OPC	Bytes 02	MOS 6502	R6501Q R6511Q R6500/13	GTE65SC02 CMD65SC02	R65C12	R65C02 R65C102/112	WDC65C02	WDC65C816 WDC65C802	CSG65CE02	HuC6280	Renesas 740	Bytes 816	Bytes 740
00	1	BRK	BRK	BRK	BRK	BRK	BRK	BRK	BRK	BRK	BRK	1	1
01	2	ORA (zp,X)	ORA (zp,X)	ORA (zp,X)	ORA (zp,X)	ORA (zp,X)	ORA (zp,X)	ORA (dp,X)	ORA (bp,X)	ORA (zp,X)	ORA (zp, X)	2	2
02		-	-	-	-	-	-	COP const	* CLE	SXY	JSR (zp)	1	2
03	2	-	-	-	-	-	-	ORA sr,S	* SEE	-	BBS 0, A, rel	2	2
04	2	-	-	TSB zp	TSB zp	TSB zp	TSB zp	TSB dp	TSB bp	TSB zp	-	2	
05	2	ORA zp	ORA zp	ORA zp	ORA zp	ORA zp	ORA zp	ORA dp	ORA bp	ORA zp	ORA zp	2	2
06	2	ASL zp	ASL zp	ASL zp	ASL zp	ASL zp	ASL zp	ASL dp	ASL bp	ASL zp	ASL zp	2	2
07	2		RMB 0,zp	-	-	RMB 0,zp	RMB 0,zp	ORA (dp)	RMB 0,bp	RMB 0,zp	BBS 0, zp, rel	2	3
08	1	PHP	PHP	PHP	PHP	PHP	PHP	PHP	PHP	PHP	PHP	1	1
09	2	ORA #n	ORA #n	ORA #n	ORA #n	ORA #n	ORA #n	* ORA #const	ORA #n	ORA #n	ORA #n	2(3)	2
0A	1	ASL A	ASL A	ASL A	ASL A	ASL A	ASL A	ASL A	ASL A	ASL A	ASL A	1	1
0B		-	-	-	-	-	-	PHD	* TSY	-	SEB 0, A	1	1
0C	3	-	-	TSB abs	TSB abs	TSB abs	TSB abs	TSB abs	TSB abs	TSB abs	-	3	
0D	3	ORA abs	ORA abs	ORA abs	ORA abs	ORA abs	ORA abs	ORA abs	ORA abs	ORA abs	ORA abs	3	3
0E	3	ASL abs	ASL abs	ASL abs	ASL abs	ASL abs	ASL abs	ASL abs	ASL abs	ASL abs	ASL abs	3	3
0F	3	-	BBR 0,zp,rel	-	-	BBR 0,zp,rel	BBR 0,zp,rel	* ORA long	BBR 0,bp,rel	BBR 0,zp,rel	SEB 0, zp	4	2
10	2	BPL rel	BPL rel	BPL rel	BPL rel	BPL rel	BPL rel	BPL rel	BPL rel	BPL rel	BPL rel	2	2
11	2	ORA (zp),Y	ORA (zp),Y	ORA (zp),Y	ORA (zp),Y	ORA (zp),Y	ORA (zp),Y	ORA (dp),Y	ORA (bp),Y	ORA (zp),Y	ORA (zp), Y	2	2
12	2	-	-	ORA (zp)	ORA (zp)	ORA (zp)	ORA (zp)	ORA (dp)	ORA (bp)	ORA (zp)	CLT	2	1
13		-	-	-	-	-	-	ORA (sr,S),Y	* BPL rel16	ST1	BBC 0, A, rel	2	2
14	2	-	-	TRB zp	TRB zp	TRB zp	TRB zp	TRB dp	TRB bp	TRB zp	-	2	
15	2	ORA zp,X	ORA zp,X	ORA zp,X	ORA zp,X	ORA zp,X	ORA zp,X	ORA dp,X	ORA bp,X	ORA zp,X	ORA zp, X	2	2
16	2	ASL zp,X	ASL zp,X	ASL zp,X	ASL zp,X	ASL zp,X	ASL zp,X	ASL dp,X	ASL bp,X	ASL zp,X	ASL zp, X	2	2
