re VDP st Work a	ound color ound color color for PAINT t screen mode 0 - 40*24 text 1 - 32*24 text 2 - hiresolution graphics 3 - Multicolor graphics of current name table of current sprite pattern table of current sprite pattern table f current sprite attribute tabl count switch flag to suppress multiple key of gister #0 save area actus register	
- MSXIO 649 650 651 652 653 654 655 656 657 658 659 660 661 662 663 664 665 664 665 666 667 668 669 671 672 673 674 675		;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;;	Ext	CGTABL ernal variables FORCLR BARCLR BDRCLR SCRMOD OLDSCR NAMBAS CGPBAS PATBAS ATRBAS JIFFY CLIKSW CLIKFL RG0SAV RG1SAV RG1SAV STATFL PATWRK	Charac s Foregr Backgr Curren ; ; ; Base c Base c Base c Base c Jiffy Click VDP re VDP re VDP st Work a	ound color ound color color for PAINT t screen mode 0 - 40*24 text 1 - 32*24 text 2 - hiresolution graphics 3 - Multicolor graphics of current name table of current sprite pattern table of current sprite pattern table f current sprite attribute tabl count switch flag to suppress multiple key of gister #0 save area actus register	

- MSXIO		BIOS) Macro- available RAM	-80 3.44	01-Jan-85	PAGE 16	28
680						
681	02D7		CHKRAM:			
682			;		aux Dhu	
683 684			•		CHKRAM	-
			; 			
685					FFFFH to C000H, and set system work	
686 687					se RAM as work area nor perform	
688					do not yet know where the available	
689				til the RAM is f	to be done inside ROM and CPU's	
690			; register und	CII GIE NAM 15, 1	.ouna.	
691	02D7	3E 82	, LD	A,82H	;Port A - output (mode 0)	
692	02D9	D3 AB	-OUT	(PPI.CM),A	;Port B - input (mode 0)	
693	02DB	AF	XOR	A	;Port C - output (mode 0)	
694	02DC	D3 A8	OUT	(PPI.AW),A	;Select slot 0 for all addresses	
695	02DE	3E 50	LD	A, 'P'	;Disable all cassette related outputs	3
696	02E0	D3 AA	OUT	(PPI.CW),A	;Motor off	
697			;			
698			; Start search	hing		
699			;			
700			; Register usa			
701					cking secondary slot	
702				# of the biggest		
703					the biggest RAM block (if any) biggest RAM block ever found	
705				xpanded' flag	biggest with block ever found	
706			1 2100 0	npunded frug		
707			; 0000xxxx			
708			; [[]]			
709			; +- s	lot #3 expanded		
710			; + s	lot #2 expanded		
		BIOS) Macro	-80 3.44	01-Jan-85	PAGE 16-1	29
		BIOS) Macro available RAM	-80 3.44	-	PAGE 16-1	29
- MSXIC 711			-80 3.44 ; [+ s]	01-Jan-85 lot #1 expanded	PAGE 16-1	29
- MSXIC 711 712			-80 3.44 ; [+ s] ; + s]	01-Jan-85	PAGE 16-1	29
- MSXIC 711 712 713) - Find	available RAM	80 3.44 ; [+ s] ; + s]	01-Jan-85 lot #1 expanded lot #0 expanded		29
- MSXIC 711 712 713 714) - Find 02E2	available RAM 11 FFFF	80 3.44 ; + sl ; + sl ; LD	01-Jan-85 lot #1 expanded lot #0 expanded DE,0FFFFH	;Initialize l <i>o</i> west address ever found	29
- MSXIC 711 712 713 714 715	0 - Find 02E2 02E5	available RAM 11 FFFF AF	-80 3.44 ; + s1 ; + s1 ; LD XOR	01-Jan-85 lot #1 expanded lot #0 expanded DE,0FFFFH A	;Initialize lowest address ever found ;Start from slot #0	29
- MSXIC 711 712 713 714) - Find 02E2	available RAM 11 FFFF	80 3.44 ; + sl ; + sl ; LD	01-Jan-85 lot #1 expanded lot #0 expanded DE,0FFFFH	;Initialize l <i>o</i> west address ever found	29
- MSXIC 711 712 713 714 715 716	0 - Find 02E2 02E5 02E6	available RAM 11 FFFF AF	-80 3.44 ; + s1 ; + s1 ; LD XOR LD	01-Jan-85 lot #1 expanded lot #0 expanded DE,0FFFFH A	;Initialize lowest address ever found ;Start from slot #0	29
- MSXIC 711 712 713 714 715 716 717 718 719	0 - Find 02E2 02E5 02E6 02E7 02E7 02E7 02E9	available RAM ll FFFF AF 4F DJ A6 CB 21	-80 3.44 ; + s1 ; + s1 ; LD XOR LD CKRM05; OUT SLA	01-Jan-85 lot #1 expanded lot #0 expanded DE,OFFFFH A C,A (PPI.AW),A C	;Initialize lowest address ever found ;Start from slot #0 ;Clear bit pattern ;Select the slot ;Shift bit pattern	29
- MSXIC 711 712 713 714 715 716 717 718 719 719 720	0 - Find 02E2 02E5 02E6 02E7 02E7 02E9 02E9 02EB	available RAM 11 FFFF AF 4F D3 A8 CB 21 06 00	80 3.44 ; + sl ; + sl ; LD XOR LD CKRM05; OUT SLA LD	01-Jan-85 lot #1 expanded lot #0 expanded DE,OFFFFH A C,A (PPI.AW),A C B,0	;Initialize lowest address ever found ;Start from slot #0 ;Clear bit pattern ;Select the slot ;Shift bit pattern ;Assume this slot is not expanded	
- MSXIC 711 712 713 714 715 716 717 718 719 720 721	0 - Find 02E2 02E5 02E6 02E7 02E7 02E9 02EB 02ED	available RAM 11 FFFF AF 4F D3 A8 CB 21 06 00 21 FFFF	-80 3.44 ; + sl ; + sl ; LD CKRM05: OUT SLA LD LD	01-Jan-85 lot #1 expanded lot #0 expanded DE,0FFFFH A C,A (PPI.AW),A C B,0 HL,0FFFFH	;Initialize lowest address ever found ;Start from slot ∉0 ;Clear bit pattern ;Select the slot ;Shift bit pattern ;Assume this slot is not expanded ;Read from possible expansion slot reg	
- MSXIC 711 712 713 714 715 716 717 718 719 720 721 722	0 - Find 02E2 02E5 02E6 02E7 02E7 02E9 02E9 02E0 02E0	available RAM 11 FFFF AF 4F D3 A8 CB 21 06 00 21 FFFF 36 F0	-80 3.44 ; + sl ; + sl ; LD XOR LD CKRM05: OUT SLA LD LD LD	01-Jan-85 lot #1 expanded lot #0 expanded DE,OFFFFH A C,A (PPI.AW),A C B,O HL,OFFFFH (HL),OFFH	;Initialize lowest address ever found ;Start from slot #0 ;Clear bit pattern ;Select the slot ;Shift bit pattern ;Assume this slot is not expanded	
- MSXIC 711 712 713 714 715 716 719 719 719 720 721 722 723	0 - Find 02E2 02E5 02E6 02E7 02E7 02E9 02EB 02EB 02ED 02F0 02F0 02F2	available RAM 11 FFFF AF 4F D3 A8 CB 21 06 00 21 FFFF 36 F0 7E	80 3.44 ; + sl ; + sl ; LD XOR LD CKRM05; OUT SLA LD LD LD LD	01-Jan-85 lot #1 expanded lot #0 expanded DE,OFFFFH A C,A (PPI.AW),A C B,O HL,OFFFFH (HL),OFOFH A,(HL)	;Initialize lowest address ever found ;Start from slot #0 ;Clear bit pattern ;Select the slot ;Shift bit pattern ;Assume this slot is not expanded ;Read from possible expansion slot reg ;Write a binary lll0000	
- MSXIC 711 712 713 714 715 716 717 718 719 720 721 722 723 724	0 - Find 02E2 02E5 02E6 02E7 02E7 02E9 02E9 02E9 02E0 02E0 02F0 02F2 02F3	available RAM 11 FFFF AF 4F D3 A8 CB 21 06 00 21 FFFF 36 F0 7E D6 0F	-80 3.44 ; + sl ; + sl ; LD CKRM05: OUT SLA LD LD LD LD LD SUB	01-Jan-85 lot #1 expanded lot #0 expanded DE,OFFFFH A C,A (PPI.AW),A C B,0 HL,OFFFFH (HL),OFOH A,(HL) OFH	<pre>;Initialize lowest address ever found ;Start from slot #0 ;Clear bit pattern ;Select the slot ;Shift bit pattern ;Assume this slot is not expanded ;Read from possible expansion slot reg ;Write a binary llll0000 ;Read back as 00001111?</pre>	
- MSXIC 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725	0 - Find 02E2 02E5 02E6 02E7 02E9 02E9 02E9 02E0 02F0 02F2 02F3 02F5	available RAM 11 FFFF AF 4F D3 A8 CB 21 06 00 21 FFFF 36 F0 7E D6 0F 20 0B	-80 3.44 ; + sl ; + sl ; LD XOR LD CKRM05: OUT SLA LD LD LD LD LD SUB JR	01-Jan-85 lot #1 expanded lot #0 expanded DE,OFFFFH A C,A (PPI.AW),A C B,O HL,OFFFFH (HL),OFFH (HL),OFFH A,(HL) OFH NZ,CKRM15	<pre>;Initialize lowest address ever found ;Start from slot #0 ;Clear bit pattern ;Select the slot ;Shift bit pattern ;Assume this slot is not expanded ;Read from possible expansion slot reg ;Write a binary llll0000 ;Read back as 0000llll? ;Nop, this is not an expanded slot</pre>	
- MSXIC 711 712 713 714 715 716 719 719 720 721 722 723 724 724 725 726	 Find 02E2 02E5 02E6 02E7 02E9 02E0 02F0 02F1 02F3 02F5 02F7 	available RAM 11 FFFF AF 4F D3 A8 CB 21 06 00 21 FFFF 36 F0 77	80 3.44 ; + s] ; s] ; LD XOR LD CKRM05; OUT SLA LD LD LD LD SUB JR LD	01-Jan-85 lot #1 expanded lot #0 expanded DE,OFFFFH A C,A (PPI.AW),A C B,O HL,OFFFFH (HL),OFOH A,(HL) OFH NZ,CKRM15 (HL),A	<pre>;Initialize lowest address ever found ;Start from slot #0 ;Clear bit pattern ;Select the slot ;Shift bit pattern ;Assume this slot is not expanded ;Read from possible expansion slot reg ;Write a binary llll0000 ;Read back as 00001111?</pre>	
