Machine Language to Assembly Language Conversion Table

Hex Code	Mnemonic Code	Mnemonic Description	Mode	Number of Bytes
00	*			
01	NOP	No operation	Inherent	1
02	*			
03	*			
04	*			
05	*			
06	TAP	Transfer from accumulator A to process code register	Inherent	1
07	TPA	Transfer from process code register to accumulator A	Inherent	1
08	INX	Increment index register	Inherent	1
09	DEX	Decrement index register	Inherent	1
0A	CLV	Clear 2's complement overflow bit	Inherent	1
0B	SEV	Set 2's complement overflow bit	Inherent	1
0C	CLC	Clear carry	Inherent	1
0D	SEC	Set carry	Inherent	1
0E	CLI	Clear interrupt mask	Inherent	1
0F	SEI	Set interrupt mask	Inherent	1
10	SBA	Subtract accumulator	Inherent	1
11	CBA	Compare accumulator	Inherent	1
12	*			
13	*			
14	*			
15	*			
16	TAB	Transfer from A to B	Inherent	1
17	TBA	Transfer from B to A	Inherent	1
18	*			
19	DAA	Decimal Adjust (A)	Inherent	1
1A	*			
1B	ADA	Add B to A	Inherent	1
1C	*			
1D	*			
1E	*			
1F	*			
20	BRA	Branch always	Relative	2
21	*			
23	BLS	Branch if lower or same	Relative	2
24	BCC	Branch if carry clear	Relative	2
25	BCS	Branch if carry set	Relative	2
26	BNE	Branch if not equal	Relative	2
27	BEQ	Branch if equal	Relative	2
28	BVC	Branch if overflow clear	Relative	2
29	BVS	Branch if overflow set	Relative	2
2A	BPL	Branch if plus	Relative	2
2B	BMI	Branch if minus	Relative	2
2C	BGE	Branch if greater than zero or equal to	Relative	2
2D	BLT	Branch if less than zero	Relative	2
2E	BGT	Branch if greater than zero	Relative	2
2F	BLE	Branch if less than zero or equal to	Relative	2

Machine Language to Assembly Language Conversion Table (continued)

Hex Code	Mnemonic Code	Mnemonic Description	Mode	Number of Bytes
30	TSX	Transfer from stack pointer to index register	Inherent	1
31	INS	Increment stack pointer	Inherent	1
32	PUL(A)	Pull data from stack		1
33	PUL(B)	Pull -data from stack		1
34	DES	Decrement stack pointer	Inherent	1
35	TXS	Transfer from index register to stack pointer	Inherent	1
36	PSH(A)	Push data on stack		1
37	PSH(B)	Push data on stack		1
38	*			
39	RTS	Return from subroutine	Inherent	1
3A	*			
3B	RTI	Return from interrupt	Inherent	1
3C	*			
3D	*			
3E	WAI	Wait for interrupt	Inherent	1
3F	SWI	Software interrupt	Inherent	1
40	NEG(A)	Negate		1
41	*			
42	*			
43	COM(A)	Complement		1
44	LSR(A)	Logical shift right		1
45	*			
46	ROR(A)	Rotate right		1
47	ASR(A)	Arithmetic right shift		1
48	ASL(A)	Arithmetic shift left		1
49	ROL(A)	Rotate left		1
4A	DEC(A)	Decrement		1
4B	*			
4C	INC(A)	Increment		1
4D	TST(A)	Test		1
4E	*			
4F	CLR(A)	Clear		1
50	NEG(B)	Negate		1
51	*			
52	*			
53	COM(B)	Complement		1
54	LSR(B)	Logical shift right		1
55	*			
56	ROR(B)	Rotate right		1
57	ASR(B)	Arithmetic shift right		1
58	ASL(B)	Arithmetic shift left		1
59	ROL(B)	Rotate left		1
5A	DEC(B)	Decrement		1
5B	*			
5C	INC(B)	Increment		1
5D	TST(B)	Test		1
5E	*			
5F	CLR(B)	Clear		1

Machine Language to Assembly Language Conversion Table (continued)

