19 Instruction Lists

Opcode			Available on:			# of	# of
Hex	*Mnemonic*	*Addressing Mode*	*6502*	*65C02*	*65802/816*	*Bytes*	*Cycles*
00	BRK	Stack/Interrupt	x	x	x	2**	7[9]
01	ORA	DP Indexed Indirect,X	x	x	x	2	6[1,2]
02	COP	Stack/Interrupt			x	2**	7[9]
03	ORA	Stack Relative			x	2	4[1]
04	TSB	Direct Page		x	x	2	5[2,5]
05	ORA	Direct Page	x	x	x	2	3[1,2]
06	ASL	Direct Page	x	x	x	2	5[2,5]
07	ORA	DP Indirect Long			x	2	6[1,2]
08	PHP	Stack (Push)	x	x	x	1	3
09	ORA	Immediate	x	x	x	2*	2[1]
0A	ASL	Accumulator	x	x	x	1	2
0B	PHD	Stack (Push)			x	1	4
0C	TSB	Absolute		x	x	3	6[5]
0D	ORA	Absolute	x	x	x	3	4[1]
0E	ASL	Absolute	x	x	x	3	6[5]
0F	ORA	Absolute Long			x	4	5[1]
10	BPL	Program Counter Relative	x	x	x	2	2[7,8]
11	ORA	DP Indirect Indexed,Y	x	x	x	2	5[1,2,3]
12	ORA	DP Indirect		x	x	2	5[1,2]
13	ORA	SR Indirect Indexed,Y			x	2	7[1]
14	TRB	Direct Page		x	x	2	5[2,5]
15	ORA	DP Indexed,X	x	x	x	2	4[1,2]
16	ASL	DP Indexed,X	x	x	x	2	6[2,5]
17	ORA	DP Indirect Long Indexed,Y			x	2	6[1,2]
18	CLC	Implied	x	x	x	1	2
19	ORA	Absolute Indexed,Y	x	x	x	3	4[1,3]
1A	INC	Accumulator		x	x	1	2
1B	TCS	Implied			x	1	2
1C	TRB	Absolute		x	x	3	6[5]
1D	ORA	Absolute Indexed,X	x	x	x	3	4[1,3]
1E	ASL	Absolute Indexed,X	x	x	x	3	7[5,6]
1F	ORA	Absolute Long Indexed,X			x	4	5[1]
20	JSR	Absolute	x	x	x	3	6
21	AND	DP Indexed Indirect,X	x	x	x	2	6[1,2]

Continued.

Opcode			Available on:			# of	# of
Hex	Mnemonic	Addressing Mode	6502	65C02	65802/816	Bytes	Cycles
22	JSR	Absolute Long			x	4	8
23	AND	Stack Relative			x	2	4[1]
24	BIT	Direct Page	x	x	x	2	3[1,2]
25	AND	Direct Page	x	x	x	2	3[1,2]
26	ROL	Direct Page	x	x	x	2	5[2,5]
27	AND	DP Indirect Long			x	2	6[1,2]
28	PLP	Stack (Pull)	x	x	x	1	4
29	AND	Immediate	x	x	x	2*	2[1]
2A	ROL	Accumulator	x	x	x	1	2
2B	PLD	Stack (Pull)			x	1	5
2C	BIT	Absolute	x	x	x	3	4[1]
2D	AND	Absolute	x	x	x	3	4[1]
2E	ROL	Absolute	x	x	x	3	6[5]
2F	AND	Absolute Long			x	4	5[1]
30	BMI	Program Counter Relative	x	x	x	2	2[7,8]
31	AND	DP Indirect Indexed,Y	x	x	x	2	5[1,2,3]
32	AND	DP Indirect		x	x	2	5[1,2]
33	AND	SR Indirect Indexed,Y			x	2	7[1]
34	BIT	DP Indexed,X		x	x	2	4[1,2]
35	AND	DP Indexed,X	x	x	x	2	4[1,2]
36	ROL	DP Indexed,X	x	x	x	2	6[2,5]
37	AND	DP Indirect Long Indexed,Y			x	2	6[1,2]
38	SEC	Implied	x	x	x	1	2
39	AND	Absolute Indexed,Y	x	x	x	3	4[1,3]
3A	DEC	Accumulator		x	x	1	2
3B	TSC	Implied			x	1	2
3C	BIT	Absolute Indexed,X		x	x	3	4[1,3]
3D	AND	Absolute Indexed,X	x	x	x	3	4[1,3]
3E	ROL	Absolute Indexed,X	x	x	x	3	7[5,6]
3F	AND	Absolute Long Indexed,X			x	4	5[1]
40	RTI	Stack/RTI	x	x	x	1	6[9]
41	EOR	DP Indexed Indirect,X	x	x	x	2	6[1,2]
42	WDM				x	2[16]	[16]
43	EOR	Stack Relative			x	2	4[1]
44	MVP	Block Move			x	3	[13]
45	EOR	Direct Page	x	x	x	2	3[1,2]
46	LSR	Direct Page	x	x	x	2	5[2,5]
47	EOR	DP Indirect Long			x	2	6[1,2]
48	PHA	Stack (Push)	x	x	x	1	3[1]
49	EOR	Immediate	x	x	x	2*	2[1]
4A	LSR	Accumulator	x	x	x	1	2