- MSXIC 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 726 727	 Find 02E2 02E5 02E6 02E7 02E9 02EB 02ED 02F0 02F5 02F5 02F5 02F7 02F8 	available RAM 11 FFFF AF 4F D3 A8 CB 21 06 00 21 FFFF 36 F0 76 F0 76 D6 0F 20 0B 77 7E	-80 3.44 ; + sl ; + sl ; LD XOR LD CKRM05: OUT SLA LD LD LD LD LD SUB JR	01-Jan-85 lot #1 expanded lot #0 expanded DE,OFFFFH A C,A (PPI.AW),A C B,O HL,OFFFFH (HL),OFFH (HL),OFFH A,(HL) OFH NZ,CKRM15	<pre>;Initialize lowest address ever found ;Start from slot #0 ;Clear bit pattern ;Select the slot ;Shift bit pattern ;Assume this slot is not expanded ;Read from possible expansion slot reg ;Write a binary llll0000 ;Read back as 00001lll? ;Nop, this is not an expanded slot ;Write 0000000</pre>	
- MSXIC 711 712 713 714 715 716 719 719 720 721 722 723 724 724 725 726	 Find 02E2 02E5 02E6 02E7 02E9 02E0 02F0 02F1 02F3 02F5 02F7 	available RAM 11 FFFF AF 4F D3 A8 CB 21 06 00 21 FFFF 36 F0 77	-80 3.44 ; + sl ; + sl ; LD CKRM05: OUT SLA LD LD LD LD SUB JR LD LD LD	01-Jan-85 lot #1 expanded lot #0 expanded DE,OFFFFH A C,A (PPI.AW),A C B,O HL,OFFFFH (HL),OFOH A,(HL) OFH NZ,CKRM15 (HL),A A,(HL) A	<pre>;Initialize lowest address ever found ;Start from slot #0 ;Clear bit pattern ;Select the slot ;Shift bit pattern ;Assume this slot is not expanded ;Read from possible expansion slot reg ;Write a binary 11110000 ;Read back as 00001117 ;Nop, this is not an expanded slot ;Write 00000000 ;Read back as 11111117</pre>	
- MSXIC 711 712 713 714 715 716 717 718 719 720 721 722 723 723 724 725 726 727 728	 Find 02E2 02E5 02E6 02E7 02E9 02E0 02F0 02F2 02F3 02F5 02F7 02F8 02F9 02F8 02F9 	available RAM 11 FFFF AF 4F D3 A8 CB 21 06 00 21 FFFF 36 F0 7E D6 0F 20 0B 77 7E 3C	-80 3.44 ; + sl ; + sl ; LD XOR LD CKRM05: OUT SLA LD LD LD LD SUB JR LD LD LD LD LD LD LD LD LD LD	01-Jan-85 lot #1 expanded lot #0 expanded DE,OFFFFH A C,A (PPI.AW),A C B,O HL,OFFFFH (HL),OFFFH (HL),OFH NZ,CKRM15 (HL),A A,(HL)	<pre>;Initialize lowest address ever found ;Start from slot #0 ;Clear bit pattern ;Select the slot ;Shift bit pattern ;Assume this slot is not expanded ;Read from possible expansion slot reg ;Write a binary llll0000 ;Read back as 00001lll? ;Nop, this is not an expanded slot ;Write 0000000</pre>	
- MSXIC 711 712 713 714 715 716 719 719 720 721 722 723 724 724 725 726 727 728 729	 Find 02E2 02E5 02E6 02E7 02E9 02E9 02E0 02F0 02F2 02F3 02F5 02F7 02F8 02F9 02F8 02F9 02F4 	available RAM 11 FFFF AF 4F D3 A8 CB 21 06 00 21 FFFF 36 F0 77 D6 0F 20 0B 77 7E 3C 20 06	-80 3.44 ; [+ s] ; s] ; LD XOR LD CKRM05: OUT SLA LD LD LD LD LD LD LD LD LD LD LD LD LD	01-Jan-85 lot #1 expanded lot #0 expanded DE,OFFFFH A C,A (PPI.AW),A C B,O HL,OFFFFH (HL),OFOH A,(HL) OFH NZ,CKRM15 (HL),A A,(HL) A NZ,CKRM15	<pre>;Initialize lowest address ever found ;Start from slot #0 ;Clear bit pattern ;Select the slot ;Shift bit pattern ;Assume this slot is not expanded ;Read from possible expansion slot reg ;Write a binary llll0000 ;Read back as 0000llll? ;Nop, this is not an expanded slot ;Write 00000000 ;Read back as lllllll? ;Nop, not expanded slot</pre>	
- MSXIC 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732	 Find 02E2 02E5 02E6 02E7 02E9 02E8 02E0 02F0 02F2 02F3 02F5 02F7 02F8 02F9 02F8 02F9 02F4 02F5 	available RAM 11 FFFF AF 47 D3 A8 CB 21 06 00 21 FFFF 36 F0 7E D6 0F 20 0B 77 7E 3C 20 06 04	-80 3.44 ; + sl ; + sl ; LD CKRM05: OUT SLA LD LD LD LD LD LD LD LD LD LD LD LD LD	01-Jan-85 lot #1 expanded lot #0 expanded DE,OFFFFH A C,A (PPI.AW),A C B,0 HL,OFFFFH (HL),A OFH NZ,CKRM15 (HL) A NZ,CKRM15 B	<pre>;Initialize lowest address ever found ;Start from slot #0 ;Clear bit pattern ;Select the slot ;Shift bit pattern ;Assume this slot is not expanded ;Read from possible expansion slot reg ;Write a binary 11110000 ;Read back as 0000111? ;Nop, this is not an expanded slot ;Write 00000000 ;Read back as 1111111? ;Nop, not expanded slot ;We're checking expanded slot</pre>	
- MSXIC 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 725 726 727 728 729 729 720 721 722 723 724 725 726 727 728 729 730	 Find 02E2 02E5 02E6 02E7 02E9 02E8 02E10 02F0 02F3 02F5 02F7 02F8 02F9 02F4 02F2 02F3 02F4 <l< td=""><td>available RAM 11 FFFF AF 47 D3 A8 CB 21 06 00 21 FFFF 36 F0 7E D6 0F 20 0B 77 7E 3C 20 06 04</td><td>-80 3.44 ; + sl ; + sl ; LD XOR LD CKRM05: OUT SLA LD LD LD LD LD LD LD LD LD SUB JR LD LD LD SUB JR INC SET CKRM10: ;</td><td>01-Jan-85 lot #1 expanded lot #0 expanded DE,OFFFFH A C,A (PPI.AW),A C B,O HL,OFFFFH (HL),A OFH NZ,CKRM15 (HL),A A,(HL) A NZ,CKRM15 B 0,C</td><td><pre>;Initialize lowest address ever found ;Start from slot #0 ;Clear bit pattern ;Select the slot ;Shift bit pattern ;Assume this slot is not expanded ;Read from possible expansion slot reg ;Write a binary 11110000 ;Read back as 00001111? ;NOp, this is not an expanded slot ;Write 00000000 ;Read back as 1111111? ;NOp, not expanded slot ;We're checking expanded slot ;Say this slot is expanded</pre></td><td></td></l<>	available RAM 11 FFFF AF 47 D3 A8 CB 21 06 00 21 FFFF 36 F0 7E D6 0F 20 0B 77 7E 3C 20 06 04	-80 3.44 ; + sl ; + sl ; LD XOR LD CKRM05: OUT SLA LD LD LD LD LD LD LD LD LD SUB JR LD LD LD SUB JR INC SET CKRM10: ;	01-Jan-85 lot #1 expanded lot #0 expanded DE,OFFFFH A C,A (PPI.AW),A C B,O HL,OFFFFH (HL),A OFH NZ,CKRM15 (HL),A A,(HL) A NZ,CKRM15 B 0,C	<pre>;Initialize lowest address ever found ;Start from slot #0 ;Clear bit pattern ;Select the slot ;Shift bit pattern ;Assume this slot is not expanded ;Read from possible expansion slot reg ;Write a binary 11110000 ;Read back as 00001111? ;NOp, this is not an expanded slot ;Write 00000000 ;Read back as 1111111? ;NOp, not expanded slot ;We're checking expanded slot ;Say this slot is expanded</pre>	
- MSXIC 711 712 713 714 715 716 717 718 719 720 721 722 723 723 724 725 726 726 727 728 729 730 731 732 733	 Find 02E2 02E5 02E6 02E7 02E9 02E8 02E10 02F0 02F3 02F5 02F7 02F8 02F9 02F4 02F2 02F3 02F4 <l< td=""><td>available RAM 11 FFFF AF 47 D3 A8 CB 21 06 00 21 FFFF 36 F0 7E D6 0F 20 0B 77 7E 3C 20 06 04</td><td>-80 3.44 ; [+ s] ; s] ; LD XOR LD CKRM05: OUT SLA LD LD LD LD LD LD LD LD LD SUB JR LD LD LD LD SUB JR INC SET CKRM10; ;</td><td>01-Jan-85 lot #1 expanded lot #0 expanded DE,OFFFFH A C,A (PPI.AW),A C B,0 HL,OFFFFH (HL),A OFH NZ,CKRM15 (HL) A NZ,CKRM15 B</td><td><pre>;Initialize lowest address ever found ;Start from slot #0 ;Clear bit pattern ;Select the slot ;Shift bit pattern ;Assume this slot is not expanded ;Read from possible expansion slot reg ;Write a binary 11110000 ;Read back as 00001111? ;NOp, this is not an expanded slot ;Write 00000000 ;Read back as 1111111? ;NOp, not expanded slot ;We're checking expanded slot ;Say this slot is expanded</pre></td><td></td></l<>	available RAM 11 FFFF AF 47 D3 A8 CB 21 06 00 21 FFFF 36 F0 7E D6 0F 20 0B 77 7E 3C 20 06 04	-80 3.44 ; [+ s] ; s] ; LD XOR LD CKRM05: OUT SLA LD LD LD LD LD LD LD LD LD SUB JR LD LD LD LD SUB JR INC SET CKRM10; ;	01-Jan-85 lot #1 expanded lot #0 expanded DE,OFFFFH A C,A (PPI.AW),A C B,0 HL,OFFFFH (HL),A OFH NZ,CKRM15 (HL) A NZ,CKRM15 B	<pre>;Initialize lowest address ever found ;Start from slot #0 ;Clear bit pattern ;Select the slot ;Shift bit pattern ;Assume this slot is not expanded ;Read from possible expansion slot reg ;Write a binary 11110000 ;Read back as 00001111? ;NOp, this is not an expanded slot ;Write 00000000 ;Read back as 1111111? ;NOp, not expanded slot ;We're checking expanded slot ;Say this slot is expanded</pre>	