Hex Code	Mnemonic Code	Mnemonic Description	Mode	Number of Bytes
60	NEG	Negate	Indexed	2
61	*			
62	*			
63	COM	Complement	Indexed	2
64	LSR	Logical shift right	Indexed	2
65	*			
66	ROR	Rotate Right	Indexed	2
67	ASR	Arithmetic shift right	Indexed	2
68	ASL	Arithmetic shift left	Indexed	2
69	ROL	Rotate left	Indexed	2
6A	DEC	Decrement	Indexed	2
6B	*			
6C	INC	Increment	Indexed	2
6D	TST	Test	Indexed	2
6E	JMP	Jump	Indexed	2
6F	CLR	Clear	Indexed	2
70	NEG	Negate	Extended	3
71	*			
72	*			
73	COM	Complement	Extended	3
74	LSR	Logical shift right	Extended	3
9F	STS	Store stack pointer	Direct	2
A0	SUB(A)	Subtract	Indexed	2
A1	CMP(A)	Compare	Indexed	2
A2	SBC(A)	Subtract with carry	Indexed	2
A3	*			
A4	AND(A)	Logical AND	Indexed	2
A5	BIT(A)	Bit test	Indexed	2
A6	LDA(A)	Load accumulator	Indexed	2
A7	STA(A)	Store accumulator	Indexed	2
A8	EOR(A)	Exclusive OR	Indexed	2
A9	ADC(A)	Add with carry	Indexed	2
AA	ORA(A)	Inclusive OR	Indexed	2
AB	ADD(A)	Add without carry	Indexed	2
AC	CPX	Compare index register	Indexed	2
AD	JSR	Jump to subroutine	Indexed	2
AE	LDS	Load stack pointer	Indexed	2
AF	STS	Store stack pointer	Indexed	2
B0	SUB(A)	Subtract	Extended	3
B1	CMP(A)	Compare	Extended	3
B2	SBC(A)	Subtract with carry	Extended	3
B3	*			
B4	AND(A)	Logical AND	Extended	3
B5	BIT(A)	Bit test	Extended	3
B6	LDA(A)	Load accumulator	Extended	3
B7	STA(A)	Store accumulator	Extended	3
B8	EOR(A)	Exclusive OR	Extended	3
B9	ADC(A)	Add with carry	Extended	3
BA	ORA(A)	Inclusive OR	Extended	3
BB	ADD(A)	Add without carry	Extended	3
BC	CPX	Compare index register	Extended	3
BD	JSR	Jump to subroutine	Extended	3
BE	LDS(A)	Load stack pointer	Extended	3
BF	STS(A)	Store stack pointer	Extended	3

Machine Language to Assembly Language Conversion Table (continued)

Hex Code	Mnemonic Code	Mnemonic Description	Mode	Number of Bytes
C0	SUM(B)	Subtract	Immediate	3
C1	CMP(B)	Compare	Immediate	2
C2	SBC(B)	Subtract with carry	Immediate	2
C3	*			
C4	AND(B)	Logical AND	Immediate	2
C5	BIT(B)	Bit test	Immediate	2
C6	LDA(B)	Load accumulator	Immediate	2
C7	*			
C8	EOR(B)	Exclusive OR	Immediate	2
C9	ADC(B)	Add with carry	Immediate	2
CA	ORA(B)	Inclusive OR	Immediate	2
CB	ADD(B)	Add without carry	Immediate	2
CC	*			
CD	*			
CE	LDX	Load index register	Immediate	3
CF	*			
D0	SUB(B)	Subtract	Direct	2
D1	CMP(B)	Compare	Direct	2
D2	SBC(B)	Subtract with carry	Direct	2
D3	*			
D4	AND(B)	Logical AND	Direct	2
D5	BIT(B)	Bit test	Direct	2
D6	LDA(B)	Load accumulator	Direct	2
D7	STA(B)	Store accumulator	Direct	2
D8	EOR(B)	Exclusive OR	Direct	2
D9	ADC(B)	Add with carry	Direct	2
DA	ORA(B)	Inclusive OR	Direct	2
DB	ADD(B)	Add without carry	Direct	2
DC	*			
DD	*			
DE	LDX	Load index register	Direct	2
DF	STX	Store index register	Direct	2
E0	SUB(B)	Subtract	Indexed	2
E1	CMP(B)	Compare	Indexed	2
E2	SBC(B)	Subtract with carry	Indexed	2
E3	*			
E4	AND(B)	Logical AND	Indexed	2
E5	BIT(B)	Bit test	Indexed	2
E6	LDA(B)	Load accumulator	Indexed	2
E7	STAB)	Store accumulator	Indexed	2
E8	EOR(B)	Exclusive OR	Indexed	2
E9	ADC(B)	Add with carry	Indexed	2
EA	ORA(B)	Inclusive OR	Indexed	2
EB	ADD(B)	Add without carry	Indexed	2
EC	*			
ED	*			
EE	LDX	Load index register	Indexed	2
EF	STX	Store index register	Indexed	2

Machine Language to Assembly Language Conversion Table (continued)

Hex Code	Mnemonic Code	Mnemonic Description	Mode	Number of Bytes
F0	SUB(B)	Subtract	Extended	3
F1	CMP(B)	Compare	Extended	3
F2	SBC(B)	Subtract with carry	Extended	3
F3	*			
F4	AND(B)	Logical AND	Extended	3
F5	BIT(B)	Bit test	Extended	3
F6	LDA(B)	Load accumulator	Extended	3
F7	STA(B)	Store accumulator	Extended	3
F8	EOR(B)	Exclusive	Extended	3
F9	ADC(B)	Add with carry	Extended	3
FA	ORA(B)	Inclusive OR	Extended	3
FB	ADD(B)	Add without carry	Extended	3
FC	*			
FD	*			
FE	LDX	Load index register	Extended	3
FF	STX	Store index register	Extended	3