Opcode			Available on:			# of	# of
Hex	*Mnemonic*	*Addressing Mode*	*6502*	*65C02*	*65802/816*	*Bytes*	*Cycles*
4B	PHK	Stack (Push)			x	1	3
4C	JMP	Absolute	x	x	x	3	3
4D	EOR	Absolute	x	x	x	3	4[1]
4E	LSR	Absolute	x	x	x	3	6[5]
4F	EOR	Absolute Long			x	4	5[1]
50	BVC	Program Counter Relative	x	x	x	2	2[7,8]
51	EOR	DP Indirect Indexed,Y	x	x	x	2	5[1,2,3]
52	EOR	DP Indirect		x	x	2	5[1,2]
53	EOR	SR Indirect Indexed,Y			x	2	7[1]
54	MVN	Block Move			x	3	[13]
55	EOR	DP Indexed,X	x	x	x	2	4[1,2]
56	LSR	DP Indexed,X	x	x	x	2	6[2,5]
57	EOR	DP Indirect Long Indexed,Y			x	2	6[1,2]
58	CLI	Implied	x	x	x	1	2
59	EOR	Absolute Indexed,Y	x	x	x	3	4[1,3]
5A	PHY	Stack (Push)		x	x	1	3[10]
5B	TCD	Implied			x	1	2
5C	JMP	Absolute Long			x	4	4
5D	EOR	Absolute Indexed,X	x	x	x	3	4[1,3]
5E	LSR	Absolute Indexed,X	x	x	x	3	7[5,6]
5F	EOR	Absolute Long Indexed,X			x	4	5[1]
60	RTS	Stack (RTS)	x	x	x	1	6
61	ADC	DP Indexed Indirect,X	x	x	x	2	6[1,2,4]
62	PER	Stack (PC Relative Long)			x	3	6
63	ADC	Stack Relative			x	2	4[1,4]
64	STZ	Direct Page		x	x	2	3[1,2]
65	ADC	Direct Page	x	x	x	2	3[1,2,4]
66	ROR	Direct Page	x	x	x	2	5[2,5]
67	ADC	DP Indirect Long			x	2	6[1,2,4]
68	PLA	Stack (Pull)	x	x	x	1	4[1]
69	ADC	Immediate	x	x	x	2*	2[1,4]
6A	ROR	Accumulator	x	x	x	1	2
6B	RTL	Stack (RTL)			x	1	6
6C	JMP	Absolute Indirect	x	x	x	3	5[11,12]
6D	ADC	Absolute	x	x	x	3	4[1,4]
6E	ROR	Absolute	x	x	x	3	6[5]
6F	ADC	Absolute Long			x	4	5[1,4]
70	BVS	Program Counter Relative	x	x	x	2	2[7,8]
71	ADC	DP Indirect Indexed,Y	x	x	x	2	5[1,2,3,4]
72	ADC	DP Indirect		x	x	2	5[1,2,4]

Continued.

Opcode			Available on:			# of	# of
Hex	Mnemonic	Addressing Mode	6502	65C02	65802/816	Bytes	Cycles
73	ADC	SR Indirect Indexed,Y			x	2	7[1,4]
74	STZ	Direct Page Indexed,X		x	x	2	4[1,2]
75	ADC	DP Indexed,X	x	x	x	2	4[1,2,4]
76	ROR	DP Indexed,X	x	x	x	2	6[2,5]
77	ADC	DP Indirect Long Indexed,Y			x	2	6[1,2,4]
78	SEI	Implied	x	x	x	1	2
79	ADC	Absolute Indexed,Y	x	x	x	3	4[1,3,4]
7A	PLY	Stack/Pull		x	x	1	4[10]
7B	TDC	Implied			x	1	2
7C	JMP	Absolute Indexed Indirect		x	x	3	6
7D	ADC	Absolute Indexed,X	x	x	x	3	4[1,3,4]
7E	ROR	Absolute Indexed,X	x	x	x	3	7[5,6]
7F	ADC	Absolute Long Indexed,X			x	4	5[1,4]
80	BRA	Program Counter Relative		x	x	2	3[8]
81	STA	DP Indexed Indirect,X	x	x	x	2	6[1,2]
82	BRL	Program Counter Relative Long			x	3	4
83	STA	Stack Relative			x	2	4[1]
84	STY	Direct Page	x	x	x	2	3[2,10]
85	STA	Direct Page	x	x	x	2	3[1,2]
86	STX	Direct Page	x	x	x	2	3[2,10]
87	STA	DP Indirect Long			x	2	6[1,2]
88	DEY	Implied	x	x	x	1	2
89	BIT	Immediate		x	x	2*	2[1]
8A	TXA	Implied	x	x	x	1	2
8B	PHB	Stack (Push)			x	1	3
8C	STY	Absolute	x	x	x	3	4[10]
8D	STA	Absolute	x	x	x	3	4[1]
8E	STX	Absolute	x	x	x	3	4[10]
8F	STA	Absolute Long			x	4	5[1]
90	BCC	Program Counter Relative	x	x	x	2	2[7,8]
91	STA	DP Indirect Indexed,Y	x	x	x	2	6[1,2]
92	STA	DP Indirect		x	x	2	5[1,2]
93	STA	SR Indirect Indexed,Y			x	2	7[1]
94	STY	Direct Page Indexed,X	x	x	x	2	4[2,10]
95	STA	DP Indexed,X	x	x	x	2	4[1,2]
96	STX	Direct Page Indexed,Y	x	x	x	2	4[2,10]
97	STA	DP Indirect Long Indexed,Y			x	2	6[1,2]
98	TYA	Implied	x	x	x	1	2
99	STA	Absolute Indexed,Y	x	x	x	3	5[1]
9A	TXS	Implied	x	x	x	1	2
9B	TXY	Implied			x	1	2