- MSXIC 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735) - Find 02E2 02E5 02E6 02E7 02E9 02E9 02E9 02E0 02F0 02F2 02F3 02F7 02F8 02F7 02F8 02F7 02F8 02F7 02F8 02F7 02F8 02F5 02F5 02F5 02F5 02F5 02E7 02E7 02E7 02E7 02E7 02E7 02E7 02E7 02E7 02E7 02E7 02E7 02E7 02F7 02F8 02F7 02F8 02F7 02F8 02F7 02F8 02F7 02F7 02F8 02F7 02F7 02F8 02F7 02F7 02F8 02F7 02F7 02F8 02F7 02F7 02F8 02F7 02F7 02F8 02F7 0	available RAM 11 FFFF AF 4F D3 A8 CB 21 06 00 21 FFFF 36 F0 7E D6 0F 20 0B 77 7E 3C 20 06 04 CB C1	-80 3.44 ; [+ s] ; s] ; LD XOR LD CKRM05: OUT SLA LD LD LD LD LD LD SUB JR LD LD LD SUB JR SUB JR SUB JR SET CKRM10: ; ; Start from e	01-Jan-85 lot #1 expanded lot #0 expanded DE,OFFFFH A C,A (PPI.AW),A C B,O HL,OFFFFH (HL),OF0H A,(HL) OFH NZ,CKRM15 (HL),A A,(HL) A NZ,CKRM15 B 0,C expansion slot #0	<pre>;Initialize lowest address ever found ;Start from slot #0 ;Clear bit pattern ;Select the slot ;Shift bit pattern ;Assume this slot is not expanded ;Read from possible expansion slot reg ;Write a binary llll0000 ;Read back as 0000lll1? ;Nop, this is not an expanded slot ;Write 00000000 ;Read back as lllllll1? ;Nop, not expanded slot ;We're checking expanded slot ;Say this slot is expanded</pre>	
- MSXIC 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736	 Find 02E2 02E5 02E6 02E7 02E9 02E8 02E0 02F2 02F3 02F5 02F7 02F8 02F7 02F0 02F7 02F7 02F6 02F7 02F7 02F8 02F8 02F9 02F8 02F8 02F8 02F9 02F8 02F8 02F9 02F8 02F8 02F9 <li< td=""><td>available RAM 11 FFFF AF 47 D3 A8 CB 21 06 00 21 FFFF 36 F0 7E D6 0F 20 0B 77 7E 3C 20 06 04</td><td>-80 3.44 ; [+ s] ; s] ; LD XOR LD CKRM05: OUT SLD LD LD LD LD LD LD LD SUB JR JR JR JR INC SET CKRM10: ; ; Start from e ; LD</td><td>01-Jan-85 lot #1 expanded lot #0 expanded DE,OFFFFH A C,A (PPI.AW),A C B,O HL,OFFFFH (HL),A OFH NZ,CKRM15 (HL),A A,(HL) A NZ,CKRM15 B 0,C</td><td><pre>;Initialize lowest address ever found ;Start from slot #0 ;Clear bit pattern ;Select the slot ;Shift bit pattern ;Assume this slot is not expanded ;Read from possible expansion slot reg ;Write a binary 11110000 ;Read back as 00001111? ;NOp, this is not an expanded slot ;Write 00000000 ;Read back as 1111111? ;NOp, not expanded slot ;We're checking expanded slot ;Say this slot is expanded</pre></td><td></td></li<>	available RAM 11 FFFF AF 47 D3 A8 CB 21 06 00 21 FFFF 36 F0 7E D6 0F 20 0B 77 7E 3C 20 06 04	-80 3.44 ; [+ s] ; s] ; LD XOR LD CKRM05: OUT SLD LD LD LD LD LD LD LD SUB JR JR JR JR INC SET CKRM10: ; ; Start from e ; LD	01-Jan-85 lot #1 expanded lot #0 expanded DE,OFFFFH A C,A (PPI.AW),A C B,O HL,OFFFFH (HL),A OFH NZ,CKRM15 (HL),A A,(HL) A NZ,CKRM15 B 0,C	<pre>;Initialize lowest address ever found ;Start from slot #0 ;Clear bit pattern ;Select the slot ;Shift bit pattern ;Assume this slot is not expanded ;Read from possible expansion slot reg ;Write a binary 11110000 ;Read back as 00001111? ;NOp, this is not an expanded slot ;Write 00000000 ;Read back as 1111111? ;NOp, not expanded slot ;We're checking expanded slot ;Say this slot is expanded</pre>	
- MSXIC 711 712 713 714 715 716 716 717 718 720 721 722 723 724 725 726 727 728 729 729 730 731 731 732 733 734 735 736 737	0 - Find 02E2 02E5 02E6 02E7 02E7 02E9 02E0 02F0 02F7 02F9 02F7 02F8 02F7 02F8 02F7 02F8 02F7 02F8 02F7 02F8 02F7 02F8 02F5 02F7 00	available RAM 11 FFFF AF 4F D3 A8 CB 21 06 00 21 FFFF 36 F0 7E D6 0F 20 0B 77 7E 20 06 04 CB C1 32 FFFF	-80 3.44 ; + sl ; + sl ; LD LD CKRM05: OUT SLA LD LD LD LD LD LD LD LD LD SUB JR LD LD LD LD SUB JR LD LD LD LD LD LD LD LD LD LD LD LD LD	01-Jan-85 lot #1 expanded lot #0 expanded DE,OFFFFH A C,A (PPI.AW),A C B,0 HL,OFFFFH (HL),OFOH A,(HL) OFH NZ,CKRM15 (HL),A A,(HL) A NZ,CKRM15 B 0,C expansion slot #0 (OFFFFH),A	<pre>;Initialize lowest address ever found ;Start from slot #0 ;Clear bit pattern ;Select the slot ;Shift bit pattern ;Assume this slot is not expanded ;Read from possible expansion slot reg ;Write a binary llll0000 ;Read back as 0000111? ;Nop, this is not an expanded slot ;Write 00000000 ;Read back as lllllll? ;Nop, not expanded slot ;We're checking expanded slot ;Say this slot is expanded ;Select the expanded slot</pre>	
- MSXIC 711 712 713 714 715 716 717 718 719 720 721 722 723 724 725 726 727 728 729 730 731 732 733 734 735 736 737 738	0 - Find 02E2 02E5 02E6 02E7 02E9 02E9 02E9 02E0 02F0 02F2 02F3 02F7 02F8 02F7 02F8 02F7 02F8 02F7 02F8 02F7 02F8 02F7 02F8 02F5 02F7 02	available RAM 11 FFFF AF 4F D3 A8 CB 21 06 00 21 FFFF 36 F0 7E D6 0F 20 0B 77 7E 3C 20 06 04 CB C1	-80 3.44 ; [+ s] ; s] ; LD XOR LD CKRM05: OUT SLA LD LD LD LD LD SUB JR LD LD LD SUB JR SUB JR LD LD LD LD LD LD LD LD LD LD LD LD LD	01-Jan-85 lot #1 expanded lot #0 expanded DE,OFFFFH A C,A (PPI.AW),A C B,O HL,OFFFFH (HL),OF0H A,(HL) OFH NZ,CKRM15 (HL),A A,(HL) A NZ,CKRM15 B 0,C expansion slot #0	<pre>;Initialize lowest address ever found ;Start from slot #0 ;Clear bit pattern ;Select the slot ;Shift bit pattern ;Assume this slot is not expanded ;Read from possible expansion slot reg ;Write a binary llll0000 ;Read back as 0000lll1? ;Nop, this is not an expanded slot ;Write 00000000 ;Read back as lllllll1? ;Nop, not expanded slot ;We're checking expanded slot ;Say this slot is expanded</pre>	
- MSXIC 711 712 713 714 715 716 716 717 718 720 721 722 723 724 725 726 727 728 729 729 730 731 731 732 733 734 735 736 737	0 - Find 02E2 02E5 02E6 02E7 02E7 02E9 02E0 02F0 02F7 02F9 02F7 02F8 02F7 02F8 02F7 02F8 02F7 02F8 02F7 02F8 02F7 02F8 02F5 02F7 00	available RAM 11 FFFF AF 4F D3 A8 CB 21 06 00 21 FFFF 36 F0 7E D6 0F 20 0B 77 7E 20 06 04 CB C1 32 FFFF	-80 3.44 ; + sl ; + sl ; LD LD CKRM05: OUT SLA LD LD LD LD LD LD LD LD LD SUB JR LD LD LD LD SUB JR LD LD LD LD LD LD LD LD LD LD LD LD LD	01-Jan-85 lot #1 expanded lot #0 expanded DE,OFFFFH A C,A (PPI.AW),A C B,0 HL,OFFFFH (HL),OFOH A,(HL) OFH NZ,CKRM15 (HL),A A,(HL) A NZ,CKRM15 B 0,C expansion slot #0 (OFFFFH),A	<pre>;Initialize lowest address ever found ;Start from slot #0 ;Clear bit pattern ;Select the slot ;Shift bit pattern ;Asume this slot is not expanded ;Read from possible expansion slot reg ;Write a binary llll0000 ;Read back as 0000111? ;Nop, this is not an expanded slot ;Write 00000000 ;Read back as lllllll? ;Nop, not expanded slot ;We're checking expanded slot ;Say this slot is expanded ;Select the expanded slot</pre>	

-			BIOS) Macro-	80	3.44	01-Jan-85	PAGE	16-2	30
	MSX10 -	Find	available RAM						
	742	0307	77		LD	(HL),A			
	743	0308	BE		CP	(HL)			
	744	0309	2F		CPL				
	745	030A	77		LD	(HL),A			
	746	030B	20 07		JR	NZ,CKRM25		t equipped in this page	
	747	030D	2C		INC	L	;Make s	ure it's not a coincidence	
	748	030E	20 F5		JR	NZ,CKRM20	;Check	more	
	749	0310	25		DEC	Н			
	750	0311	FA 0305		JP	M,CKRM20	;Check	next page	
	751	0314		CKRM25:					
	752	0314	2E 00		LD	L,0			
	753	0316	24		INC	н			
	754	0317	7D		LD	A,L	;Below	the one ever found	
	755	0318	93		SUB	Е			
	756	0319	7C		LD	A,H			
	757	031A	9A		SBC	A,D			
	758	031B	30 OA		JR	NC,CKRM30	; NO		
	759	031D	EB		EX	DE,HL	;Regist	er this address as the lowest	
	760	031E	3A FFFF		LD	A,(OFFFFH)	;Set po	ssible secondary slot #	
	761	0321	2F		CPL				
	762	0322	6F		LD	L,A			
	763	0323	DB A8		IN	A,(PPI.AR)	;Set pr	imary slot #	
	764	0325	67		LD	H,A			
	765	0326	F9		LD	SP, HL	;Regist	ter these slot #'s	
	766	0327		CKRM30:					
	767	0327	78		LD	А, В			
	768	0328	A7		AND	A		e checking secondary slot	
	769	0329	28 OA		JR	Z,CKRM35	; No		
	770	032B	3A FFFF		LD	A,(OFFFFH)			
	771	032E	2F		CPL				
	772	032F	C6 10		ADD	A,10H	;Prepar	re to select next secondary slot	5
(MSX ROM	BASIC I	BIOS) Macro-	80	3.44	01-Jan-85	PAGE	16-3	
			available RAM			or oun of	THOL	10-3	31
	773	0331	FE 40		CP	01000000B			
	774	0333	38 CA		JR	C,CKRM10	;Continu	e if more secondary slots remain	1 . 1
	775	0335		CKRM35:	JR	C,CKRM10	;Continu	e if more secondary slots remain	ı .
