



Display HEXCHS (Offset)
 Extract least significant four bits of Number
 Display HEXCHS (Offset)

The final routine needed for handling input and output is the PUTCR subroutine. This is straightforward, and the final coded form is self-explanatory. Having coded all the necessary routines, we can now design the I/O module itself.

THE INPUT/OUTPUT MODULE

Process:

GetCommand will return Offset in B, which can be used as an offset into a jump table
 GetAddress leaves return address in D
 GetValue leaves return value in B, flag in A
 DisplayValue is passed in B
 DisplayAddress is passed in D

The final coded form of the I/O module is given on the following page. Now we can return to the Breakpoint module that we began in the last instalment (see page 758). We have already given the code for the second process in this module, which sets up breakpoints. We put aside the problem of coding the first process (inserting breakpoints) because it involved getting an address. Having now dealt with this task in the routines given here, we can proceed to give the coded version of the process — which incorporates a branch to the GETADD subroutine.

Notice that in the code, the command INC NUMBP, PCR adds 1 to the Number-Of-Breakpoints. At this point, A is one less than the Number-Of-Breakpoints, which is the correct offset into the breakpoint table. However, the address is returned in D, and this is going to destroy

The GETHX2 Routine

GETHX2	LDB	#18	Number-Of-Valid-Chars
HX4	PSHS	X	Save used register
	LEAX	HEXCHS,PCR	Get address of Valid-Chars in X
	BSR	GETCH	Get Next-Character
IF00	CMPB	#16	If Offset = 16
	LDA	#SFF	Set flag to -1 (in two's complement)
	BRA	ENDFOO	
	CMPB	#17	If Offset = 17
	LDA	#1	Set flag to 1
	BRA	ENDFOO	
	LSLB		Shift B left four places to form most significant digit; B holds offset in HEXCHS and hence the binary value
	LSLB		
	LSLB		
	LSLB		
	PSHS	B	Save B temporarily
	LDB	#16	Only hex digits now valid
	BSR	GETCH	Next-Character
	ADDB	1,S+	Construct eight-bit number and lose temporary B
	PULS	X,PC	

The PUTCR Routine

PUTCR	PSHS	A	Save A
	LDA	#13	ASCII code for Return
	BSR	OUTCH	Display it
	PULS	A,PC	



The GETHX4 Routine

GETHX4	LDB	#16	
	BSR	HX4	Get Most-Significant-Byte
	PSHS	B	Save Most-Significant-Byte temporarily
	LDB	#16	
	BSR	HX4	Get Least-Significant-Byte in B
	PULS	A	Get Most-Significant-Byte back in A
	RTS		Required value is in D

Display-Breakpoint Routine

BPLABS	FCC	'12345678910111213141516'	
SPACE	FCB	32	ASCII code for a Space
DISPBP	PSHS	A,B,X,Y	
	LEAX	BPTAB,PCR	Address of Breakpoint-Table
	LEAY	BPLABS,PCR	Address of labels
	CLRB		Set Breakpoint-Number to zero offset
WHILO1	CMBP	NUMBP,PCR	While Breakpoint-Number <= Number-Of-Breakpoints
	BGT	ENDW01	
	LDA	,Y+	Display label
	BSR	OUTCH	
	LDA	,Y+	
	BSR	OUTCH	
	LDA	SPACE,PCR	Display a Space
	BSR	OUTCH	
	PSHS	B	Save B temporarily
	LDD	,X++	Save Address
	BSR	DSPADD	
	PULS	B	Restore B
	BRA	WHILO1	
ENDW01	PULS	A,B,X,Y	Restore and return