BASIC Self-modifying Code Initialisation

BASIC EPROM	E046	E047	E048	E049	E050	E051	E052	E053	E054	E055	E056	E057	E058	E059	E060	E061	E062	E063	E064	E065	E066	E067	E068	E069
Zero Page	**E2**	**E3**	**E4**	**E5**	**E6**	**E7**	**E8**	**E9**	**EA**	**EB**	**EC**	**ED**	**EE**	**EF**	**F0**	**F1**	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**F8**	**F9**
Self Modifying Code	E6	E9	D0	02	E6	EA	AD	60	EA	C9	3A	B0	0A	C9	20	F0	EF	38	E9	30	38	E9	D0	60
	INC $E9		BNE $E8		INC $EA		LDA $EA60			CMP #3A		BCS $F9		CMP #20		BEQ $E2		SEC	SBC #30		SEC	SBC #D0		RTS
DBASIC Initialisation										C9	20	F0	F3	4C	00	A8								
										CMP #20		BEQ $E2		JMP $A800			*	*						
HRG Toolkit Initialisation																		20	8F	E8	4C	63	E0	
																		JSR $E88F			JMP $E063			

Default BASIC Self Modifying Code

The BASIC interpreter progresses the interpretation of each character located in a BASIC program or in the direct command input buffer in an area of self modifying code in the zero page between $E2 and $F9. When BASIC is started, this code is loaded in from the BASIC EPROM locations $E046-$E069 to the zero page during its initialisation. The key locations are $E9 and $EA as these hold the current memory location of the character being inspected in either the BASIC program or the input buffer. The first part of the code ($E2-$EA) increments this address and loads the character at this address into the Accumulator. The code between $EB and $F2 checks if the character is a Space (program jumps back to $E2 to load the next character) or not a number (program exits the routine). The code between $F3 and $F9 converts the number to a form that the BASIC program can use and exits. (Note that this last piece of code is the same as that located at $E063-$E069).

DBASIC Initialisation

One task of the DBASIC utility program is to modify the default self-modifying code between $EB and $F1. After checking to see if the character is a Space, it jumps to $A800 in the DBASIC EPROM code to check if the character corresponds with a DBASIC command and process it. The $A800 routine either exits with an RTS or if the character is a number, jumps to $E063 to execute the final part of the default code described above. It should be noted that the DBASIC code also uses, and therefore corrupts the contents of the zero page locations $F2 and $F3.

HRG Toolkit Initialisation

The HRG Toolkit initialisation routine modifies the default self-modifying code between $F3 and $F8. After the default code has checked for a Space, it jumps at $F3 to the HRG toolkit sub-routine $E88F to check if the character corresponds with a HRG plot command and process it and then returns to $F6 where it jumps to $E063 to complete the latter part of the default code. The BASIC Toolkit initialisation is the same except it has a difference sub-routine location. It can be seen that if the default code has already been modified by the DBASIC modification, then the HRG $E88F sub-routine (or the BASIC Toolkit sub-routine) would never be called.

BASIC Self-modifying Code Initialisation

BASIC EPROM	E046	E047	E048	E049	E050	E051	E052	E053	E054	E055	E056	E057	E058	E059	E060	E061	E062	E063	E064	E065	E066	E067	E068	E069
Zero Page	**E2**	**E3**	**E4**	**E5**	**E6**	**E7**	**E8**	**E9**	**EA**	**EB**	**EC**	**ED**	**EE**	**EF**	**F0**	**F1**	**F2**	**F3**	**F4**	**F5**	**F6**	**F7**	**F8**	**F9**
Self Modifying Code	E6	E9	D0	02	E6	EA	AD	60	EA	C9	3A	B0	0A	C9	20	F0	EF	38	E9	30	38	E9	D0	60
	INC $E9		BNE $E8		INC $EA		LDA $EA60			CMP #3A		BCS $F9		CMP #20		BEQ $E2		SEC	SBC #30		SEC	SBC #D0		RTS
DBASIC Solution										C9	20	F0	F3	4C	00	A8			EA	EA	EA	4C	63	E0
										CMP #20		BEQ $E2		JMP $A800			*	*	NOP	NOP	NOP	JMP $E063		
HBASIC Solution										C9	20	F0	F3	4C	00	A8			20	8F	E8	4C	63	E0
										CMP #20		BEQ $E2		JMP $A800			*	*	JSR $E88F			JMP $E063		
TBASIC Solution										C9	20	F0	F3	4C	00	A8			20	34	E8	4C	63	E0
										CMP #20		BEQ $E2		JMP $A800			*	*	JSR $E834			JMP $E063		

DBASIC Solution

The solution to avoid any corruption at location $F3 is to move the jumps to the toolkit sub-routines one location to the right commencing at $F4. So, after completing the DBASIC $A800 routine the program needs to jump back to $F4 instead of jumping to $E063. To do this the $A800 EPROM code needs to be modified at $A8FB from 4C 63 E0 to 4C F4 00. The DBASIC initialisation has no other Toolkit routine to call so the first 3 locations starting at $F4 are filled with NOP. The self-modifying code then makes the jump to $E063 at $F7. The only change that needs to be made to the DBASIC utility program is to modify the data table used to modify the self-modifying code between $EB and $F9 as shown above.

HBASIC Solution

The HBASIC utility program has to initialise the Microtan system for BASIC, and to recognise both the extra BASIC TANDOS commands and the HRG Toolkit commands. It is very similar to the DBASIC program except it also needs to call the HRG initialisation routine at $E800 and further modify the Self-modifying code as shown above to ensure the $E88F sub-routine is called when BASIC code is being interpreted.

TBASIC Solution

The TBASIC utility program has to initialise the Microtan system for BASIC, and to recognise both the extra BASIC TANDOS commands and the extra BASIC Toolkit commands. It will be very similar to the HBASIC program described above except the self-modifying code needs to be amended to call the $E834 sub-routine instead.