	775 776	0335 0335	DB A8	CKRM35:	JR IN	C,CKRM10 A,(PPI.AR)			ì
	775 776 777	0335 0335 0337	DB A8 C6 50	CKRM35:	JR IN ADD	C,CKRM10 A,(PPI.AR) A,01010000B	;Prepare	to select next slot	1
	775 776	0335 0335	DB A8	CKRM35:	JR IN	C,CKRM10 A,(PPI.AR)	;Prepare		י י י
	775 776 777	0335 0335 0337	DB A8 C6 50	CKRM35:	JR IN ADD	C,CKRM10 A,(PPI.AR) A,01010000B	;Prepare	to select next slot	1
	775 776 777 778	0335 0335 0337 0339	DB A8 C6 50	CKRM35:	JR IN ADD	C,CKRM10 A,(PPI.AR) A,01010000B	;Prepare	to select next slot	
	775 776 777 778 (MSX ROM	0335 0335 0337 0339	DB A8 C6 50 30 AC	CKRM35:	JR IN ADD JR	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05	;Prepare ;Continu	to select next slot e if more primary slots remain	32
	775 776 777 778 (MSX ROM - MSX10 -	0335 0335 0337 0339	DB A8 C6 50 30 AC BIOS) Macro-	CKRM35:	JR IN ADD JR	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05	;Prepare ;Continu	to select next slot e if more primary slots remain	
	775 776 777 778 (MSX ROM - MSXIO - 779	0335 0335 0337 0339	DB A8 C6 50 30 AC BIOS) Macro-	CKRM35:	JR IN ADD JR	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05	;Prepare ;Continu	to select next slot e if more primary slots remain	
	775 776 777 778 (MSX ROM - MSXIO - 779 780	0335 0335 0337 0339	DB A8 C6 50 30 AC BIOS) Macro-	CKRM35: -80 ;	JR IN ADD JR 3.44	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05 01-Jan-85	;Prepare ;Continu PAGE	to select next slot e if more primary slots remain 17	
	775 776 777 778 (MSX ROM - MSXIO - 779 780 781	0335 0335 0337 0339	DB A8 C6 50 30 AC BIOS) Macro-	CKRM35: -80 ; ; Check	JR IN ADD JR 3.44	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05	;Prepare ;Continu PAGE	to select next slot e if more primary slots remain 17	
	775 776 777 778 (MSX ROM - MSXIO - 779 780 781 782	0335 0335 0337 0339 4 BASIC - Find	DB A8 C6 50 30 AC BIOS) Macro- available RAM	CKRM35: -80 ;	JR IN ADD JR 3.44 is don	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05 01-Jan-85 ne, select the b	;Prepare ;Continu PAGE	to select next slot e if more primary slots remain 17	
	775 776 777 778 (MSX ROM - MSXIO - 779 780 781 782 783	0335 0335 0337 0339 4 BASIC - Find 033B	DB A8 C6 50 30 AC BIOS) Macro- available RAM 21 Q000	CKRM35: -80 ; ; Check	JR IN ADD JR 3.44 is doi LD	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05 01-Jan-85 ne, select the b HL,0	;Prepare ;Continu PAGE	to select next slot e if more primary slots remain 17	
	775 776 777 778 (MSX ROM - MSXIO - 779 780 781 782 783 784	0335 0335 0337 0339 4 BASIC - Find 033B 033E	DB A8 C6 50 30 AC BIOS) Macro- available RAM 21 Q000 39	CKRM35: -80 ; ; Check	JR IN ADD JR 3.44 is don LD ADD	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05 01-Jan-85 ne, select the b HL,0 HL,SP	;Prepare ;Continu PAGE	to select next slot e if more primary slots remain 17	
	775 776 777 778 (MSX ROM- MSXIO - 779 780 781 782 783 784 785	0335 0335 0337 0339 4 BASIC - Find 033B 033E 033F	DB A8 C6 50 30 AC BIOS) Macro- available RAM 21 Q000 39 7C	CKRM35: -80 ; ; Check	JR IN ADD JR 3.44 is don LD ADD LD	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05 01-Jan-85 ne, select the b HL,0 HL,SP A,H	;Prepare ;Continu PAGE iggest one	to select next slot e if more primary slots remain 17	
	775 776 777 778 (MSX ROM - MSXIO - 779 780 781 782 783 784 783 784 785 786	0335 0335 0337 0339 4 BASIC - Find 033B 033E 033F 0340	DB A8 C6 50 30 AC BIOS) Macro- available RAM 21 Q000 39 7C D3 A8	CKRM35: -80 ; ; Check	JR IN ADD JR 3.44 is don LD ADD LD OUT	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05 01-Jan-85 ne, select the b HL,0 HL,SP A,H (PPI.AW),A	;Prepare ;Continu PAGE iggest one	to select next slot e if more primary slots remain 17	
	775 776 777 778 (MSX RON- MSXIO - 779 780 781 782 783 784 783 784 785 786 787	0335 0335 0337 0339 4 BASIC - Find 033B 033B 033E 033F 0340 0342	DB A8 C6 50 30 AC BIOS) Macro- available RAM 21 Q000 39 7C D3 A8 7D	CKRM35: -80 ; ; Check	JR IN ADD JR 3.44 ID ADD LD OUT LD	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05 01-Jan-85 ne, select the b HL,0 HL,SP A,H (PPI.AW),A A,L	;Prepare ;Continu PAGE iggest one ;Set pr	to select next slot e if more primary slots remain 17 : : : : : : : : : : : : : : : : : :	
	775 776 777 778 (MSX ROM- MSXIO - 779 780 781 781 783 784 785 786 786 787 788	0335 0335 0337 0339 4 BASIC - Find 033B 033E 033F 0340	DB A8 C6 50 30 AC BIOS) Macro- available RAM 21 Q000 39 7C D3 A8	CKRM35: -80 ; Check ;	JR IN ADD JR 3.44 is don LD ADD LD OUT	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05 01-Jan-85 ne, select the b HL,0 HL,SP A,H (PPI.AW),A	;Prepare ;Continu PAGE iggest one ;Set pr	to select next slot e if more primary slots remain 17	
	775 776 777 778 (MSX ROA - MSXIO - 779 780 781 781 782 783 784 783 784 785 786 786 786 787 788 789	0335 0335 0337 0339 4 BASIC - Find 033B 033B 033E 033F 0340 0342	DB A8 C6 50 30 AC BIOS) Macro- available RAM 21 Q000 39 7C D3 A8 7D	CKRM35: -80 ; Check ; ;	JR IN ADD JR 3.44 ID ADD LD UD UD LD LD	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05 01-Jan-85 ne, select the b HL,0 HL,SP A,H (PPI.AW),A A,L (0FFFFH),A	;Prepare ;Continu PAGE iggest one ;Set pr	to select next slot e if more primary slots remain 17 : : : : : : : : : : : : : : : : : :	
	775 776 777 778 (MSX RON- MSXIO - 779 780 781 780 781 782 783 784 783 784 785 786 787 786 787 786 787 789 790	0335 0335 0337 0339 4 BASIC - Find 033B 033B 033E 033F 0340 0342	DB A8 C6 50 30 AC BIOS) Macro- available RAM 21 Q000 39 7C D3 A8 7D	CKRM35: -80 ; Check ; ; Next,	JR IN ADD JR 3.44 ID ADD LD UD UD LD LD	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05 01-Jan-85 ne, select the b HL,0 HL,SP A,H (PPI.AW),A A,L	;Prepare ;Continu PAGE iggest one ;Set pr	to select next slot e if more primary slots remain 17 : : : : : : : : : : : : : : : : : :	
	775 776 777 778 (MSX ROM- MSXIO - 779 780 781 782 783 784 785 784 785 784 785 786 787 788 789 789 790 791	0335 0335 0337 0339 4 BASIC - Find 033B 033E 033F 0340 0342 0343	DB A8 C6 50 30 AC BIOS) Macro- available RAM 21 Q000 39 7C D3 A8 7D 32 FFFF	CKRM35: -80 ; Check ; ;	JR IN ADD JR 3.44 is dou LD ADD LD UD LD LD LD LD Check	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05 01-Jan-85 ne, select the b HL,0 HL,SP A,H (PPI.AW),A A,L (0FFFFH),A	;Prepare ;Continu PAGE iggest one ;Set pr	to select next slot e if more primary slots remain 17 : : : : : : : : : : : : : : : : : :	
	775 776 777 778 (MSX ROA - MSXIO - 779 780 781 781 783 784 783 784 785 786 785 786 786 787 788 789 790 791 792	0335 0335 0337 0339 4 BASIC - Find 033B 033E 0340 0342 0343 0346	DB A8 C6 50 30 AC BIOS) Macro- available RAM 21 0000 39 70 D3 A8 70 32 FFFF	CKRM35: -80 ; Check ; ; Next,	JR IN ADD JR 3.44 is don LD ADD LD OUT LD Check LD	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05 01-Jan-85 ne, select the b HL,0 HL,SP A,H (PPI.AW),A A,L (0FFFFH),A	;Prepare ;Continu PAGE iggest one ;Set pr	to select next slot e if more primary slots remain 17 : : : : : : : : : : : : : : : : : :	
	775 776 777 778 (MSX ROM- MSXIO - 779 780 781 782 783 784 783 784 785 786 785 786 786 787 788 789 790 791 792 793	0335 0335 0337 0339 4 BASIC - Find 033B 033E 0340 0342 0343 0346 0347	DB A8 C6 50 30 AC BIOS) Macro- available RAM 21 0000 39 7C D3 A8 7D 32 FFFF 79 07	CKRM35: -80 ; Check ; ; Next,	JR IN ADD JR 3.44 is dou LD ADD LD OUT LD COUT LD Check LD RLCA	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05 01-Jan-85 ne, select the b HL,0 HL,SP A,H (PPI.AW),A A,L (0FFFFH),A	;Prepare ;Continu PAGE iggest one ;Set pr	to select next slot e if more primary slots remain 17 : : : : : : : : : : : : : : : : : :	
	775 776 777 778 (MSX ROM- MSXIO - 779 780 781 782 783 784 785 786 787 788 787 788 787 788 787 789 790 791 792 793 794	0335 0335 0337 0339 4 BASIC - Find 033B 033E 033F 0340 0342 0343 0346 0347 0348	DB A8 C6 50 30 AC BIOS) Macro- available RAM 21 Q000 39 7C D3 A8 7D 32 FFFF 79 07 07	CKRM35: -80 ; Check ; ; Next,	JR IN ADD JR 3.44 is don LD ADD LD UD LD UD LD Check LD RLCA RLCA	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05 01-Jan-85 ne, select the b HL,0 HL,SP A,H (PPI.AW),A A,L (0FFFFH),A	;Prepare ;Continu PAGE iggest one ;Set pr	to select next slot e if more primary slots remain 17 : : : : : : : : : : : : : : : : : :	
	775 776 777 778 MSX RON- MSXIO - 779 780 781 781 783 784 783 784 785 785 786 787 785 786 787 788 789 790 791 792 793 794 795	0335 0335 0337 0339 4 BASIC - Find 033B 033C 0340 0342 0343 0346 0347 0346 0349	DB A8 C6 50 30 AC BIOS) Macro- available RAM 21 0000 39 7C D3 A8 7D 32 FFFF 79 07 07 07	CKRM35: -80 ; Check ; ; Next,	JR IN ADD JR 3.44 is doi LD ADD LD UD LD Check LD RLCA RLCA	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05 01-Jan-85 ne, select the b HL,0 HL,SP A,H (PPI.AW),A A,L (0FFFFH),A	;Prepare ;Continu PAGE iggest one ;Set pr	to select next slot e if more primary slots remain 17 : : : : : : : : : : : : : : : : : :	
	775 776 777 778 (MSX ROA - MSXIO - 779 780 781 783 784 783 784 783 784 785 786 785 786 786 787 786 789 790 791 792 793 794 795 796	0335 0335 0337 0339 4 BASIC - Find 033B 033E 033F 0340 0342 0343 0344 0346 0347 0348 0349 034A	DB A8 C6 50 30 AC BIOS) Macro- available RAM 21 0000 39 7C D3 A8 7D 32 FFFF 79 07 07 07 07 07	CKRM35: -80 ; Check ; ; Next,	JR IN ADD JR 3.44 is dou LD ADD LD Check LD Check LD RLCA RLCA RLCA	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05 01-Jan-85 ne, select the b HL,0 HL,SP A,H (PPI.AW),A A,L (0FFFFH),A 0C000H0FFFFH A,C	;Prepare ;Continu PAGE iggest one ;Set pr	to select next slot e if more primary slots remain 17 : : : : : : : : : : : : : : : : : :	
	775 776 777 778 (MSX RON- MSXIO - 779 780 781 782 783 784 785 783 784 785 786 787 786 787 786 787 789 790 791 792 793 794 795 796 797	0335 0335 0337 0339 4 BASIC - Find 033B 033E 0340 0342 0343 0344 0344 0348 0349 034B	DB A8 C6 50 30 AC BIOS) Macro- available RAM 21 Q000 39 7C D3 A8 7D 32 FFFF 32 FFFF 	CKRM35: -80 ; Check ; ; Next,	JR IN ADD JR 3.44 is dou LD LD LD LD LD LD LD LD LD LD LD LD LD	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05 01-Jan-85 ne, select the b HL,0 HL,SP A,H (PPI.AW),A A,L (OFFFFH),A 0C000H0FFFFH A,C C,A	;Prepare ;Continu PAGE iggest one ;Set pr ;Set po	to select next slot e if more primary slots remain 17 : : : : : : : : : : : : : : : : : :	