Opcode Hex	Mnemonic	Addressing Mode	Available on: 6502	65C02	65802/816	# of Bytes	# of Cycles
9C	STZ	Absolute		x	x	3	4[1]
9D	STA	Absolute Indexed,X	x	x	x	3	5[1]
9E	STZ	Absolute Indexed,X		x	x	3	5[1]
9F	STA	Absolute Long Indexed,X			x	4	5[1]
A0	LDY	Immediate	x	x	x	2+	2[10]
A1	LDA	DP Indexed Indirect,X	x	x	x	2	6[1,2]
A2	LDX	Immediate	x	x	x	2+	2[10]
A3	LDA	Stack Relative			x	2	4[1]
A4	LDY	Direct Page	x	x	x	2	3[2,10]
A5	LDA	Direct Page	x	x	x	2	3[1,2]
A6	LDX	Direct Page	x	x	x	2	3[2,10]
A7	LDA	DP Indirect Long			x	2	6[1,2]
A8	TAY	Implied	x	x	x	1	2
A9	LDA	Immediate	x	x	x	2*	2[1]
AA	TAX	Implied	x	x	x	1	2
AB	PLB	Stack (Pull)			x	1	4
AC	LDY	Absolute	x	x	x	3	4[10]
AD	LDA	Absolute	x	x	x	3	4[1]
AE	LDX	Absolute	x	x	x	3	4[10]
AF	LDA	Absolute Long			x	4	5[1]
B0	BCS	Program Counter Relative	x	x	x	2	2[7,8]
B1	LDA	DP Indirect Indexed,Y	x	x	x	2	5[1,2,3]
B2	LDA	DP Indirect		x	x	2	5[1,2]
B3	LDA	SR Indirect Indexed,Y			x	2	7[1]
B4	LDY	DP Indexed,X	x	x	x	2	4[2,10]
B5	LDA	DP Indexed,X	x	x	x	2	4[1,2]
B6	LDX	DP Indexed,Y	x	x	x	2	4[2,10]
B7	LDA	DP Indirect Long Indexed,Y			x	2	6[1,2]
B8	CLV	Implied	x	x	x	1	2
B9	LDA	Absolute Indexed,Y	x	x	x	3	4[1,3]
BA	TSX	Implied	x	x	x	1	2
BB	TYX	Implied			x	1	2
BC	LDY	Absolute Indexed,X	x	x	x	3	4[3,10]
BD	LDA	Absolute Indexed,X	x	x	x	3	4[1,3]
BE	LDX	Absolute Indexed,Y	x	x	x	3	4[3,10]
BF	LDA	Absolute Long Indexed,X			x	4	5[1]
C0	CPY	Immediate	x	x	x	2+	2[10]
C1	CMP	DP Indexed Indirect,X	x	x	x	2	6[1,2]
C2	REP	Immediate			x	2	3
C3	CMP	Stack Relative			x	2	4[1]

Continued.

Opcode Hex	Mnemonic	Addressing Mode	Available on: 6502	65C02	65802/816	# of Bytes	# of Cycles
C4	CPY	Direct Page	x	x	x	2	3[2,10]
C5	CMP	Direct Page	x	x	x	2	3[1,2]
C6	DEC	Direct Page	x	x	x	2	5[2,5]
C7	CMP	DP Indirect Long			x	2	6[1,2]
C8	INY	Implied	x	x	x	1	2
C9	CMP	Immediate	x	x	x	2*	2[1]
CA	DEX	Implied	x	x	x	1	2
CB	WAI	Implied			x	1	3[15]
CC	CPY	Absolute	x	x	x	3	4[10]
CD	CMP	Absolute	x	x	x	3	4[1]
CE	DEC	Absolute	x	x	x	3	6[5]
CF	CMP	Absolute Long			x	4	5[1]
D0	BNE	Program Counter Relative	x	x	x	2	2[7,8]
D1	CMP	DP Indirect Indexed,Y	x	x	x	2	5[1,2,3]
D2	CMP	DP Indirect		x	x	2	5[1,2]
D3	CMP	SR Indirect Indexed,Y			x	2	7[1]
D4	PEI	Stack (Direct Page Indirect)			x	2	6[2]
D5	CMP	DP Indexed,X	x	x	x	2	4[1,2]
D6	DEC	DP Indexed,X	x	x	x	2	6[2,5]
D7	CMP	DP Indirect Long Indexed,Y			x	2	6[1,2]
D8	CLD	Implied	x	x	x	1	2
D9	CMP	Absolute Indexed,Y	x	x	x	3	4[1,3]
DA	PHX	Stack (Push)		x	x	1	3[10]
DB	STP	Implied			x	1	3[14]
DC	JMP	Absolute Indirect Long			x	3	6
DD	CMP	Absolute Indexed,X	x	x	x	3	4[1,3]
DE	DEC	Absolute Indexed,X	x	x	x	3	7[5,6]
DF	CMP	Absolute Long Indexed,X			x	4	5[1]
E0	CPX	Immediate	x	x	x	2+	2[10]
E1	SBC	DP Indexed Indirect,X	x	x	x	2	6[1,2,4]
E2	SEP	Immediate			x	2	3
E3	SBC	Stack Relative			x	2	4[1,4]
E4	CPX	Direct Page	x	x	x	2	3[2,10]
E5	SBC	Direct Page	x	x	x	2	3[1,2,4]
E6	INC	Direct Page	x	x	x	2	5[2,5]
E7	SBC	DP Indirect Long			x	2	6[1,2,4]
E8	INX	Implied	x	x	x	1	2
E9	SBC	Immediate	x	x	x	2*	2[1,4]
EA	NOP	Implied	x	x	x	1	2
EB	XBA	Implied			x	1	3
EC	CPX	Absolute	x	x	x	3	4[10]

Opcode Hex	Mnemonic	Addressing Mode	Available on: 6502	65C02	65802/816	# of Bytes	# of Cycles
ED	SBC	Absolute	x	x	x	3	4[1,4]
EE	INC	Absolute	x	x	x	3	6[5]
EF	SBC	Absolute Long			x	4	5[1,4]
F0	BEQ	Program Counter Relative	x	x	x	2	2[7,8]
F1	SBC	DP Indirect Indexed,Y	x	x	x	2	5[1,2,3,4]
F2	SBC	DP Indirect		x	x	2	5[1,2,4]
F3	SBC	SR Indirect Indexed,Y			x	2	7[1,4]
F4	PEA	Stack (Absolute)			x	3	5
F5	SBC	DP Indexed,X	x	x	x	2	4[1,2,4]
F6	INC	DP Indexed,X	x	x	x	2	6[2,5]
F7	SBC	DP Indirect Long Indexed,Y			x	2	6[1,2,4]
F8	SED	Implied	x	x	x	1	2
F9	SBC	Absolute Indexed,Y	x	x	x	3	4[1,3,4]
FA	PLX	Stack/Pull		x	x	1	4[10]
FB	XCE	Implied			x	1	2
FC	JSR	Absolute Indexed Indirect			x	3	8
FD	SBC	Absolute Indexed,X	x	x	x	3	4[1,3,4]
FE	INC	Absolute Indexed,X	x	x	x	3	7[5,6]
FF	SBC	Absolute Long Indexed,X			x	4	5[1,4]

