



Initialisation Procedure

START	RMB	2	To save the start address	LDA	,Y+	Second character of label
OPJMP	FCB	\$0E	JMP-opcode	BSR	OUTCH	Display it
INIT	LDX	\$FFFA	Get Vector-Address	LDA	SPACE,P-CR	Display space
	LDA	OPJMP,PCR	Get JMP-opcode and	BSR	OUTCH	
	STA	,X+	save it at Vector-Address	LDD	,X++	Get next register into Two-Byte-Value
	LDY	1,S	Get Entry-Address from the stack			-Value
	STY	,X	Save it at Vector-Address + 1	BSR	DSPADD	Display Two-Byte-Value
	BSR	GETADD	Get Start-Address from keyboard	PULS	A,PC	Restore A and return
	STD	START,PCR	Save it			
	RTS		Return			

Command R

STACKP	RMB	2	Stack-Pointer
LABELS	FCC	'CC A BDP X Y UPC S'	
SPACE	FCB	32	ASCII code for space
CMDR	PSHS	A,B,X,Y	Save used registers
	LDX	STACKP,PCR	Get Stack-Pointer
	LEAY	LABELS,PCR	Use Y to point to label
	LDA	#4	Number of single byte registers
FOR01	BSR	CMDR1	Display next register four times
	DECA		
	BGT	FOR01	
	LDA	#4	Number of two byte registers
FOR02	BSR	CMDR2	Display next register four times
	DECA		
	BGT	FOR02	
	LDA	,Y+	First character of label
	BSR	OUTCH	Display it
	LDA	,Y+	Second character of label
	BSR	OUTCH	Display it
	LDA	SPACE,PCR	Display space
	BSR	OUTCH	
	TFR	X,D	X now contains the required value of S
	BSR	DSPADD	Display S
	PULS	A,B,X,Y,PC	Restore and return

*Subroutine to display a single byte register

CMDR1	PSHS	A	Save A
	LDA	,Y+	First character of label
	BSR	OUTCH	Display it
	LDA	,Y+	Second character of label
	BSR	OUTCH	Display it
	LDA	SPACE,PCR	Display space
	BSR	OUTCH	
	LDB	,X+	Get next register into Single-Byte-Value
			-Value
	BSR	DSPVAL	Display Single-Byte-Value
	PULS	A,PC	Restore A and return

*Subroutine to display a two byte register

CMDR2	PSHS	A	Save A
	LDA	,Y+	First character of label
	BSR	OUTCH	Display it

Command G

BPTAB	RMB	32	Breakpoint-Table
REMTAB	RMB	16	Removed-Values
NEXTBP	RMB	1	Next-Breakpoint
GMDG	PSHS	A	Save A in case we do a normal return
	LDA	NEXTBP,PCR	Next-Breakpoint
IF04	BLE	ENDF04	If Next-Breakpoint > 0
	CMPA	MAXBP,PCR	and <=16
	BGT	ENDF04	(maximum number of breakpoints)
	DECA		Convert to offset into table
	LEAX	BPTAB,PCR	Address of Breakpoint-Table
	LEAY	REMTAB,PCR	Address of Removed-Values
	LDB	A,Y	Get Removed-Value
	LSLA		Convert A to offset for 16-bit table
	STB	[A,X]	Store it at address in Breakpoint-Table
	LDS	STACKP,PCR	Get Stack-Pointer into S
	DEC	10,S	Adjust value of PC on stack
	INC	NEXTBP,PCR	Increment Next-Breakpoint
	RTI		Return from interrupt
ENDF04	PULS	A,PC	Restore and return