	775 776 777 778 MSX RON- MSXIO - 779 780 781 781 783 784 783 784 785 786 785 786 787 788 789 790 791 792 793 794 795 796 797 798	0335 0335 0337 0339 4 BASIC - Find 033E 033F 0340 0342 0343 0346 0347 0348 0349 034A 0348 0342	DB A8 C6 50 30 AC BIOS) Macro- available RAM 21 0000 39 7C D3 A8 7D 32 FFFF 79 07 07 07 07 07 07 07 07 07 07 07 07 07	CKRM35: -80 ; Check ; ; Next,	JR IN ADD JR 3.44 is doi LD ADD LD Check LD RLCA RLCA RLCA RLCA LD LD	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05 01-Jan-85 ne, select the b HL,0 HL,0 HL,SP A,H (PPI.AW),A A,L (OFFFFH),A 0C000H0FFFFH A,C C,A DE,0FFFFH	;Prepare ;Continu PAGE iggest one ;Set pr ;Set po ;Set po	to select next slot e if more primary slots remain 17 : : : : : : : : : : : : : : : : : :	
	775 776 777 778 MSX ROA - MSX IO - 779 780 781 783 784 783 784 785 786 786 787 786 789 790 791 792 793 794 795 795 796 797 798 799	0335 0335 0337 0339 4 BASIC - Find 033B 033E 034C 0343 0340 0342 0343 0344 0344 0348 0348 0348 0348 034A	DB A8 C6 50 30 AC BIOS) Macro- available RAM 21 Q000 39 7C D3 A8 7D 32 FFFF 79 07 07 07 07 07 07 07 07 07 07 07 07 07	CKRM35: -80 ; Check ; ; Next,	JR IN ADD JR 3.44 is don LD ADD LD OUT LD LD Check LD RLCA RLCA RLCA RLCA LD LD LD IN	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05 01-Jan-85 ne, select the b HL,0 HL,SP A,H (PPI.AW),A A,L (0FFFFH),A 0C000H0FFFFH A,C C,A DE,0FFFFH A,(PPI.AR)	;Prepare ;Continu PAGE iggest one ;Set pr ;Set po ;Set po	to select next slot e if more primary slots remain 17 : : : : : : : : : : : : : : : : : :	
	775 776 777 778 (MSX ROM- MSXIO - 779 780 781 782 783 784 783 784 785 786 787 786 787 786 787 788 790 791 792 793 794 795 796 797 795 796 797 798 800	0335 0335 0337 0339 4 BASIC - Find 0348 0346 0347 0348 0344 0348 0348 0348 0346 0347	DB A8 C6 50 30 AC BIOS) Macro- available RAM 21 0000 39 7C D3 A8 7D 32 FFFF 79 07 07 07 07 07 07 07 07 07 07 07 07 07	CKRM35: -80 ; Check ; ; Next, ;	JR IN ADD JR 3.44 is dou LD LD LD LD LD LD LD LD LD LD LD LD LD	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05 01-Jan-85 ne, select the b HL,0 HL,0 HL,SP A,H (PPI.AW),A A,L (OFFFFH),A 0C000H0FFFFH A,C C,A DE,0FFFFH	;Prepare ;Continu PAGE iggest one ;Set pr ;Set po ;Set po	to select next slot e if more primary slots remain 17 : : : : : : : : : : : : : : : : : :	
	775 776 777 778 MSX RON- MSXIO - 779 780 781 781 783 784 783 784 785 786 787 785 786 787 788 789 790 791 792 793 794 795 796 797 798 798 799 800	0335 0335 0337 0339 4 BASIC - Find 033E 033F 0340 0342 0343 0344 0344 0349 034A 0349 034A 0349 034A 0345 0345 0353	DB A8 C6 50 30 AC BIOS) Macro- available RAM 21 0000 39 7C D3 A8 7D 32 FFFF 79 07 07 07 07 07 07 07 07 07 07 07 07 07	CKRM35: -80 ; Check ; ; Next,	JR IN ADD JR 3.44 is doi LD ADD LD Check LD RLCA RLCA RLCA RLCA RLCA RLCA ID IN AND	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05 01-Jan-85 ne, select the b HL,0 HL,SP A,H (PPI.AW),A A,L (OFFFFH),A, 0C0000H0FFFFH A,C C,A DE,OFFFFH A,(PPI.AR) 00111111B	;Prepare ;Continu PAGE ;Set pr ;Set po ;Set po ;Set po ;Set art	to select next slot e if more primary slots remain 17 : immary slot register ossible secondary slot register alize lowest address ever found from slot #0	
	775 776 777 778 MSX ROA MSXIO - 779 780 781 783 784 783 784 783 784 783 784 785 786 787 789 790 791 792 793 794 795 795 795 795 796 797 797 798 799 800 801 802	0335 0335 0337 0339 4 BASIC - Find 033E 033E 0340 0342 0343 0340 0342 0343 0344 0347 0348 0347 0348 0347 0348 0347 0348 0346 0347	DB A8 C6 50 30 AC BIOS) Macro- available RAM 21 Q000 39 7C D3 A8 7D 32 FFFF 07 07 07 07 07 07 07 07 07 07 07 07 07	CKRM35: -80 ; Check ; ; Next, ;	JR IN ADD JR 3.44 is don LD ADD LD OUT LD Check LD RLCA RLCA RLCA RLCA RLCA LD LD LD LD Check UD RLCA RLCA RLCA RLCA RLCA RLCA RLCA RLCA	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05 01-Jan-85 ne, select the b HL,0 HL,SP A,H (PPI.AW),A A,L (OFFFFH),A, 0C000HOFFFFH A,C C,A DE,OFFFFH A,(PPI.AR) 0011111B (PPI.AW),A	;Prepare ;Continue PAGE iggest one ;Set pr ;Set po ;Set po ;Set po ;Set start ;Start	to select next slot e if more primary slots remain 17 timary slot register pssible secondary slot register enlize lowest address ever found from slot #0 t the slot	
	775 776 777 778 MSX10 - 779 780 781 783 784 783 784 783 784 785 786 787 786 787 786 787 788 789 790 791 792 793 794 795 795 796 797 795 796 797 795 796 797 799 800 801 802 803	0335 0335 0337 0339 4 BASIC - Find 0348 0340 0342 0343 0346 0347 0348 0347 0348 0348 0349 034A 034B 034C 0345 0353 0355	DB A8 C6 50 30 AC BIOS) Macro- available RAM 21 Q000 39 7C D3 A8 7D 32 FFFF 79 07 07 07 07 07 07 07 07 07 07 07 07 07	CKRM35: -80 ; Check ; ; Next, ;	JR IN ADD JR 3.44 is dou LD LD LD LD LD LD LD LD LD LD	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05 01-Jan-85 01-Jan-85 he, select the b HL,0 HL,SP A,H (PPI.AW),A A,L (OFFFFH),A, 0C000H0FFFFH A,C C,A DE,0FFFFH A,(PPI.AR) 0011111B (PPI.AW),A B,0	;Prepare ;Continue PAGE iggest one ;Set pr ;Set po ;Set po ;Set po ;Set at ;Setect ;Assume	to select next slot e if more primary slots remain 17	
	775 776 777 778 MSX RON- MSXIO - 779 780 781 781 783 784 783 784 783 784 785 786 787 785 786 787 788 789 790 791 792 793 794 795 796 797 795 796 797 798 799 800 801 802 803 804	0335 0335 0337 0339 4 BASIC - Find 033E 033F 0340 0342 0343 0344 0344 0344 0348 0346 0347 0348 0349 0346 0347 0348 0349 0346 0345 0353 0353 0353 0353 0355	DB A8 C6 50 30 AC BIOS) Macro- available RAM 21 0000 39 7C D3 A8 7D 32 FFFF 07 07 07 07 07 07 07 07 07 07 07 07 07	CKRM35: -80 ; Check ; ; Next, ;	JR IN ADD JR 3.44 is doi LD ADD LD Check LD RLCA RLCA RLCA RLCA IN AND OUT LD IN AND OUT LD IN ADD ID ADD AD	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05 01-Jan-85 ne, select the b HL,0 HL,SP A,H (PPI.AW),A A,L (OFFFFH),A, 0C000H0FFFFH A,C C,A DE,OFFFFH A,(PPI.AR) 00111111B (PPI.AW),A B,0 C	;Prepare ;Continu PAGE iggest one ;Set pr ;Set po ;Set po ;Set po ;Set so ;Set	to select next slot e if more primary slots remain 17 : imary slot register ossible secondary slot register dlize lowest address ever found from slot #0 : the slot = this slot is not expanded bit pattern	
	775 776 777 778 MSX ROA MSX ROA MSXIO - 779 780 781 783 784 783 784 783 784 783 784 785 786 787 789 796 791 792 793 794 795 795 795 795 795 795 795 795 795 795	0335 0335 0337 0339 4 BASIC - Find 033B 033E 0340 0342 0343 0340 0342 0343 0344 0344	DB A8 C6 50 30 AC BIOS) Macro- available RAM 21 Q000 39 7C D3 A8 7D 32 FFFF 07 07 07 07 07 07 07 07 07 07 07 07 07	CKRM35: -80 ; Check ; ; Next, ;	JR IN ADD JR 3.44 is don LD ADD LD OUT LD Check LD RLCA RLCA RLCA RLCA RLCA IN AND OUT LD LD LD Check LD RLCA R	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05 01-Jan-85 ne, select the b HL,0 HL,SP A,H (PPI.AW),A A,L (OFFFFH),A, 0C000H0FFFFH A,C C,A DE,0FFFFH A,(PPI.AR) 0011111B (PPI.AW),A B,0 C NC,CKRM60	;Prepare ;Continue PAGE iggest one ;Set pr ;Set po ;Set po ;Set po ;Set po ;Set se ;Set po ;Set se ;Set po ;Set po ;Set po ;Set po	to select next slot e if more primary slots remain 17 timary slot register pssible secondary slot register elize lowest address ever found from slot #0 the slot bit pattern lot is not expanded	
	775 776 777 778 MSX ROM- MSX IO - 779 780 781 782 783 784 785 786 785 786 787 788 788 789 790 791 792 793 794 799 799 799 799 799 799 799 799 799	0335 0335 0337 0339 4 BASIC - Find 0348 0346 0342 0343 0346 0347 0348 0346 0347 0348 0346 0347 0348 0345 0345 0355 0355 0355	DB A8 C6 50 30 AC BIOS) Macro- available RAM 21 Q000 39 7C D3 A8 7D 32 FFFF 03 77 07 07 07 07 07 07 07 07 07 07 07 07	CKRM35: -80 ; Check ; ; Next, ;	JR IN ADD JR 3.44 is dou LD LD LD LD LD LD LD LD LD LD	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05 01-Jan-85 01-Jan-85 01-Jan-85 he, select the b HL,0 HL,SP A,H (PPI.AW),A A,L (OFFFFH),A, 0C000H0FFFFH A,C C,A DE,0FFFFH A,(PPI.AR) 00111111B (PPI.AW),A B,0 C NC,CKRM60 B	;Prepare ;Continue PAGE iggest one ;Set pr ;Set po ;Set po ;Set po ;Set po ;Set se ;Set po ;Set se ;Set po ;Set po ;Set po ;Set po	to select next slot e if more primary slots remain 17 : imary slot register ossible secondary slot register dlize lowest address ever found from slot #0 : the slot = this slot is not expanded bit pattern	
	775 776 777 778 MSX RON- MSXIO - 779 780 781 781 783 784 783 784 783 784 785 786 787 785 786 787 788 789 790 791 792 793 794 795 796 797 795 796 797 798 799 800 801 801 802 803 804 805 806 807	0335 0335 0337 0339 4 BASIC - Find 033E 033F 0340 0342 0343 0344 0343 0346 0347 0348 0349 0344 0349 0344 0345 0345 0353 0353 0355 0357 0358 0356	DB A8 C6 50 30 AC BIOS) Macro- available RAM 21 0000 39 7C D3 A8 7D 32 FFFF 07 07 07 07 07 07 07 07 07 07 07 07 07	CKRM35: -80 ; Check ; ; Next, ;	JR IN ADD JR 3.44 is doi LD ADD LD OUT LD Check LD RLCA RLCA RLCA RLCA RLCA ID IN AND OUT LD ID RLCA IN AND JR IN ADD ID ID ID ID ID ID ID ID ID	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05 01-Jan-85 ne, select the b HL,0 HL,SP A,H (PPI.AW),A A,L (OFFFFH),A, 0C000H0FFFFH A,C C,A DE,0FFFFH A,(PPI.AR) 0011111B (PPI.AW),A B,0 C NC,CKRM60	;Prepare ;Continue PAGE iggest one ;Set pr ;Set po ;Set po ;Set po ;Set po ;Set se ;Set po ;Set se ;Set po ;Set po ;Set po ;Set po	to select next slot e if more primary slots remain 17 timary slot register pssible secondary slot register elize lowest address ever found from slot #0 the slot bit pattern lot is not expanded	
	775 776 777 778 MSX ROM- MSX IO - 779 780 781 782 783 784 785 786 785 786 787 788 788 789 790 791 792 793 794 799 799 799 799 799 799 799 799 799	0335 0335 0337 0339 4 BASIC - Find 0348 0346 0342 0343 0346 0347 0348 0346 0347 0348 0346 0347 0348 0345 0345 0355 0355 0355	DB A8 C6 50 30 AC BIOS) Macro- available RAM 21 Q000 39 7C D3 A8 7D 32 FFFF 03 77 07 07 07 07 07 07 07 07 07 07 07 07	CKRM35: -80 ; Check ; ; Next, ;	JR IN ADD JR 3.44 is dou LD LD LD LD LD LD LD LD LD LD	C,CKRM10 A,(PPI.AR) A,01010000B NC,CKRM05 01-Jan-85 01-Jan-85 01-Jan-85 he, select the b HL,0 HL,SP A,H (PPI.AW),A A,L (OFFFFH),A, 0C000H0FFFFH A,C C,A DE,0FFFFH A,(PPI.AR) 00111111B (PPI.AW),A B,0 C NC,CKRM60 B	;Prepare ;Continue PAGE iggest one ;Set pr ;Set po ;Set po ;Set po ;Set po ;Set se ;Set po ;Set se ;Set po ;Set po ;Set po ;Set po	to select next slot e if more primary slots remain 17 timary slot register pssible secondary slot register elize lowest address ever found from slot #0 the slot bit pattern lot is not expanded	

		BIOS) Mac available B		3.44	01-Jan-85	PAGE 17-1	33
810	0362		CKRM55:				
811	0362	32 FFFF		LD	(OFFFFH),A	;Select the expanded slot	
812	0365		CKRM60:				
813 814	0365 0368	21 FE00	CKDWCE .	LD	HL,0FE00H	;Start checking from OFE00H to OC000H	
815	0368	7E	CKRM65:	LD	x (117.)		