* Add 1 byte if m = 0 (16-bit memory/accumulator)

** opcode is 1 byte, but program counter value pushed onto stack is incremented by 2 allowing for optional signature byte

\+ Add 1 byte if x = 0 (16-bit index registers)

[1] Add 1 cycle if m = 0 (16-bit memory/accumulator)

[2] Add 1 cycle if low byte of Direct Page register is other than zero (DL< >0)

[3] Add 1 cycle if adding index crosses a page boundary

[4] Add 1 cycle if 65C02 and d = 1 (decimal mode, 65C02)

[5] Add 2 cycles if m = 0 (16-bit memory/accumulator)

[6] Subtract 1 cycle if 65C02 and no page boundary crossed

[7] Add 1 cycle if branch is taken

[8] Add 1 more cycle if branch taken crosses page boundary on 6502, 65C02, or 65816/65802's 6502 emulation mode (e = 1)

[9] Add 1 cycle for 65802/65816 native mode (e = 0)

[10] Add 1 cycle if x = 0 (16-bit index registers)

[11] Add 1 cycle if 65C02

[12] 6502: If low byte of operand is $FF (i.e., operand is $xxFF): yields incorrect result

[13] 7 cycles per byte moved

[14] Uses 3 cycles to shut the processor down; additional cycles are required by reset to restart it

[15] Uses 3 cycles to shut the processor down; additional cycles are required by interrupt to restart it

[16] Byte and cycle counts subject to change in future processors which expand WDM into 2-byte opcode portions of instructions of varying lengths

Opcodes Reference Chart

Mnemonic	Operation	Immediate # const 1 ~ #	Absolute addr 2 ~ #	Absolute* Long long 3 ~ #	Direct Page (DP) dp 4 ~2 #	Accumulator A 5 ~ #	Implied 6 ~ #	DP Indirect Indexed,Y (dp), Y 7 ~2 #	DP* Indirect Long Indexed,Y [dp], Y 8 ~2 #	DP Indexed Indirect,X (dp, X) 9 ~2 #	DP Indexed,X dp, X 10 ~2 #	DP Indexed,Y dp, Y 11 ~2 #	Absolute Indexed,X addr, X 12 ~ #
ADC	A+M+C→A (1)(4)	69 2 2*	6D 4 3	6F 5 4	65 3 2			71 5[3] 2	77 6 2	61 6 2	75 4 2		7D 4[3] 3
AND	A∧M→A (1)	29 2 2*	2D 4 3	2F 5 4	25 3 2			31 5[3] 2	37 6 2	21 6 2	35 4 2		3D 4[3] 3
ASL	C← 15/7 0 ←0		0E 6[5] 3		06 5[5] 2	0A 2 1					16 6[5] 2		1E 7[5,6] 3
BCC	Branch if C = 0 (BLT) (7)												
BCS	Branch if C = 1 (BGE) (7)												
BEQ	Branch if Z = 1 (7)												
BIT	A∧M (1)	89● 2 2*	2C 4 3		24 3 2						34● 4 2		3C● 4[3] 3
BMI	Branch if N = 1 (7)												
BNE	Branch if Z = 0 (7)												
BPL	Branch if N = 0 (7)												
BRA	Branch always												
BRK	Break (9)												
BRL	Branch long always												
BVC	Branch if V = 0 (7)												
BVS	Branch if V = 1 (7)												
CLC	0→C						18 2 1						
CLD	0→D						D8 2 1						
CLI	0→I						58 2 1						
CLV	0→V						B8 2 1						
CMP	A – M (1)	C9 2 2*	CD 4 3	CF 5 4	C5 3 2			D1 5[3] 2	D7 6 2	C1 6 2	D5 4 2		DD 4[3] 3
COP	Co-processor (9)												
CPX	X – M (10)	E0 2 2+	EC 4 3		E4 3 2								
CPY	Y – M (10)	C0 2 2+	CC 4 3		C4 3 2								
DEC	Decrement		CE 6[5] 3		C6 5[5] 2	3A● 2 1					D6 6[5] 2		DE 7[5,6] 3
DEX	X – 1→X						CA 2 1						
DEY	Y – 1→Y						88 2 1						
EOR	A∀M→A (1)	49 2 2*	4D 4 3	4F 5 4	45 3 2			51 5[3] 2	57 6 2	41 6 2	55 4 2		5D 4[3] 3
INC	Increment		EE 6[5] 3		E6 5[5] 2	1A● 2 1					F6 6[5] 2		FE 7[5,6] 3
INX	X + 1→X						E8 2 1						
INY	Y + 1→Y						C8 2 1						
JMP	Jump to new location		4C 3 3	5C 4 4									
JSL	Jump long to subroutine			22 8 4									
JSR	Jump to subroutine		20 6 3										
LDA	M→A (1)	A9 2 2*	AD 4 3	AF 5 4	A5 3 2			B1 5[3] 2	B7 6 2	A1 6 2	B5 4 2		BD 4[3] 3
LDX	M→X (10)	A2 2 2+	AE 4 3		A6 3 2							B6 4 2	
LDY	M→Y (10)	A0 2 2+	AC 4 3		A4 3 2						B4 4 2		BC 4[3] 3
LSR	0→ 15/7 0 → C		4E 6[5] 3		46 5[5] 2	4A 2 1					56 6[5] 2		5E 7[5,6] 3
MVN	M→M Backward (start byte first)												
MVP	M→M Forward (end byte first)												
NOP	No operation						EA 2 1						
ORA	A∨M→A (1)	09 2 2*	0D 4 3	0F 5 4	05 3 2			11 5[3] 2	17 6 2	01 6 2	15 4 2		1D 4[3] 3
PEA	M(pc + 2)→M(s); M(pc + 1)→M(s – 1) S – 2 → S												
PEI	M(d + 1)→M(s); M(d)→M(s – 1) (2) S – 2 → S												
PER	PC + rl + 3→M(s), M(s – 1) S – 2 → S												
PHA	A→M(s); S – 2→S or S – 1→S (1)												
PHB	DBR→M(s); S – 1→S												
PHD	D→M(s), M(s – 1); S – 2→S												
PHK	PBR→M(s); S – 1→S												
PHP	P→M(s); S – 1→S												
PHX	X→M(s); S – 1→S or S – 2→S (10)												
PHY	Y→M(s); S – 1→S or S – 2→S (10)												
PLA	S + 1→S or S + 2→S; M(s)→A (1)												
PLB	S + 1→S; M(s)→DBR												
PLD	S + 2→S; M(s – 1), M(s)→D												
PLP	S + 1→S; M(s)→P												
PLX	S + 1→S or S + 2→S; M(s)→X (10)												
PLY	S + 1→S or S + 2→S; M(s)→Y (10)												
REP	M(pc + 1)∧P → P	C2 3 2											
ROL	15/7 0 ← C		2E 6[5] 3		26 5[5] 2	2A 2 1					36 6[5] 2		3E 7[5,6] 3
ROR	C → 15/7 0		6E 6[5] 3		66 5[5] 2	6A 2 1					76 6[5] 2		7E 7[5,6] 3
RTI	Return from interrupt (9)												
RTL	Return from subroutine long												
RTS	Return from subroutine												
SBC	A – M – C̄→A (1)(4)	E9 2 2*	ED 4 3	EF 5 4	E5 3 2			F1 5[3] 2	F7 6 2	E1 6 2	F5 4 2		FD 4[3] 3
SEC	1 → C						38 2 1						
SED	1 → D						F8 2 1						
SEI	1 → I						78 2 1						
SEP	M(pc + 1) V P → P	E2 3 2											
STA	A → M (1)		8D 4 3	8F 5 4	85 3 2			91 6 2	97 6 2	81 6 2	95 4 2		9D 5 3

