

INDEX REGISTER AND STACK			IMMED			DIRECT			INDEX			EXTND			INHER			BOOLEAN/ARITHMETIC OPERATION						
OPERATIONS	MNEMONIC	OP	~	#	OP	~	#	OP	~	#	OP	~	#	OP	~	#	H	I	N	Z	V	C		
Compare Index Reg	CPX	8C	3	3	9C	4	2	AC	6	2	BC	5	3	09	4	1	(X _H /X _L) - (M/M + 1)	•	•	•	⑦	↑	⑧	•
Decrement Index Reg	DEX													34	4	1	X - 1 → X	•	•	•	•	•	•	•
Decrement Stack Pntr	DES													08	4	1	SP - 1 → SP	•	•	•	•	•	•	•
Increment index Reg	INX													31	4	1	X + 1 → X	•	•	•	•	•	•	•
Increment Stack Pntr	INS																SP + 1 → SP	•	•	•	•	•	•	•
Load Index Reg	LDX	CE	3	3	DE	4	2	EE	6	2	FE	5	3				M → X _H , (M + 1) → X _L	•	•	•	⑨	↑	R	•
Load Stack Pntr	LDS	8E	3	3	9E	4	2	AE	6	2	BE	5	3				M → SP _H , (M + 1) → SP _L	•	•	•	⑨	↑	R	•
Store Index Reg	STX				DF	5	2	EF	7	2	FF	6	3				X _H → M, X _L → (M + 1)	•	•	•	⑨	↑	R	•
Store Stack Pntr	STS				9F	5	2	AF	7	2	BF	6	3				SP _H → M, SP _L → (M + 1)	•	•	•	⑨	↑	R	•
Indx Reg → Stack Pntr	TXS													35	4	1	X - 1 → SP	•	•	•	•	•	•	•
Stack Pntr → Indx Reg	TSX													30	4	1	SP + 1 → X	•	•	•	•	•	•	•

JUMP AND BRANCH			RELATIVE			INDEX			EXTND			INHER			BRANCH TEST							
OPERATIONS	MNEMONIC	OP	~	#	OP	~	#	OP	~	#	OP	~	#	H	I	N	Z	V	C			
Branch Always	BRA	20	4	2																		
Branch If Carry Clear	BCC	24	4	2																		
Branch If Carry Set	BCS	25	4	2																		
Branch If = Zero	BEQ	27	4	2																		
Branch If ≥ Zero	BGE	2C	4	2																		
Branch If > Zero	BGT	2E	4	2																		
Branch If Higher	BHI	22	4	2																		
Branch If ≤ Zero	BLE	2F	4	2																		
Branch If Lower Or Same	BLS	23	4	2																		
Branch If < Zero	BLT	2D	4	2																		
Branch If Minus	BMI	2B	4	2																		
Branch If Not Equal Zero	BNE	26	4	2																		
Branch If Overflow Clear	BVC	28	4	2																		
Branch If Overflow Set	BVS	29	4	2																		
Branch If Plus	BPL	2A	4	2																		
Branch To Subroutine	BSR	8D	8	2																		
Jump	JMP				6E	4	2	7E	3	3												
Jump To Subroutine	JSR				AD	8	2	BD	9	3												
No Operation	NOP													01	2	1						
Return From Interrupt	RTI													3B	10	1						
Return From Subroutine	RTS													39	5	1						
Software Interrupt	SWI													3F	12	1						
Wait for Interrupt	WAI													3E	9	1						

CONDITIONS CODE REGISTER			INHER			BOOLEAN OPERATION						5	4	3	2	1	0
OPERATIONS	MNEMONIC	OP	~	#	OP	H	I	N	Z	V	C						
Clear Carry	CLC	0C	2	1	0 → C	•	•	•	•	•	R						
Clear Interrupt Mask	CLI	0E	2	1	0 → I	•	R	•	•	•	•						
Clear Overflow	CLV	0A	2	1	0 → V	•	•	•	•	•	R						
Set Carry	SEC	0D	2	1	1 → C	•	•	•	•	•	S						
Set Interrupt Mask	SEI	0F	2	1	1 → I	•	S	•	•	•	•						
Set Overflow	SEV	0B	2	1	1 → V	•	•	•	•	•	S						
Acmltr A → CCR	TAP	06	2	1	A → CCR						⑫						
CCR → Acmltr A	TPA	07	2	1	CCR → A	•	•	•	•	•	•						

CONDITION CODE REGISTER NOTES:

(Bit set if test is true and cleared otherwise)

- ① (Bit V) Test: Result = 10000000?
- ② (Bit C) Test: Result = 00000000?
- ③ (Bit C) Test: Decimal value of most significant BCD Character greater than nine?
(Not cleared if previously set.)
- ④ (Bit V) Test: Operand = 10000000 prior to execution?
- ⑤ (Bit V) Test: Operand = 01111111 prior to execution?
- ⑥ (Bit V) Test: Set equal to result of N + C after shift has occurred.
- ⑦ (Bit N) Test: Sign bit of most significant (MS) byte of result = 1?
- ⑧ (Bit V) Test: 2's complement overflow from subtraction of LS bytes?
- ⑨ (Bit N) Test: Result less than zero? (Bit 15 = 1)
- ⑩ (All) Load Condition Code Register from Stack. (See Special Operations)
- ⑪ (Bit I) Set when interrupt occurs. If previously set, a Non-Maskable Interrupt is required to exit the wait state.
- ⑫ (ALL) Set according to the contents of Accumulator A.

LEGEND:

- 00 Byte = Zero;
- OP Operation Code (Hexadecimal);
<li