816	0369	2F		CPL	A,(HL)		
817	036A	77		LD	(HL),A		
818	036B	BE		CP	(HL)		
819	036C	2F		CPL	•		
820	036D	77		LD	(HL),A		
821 822	036E	20 09		JR	NZ,CKRM70	;RAM not equipped in this page	
823	0370 0371	2C 20 F5		INC JR	L Ng GKDWG 5	Make sure it's not a coincidence	
824	0373	25		DEC	NZ,CKRM65 H	;Check more	
825	0374	7C		LD	л.н А,Н		
826	0375	FE CO		CP	ОСОН		
827	0377	30 EF		JR	NC , CKRM6 5	;Check next page	
828 829	0379	27.00	CKRM70:				
830	0379 037в	2E 00 24		LD	L,0		
831	037C	7D		INC LD	H A,L	Delay the second of a	
832	037D	93		SUB	E	;Below the one ever found	
833	037E	7C		LD	А,Н		
834	037F	9A		SBC	A,D		
835	0380	30 OA		JR	NC, CKRM75	; NO	
836 837	0382 0383	EB		EX	DE,HL	;Register this address as the lowest	
838	0385	3A FFFF 2F		LD CPL	A,(OFFFFH)	;Set possible secondary slot #	
839	0387	6F		LD	L,A		
840	0388	DB A8		IN	A,(PPI.AR)	;Set primary slot #	
					, (111	, Set primary SIDL #	
		BIOS) Mac available R		3.44	01-Jan-85	PAGE 17-2	34
841	038A	67		LD	H,A		
842	038B	F9		LD	SP, HL	;Register these slot #'s	
843	038C		CKRM75:				
844 845	038C 038D	78		LD	А,В		
845	038D 038E	A7 . 28 08		AND JR	A Z,CKRM80	;Are we checking secondary slot ;No	
847	0390	JA FFFF		LD	A,(OFFFFH)		
848	0393	2F		CPL	,		
849	0394	C6 40		ADD	A,01000000B	;Prepare to select next secondary slot	
850	0396	30 CA		JR	NC,CKRM55	;Continue if more secondary slots remain	
851	0398		CKRM80:				
852 853	0398 039a	DB A8 C6 40		IN	A,(PPI.AR)		
854	039C	30 B5		ADD JR	A,01000000B NC,CKRM50	;Prepare to select next slot ;Continue if more primary slots remain	
855			SUBTTL ·				
(MSX ROM - MSXIO -		BIOS) Mac attribute s		3.44	01-Jan-85	PAGE 18	35
856							
857			;				
858			; Check	is don	e, select the bi	ggest one	
859	030-	a1	;				
860 861	039E 03Al	21 0000 39		LD	HL,0		
862	03A1	39 7C		ADD LD	HL,SP A,H		
863	03A3	D3 A8		OUT	(PPI.AW),A	;Set primary slot register	
864	03A5	7D		LD	A,L	, see primary side register	
865	03A6	32 FFFF		LD	(OFFFFH),A	;Set possible secondary slot register	
866	03A9	79		LD	A,C	;Set 'slot expanded' flag	
867			;			•	
868			; Clear	work a	rea with zero		
869 870	0.2.8.8	01 0040	;				
870 871	03AA 03AD	01 0C49 11 F381		LD	BC,0C49H	;length of work area	
871	0380	21 F381		LD LD	DE,RAMLOW+1 HL,RAMLOW	beginning of the	
873	03B3	36 00		LD	(HL),0	;beginning of work ;init first byte	
874	03B5	ED BO		LDIR	(1117)0	; init first byte ; transfer it to rest of area	
875			;			, is to rest or urea	
876			; Set E	(PTBL			
877			;				
878	03B7	4F		LD	C,A	;Get 'slot-expanded' flag	
879 880	03B8 03BA	06 04 21 FCC4		LD	B,4	;Loop 4 times .	
881	03BA 03BD	AL FUU4	SSLTLP:	LD	HL, EXPTBL+3		
882	03BD	CB 19	5551621	RR	с	;Set carry if LSB is set	
883	03BF	9F		SBC	А,А	;[Acc]=255 if expanded, 0 if not expanded	
884	03C0	E6 80		AND	80H	;Affects only MSB	
885	03C2	77		LD	(HL),A	;Set table for each slot	
886	03C3	2B		DEC	HL		

- MSXIO	- Slot	attribute se	tup			
887	03C4	10 F7	DJNZ	SSLTLP		
888			;			
889			; Set SLTTBL			
890			;			
891	03C6	DB A8	IN	A,(PPI.AR)	;Remember primary slot register's content	
892	03C8	4F	LD	C,A	· · · · ·	
893	03C9	AF	XOR	A	;Read from slot #0	
894	03CA	D3 A8	OUT	(PPI.AW),A		
895	03CC	3A FFFF	LD	A,(OFFFFH)		
896	03CF	2F	CPL			
897	03D0	6F	LD	L,A		
898	03D1	3E 40	LD	A,01000000B	;Read from slot #1	
899	03D3	D3 A8	OUT	(PPI.AW),A		
900	03D5	3A FFFF	LD	A,(OFFFFH)		
901	03D8	2F	CPL			
902	03D9	67	LD	H,A		
903	03DA	3E 80	LD	A,80H	;Read from slot #2	
904 905	03DC 03DF	D3 A8	OUT	(PPI.AW),A		
	03DE	3A FFFF 2F	LD	A,(OFFFFH)		
906 907	03E1	2F 5F	CPL			
907	03E2	5F	LD	E,A		
908	03E3	3E C0	LD	A,OCOH	;Read from slot #3	
909	03E5 03E7	D3 A8	OUT	(PPI.AW),A		
910		3A FFFF 2F	LD	A,(OFFFFH)		
911	03EA 03EB	2F 57	CPL			
912	03EB	57	LD	D,A	N	
	03EC	79 D2 N8	LD	A,C	Restore primary slot register;	
914	03ED	D3 A8	OUT	(PPI.AW),A		
915 916	03EF	22 FCC5	LD	(SLTTBL),HL	;Set SLTTBL	
916 917	03F2	EB 22 FCC7	EX	DE,HL		
211	03F3	ee ruu/	LD	(SLTTBL+2),HL		
MSXIO	- Slot 03F6	BIOS) Macro attribute se ED 56 C3 2680	etup IM	01-Jan-85 l INIT	PAGE 18-2	37
MSXIO	- Slot	attribute se	etup IM JP		;IM 1	37
MSXIO 918 919 920	- Slot 03F6 03F8	attribute se ED 56	IM JP SUBTTL - MSXIO	l Init	;IM 1	37
MSXIO 918 919 920 (MSX R	- Slot 03F6 03F8 0M BASIC	attribute se ED 56 C3 2680	IM JP SUBTTL - MSXIO 0-80 3.44	l INIT - Control-[C]	;IM 1 processing	37 38
MSXIO 918 919 920 (MSX R - MSXIO 921	O - Slot 03F6 03F8 OM BASIC - Contr	attribute se ED 56 C3 2680 BIOS) Macro	IM JP SUBTTL - MSXIO o-80 3.44 ssing	l INIT - Control-[C]	;IM 1 processing	
MSXIO 918 919 920 (MSX R - MSXIO 921 922	O - Slot 03F6 03F8 OM BASIC - Contr 03FB	attribute se ED 56 C3 2680 BIOS) Macro col-[C] proces	IM JP SUBTTL - MSXIO o-80 3.44 ssing ISCNTC:	l INIT - Control-[C] y 0l-Jan-85	;IM 1 processing PAGE 19	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923	OH BASIC OM BASIC OM BASIC OM BASIC	Attribute se ED 56 C3 2680 BIOS) Macro col-[C] proces 3A FBB1	IM JP SUBTTL - MSXIO o-80 3.44 ssing ISCNTC: LD	l INIT - Control-[C] ; 0l-Jan-85 A,(BASROM)	;IM 1 processing	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923 924	OM BASIC O3F8 OM BASIC - Contr 03FB 03FB 03FB	attribute se ED 56 C3 2680 BIOS) Macro col-[C] proces 3A FBB1 A7	IM JP SUBTTL - MSXIO o-80 3.44 ssing ISCNTC: LD AND	l INIT - Control-[C] j 01-Jan-85 A,(BASROM) A	;IM 1 processing PAGE 19 ;Is BASIC text in ROM	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923 924 925	OM BASIC OJF8 OM BASIC - Contr 03FB 03FB 03FB 03FF	attribute se ED 56 C3 2680 BIOS) Macr. col-(C) proces 3A FBB1 A7 C0	IM JP SUBTTL - MSXIO o-80 3.44 ssing ISCNTC: LD AND RET	l INIT - Control-[C] y 0l-Jan-85 A,(BASROM) A NZ	;IM 1 processing PAGE 19	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923 924 925 926	OM BASIC O3F8 OM BASIC - Contr O3FB O3FB O3FB O3FE O3FF O400	attribute se ED 56 C3 2680 BIOS) Macro col-[C] proces 3A FBB1 A7 C0 E5 ·	IM JP SUBTTL - MSXIO o-80 3.44 ssing ISCNTC: LD AND RET PUSH	l INIT - Control-[C] p 01-Jan-85 A,(BASROM) A NZ HL	; IM 1 processing PAGE 19 ; Is BASIC text in ROM ; Yes	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923 924 925 926 927	- Slot 03F6 03F8 04 BASIC - Contr 03FB 03FB 03FB 03FB 03FB 03FB 03FB 03FB	attribute se ED 56 C3 2680 BIOS) Macro col-[C] proces 3A FBB1 A7 C0 E5 · 21 FC9B	Etup IM JP SUBTTL - MSXIO o-80 3.44 ssing ISCNTC: LD AND RET PUSH LD	l INIT - Control-[C] y 0l-Jan-85 A,(BASROM) A NZ	;IM 1 processing PAGE 19 ;Is BASIC text in ROM	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923 924 925 926 927 928	- Slot 03F6 03F8 04 BASIC - Contr 03FB 03FB 03FB 03FF 03FF 0400 0401	attribute se ED 56 C3 2680 BIOS) Macro col-[C] proces 3A FBB1 A7 C0 E5 - 21 FC9B F3	Etup IM JP SUBTTL - MSXIO o-80 3.44 ssing ISCNTC: LD AND RET PUSH LD DI	l INIT - Control-[C] p 01-Jan-85 A,(BASROM) A NZ HL HL,INTFLG	; IM 1 processing PAGE 19 ; Is BASIC text in ROM ; Yes	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923 924 925 926 927 928 929	- Slot 03F6 03F8 04 03F8 03F8 03F8 03F8 03F8 03F8 03F8 03F8	attribute se ED 56 C3 2680 BIOS) Macro col-[C] proces 3A FBB1 A7 C0 E5 . 21 FC9B F3 7E	IM JP SUBTTL - MSXIO o-80 3.44 ssing ISCNTC: LD AND RET PUSH LD DI LD DI LD	l INIT - Control-[C] y 0l-Jan-85 A,(BASROM) A NZ HL HL,INTFLG A,(HL)	; IM 1 processing PAGE 19 ; Is BASIC text in ROM ; Yes	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923 924 925 926 927 928 929 930	 Slot 03F6 03F8 04 BASIC Contr 03FB 03FB 03FB 03FB 03FB 03FB 03FB 0400 0401 0404 0405 	attribute se ED 56 C3 2680 BIOS) Macro col-[C] proces 3A FBB1 A7 C0 E5 · 21 FC9B F3 7E 36 00	Etup IM JP SUBTTL - MSXIO o-80 3.44 ssing ISCNTC: LD AND RET PUSH LD DI LD LD	1 INIT - Control-[C] ; 01-Jan-85 A,(BASROM) A NZ HL HL,INTFLG A,(HL) (HL),0	; IM 1 processing PAGE 19 ; Is BASIC text in ROM ; Yes	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923 924 925 926 927 928 929 931	- Slot 03F6 03F8 04 BASIC - Contr 03FB 03FB 03FB 03FF 0400 0401 0404 0405 0406 0408	attribute se ED 56 C3 2680 BIOS) Macro col-[C] proces 3A FBB1 A7 C0 E5 · 21 FC9B F3 7E 36 00 E1	Etup SUBTTL - MSXIO o-80 3.44 ssing ISCNTC: LD AND RET PUSH LD DI LD DI LD POP	l INIT - Control-[C] y 0l-Jan-85 A,(BASROM) A NZ HL HL,INTFLG A,(HL)	; IM 1 processing PAGE 19 ; Is BASIC text in ROM ; Yes	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923 924 925 926 927 928 929 930 931 932	- Slot 03F6 03F8 03F8 03FB 03FB 03FB 03FB 03FB 03FB 03FB 03FB	attribute se ED 56 C3 2680 BIOS) Macro col-[C] proces 3A FBB1 A7 C0 E5 . 