Opcodes Reference Chart

Absolute * Long Indexed,X	Absolute Indexed,Y	PC Relative	Absolute* Indirect Long	Absolute Indirect	DP● Indirect	DP* Indirect Long	Absolute Indexed Indirect	Stack	Stack * Relative (SR)	SR* Indirect Indexed,Y	Block Move	Processor Status Code								Mne-monic
												7	6	5	4	3	2	1	0	
long,X	addr,Y	label	[addr]	(addr)	(dp)	[dp]	(addr,X)		sr,S	(sr,S),Y	srcbk,destbk	N	V	M	X	D	I	Z	C	E = 0
13 ~ #	14 ~ #	15 ~8 #	17 ~ #	17 ~ #	18 ~2 #	19 ~2 #	20 ~ #	21 ~ #	22 ~ #	23 ~ #	24 ~13 #	N	V	1	B	D	I	Z	C	E = 1
7F 5 4	79 4[3] 3				72 5 2	67 6 2			63 4 2	73 7 2		N	V	.	.	.	.	Z	C	ADC
3F 5 4	39 4[3] 3				32 5 2	27 6 2			23 4 2	33 7 2		N	.	.	.	.	.	Z	.	AND
												N	.	.	.	.	.	Z	C	ASL
		90 2 2										.	.	.	.	.	.	.	.	BCC
		B0 2 2										.	.	.	.	.	.	.	.	BCS
		F0 2 2										.	.	.	.	.	.	.	.	BEQ
												M7/15	M6/14	.	.	.	.	Z	.	(17) BIT
		30 2 2										.	.	.	.	.	.	.	.	BMI
		D0 2 2										.	.	.	.	.	.	.	.	BNE
		10 2 2										.	.	.	.	.	.	.	.	BPL
		80 3 2										.	.	.	.	.	.	.	.	● BRA
								00 7 2**				.	.	.	B	0	1	.	.	(18) BRK
		82 4 3										.	.	.	.	.	.	.	.	★ BRL
		50 2 2										.	.	.	.	.	.	.	.	BVC
		70 2 2										.	.	.	.	.	.	.	.	BVS
												.	.	.	.	.	.	.	0	CLC
												.	.	.	.	0	.	.	.	CLD
												.	.	.	.	.	0	.	.	CLI
												.	0	.	.	.	.	.	.	CLV
DF 5 4	D9 4[3] 3				D2 5 2	C7 6 2			C3 4 2	D3 7 2		N	.	.	.	.	.	Z	C	CMP
								02 7 2**				.	.	.	.	0	1	.	.	★ COP
												N	.	.	.	.	.	Z	C	CPX
												N	.	.	.	.	.	Z	C	CPY
												N	.	.	.	.	.	Z	.	DEC
												N	.	.	.	.	.	Z	.	DEX
												N	.	.	.	.	.	Z	.	DEY
5F 5 4	59 4[3] 3				52 5 2	47 6 2			43 4 2	53 7 2		N	.	.	.	.	.	Z	.	EOR
												N	.	.	.	.	.	Z	.	INC
												N	.	.	.	.	.	Z	.	INX
												N	.	.	.	.	.	Z	.	INY
			DC 6 3	6C 5[11,12] 3			7C● 6 3					.	.	.	.	.	.	.	.	JMP
												.	.	.	.	.	.	.	.	★ JSL
							FC* 8 3					.	.	.	.	.	.	.	.	JSR
BF 5 4	B9 4[3] 3				B2 5 2	A7 6 2			A3 4 2	B3 7 2		N	.	.	.	.	.	Z	.	LDA
	BE 4[3] 3											N	.	.	.	.	.	Z	.	LDX
												N	.	.	.	.	.	Z	.	LDY
												0	.	.	.	.	.	Z	C	LSR
											54 7/n 3	.	.	.	.	.	.	.	.	★ MVN
											44 7/n 3	.	.	.	.	.	.	.	.	★ MVP
												.	.	.	.	.	.	.	.	NOP
1F 5 4	19 4[3] 3				12 5 2	07 6 2			03 4 2	13 7 2		N	.	.	.	.	.	Z	.	ORA
								F4 5 3				.	.	.	.	.	.	.	.	★ PEA
								D4 6 2				.	.	.	.	.	.	.	.	★ PEI
								62 6 3				.	.	.	.	.	.	.	.	★ PER
								48 3 1				.	.	.	.	.	.	.	.	PHA
								8B 3 1				.	.	.	.	.	.	.	.	★ PHB
								0B 4 1				.	.	.	.	.	.	.	.	★ PHD
								4B 3 1				.	.	.	.	.	.	.	.	★ PHK
								08 3 1				.	.	.	.	.	.	.	.	PHP
								DA 3 1				.	.	.	.	.	.	.	.	● PHX
								5A 3 1				.	.	.	.	.	.	.	.	● PHY
								68 4 1				N	.	.	.	.	.	Z	.	PLA
								AB 4 1				N	.	.	.	.	.	Z	.	★ PLB
								2B 5 1				N	.	.	.	.	.	Z	.	★ PLD
								28 4 1				N	V	M	X	D	I	Z	C	PLP
								FA 4 1				N	.	.	.	.	.	Z	.	● PLX
								7A 4 1				N	.	.	.	.	.	Z	.	● PLY
												N	V	M	X	D	I	Z	C	★ REP
												N	.	.	.	.	.	Z	C	ROL
												N	.	.	.	.	.	Z	C	ROR
								40 6 1				N	V	M	X	D	I	Z	C	RTI
								6B 6 1				.	.	.	.	.	.	.	.	★ RTL
								60 6 1				.	.	.	.	.	.	.	.	RTS
FF 5 4	F9 4[3] 3				F2 5 2	E7 6 2			E3 4 2	F3 7 2		N	V	.	.	.	.	Z	C	SBC
												.	.	.	.	.	.	.	1	SEC
												.	.	.	.	1	.	.	.	SED
												.	.	.	.	.	1	.	.	SEI
												N	V	M	X	D	I	Z	C	★ SEP
9F 5 4	99 5 3				92 5 2	87 6 2			83 4 2	93 7 2		.	.	.	.	.	.	.	.	STA

