



the value in A because the D register comprises the A and B registers. Therefore, we use Y to hold the actual address.

Having coded the first Breakpoint process, there are three processes remaining. Two of these reverse the two processes we have coded so far: Uninsert-Breakpoint will remove a breakpoint from the table, and Unset-Breakpoint takes out the SWI-Opcode and puts back the original value. These two routines will be looked at in the next instalment. The third remaining routine, to display all the breakpoints, is the last routine that we will code here.

DISPLAY-BREAKPOINTS

Data:

Breakpoint-Number is an eight-bit counter to run through the Breakpoint table in B

Current-Breakpoint is the address to be displayed
Breakpoint-Labels are two-digit (decimal) numbers to label the addresses as they are displayed

Space is the space character that separates a label from an address

Process 3: Display-Breakpoints

Set Breakpoint-Number to 1 (an actual offset of zero)

While Breakpoint-Number <= Number-Of-Breakpoints

Display Breakpoint-Labels(Breakpoint-Number)

Display Breakpoint-Table(Breakpoint-Number)

Increment Breakpoint-Number

Endwhile

End of Process 3

The PUTHEX Routine

PUTHEX	PSHS	A,B,X	Save used registers
	PSHS	B	Save B temporarily
	LEAX	HEXCHS,PCR	Address of HEXCHS in X
	LSRB		Four right shifts for most significant four bits
	LSRB		
	LSRB		
	LSRB		
	LDA	B,X	Get appropriate character in A
	BSR	OUTCH	Display it
	PULS	B	Get B back
	ANDB	#%00001111	Mask off most significant four bits
	LDA	B,X	Second character
	BSR	OUTCH	
	PULS	A,B,X,PC	Restore and return

The Input/Output Module

HEXCHS	FCC	'0123456789ABCDEF'	
DOT	FCB	'.'	
RETURN	FCB	13	ASCII code for Return
COMNDS	FCC	'BUDSGRMQ'	
GETCOM	PSHS	A,X	Save A and X contents
	LEAX	COMNDS,PCR	Address of command characters in X
	LDB	#8	Number-Of-Valid-Characters
	BSR	GETCH	Get Character
	PULS	A,X,PC	Return
GETADD	BSR	GETHX4	
	BSR	PUTCR	
	RTS		
GETVAL	BSR	GETHX2	
	BSR	PUTCR	
	RTS		
DSPVAL	BSR	PUTHEX	
	BSR	PUTCR	
	RTS		
DSPADD	PSHS	B	Save B temporarily
	TFR	A,B	Most-Significant-Byte in B
	BSR	PUTHEX	
	PULS	B	Retrieve B
	BSR	PUTHEX	
	PULS	B	
	BSR	PUTHEX	
	BSR	PUTCR	
	RTS		

Insert-Breakpoint Routine

BP01	PSHS	A,B,X,Y	Save used registers
	LDX	BPTAB	Address of Breakpoint-Table
	LDA	NUMBP,PCR	
IF01	CMPA	MAXBP,PCR	If Number-Of-Breakpoints < Max
	BGE	ENDF01	
	INC	NUMBP,PCR	Add 1 to Number-Of-Breakpoints
	LSLA		Multiply Offset by two for 16-bit table
	LEAY	A,X	
	BSR	GETADD	Get the address
	STD	,Y	Store address in Breakpoint-Table
ENDF01	PULS	A,B,X,Y	Restore and return