21 FC9B F3 7E 36 00 E1 FB	Etup IM JP SUBTTL - MSXIO o-80 3.44 ssing ISCNTC: LD AND AND RET PUSH LD DI LD DI LD POP EI	l INIT - Control-[C] p 0l-Jan-85 A,(BASROM) A NZ HL HL,INTFLG A,(HL) (HL),0 HL	; IM 1 processing PAGE 19 ; Is BASIC text in ROM ; Yes	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923 924 925 926 927 926 927 928 929 930 931 932 933	- Slot 03F6 03F8 04 BASIC - Contr 03FB 03FB 03FB 03FB 03FB 03FB 0400 0401 0404 0405 0406 0408 0408 0409 040A	attribute se ED 56 C3 2680 BIOS) Macro col-[C] proces 3A FBB1 A7 C0 E5 · 21 FC9B F3 7E 36 00 E1 FB A7	IM JP SUBTTL - MSXIO o-80 3.44 ssing ISCNTC: LD AND RET PUSH LD DI LD LD LD EI AND	1 INIT - Control-[C] p 01-Jan-85 A,(BASROM) A NZ HL HL,INTFLG A,(HL) (HL),0 HL	;IM 1 processing PAGE 19 ;Is BASIC text in ROM ;Yes ;Seen any interesting key	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923 924 925 926 927 928 929 930 931 932 931 932 933 934	- Slot 03F6 03F8 04 BASIC - Contr 03FB 03FB 03FB 03FF 0400 0401 0404 0405 0406 0408 0409 040A	attribute se ED 56 C3 2680 BIOS) Macro col-[C] proces 3A FBB1 A7 C0 E5 · 21 FC9B F3 7E 36 00 E1 FB A7 C8	Etup SUBTTL - MSXIO o-80 3.44 ssing ISCNTC: LD AND RET PUSH LD DI LD POP EI AND RET	l INIT - Control-[C] y 0l-Jan-85 A,(BASROM) A NZ HL HL,INTFLG A,(HL) (HL),0 HL A Z	;IM 1 processing PAGE 19 ;Is BASIC text in ROM ;Yes ;Seen any interesting key ;No	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923 924 925 926 927 928 929 931 932 933 933 933 935	- Slot 03F6 03F8 03F8 03FB 03FB 03FB 03FB 03FB 03FB 03FB 03FB	attribute se ED 56 C3 2680 BIOS) Macro col-[C] proces 3A FBB1 A7 C0 E5 · 21 FC9B F3 7E 36 00 E1 FB A7 C8 FE 03	Etup SUBTTL - MSXIO SUBTTL - MSXIO 0-80 3.44 ssing ISCNTC: LD AND AND RET PUSH LD DI LD DI LD DI LD DI LD RET PUSH CP	l INIT - Control-[C] j 0l-Jan-85 A,(BASROM) A NZ HL HL,INTFLG A,(HL) (HL),0 HL A Z 3	;IM 1 processing PAGE 19 ;Is BASIC text in ROM ;Yes ;Seen any interesting key ;No ;Is it ctrl-stop?	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936	- Slot 03F6 03F8 04 BASIC - Contr 03FB 03FB 03FB 03FF 0400 0401 0404 0405 0406 0408 0409 040A	attribute se ED 56 C3 2680 BIOS) Macro col-[C] proces 3A FBB1 A7 C0 E5 · 21 FC9B F3 7E 36 00 E1 FB A7 C8	Etup SUBTTL - MSXIO o-80 3.44 ssing ISCNTC: LD AND RET PUSH LD DI LD POP EI AND RET	l INIT - Control-[C] y 0l-Jan-85 A,(BASROM) A NZ HL HL,INTFLG A,(HL) (HL),0 HL A Z	;IM 1 processing PAGE 19 ;Is BASIC text in ROM ;Yes ;Seen any interesting key ;No	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923 924 925 926 927 928 929 930 931 933 934 935 936 931 935 936 937	- Slot 03F6 03F8 03F8 03FB 03FB 03FB 03FB 03FB 03FB 03FB 03FB	attribute se ED 56 C3 2680 BIOS) Macro col-[C] proces 3A FBB1 A7 C0 E5 · 21 FC9B F3 7E 36 00 E1 FB A7 C8 FE 03	SUBTTL - MSXIO SUBTTL - MSXIO SUBTTL - MSXIO 3.44 ssing ISCNTC: LD AND RET DI LD DI LD POP EI LD POP EI AND RET CP JR	1 INIT - Control-[C] 01-Jan-85 A,(BASROM) A NZ HL HL,INTFLG A,(HL) (HL),0 HL A Z 3 2,EXCABO	;IM 1 processing PAGE 19 ;Is BASIC text in ROM ;Yes ;Seen any interesting key ;No ;Is it ctrl-stop? ;Yes, execution aborted	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936	- Slot 03F6 03F8 03F8 03FB 03FB 03FB 03FB 03FB 03FB 03FB 03FB	attribute se ED 56 C3 2680 BIOS) Macro col-[C] proces 3A FBB1 A7 C0 E5 · 21 FC9B F3 7E 36 00 E1 FB A7 C8 FE 03	SUBTTL - MSXIO SUBTTL - MSXIO SUBTTL - MSXIO 3.44 ssing ISCNTC: LD AND RET DI LD DI LD POP EI LD POP EI AND RET CP JR	l INIT - Control-[C] j 0l-Jan-85 A,(BASROM) A NZ HL HL,INTFLG A,(HL) (HL),0 HL A Z 3	;IM 1 processing PAGE 19 ;Is BASIC text in ROM ;Yes ;Seen any interesting key ;No ;Is it ctrl-stop? ;Yes, execution aborted	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939	- Slot 03F6 03F8 03F8 03FB 03FB 03FB 03FB 03FF 03FF 03FF 0400 0401 0404 0405 0406 0408 0409 040A 0408 0408	attribute se ED 56 C3 2680 BIOS) Macro col-[C] proces 3A FBB1 A7 C0 E5 · 21 FC9B F3 7E 36 00 E1 FB A7 C8 FE 03	SUBTTL - MSXIO SUBTTL - MSXIO SUBTTL - MSXIO O-80 3.44 Ssing ISCNTC: LD AND RET UD DI LD DI LD POP EI AND RET CP JR ; ; Pause until ;	1 INIT - Control-[C] 01-Jan-85 A,(BASROM) A NZ HL HL,INTFLG A,(HL) (HL),0 HL A Z 3 2,EXCABO	;IM 1 processing PAGE 19 ;Is BASIC text in ROM ;Yes ;Seen any interesting key ;No ;Is it ctrl-stop? ;Yes, execution aborted	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923 924 925 926 927 928 929 930 931 932 933 931 932 933 935 936 937 938	- Slot 03F6 03F8 04 BASIC - Contr 03FB 03FB 03FB 03FB 03FB 03FB 0400 0401 0404 0405 0406 0408 0408 0408 0408 0400 0408	attribute se ED 56 C3 2680 BIOS) Macro col-[C] proces 3A FBB1 A7 C0 E5 · 21 FC9B F3 7E 36 00 E1 FB A7 C8 FE 03	Etup IM JP SUBTTL - MSXIO 0-80 3.44 ssing ISCNTC: LD AND RET PUSH LD DI LD DI LD EI AND RET PUSH ZD JT ; ; Pause until	1 INIT - Control-[C] 01-Jan-85 A,(BASROM) A NZ HL HL,INTFLG A,(HL) (HL),0 HL A Z 3 2,EXCABO	;IM 1 processing PAGE 19 ;Is BASIC text in ROM ;Yes ;Seen any interesting key ;No ;Is it ctrl-stop? ;Yes, execution aborted	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923 924 925 926 927 926 927 928 929 930 931 932 933 934 935 936 937 938 939	- Slot 03F6 03F8 03F8 03FB 03FB 03FB 03FB 03FF 03FF 03FF 0400 0401 0404 0405 0406 0408 0409 040A 0408 0408	attribute se ED 56 C3 2680 BIOS) Macro col-[C] process 3A FBB1 A7 C0 E5 · 21 FC9B F3 7E 36 00 E1 FB A7 C8 FE B3 21 FC9B F3 7E 36 00 E1 FB A7 C8 FE B3 A7 C9 B1 C9 C9 C9 C0 E5 · C1 C1 C1 C1 C1 C2 C2 C2 C2 C2 C2 C2 C2 C2 C2	SUBTTL - MSXIO SUBTTL - MSXIO SUBTTL - MSXIO O-80 3.44 Ssing ISCNTC: LD AND RET UD DI LD DI LD POP EI AND RET CP JR ; ; Pause until ;	l INIT - Control-[C] p 0l-Jan-85 A,(BASROM) A NZ HL HL,INTFLG A,(HL) (HL),0 HL HL, O HL A Z Z,EXCABO next STOP is pres	; IM 1 processing PAGE 19 ; Is BASIC text in ROM ;Yes ;Seen any interesting key ;No ;Is it ctrl-stop? ;Yes, execution aborted ssed	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940	- Slot 03F6 03F8 04 BASIC - Contr 03FB 03FB 03FB 03FB 03FB 03FB 0400 0401 0404 0405 0406 0408 0408 0408 0408 0400 0408	attribute se ED 56 C3 2680 BIOS) Macro col-[C] proces 3A FBB1 A7 C0 E5 · 21 FC9B F3 7E 36 00 E1 FB A7 C8 FE 03 28 1C	SUBTTL - MSXIO SUBTTL - MSXIO SUBTTL - MSXIO 3.44 Ssing ISCNTC: LD AND RET UD LD DI LD POP EI AND RET CP JR ; ; Pause until ; PUSH	1 INIT - Control-[C] ; 01-Jan-85 A,(BASROM) A NZ HL HL,INTFLG A,(HL) (HL),0 HL A Z 3 Z,EXCABO next STOP is pre HL	; IM 1 processing PAGE 19 ; Is BASIC text in ROM ;Yes ;Seen any interesting key ;No ;Is it ctrl-stop? ;Yes, execution aborted ssed	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923 924 925 926 927 928 929 930 931 932 933 931 932 933 935 936 935 936 939 939 941	- Slot 03F6 03F8 03F8 03FB 03FB 03FB 03FB 03FB 03FB 03FB 03FB	attribute se ED 56 C3 2680 BIOS) Macro col-[C] proces 3A FBB1 A7 C0 E5 . 