Assembly Language to Machine Language Conversion Table

Mnemonic Code	Hex Code	Mnemonic Description	Mode	Number of Bytes
ADA	1B	Add B to A	Inherent	1
ADC(A)	A9	Add with carry	Indexed	2
ADC(A)	B9	Add with carry	Extended	3
ADC(B)	C9	Add with carry	Immediate	2
ADC(B)	D9	Add with carry	Direct	2
ADC(B)	E9	Add with carry	Indexed	2
ADC(B)	F9	Add with carry	Extended	3
ADD(A)	AB	Add without carry	Indexed	2
ADD(A)	BB	Add without carry	Extended	3
ADD(B)	CB	Add without carry	Immediate	2
ADD(B)	DB	Add without carry	Direct	2
ADD(B)	EB	Add without carry	Indexed	2
ADD(B)	FB	Add without carry	Extended	3
AND(A)	A4	Logical AND	Indexed	2
AND(A)	B4	Logical AND	Extended	3
AND(B)	C4	Logical AND	Immediate	2
AND(B)	D4	Logical AND	Direct	2
AND(B)	E4	Logical AND	Indexed	2
AND(B)	F4	Logical AND	Extended	3
ASL	68	Arithmetic shift left	Indexed	2
ASL(A)	48	Arithmetic shift left		1
ASL(B)	58	Arithmetic shift left		1
ASR	67	Arithmetic shift right	Indexed	2
ASR(A)	47	Arithmetic right shift		1
ASR(B)	57	Arithmetic shift right		1
BCC	24	Branch if carry clear	Relative	2
BCS	25	Branch if carry set	Relative	2
BEQ	27	Branch if equal	Relative	2
BGE	2C	Branch if greater than zero or equal to	Relative	2
BGT	2E	Branch if greater than zero	Relative	2
BHI	22	Branch if higher	Relative	2
BIT(A)	A5	Bit test	Indexed	2
BIT(A)	B5	Bit test	Extended	3
BIT(B)	C5	Bit test	Immediate	2
BIT(B)	D5	Bit test	Direct	2
BIT(B)	E5	Bit test	Indexed	2
BIT(B)	F5	Bit test	Extended	3
BLE	2F	Branch if less than zero or equal to	Relative	2
BLS	23	Branch if lower or same	Relative	2
BLT	2D	Branch if less than zero	Relative	2
BMI	2B	Branch if minus	Relative	2
BNE	26	Branch if not equal	Relative	2
BPL	2A	Branch if plus	Relative	2
BRA	20	Branch always	Relative	2
BVC	28	Branch if overflow clear	Relative	2
BVS	29	Branch if overflow set	Relative	2

Assembly Language to Machine Language Conversion Table (continued)

Mnemonic Code	Hex Code	Mnemonic Description	Mode	Number of Bytes
CBA	11	Compare accumulator	Inherent	1
CLC	0C	Clear carry	Inherent	1
CLI	0E	Clear interrupt mask	Inherent	1
CLR	6F	Clear	Indexed	2
CLR(A)	4F	Clear		1
CLR(B)	5F	Clear		1
CLV	0A	Clear 2's complement overflow bit	Inherent	1
CMP(A)	A1	Compare	Indexed	2
CMP(A)	B1	Compare	Extended	3
CMP(B)	C1	Compare	Immediate	2
CMP(B)	D1	Compare	Direct	2
CMP(B)	E1	Compare	Indexed	2
CMP(B)	F1	Compare	Extended	3
COM	63	Complement	Indexed	2
COM	73	Complement	Extended	3
COM(A)	43	Complement		1
COM(B)	53	Complement		1
CPX	AC	Compare index register	Indexed	2
CPX	BC	Compare index register	Extended	3
DAA	19	Decimal Adjust (A)	Inherent	1
DEC	6A	Decrement	Indexed	2
DEC(A)	4A	Decrement		1
DEC(B)	5A	Decrement		1
DES	34	Decrement stack pointer	Inherent	1
DEX	09	Decrement index register	Inherent	1
EOR(A)	A8	Exclusive OR	Indexed	2
EOR(A)	B8	Exclusive OR	Extended	3
EOR(B)	C8	Exclusive OR	Immediate	2
EOR(B)	D8	Exclusive OR	Direct	2
EOR(B)	E8	Exclusive OR	Indexed	2
EOR(B)	F8	Exclusive	Extended	3
INC	6C	Increment	Indexed	2
INC(A)	4C	Increment		1
INC(B)	5C	Increment		1
INS	31	Increment stack pointer	Inherent	1
INX	08	Increment index register	Inherent	1
JMP	6E	Jump	Indexed	2
JSR	AD	Jump to subroutine	Indexed	2
JSR	BD	Jump to subroutine	Extended	3
LDA(A)	A6	Load accumulator	Indexed	2
LDA(A)	B6	Load accumulator	Extended	3
LDA(B)	C6	Load accumulator	Immediate	2
LDA(B)	D6	Load accumulator	Direct	2
LDA(B)	E6	Load accumulator	Indexed	2
LDA(B)	F6	Load accumulator	Extended	3
LDS	AE	Load stack pointer	Indexed	2
LDS(A)	BE	Load stack pointer	Extended	3