Opcodes Reference Chart—(continued)

Mne-monic	Operation		Immediate # const 1 ~ #	Absolute addr 2 ~ #	Absolute * Long long 3 ~ #	Direct Page (DP) dp 4 ~[2] #	Accumulator A 5 ~ #	Implied 6 ~ #	DP Indirect Indexed,Y (dp), Y 7 ~[2] #	DP * Indirect Long Indexed,Y [dp], Y 8 ~[2] #	DP Indexed Indirect,X (dp, X) 9 ~[2] #	DP Indexed,X dp, X 10 ~[2] #	DP Indexed,Y dp, Y 11 ~[2] #	Absolute Indexed,X addr, X 12 ~ #
STP	Stop (1 → ϕ2)	(14)						DB 3 1						
STX	X → M	(10)		8E 4 3		86 3 2							96 4 2	
STY	Y → M	(10)		8C 4 3		84 3 2						94 4 2		
STZ	00 → M	(1)		9C 4 3		64 3 2						74 4 2		9E 5 3
TAX	A → X							AA 2 1						
TAY	A → Y							A8 2 1						
TCD	C → D							5B 2 1						
TCS	C → S							1B 2 1						
TDC	D → C							7B 2 1						
TRB	$\bar{A}$∧M → M	(5)		1C 6 3		14 5 2								
TSB	AVM → M	(5)		0C 6 3		04 5 2								
TSC	S → C							3B 2 1						
TSX	S → X							BA 2 1						
TXA	X → A							8A 2 1						
TXS	X → S							9A 2 1						
TXY	X → Y							9B 2 1						
TYA	Y → A							98 2 1						
TYX	Y → X							BB 2 1						
WAI	0 → RDY	(15)						CB 3 1						
WDM	No operation (reserved)	(16)						42 2^{16}						
XBA	B ↔ A							EB 3 1						
XCE	C ↔ E							FB 2 1						

Processor

• Opcode or instruction first introduced on the 65C02

* Opcode or instruction first introduced on the 65816/65802
(not marked: first introduced on the NMOS 6502)

Addressing mode box:

Immediate	Addressing mode
# const	Assembler operand syntax
1 ~ #	

- # — Number of bytes
- ~ — Number of cycles
- 1 — Key to detailed instruction operation chart (see Appendix E: 65816 Data Sheet)

Operation column:

A Accumulator
X Index register X
Y Index register Y
M Contents of memory location specified by effective address
M(d) Contents of direct page memory location pointed to by operand
M(s) Contents of memory location pointed to by stack pointer
M(pc) Current opcode pointed to by the program counter
PC Memory location of current opcode pointed to by the program counter
rl Two-byte operand of relative long addressing mode instruction
+ Add
− Subtract
∧ And
∨ Or
⊻ Exclusive Or
— Logical complement of a value or status bit ($\bar{A}$ indicates the complement of the value in the accumulator)
ϕ2 Phase 2 clock (hardware signal)
RDY Ready (hardware signal)

Opcodes Reference Chart—(continued)