17	3	-	RMB 1,zp	-	-	RMB 1,zp	RMB 1,zp	* ORA (dp),Y	RMB 1,bp	RMB 1,zp	BBC 0, zp, rel	2	3
18	1	CLC	CLC	CLC	CLC	CLC	CLC	CLC	CLC	CLC	CLC	1	1
19	3	ORA abs,Y	ORA abs,Y	ORA abs,Y	ORA abs,Y	ORA abs,Y	ORA abs,Y	ORA abs,Y	ORA abs,Y	ORA abs,Y	ORA abs, Y	3	3
1A	1	-	-	INC A	INC A	INC A	INC A	INC A	INC A	INC A	DEC A	1	1
1B	3	-	-	-	-	-	-	TCS	* INZ	-	CLB 0, A	1	1
1C	3	-	-	TRB abs	TRB abs	TRB abs	TRB abs	TRB abs	TRB abs	TRB abs	-	3	
1D	3	ORA abs,X	ORA abs,X	ORA abs,X	ORA abs,X	ORA abs,X	ORA abs,X	ORA abs,X	ORA abs,X	ORA abs,X	ORA abs, X	3	3
1E	3	ASL abs,X	ASL abs,X	ASL abs,X	ASL abs,X	ASL abs,X	ASL abs,X	ASL abs,X	ASL abs,X	ASL abs,X	ASL abs, X	3	3
1F	3	-	BBR 1,zp,rel	-	-	BBR 1,zp,rel	BBR 1,zp,rel	* ORA long,X	BBR 1,bp,rel	BBR 1,zp,rel	CLB 0, zp	4	2
20	3	JSR abs	JSR abs	JSR abs	JSR abs	JSR abs	JSR abs	JSR abs	JSR abs	JSR abs	JSR abs	3	3
21	2	AND (zp,X)	AND (zp,X)	AND (zp,X)	AND (zp,X)	AND (zp,X)	AND (zp,X)	AND (dp,X)	AND (bp,X)	AND (zp,X)	AND (zp, X)	2	2
22		-	-	-	-	-	-	JSR long	* JSR (abs)	SAX	JSR $FFsp	4	2
23		-	-	-	-	-	-	AND sr,S	* JSR (ABS,X)	ST2	BBS 1, A, rel	2	2
24	2	BIT zp	BIT zp	BIT zp	BIT zp	BIT zp	BIT zp	BIT dp	BIT bp	BIT zp	BIT zp	2	2
25	2	AND zp	AND zp	AND zp	AND zp	AND zp	AND zp	AND dp	AND bp	AND zp	AND zp	2	2
26	2	ROL zp	ROL zp	ROL zp	ROL zp	ROL zp	ROL zp	ROL dp	ROL bp	ROL zp	ROL zp	2	2
27		-	RMB 2,zp	-	-	RMB 2,zp	RMB 2,zp	* AND (dp)	RMB 2,bp	RMB 2,zp	BBS 1, zp, rel	2	3
28	1	PLP	PLP	PLP	PLP	PLP	PLP	PLP	PLP	PLP	PLP	1	1
29	2	AND #n	AND #n	AND #n	AND #n	AND #n	AND #n	* AND #const	AND #n	AND #n	AND #n	2(3)	2
2A	1	ROL A	ROL A	ROL A	ROL A	ROL A	ROL A	ROL A	ROL A	ROL A	ROL A	1	1
2B		-	-	-	-	-	-	PLD	* TYS	-	SEB 1, A	1	1
2C	3	BIT abs	BIT abs	BIT abs	BIT abs	BIT abs	BIT abs	BIT abs	BIT abs	BIT abs	BIT abs	3	3
2D	3	AND abs	AND abs	AND abs	AND abs	AND abs	AND abs	AND abs	AND abs	AND abs	AND abs	3	3
2E	3	ROL abs	ROL abs	ROL abs	ROL abs	ROL abs	ROL abs	ROL abs	ROL abs	ROL abs	ROL abs	3	3