Assembly Language to Machine Language Conversion Table (continued)

Mnemonic Code	Hex Code	Mnemonic Description	Mode	Number of Bytes
LDX	CE	Load index register	Immediate	3
LDX	DE	Load index register	Direct	2
LDX	EE	Load index register	Indexed	2
LDX	FE	Load index register	Extended	3
LSR	64	Logical shift right	Indexed	2
LSR	74	Logical shift right	Extended	3
LSR(A)	44	Logical shift right		1
LSR(B)	54	Logical shift right		1
NEG	60	Negate	Indexed	2
NEG	70	Negate	Extended	3
NEG(A)	40	Negate		1
NEG(B)	50	Negate		1
NOP	01	No operation	Inherent	1
ORA(A)	AA	Inclusive OR	Indexed	2
ORA(A)	BA	Inclusive OR	Extended	3
ORA(B)	CA	Inclusive OR	Immediate	2
ORA(B)	DA	Inclusive OR	Direct	2
ORA(B)	EA	Inclusive OR	Indexed	2
ORA(B)	FA	Inclusive OR	Extended	3
PSH(A)	36	Push data on stack		1
PSH(B)	37	Push data on stack		1
PUL(A)	32	Pull data from stack		1
PUL(B)	33	Pull -data from stack		1
ROL	69	Rotate left	Indexed	2
ROL(A)	49	Rotate left		1
ROL(B)	59	Rotate left		1
ROR	66	Rotate Right	Indexed	2
ROR(A)	46	Rotate right		1
ROR(B)	56	Rotate right		1
RTI	3B	Return from interrupt	Inherent	1
RTS	39	Return from subroutine	Inherent	1
SBA	10	Subtract accumulator	Inherent	1
SBC(A)	A2	Subtract with carry	Indexed	2
SBC(A)	B2	Subtract with carry	Extended	3
SBC(B)	C2	Subtract with carry	Immediate	2
SBC(B)	D2	Subtract with carry	Direct	2
SBC(B)	E2	Subtract with carry	Indexed	2
SBC(B)	F2	Subtract with carry	Extended	3
SEC	0D	Set carry	Inherent	1
SEI	0F	Set interrupt mask	Inherent	1
SEV	0B	Set 2's complement overflow bit	Inherent	1
STA(A)	A7	Store accumulator	Indexed	2
STA(A)	B7	Store accumulator	Extended	3
STA(B)	D7	Store accumulator	Direct	2
STA(B)	F7	Store accumulator	Extended	3
STAB)	E7	Store accumulator	Indexed	2

Assembly Language to Machine Language Conversion Table (continued)

Mnemonic Code	Hex Code	Mnemonic Description	Mode	Number of Bytes
STS	AF	Store stack pointer	Indexed	2
STS	9F	Store stack pointer	Direct	2
STS(A)	BF	Store stack pointer	Extended	3
STX	DF	Store index register	Direct	2
STX	EF	Store index register	Indexed	2
STX	FF	Store index register	Extended	3
SUB(A)	A0	Subtract	Indexed	2
SUB(A)	B0	Subtract	Extended	3
SUB(B)	D0	Subtract	Direct	2
SUB(B)	E0	Subtract	Indexed	2
SUB(B)	F0	Subtract	Extended	3
SUM(B)	C0	Subtract	Immediate	3
SWI	3F	Software interrupt	Inherent	1
TAB	16	Transfer from A to B	Inherent	1
TAP	06	Transfer from accumulator A to process code register	Inherent	1
TBA	17	Transfer from B to A	Inherent	1
TPA	07	Transfer from process code register to accumulator A	Inherent	1
TST	6D	Test	Indexed	2
TST(A)	4D	Test		1
TST(B)	5D	Test		1
TSX	30	Transfer from stack pointer to index register	Inherent	1
TXS	35	Transfer from index register to stack pointer	Inherent	1
WAI	3E	Wait for interrupt	Inherent	1