Absolute* Long Indexed,X	Absolute Indexed,Y	PC Relative	Absolute Indirect Long	Absolute Indirect	DP● Indirect	DP* Indirect Long	Absolute Indexed Indirect	Stack	Stack* Relative (SR)	SR* Indirect Indexed,Y	Block Move	Processor Status Code 7	6	5	4	3	2	1	0	Mne-monic
long,X	addr,Y	label	[addr]	(addr)	(dp)	[dp]	(addr,X)		sr,S	(sr,S),Y	srcbk,destbk	N	V	M	X	D	I	Z	C	E = 0
13 ~ #	14 ~ #	15 ~[8] #	16 ~ #	17 ~ #	18 ~[2] #	19 ~[2] #	20 ~ #	21 ~ #	22 ~ #	23 ~ #	24 ~[13] #	N	V	1	B	D	I	Z	C	E = 1
												.	.	.	.	.	.	.	.	★ STP
												.	.	.	.	.	.	.	.	STX
												.	.	.	.	.	.	.	.	STY
												.	.	.	.	.	.	.	.	● STZ
												N	.	.	.	.	.	Z	.	TAX
												N	.	.	.	.	.	Z	.	TAY
												N	.	.	.	.	.	Z	.	★ TCD
												.	.	.	.	.	.	.	.	★ TCS
												N	.	.	.	.	.	Z	.	★ TDC
												.	.	.	.	.	.	Z	.	● TRB
												.	.	.	.	.	.	Z	.	● TSB
												N	.	.	.	.	.	Z	.	★ TSC
												N	.	.	.	.	.	Z	.	TSX
												N	.	.	.	.	.	Z	.	TXA
												.	.	.	.	.	.	.	.	TXS
												N	.	.	.	.	.	Z	.	★ TXY
												N	.	.	.	.	.	Z	.	TYA
												N	.	.	.	.	.	Z	.	★ TYX
												.	.	.	.	.	.	.	.	★ WAI
												.	.	.	.	.	.	.	.	★ WDM
												N	.	.	.	.	.	Z	.	★ XBA
												.	.	.	.	.	.	.	E	★ XCE

Bytes, cycles, and status codes:

* Add 1 byte if M=0 (16-bit memory/accumulator)

** opcode is 1 byte, but program counter value pushed onto stack is incremented by 2 allowing for optional signature byte

\+ Add 1 byte if x=0 (16-bit index registers)

n number of bytes moved

[1] Add 1 cycle if m=0 (16-bit memory/accumulator)

[2] Add 1 cycle if low byte of Direct Page register is other than zero (DL<>0)

[3] Add 1 cycle if adding index crosses a page boundary

[4] Add 1 cycle if 65C02 and d=1 (decimal mode, 65C02)

[5] Add 2 cycles if m=0 (16-bit memory/accumulator)

[6] Subtract 1 cycle if 65C02 and no page boundary crossed

[7] Add 1 cycle if branch is taken

[8] Add 1 more cycle if branch taken crosses page boundary on 6502, 65C02, or 65816/65802's 6502 emulation mode (e=1)

[9] Add 1 cycle for 65802/65816 native mode (e=0)

[10] Add 1 cycle if x=0 (16-bit index registers)

[11] Add 1 cycle if 65C02

[12] 6502: If low byte of *addr* is $FF (i.e., *addr* is $xxFF): yields incorrect result

[13] 7 cycles per byte moved

[14] Uses 3 cycles to shut the processor down; additional cycles are required by reset to restart it

[15] Uses 3 cycles to shut the processor down; additional cycles are required by interrupt to restart it

[16] Byte and cycle counts subject to change in future processors which expand WDM into 2-byte opcode portions of instructions of varying lengths

[17] BIT: immediate n and v flags not affected; if m=0, m(15) →n and M(14) →v; if m=1, M(7) →n and M(6) →v

[18] BRK: if b=1 in pushed status register (6502, 65C02 and emulation mode e=1), then interrupt was caused by software BRK;
if 6502, d is unaffected by BRK; if 65C02 or 65816/65802, d is 0 after BRK