2F	3	-	BBR 2,zp,rel	-	-	BBR 2,zp,rel	BBR 2,zp,rel	* AND long	BBR 2,bp,rel	BBR 2,zp,rel	SEB 1, zp	4	2
30	2	BMI rel	BMI rel	BMI rel	BMI rel	BMI rel	BMI rel	BMI rel	BMI rel	BMI rel	BMI rel	2	2
31	2	AND (zp),Y	AND (zp),Y	AND (zp),Y	AND (zp),Y	AND (zp),Y	AND (zp),Y	AND (dp),Y	AND (bp),Y	AND (zp),Y	AND (zp), Y	2	2
32	2	-	AND (zp)	AND (zp)	-	AND (zp)	AND (zp)	AND (dp)	AND (bp)	AND (zp)	SET	2	1
33		-	-	-	-	-	-	AND (sr,S),Y	* BMI rel16	-	BBC 1, A, rel	2	2
34	2	-	-	BIT zp,X	BIT zp,X	BIT zp,X	BIT zp,X	BIT dp,X	BIT bp,X	BIT zp,X	-	2	
35	2	AND zp,X	AND zp,X	AND zp,X	AND zp,X	AND zp,X	AND zp,X	AND dp,X	AND bp,X	AND zp,X	AND zp, X	2	2
36	2	ROL zp,X	ROL zp,X	ROL zp,X	ROL zp,X	ROL zp,X	ROL zp,X	ROL dp,X	ROL bp,X	ROL zp,X	ROL zp, X	2	2
37		-	RMB 3,zp	-	-	RMB 3,zp	RMB 3,zp	* AND (dp),Y	RMB 3,bp	RMB 3,zp	BBC 1, zp, rel	2	3
38	1	SEC	SEC	SEC	SEC	SEC	SEC	SEC	SEC	SEC	SEC	1	1
39	3	AND abs,Y	AND abs,Y	AND abs,Y	AND abs,Y	AND abs,Y	AND abs,Y	AND abs,Y	AND abs,Y	AND abs,Y	AND abs, Y	3	3
3A	1	-	-	DEC A	DEC A	DEC A	DEC A	DEC A	DEC A	DEC A	INC A	1	1
3B		-	-	-	-	-	-	TSC	* DEZ	-	CLB 1, A	1	1
3C	3	-	-	BIT abs,X	BIT abs,X	BIT abs,X	BIT abs,X	BIT abs,X	BIT abs,X	BIT abs,X	LDM #n, zp	3	3
3D	3	ORA abs,X	ORA abs,X	ORA abs,X	ORA abs,X	ORA abs,X	ORA abs,X	AND abs,X	ORA abs,X	ORA abs,X	AND abs, X	3	3
3E	3	ASL abs,X	ASL abs,X	ASL abs,X	ASL abs,X	ASL abs,X	ASL abs,X	ROL abs,X	ASL abs,X	ASL abs,X	ROL abs, X	3	3
3F	3	-	BBR 3,zp,rel	-	-	BBR 3,zp,rel	BBR 3,zp,rel	* AND long,X	BBR 3,bp,rel	BBR 3,zp,rel	CLB 1, zp	4	2
40	1	RTI	RTI	RTI	RTI	RTI	RTI	RTI	RTI	RTI	RTI	1	1
41	2	EOR (zp,X)	EOR (zp,X)	EOR (zp,X)	EOR (zp,X)	EOR (zp,X)	EOR (zp,X)	EOR (dp,X)	EOR (bp,X)	EOR (zp,X)	EOR (zp, X)	2	2
42		-	-	-	-	-	-	WDM	* NEG A	SAY	STP	2	1
43		-	-	-	-	-	-	EOR sr,S	* ASR A	TMA #n	BBS 2, A, rel	2	2
44		-	-	-	-	-	-	MVN sbnk,dbnk	* ASR bp	-	COM zp	3	2
45	2	EOR zp	EOR zp	EOR zp	EOR zp	EOR zp	EOR zp	EOR dp	EOR bp	EOR zp	EOR zp	2	2