21 FC9B F3 7E 36 00 E1 FB A7 C8 FE 03 28 1C E5 D5	SUBTTL - MSXIO SUBTTL - MSXIO SUBTTL - MSXIO 3.44 ssing ISCNTC: LD AND RET PUSH LD DI LD DI LD BI EI AND RET PUSH ; Pause until ; PUSH	1 INIT - Control-[C] ; 01-Jan-85 A,(BASROM) A NZ HL HL,INTFLG A,(HL) (HL),0 HL A A,(HL) (HL),0 HL Z Z 3 Z,EXCABO next STOP is pre- HL DE	; IM 1 processing PAGE 19 ; Is BASIC text in ROM ;Yes ;Seen any interesting key ;No ;Is it ctrl-stop? ;Yes, execution aborted ssed	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923 924 925 926 927 928 929 930 931 932 933 933 934 935 936 937 938 939 940 940	- Slot 03F6 03F8 03F8 03FB 03FB 03FB 03FB 03FF 03FF 03FF 03FF	attribute se ED 56 C3 2680 BIOS) Macro col-[C] proces 3A FBB1 A7 C0 E5 · 21 FC9B F3 7E 36 00 E1 FB A7 C8 FE 03 28 1C E5 D5 C5	SUBTTL - MSXIO SUBTTL - MSXIO SUBTTL - MSXIO O-80 3.44 Ssing ISCNTC: LD AND RET PUSH LD DI LD POP EI AND RET CP JR ; PUSH PUSH PUSH PUSH	1 INIT - Control-[C] p 01-Jan-85 A,(BASROM) A NZ HL HL,INTFLG A,(HL) (HL),0 HL (HL),0 HL (HL),0 HL C S Z,EXCABO next STOP is present L DE BC	<pre>;IM 1 processing PAGE 19 ;Is BASIC text in ROM ;Yes ;Seen any interesting key ;No ;Is it ctrl-stop? ;Yes, execution aborted ssed ;STOP pressed (pause) ;Display cursor if disabled</pre>	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 937 938 939 940 941 943	- Slot 03F6 03F8 03F8 03F8 03FB 03FB 03FB 03FB 03FB 03FB 03FB 03FB	attribute se ED 56 C3 2680 BIOS) Macro col-[C] proces 3A FBB1 A7 C0 E5 · 21 FC9B F3 7E 36 00 E1 FB A7 C8 FE 03 28 1C E5 D5 C5 C0 09DA	SUBTTL - MSXIO SUBTTL - MSXIO SUBTTL - MSXIO Ssing ISCNTC: LD AND RET PUSH LD DI LD POP EI AND RET CP JR ; Pause until ; PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH	1 INIT - Control-[C] ; 01-Jan-85 A,(BASROM) A NZ HL HL,INTFLG A,(HL) (HL),0 HL A Z 3 Z,EXCABO next STOP is pre- HL DE BC CKDPC0	; IM 1 processing PAGE 19 ; Is BASIC text in ROM ;Yes ;Seen any interesting key ;No ;Is it ctrl-stop? ;Yes, execution aborted ssed ;STOP pressed (pause)	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 941 942 943	- Slot 03F6 03F8 03F8 03FB 03FB 03FB 03FB 03FB 03FB 03FB 03FB	attribute se ED 56 C3 2680 BIOS) Macro col-[C] proces 3A FBB1 A7 C0 E5 · 21 FC9B F3 7E 36 00 E1 FB A7 C8 FE 03 28 1C E5 D5 C5 C0 09DA	SUBTTL - MSXIO SUBTTL - MSXIO SUBTTL - MSXIO 3.44 Ssing ISCNTC: LD AND RET PUSH LD DI LD LD DI LD LD DI LD LD LD DI LD LD LD DI LD LD LD LD LD LD LD LD LD LD LD LD LD	1 INIT - Control-[C] ; 01-Jan-85 A,(BASROM) A NZ HL HL,INTFLG A,(HL) (HL),0 HL A Z 3 Z,EXCABO next STOP is pre- HL DE BC CKDPC0	<pre>;IM 1 processing PAGE 19 ;Is BASIC text in ROM ;Yes ;Seen any interesting key ;No ;Is it ctrl-stop? ;Yes, execution aborted ssed ;STOP pressed (pause) ;Display cursor if disabled</pre>	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923 924 925 926 927 928 929 930 931 932 933 933 934 935 936 937 938 939 940 942 943 944	- Slot 03F6 03F8 03F8 03FB 03FB 03FB 03FB 03FB 03FB 03FF 03FF	attribute se ED 56 C3 2680 BIOS) Macro col-[C] process 3A FBB1 A7 C0 E5 · 21 FC9B F3 7E 36 00 E1 FB A7 C8 FE 03 28 1C E5 C5 C5 C5 C0 09DA 21 FC9B	SUBTTL - MSXIO SUBTTL - MSXIO SUBTTL - MSXIO 3.44 Ssing ISCNTC: LD AND RET PUSH LD LD DI LD POP EI AND RET CP JR ; Pause until ; PUSH PUSH PUSH PUSH PUSH PUSH PUSH PUSH	1 INIT - Control-[C] ; 01-Jan-85 A,(BASROM) A NZ HL HL,INTFLG A,(HL) (HL),0 HL A Z 3 Z,EXCABO next STOP is pre- HL DE BC CKDPC0	<pre>;IM 1 processing PAGE 19 ;Is BASIC text in ROM ;Yes ;Seen any interesting key ;No ;Is it ctrl-stop? ;Yes, execution aborted ssed ;STOP pressed (pause) ;Display cursor if disabled</pre>	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 937 938 939 940 941 943 944 945	- Slot 03F6 03F8 03F8 03FB 03FB 03FB 03FB 03FB 03FB 03FB 03FB	attribute se ED 56 C3 2680 BIOS) Macro col-(C) proces 3A FBB1 A7 C0 E5 21 FC9B F3 7E 36 00 E1 FB A7 C8 FE 03 28 1C E5 C5 C5 C5 C5 C5 C5 C5 C5 C5 C	SUBTTL - MSXIO SUBTTL - MSXIO SUBTTL - MSXIO 3.44 Ssing ISCNTC: LD AND RET DI LD DI LD POP EI AND RET CP JR ; Pause until ; Pause until ; Push PUSH DI LD POP EI AND RET CP USH DI LD POP EI AND RET CP USH DI LD DI LD POP EI AND RET CD DI LD DI LD POP EI AND RET LD DI LD DI LD POP EI AND RET LD DI LD DI LD POP EI AND RET LD DI LD DI LD POP EI AND RET LD DI LD POP EI AND RET LD DI LD POP EI AND RET LD DI LD POP EI AND RET LD DI LD POP EI AND RET LD DI LD POP EI AND RET LD DI LD POP EI AND RET CJR T LD DI LD POP EI AND RET CJR T LD DI LD DI LD POP EI AND RET CJR T LD DI LD DI LD POP EI AND RET CJR T LD DI LD DI LD POP EI AND RET CJR T LD DI LD DI LD POP EI AND RET CJR T LD DI LD DI LD POP EI AND RET CJR T LD DI LD DI LD POP EI AND RET CJR T LD DI LD DI LD POP EI AND RET CJR T LD DI LD DI LD DI LD POP EI AND RET CJR T LD DI LD DI LD DI LD DI LD DI LD DI LD D DI LDI L	1 INIT - Control-[C] p 01-Jan-85 A,(BASROM) A NZ HL HL,INTFLG A,(HL) (HL),0 HL A Z 3 Z,EXCABO next STOP is present HL DE BC CKDPC0 HL,INTFLG	<pre>;IM 1 processing PAGE 19 ;Is BASIC text in ROM ;Yes ;Seen any interesting key ;No ;Is it ctrl-stop? ;Yes, execution aborted ssed ;STOP pressed (pause) ;Display cursor if disabled</pre>	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 942 943 945 945 946 947	- Slot 03F6 03F8 03F8 03FB 03FB 03FB 03FB 03FB 03FB 03FB 03FF 0400 0401 0404 0405 0406 0406 0406 0408 0409 040A 0409 040A 0405 0406 0408 0409 040A 0409 040A 0408 0409 0404 0409 0404 0409 0404 0409 0404 0409 0404 0409 0404 0409 0404 0409 0404 0409 0404 0409 0404 0409 0404 0409 0404 0409 0404 0409 0404 0409 0409 0400 0401 0409 0409	attribute se ED 56 C3 2680 BIOS) Macro col-[C] process 3A FBB1 A7 C0 E5 · 21 FC9B F3 7E 36 00 E1 FB A7 C8 FE 03 28 1C E5 C5 C5 C0 09DA 21 FC9B F3 7E 36 00 E1 F3 5 5 5 5 5 5 5 5 5 5 5 5 5	stup SUBTTL - MSXIO SUBTTL - MSXIO 0-80 3.44 Ssing ISCNTC: LD AND RET PUSH LD DI LD POP EI AND RET CP JR T PUSH PUSH LD DI LD POSH PUSH CP JR T PUSH PUSH CP JR T DI LD DI DI DI DI DI DI DI DI DI D	1 INIT - Control-[C] ; 01-Jan-85 A,(BASROM) A NZ HL HL,INTFLG A,(HL) (HL),0 HL A,(HL) (HL),0 HL A Z Z Z,EXCABO next STOP is pre- HL DE BC CKDPC0 HL,INTFLG A,(HL)	; IM 1 processing PAGE 19 ; Is BASIC text in ROM ;Yes ;Seen any interesting key ;No ;Is it ctrl-stop? ;Yes, execution aborted ssed ;STOP pressed (pause) ;Display cursor if disabled ;Wait for next interesting key	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923 924 925 926 927 928 926 927 928 930 931 933 934 935 936 937 938 939 940 941 943 944 943	- Slot 03F6 03F8 03F8 03FB 03FB 03FB 03FB 03FB 03FB 03FB 03FB	attribute se ED 56 C3 2680 BIOS) Macro col-(C) proces 3A FBB1 A7 C0 E5 21 FC9B F3 7E 36 00 E1 FB A7 C8 FE 03 28 1C E5 C5 C5 C5 C5 C5 C5 C5 C5 C5 C	stup SUBTTL - MSXIO SUBTTL - MSXIO SUBTTL - MSXIO 3.44 Ssing ISCNTC: LD AND RET PUSH LD DI LD DI LD POP EI AND RET CP JR ; PusH LD DI DI DI DI DI DI DI DI DI D	1 INIT - Control-[C] p 01-Jan-85 A,(BASROM) A NZ HL HL,INTFLG A,(HL) (HL),0 HL A Z,EXCABO next STOP is pres HL DE BC CKDPC0 HL,INTFLG A,(HL) (HL),0	<pre>;IM 1 processing PAGE 19 ;Is BASIC text in ROM ;Yes ;Seen any interesting key ;No ;Is it ctrl-stop? ;Yes, execution aborted ssed ;STOP pressed (pause) ;Display cursor if disabled ;Wait for next interesting key ;Wait for character if SELECT pressed</pre>	
MSXIO 918 919 920 (MSX R - MSXIO 921 922 923 924 925 926 927 928 929 930 931 932 933 934 935 936 937 938 939 940 942 943 945 945 946 947	- Slot 03F6 03F8 03F8 03FB 03FB 03FB 03FB 03FB 03FB 03FB 03FF 0400 0401 0404 0405 0406 0406 0406 0408 0409 040A 0409 040A 0405 0406 0408 0409 040A 0409 040A 0408 0409 0404 0409 0404 0409 0404 0409 0404 0409 0404 0409 0404 0409 0404 0409 0404 0409 0404 0409 0404 0409 0404 0409 0404 0409 0404 0409 0404 0409 0409 0400 0401 0409 0409	attribute se ED 56 C3 2680 BIOS) Macro col-[C] process 3A FBB1 A7 C0 E5 · 21 FC9B F3 7E 36 00 E1 FB A7 C8 FE 03 28 1C E5 C5 C5 C0 09DA 21 FC9B F3 7E 36 00 E1 F3 5 5 5 5 5 5 5 5 5 5 5 5 5	stup SUBTTL - MSXIO SUBTTL - MSXIO 0-80 3.44 Ssing ISCNTC: LD AND RET PUSH LD DI LD POP EI AND RET CP JR T PUSH PUSH LD DI LD POSH PUSH CP JR T PUSH PUSH CP JR T DI LD DI DI DI DI DI DI DI DI DI D	1 INIT - Control-[C] ; 01-Jan-85 A,(BASROM) A NZ HL HL,INTFLG A,(HL) (HL),0 HL A,(HL),0 HL A Z Z Z,EXCABO next STOP is pre- HL DE BC CKDPC0 HL,INTFLG A,(HL)	; IM 1 processing PAGE 19 ; Is BASIC text in ROM ;Yes ;Seen any interesting key ;No ;Is it ctrl-stop? ;Yes, execution aborted ssed ;STOP pressed (pause) ;Display cursor if disabled ;Wait for next interesting key	