Opcode Matrix

MSD	LSD 0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	MSD
0	BRK s 2 7	ORA (d,x) 2 6	COP s 2 * 7	ORA d,s 2 * 4	TSB d 2 ● 5	ORA d 2 3	ASL d 2 5	ORA [d] 2 * 6	PHP s 1 3	ORA # 2 2	ASL A 1 2	PHD s 1 * 4	TSB a 3 ● 6	ORA a 3 4	ASL a 3 6	ORA al 4 * 5	0
1	BPL r 2 2	ORA (d),y 2 5	ORA (d) 2 ● 5	ORA (d,s),y 2 * 7	TRB d 2 ● 5	ORA d,x 2 4	ASL d,x 2 6	ORA [d],y 2 * 6	CLC i 1 2	ORA a,y 3 4	INC A 1 ● 2	TCS i 1 * 2	TRB a 3 ● 6	ORA a,x 3 4	ASL a,x 3 7	ORA al,x 4 * 5	1
2	JSR a 3 6	AND (d,x) 2 6	JSL al 4 * 8	AND d,s 2 * 4	BIT d 2 3	AND d 2 3	ROL d 2 5	AND [d] 2 * 6	PLP s 1 4	AND # 2 2	ROL A 1 2	PLD s 1 * 5	BIT a 3 4	AND a 3 4	ROL a 3 6	AND al 4 * 5	2
3	BMI r 2 2	AND (d),y 2 5	AND (d) 2 ● 5	AND (d,s),y 2 * 7	BIT d,x 2 ● 4	AND d,x 2 4	ROL d,x 2 6	AND [d],y 2 * 6	SEC i 1 2	AND a,y 3 4	DEC A 1 ● 2	TSC i 1 * 2	BIT a,x 3 ● 4	AND a,x 3 4	ROL a,x 3 7	AND al,x 4 * 5	3
4	RTI s 1 6	EOR (d,x) 2 6	WDM 2 *	EOR d,s 2 * 4	MVP xyc 3 * 7	EOR d 2 3	LSR d 2 5	EOR [d] 2 * 6	PHA s 1 3	EOR # 2 2	LSR A 1 2	PHK s 1 * 3	JMP a 3 3	EOR a 3 4	LSR a 3 6	EOR al 4 * 5	4
5	BVC r 2 2	EOR (d),y 2 5	EOR (d) 2 ● 5	EOR (d,s),y 2 * 7	MVN xyc 3 * 7	EOR d,x 2 4	LSR d,x 2 6	EOR [d],y 2 * 6	CLI i 1 2	EOR a,y 3 4	PHY s 1 ● 3	TCD i 1 * 2	JMP al 4 * 4	EOR a,x 3 4	LSR a,x 3 7	EOR al,x 4 * 5	5
6	RTS s 1 6	ADC (d,x) 2 6	PER s 3 * 6	ADC d,s 2 * 4	STZ d 2 ● 3	ADC d 2 3	ROR d 2 5	ADC [d] 2 * 6	PLA s 1 4	ADC # 2 2	ROR A 1 2	RTL s 1 * 6	JMP (a) 3 5	ADC a 3 4	ROR a 3 6	ADC al 4 * 5	6
7	BVS r 2 2	ADC (d),y 2 5	ADC (d) 2 ● 5	ADC (d,s),y 2 * 7	STZ d,x 2 ● 4	ADC d,x 2 4	ROR d,x 2 6	ADC [d],y 2 * 6	SEI i 1 2	ADC a,y 3 4	PLY s 1 ● 4	TDC i 1 * 2	JMP (a,x) 3 ● 6	ADC a,x 3 4	ROR a,x 3 7	ADC al,x 4 * 5	7
8	BRA r 2 ● 3	STA (d,x) 2 6	BRL rl 3 * 4	STA d,s 2 * 4	STY d 2 3	STA d 2 3	STX d 2 3	STA [d] 2 * 6	DEY i 1 2	BIT # 2 ● 2	TXA i 1 2	PHB s 1 * 3	STY a 3 4	STA a 3 4	STX a 3 4	STA al 4 * 5	8
9	BCC r 2 2	STA (d),y 2 6	STA (d) 2 ● 5	STA (d,s),y 2 * 7	STY d,x 2 4	STA d,x 2 4	STX d,y 2 4	STA [d],y 2 * 6	TYA i 1 2	STA a,y 3 5	TXS i 1 2	TXY i 1 * 2	STZ a 3 ● 4	STA a,x 3 5	STZ a,x 3 ● 5	STA al,x 4 * 5	9
A	LDY # 2 2	LDA (d,x) 2 6	LDX # 2 2	LDA d,s 2 * 4	LDY d 2 3	LDA d 2 3	LDX d 2 3	LDA [d] 2 * 6	TAY i 1 2	LDA # 2 2	TAX i 1 2	PLB s 1 * 4	LDY a 3 4	LDA a 3 4	LDX a 3 4	LDA al 4 * 5	A
B	BCS r 2 2	LDA (d),y 2 5	LDA (d) 2 ● 5	LDA (d,s),y 2 * 7	LDY d,x 2 4	LDA d,x 2 4	LDX d,y 2 4	LDA [d],y 2 * 6	CLV i 1 2	LDA a,y 3 4	TSX i 1 2	TYX i 1 * 2	LDY a,x 3 4	LDA a,x 3 4	LDX a,y 3 4	LDA al,x 4 * 5	B
C	CPY # 2 2	CMP (d,x) 2 6	REP # 2 * 3	CMP d,s 2 * 4	CPY d 2 3	CMP d 2 3	DEC d 2 5	CMP [d] 2 * 6	INY i 1 2	CMP # 2 2	DEX i 1 2	WAI i 1 * 3	CPY a 3 4	CMP a 3 4	DEC a 3 6	CMP al 4 * 5	C
D	BNE r 2 2	CMP (d),y 2 5	CMP (d) 2 ● 5	CMP (d,s),y 2 * 7	PEI s 2 * 6	CMP d,x 2 4	DEC d,x 2 6	CMP [d],y 2 * 6	CLD i 1 2	CMP a,y 3 4	PHX s 1 ● 3	STP i 1 * 3	JML (a) 3 * 6	CMP a,x 3 4	DEC a,x 3 7	CMP al,x 4 * 5	D
E	CPX # 2 2	SBC (d,x) 2 6	SEP # 2 * 3	SBC d,s 2 * 4	CPX d 2 3	SBC d 2 3	INC d 2 5	SBC [d] 2 * 6	INX i 1 2	SBC # 2 2	NOP i 1 2	XBA i 1 * 3	CPX a 3 4	SBC a 3 4	INC a 3 6	SBC al 4 * 5	E
F	BEQ r 2 2	SBC (d),y 2 5	SBC (d) 2 ● 5	SBC (d,s),y 2 * 7	PEA s 3 * 5	SBC d,x 2 4	INC d,x 2 6	SBC [d],y 2 * 6	SED i 1 2	SBC a,y 3 4	PLX s 1 ● 4	XCE i 1 * 2	JSR (a,x) 3 * 8	SBC a,x 3 4	INC a,x 3 7	SBC al,x 4 * 5	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

Op Code Matrix Legend

INSTRUCTION MNEMONIC	★ = New W65C816/802 Opcodes ● = New W65C02 Opcodes Blank = NMOS 6502 Opcodes	ADDRESSING MODE
BASE NO. BYTES		BASE NO. CYCLES

symbol	addressing mode	symbol	addressing mode
#	immediate	[d]	direct indirect long
A	accumulator	[d],y	direct indirect long indexed
r	program counter relative	a	absolute
rl	program counter relative long	a,x	absolute indexed (with x)
i	implied	a,y	absolute indexed (with y)
s	stack	al	absolute long
d	direct	al,x	absolute long indexed
d,x	direct indexed (with x)	d,s	stack relative
d,y	direct indexed (with y)	(d,s),y	stack relative indirect indexed
(d)	direct indirect	(a)	absolute indirect
(d,x)	direct indexed indirect	(a,x)	absolute indexed indirect
(d),y	direct indirect indexed	xyc	block move