46	2	LSR zp	LSR zp	LSR zp	LSR zp	LSR zp	LSR zp	LSR dp	LSR bp	LSR zp	LSR zp	2	2
47		-	RMB 4,zp	-	-	RMB 4,zp	RMB 4,zp	* EOR (dp)	RMB 4,bp	RMB 4,zp	BBS 2, zp, rel	2	3
48	1	PHA	PHA	PHA	PHA	PHA	PHA	PHA	PHA	PHA	PHA	1	1
49	2	EOR #n	EOR #n	EOR #n	EOR #n	EOR #n	EOR #n	* EOR #const	EOR #n	EOR #n	EOR #n	2(3)	2
4A	1	LSR A	LSR A	LSR A	LSR A	LSR A	LSR A	LSR A	LSR A	LSR A	LSR A	1	1
4B		-	-	-	-	-	-	PHK	* TAZ	-	SEB 2, A	1	1
4C	3	JMP abs	JMP abs	JMP abs	JMP abs	JMP abs	JMP abs	JMP abs	JMP abs	JMP abs	JMP abs	3	3
4D	3	EOR abs	EOR abs	EOR abs	EOR abs	EOR abs	EOR abs	EOR abs	EOR abs	EOR abs	EOR abs	3	3
4E	3	LSR abs	LSR abs	LSR abs	LSR abs	LSR abs	LSR abs	LSR abs	LSR abs	LSR abs	LSR abs	3	3
4F		-	BBR 4,zp,rel	-	-	BBR 4,zp,rel	BBR 4,zp,rel	* EOR long	BBR 4,bp,rel	BBR 4,zp,rel	SEB 2, zp	4	2
50	2	BVC rel	BVC rel	BVC rel	BVC rel	BVC rel	BVC rel	BVC rel	BVC rel	BVC rel	BVC rel	2	2
51	2	EOR (zp),Y	EOR (zp),Y	EOR (zp),Y	EOR (zp),Y	EOR (zp),Y	EOR (zp),Y	EOR (dp),Y	EOR (bp),Y	EOR (zp),Y	EOR (zp), Y	2	2
52	2	-	-	EOR (zp)	EOR (zp)	EOR (zp)	EOR (zp)	EOR (dp)	EOR (bp)	EOR (zp)	-	2	
53		-	-	-	-	-	-	EOR (sr,S),Y	* BVC rel16	TAM #n	BBC 2, A, rel	2	2
54		-	-	-	-	-	-	* MVN sbnk,dbnk	* ASR bp,X	CSL	-	3	
55	2	EOR zp,X	EOR zp,X	EOR zp,X	EOR zp,X	EOR zp,X	EOR zp,X	EOR dp,X	EOR bp,X	EOR zp,X	EOR zp, X	2	2
56	2	LSR zp,X	LSR zp,X	LSR zp,X	LSR zp,X	LSR zp,X	LSR zp,X	LSR dp,X	LSR bp,X	LSR zp,X	LSR zp, X	2	2
57		-	RMB 5,zp	-	-	RMB 5,zp	RMB 5,zp	* EOR (dp),Y	RMB 5,bp	RMB 5,zp	BBC 2, zp, rel	2	3
58	1	CLI	CLI	CLI	CLI	CLI	CLI	CLI	CLI	CLI	CLI	1	1
59	3	EOR abs,Y	EOR abs,Y	EOR abs,Y	EOR abs,Y	EOR abs,Y	EOR abs,Y	EOR abs,Y	EOR abs,Y	EOR abs,Y	EOR abs, Y	3	3
5A	1	-	-	PHY	PHY	PHY	PHY	PHY	PHY	PHY	-	1	
5B		-	-	-	-	-	-	TCD	* TAB	-	CLB 2, A	1	1
5C		-	-	-	-	-	-	JMP long	* Aug	-	-	4	
5D	3	EOR abs,X	EOR abs,X	EOR abs,X	EOR abs,X	EOR abs,X	EOR abs,X	EOR abs,X	EOR abs,X	EOR abs,X	EOR abs, X	3	3
