

Description		Mnemonic	Operands	Modifies	Status	Corrupts
Stop execution	0	STOP				
Return from trap	0	RETTR				
Move SP to A	0	MOVSPA		A		
Move NZVC flags to A	0	MOVFLGA		A		
Branch unconditional	0	BR	label			
Branch less than or equal	0	BRLE	label			
Branch less than	0	BRLT	label			
Branch equal to	0	BREQ	label			
Branch not equal to	0	BRNE	label			
Branch greater than or equal	0	BRGE	label			
Branch greater than	0	BRGT	label			
Branch if oVerflow	0	BRV	label			
Branch if Carry	0	BRC	label			
Call subprogram	0	CALL	label	SP		
Bitwise invert r	0	NOTr		r	NZ	
Negate r	0	NEGr		r	NZV	
Arithmetic shift left r	0	ASLr		r	NZVC	
Arithmetic shift right r	0	ASRr		r	NZC	
Rotate left r	0	ROLr		r	C	
Rotate right r	0	RORr		r	C	
Unary no operation trap	0	NOPn				
Nonunary no operation trap	0	NOP				
Decimal input trap	0	DECI	m,ams	m	NZV	
Decimal output trap	0	DECO	m,ams			
String output trap	0	STRO	m,ams			
Character input	0	CHARI	m,ams	m		
Character output	0	CHARO	m,ams			
Return from call with n bytes	0	RETn		SP		
Add to stack pointer	0	ADDSP	m,ams	SP	NZVC	
Subtract from stack pointer	0	SUBSP	m,ams	SP	NZVC	
Add to r	0	ADDr	m,ams	r	NZVC	
Subtract from r	0	SUBr	m,ams	r	NZVC	
Bitwise AND to r	0	ANDr	m,ams	r	NZ	
Bitwise OR to r	0	ORr	m,ams	r	NZ	
Compare r	0	CPr	m,ams		NZVC	
Load r from memory	0	LDr	m,ams	r	NZ	
Load byte from memory	0	LDBYTEr	m,ams	r	NZ	
Store r to memory	0	STr	m,ams	m		
Store byte r to memory	0	STBYTEr	m,ams	m		
		Directive				
The address of a symbol	0	.ADDRSS				
A string of ASCII bytes	0	.ASCII				
A block of bytes	0	.BLOCK				
Initiate ROM burn	0	.BURN				
A byte value	0	.BYTE				
The sentinel for the assembler	0	.END				
Equate a symbol to a constant	0	.EQUATE				
A word value	0	.WORD				

Description		Directive				
Assemble the file at this point	1	.INCLUDE	"fid"			
Assemble the file at the end	1	.APPEND	"fid"			
Guarantee a label's global scope	1	.GLOBAL	label			
Begin a macro definition	1	.MACRO				
End a macro definition	1	.MACROEND				
Description		Mnemonic	Operands	Modifies	Status	Corrupts
Branch if N	2	BRN	Label			
Branch if Z	2	BRZ	label			
Branch if not oVerflow	2	BRNV	label			
Branch if not Carry	2	BRNC	label			
Clear r	2	CLRr		r	NZVC	
Clear memory	2	CLR	m,ams	m		NZVC
Increment r	2	INCr		r	NZVC	
Decrement r	2	DECr		r	NZVC	
Increment memory	2	INC	m,ams	m		NZVC
Decrement memory	2	DEC	m,ams	m		NZVC
Bitwise invert memory	2	NOT	m,ams	m		NZVC
Negate memory	2	NEG	m,ams	m		NZVC
Add to memory; $m \leftarrow m + \text{addend}$	2	ADD	m,ams,addend,ams2	m		NZVC
Subtract from memory; $m \leftarrow m - \text{subtrahend}$	2	SUB	m,ams,subtrahend,ams2	m		NZVC
Move; $\text{destin} \leftarrow \text{source}$	2	MOVE	source,ams,destin,ams2	destin		NZVC
Test (compare to zero) r	2	TSTr			NZVC	
Test (compare to zero) memory	2	TEST	m,ams		NZVC	A
Compare	2	CPM	m1,ams,m2,ams2		NZVC	A
PUSH r	2	PUSHr		SP		NZVC
POP r	2	POP r		SP, r		NZVC
PUSH memory	2	PUSH	m,ams	SP		NZVC
POP memory	2	POP	m,ams	SP, m		NZVC
String Input ¹ to memory ("String Object")	2	STRI ¹	m,ams	m ²		A, NZVC
Decimal Input r	2	DECIR		r		NZVC
Decimal Output r	2	DECOR				
Binary Output r	2	BINOR				NZVC
Binary Output memory	2	BINO	m,ams			NZVC
Hexadecimal Output r	2	HEXOR				NZVC
Hexadecimal Output memory	2	HEXO	m,ams			NZVC
Dump (display) top portion of stack	2	DUMPS	m,ams			NZVC
Statically save r (w/RESTOREr)	2	SAVER ³				
Statically save A,X (w/RESTORE)	2	SAVE ⁴				
Restore previously saved r (w/SAVER)	2	RESTOREr ³		r		NZVC
Restore previously saved A,X (w/SAVE)	2	RESTORE ⁴		A, X		NZVC

1 Memory operand must reference a "String Object" with a capacity value stored in the "before byte descriptor".

2 The referenced "String Object" is modified. Also, A is corrupted such that it contains the count of the number of characters read but truncated from the object's result due to unavailable capacity. Thus, a zero value in A indicates that the entire input has been stored in the string.

3 SAVER and RESTOREr are complementary operations.

4 SAVE and RESTORE are complementary operations.