5E	3	LSR abs,X	LSR abs,X	LSR abs,X	LSR abs,X	LSR abs,X	LSR abs,X	LSR abs,X	LSR abs,X	LSR abs,X	LSR abs, X	3	3
5F	3	-	BBR 5,zp,rel		-	BBR 5,zp,rel	BBR 5,zp,rel	* EOR long,X	BBR 5,bp,rel	BBR 5,zp,rel	CLB 2, zp	4	2
60	1	RTS	RTS	RTS	RTS	RTS	RTS	RTS	RTS	RTS	RTS	1	1
61	2	ADC (zp,X)	ADC (zp,X)	ADC (zp,X)	ADC (zp,X)	ADC (zp,X)	ADC (zp,X)	ADC (dp,X)	ADC (bp,X)	ADC (zp,X)	ADC (zp, X)	2	2
62		-	-	-	-	-	-	PER rel16	* RTN #n	CLA	MUL zp, X	3	2
63		-	-	-	-	-	-	ADC sr,S	* BSR rel16	-	BBS 3, A, rel	2	2
64	2	-	-	STZ zp	STZ zp	STZ zp	STZ zp	STZ dp	STZ bp	STZ zp	TST zp	2	2
65	2	ADC zp	ADC zp	ADC zp	ADC zp	ADC zp	ADC zp	ADC dp	ADC bp	ADC zp	ADC zp	2	2
66	2	ROR zp	ROR zp	ROR zp	ROR zp	ROR zp	ROR zp	ROR dp	ROR bp	ROR zp	ROR zp	2	2
67		-	RMB 6,zp	-	-	RMB 6,zp	RMB 6,zp	* ADC (dp)	RMB 6,bp	RMB 6,zp	BBS 3, zp, rel	2	3
68	1	PLA	PLA	PLA	PLA	PLA	PLA	PLA	PLA	PLA	PLA	1	1
69	2	ADC #n	ADC #n	ADC #n	ADC #n	ADC #n	ADC #n	* ADC #const	ADC #n	ADC #n	ADC #n	2(3)	2
6A	1	ROR A	ROR A	ROR A	ROR A	ROR A	ROR A	ROR A	ROR A	ROR A	ROR A	1	1
6B		-	-	-	-	-	-	RTL	* TZA	-	SEB 3, A	1	1
6C	3	JMP (abs)	JMP (abs)	JMP (abs)	JMP (abs)	JMP (abs)	JMP (abs)	JMP (abs)	JMP (abs)	JMP (abs)	JMP (abs)	3	3
6D	3	ADC abs	ADC abs	ADC abs	ABC abs	ADC abs	ADC abs	ADC abs	ADC abs	ADC abs	ADC abs	3	3
6E	3	ROR abs	ROR abs	ROR abs	ROR abs	ROR abs	ROR abs	ROR abs	ROR abs	ROR abs	ROR abs	3	3
6F	3	-	BBR 6,zp,rel	-	-	BBR 6,zp,rel	BBR 6,zp,rel	* ADC long	BBR 6,bp,rel	BBR 6,zp,rel	SEB 3, zp	4	2
70	2	BVS rel	BVS rel	BVS rel	BVS rel	BVS rel	BVS rel	BVS rel	BVS rel	BVS rel	BVS rel	2	2
71	2	ADC (zp),Y	ADC (zp),Y	ADC (zp),Y	ADC (zp),Y	ADC (zp),Y	ADC (zp),Y	ADC (dp),Y	ADC (bp),Y	ADC (zp),Y	ADC (zp), Y	2	2
72	2	-	-	ADC (zp)	ADC (zp)	ADC (zp)	ADC (zp)	ADC (dp)	ADC (bp)	ADC (zp)	-	2	
73		-	-	-	-	-	-	ADC (sr,S),Y	* BVS rel16	TII s,d,l	BBC 3, A, rel	2	2
74	2	-	-	STZ zp,X	STZ zp,X	STZ zp,X	STZ zp,X	STZ dp,X	STZ bp,X	STZ zp,X	-	2	
75	2	ADC zp,X	ADC zp,X	ADC zp,X	ADC zp,X	ADC zp,X	ADC zp,X	ADC dp,X	ADC bp,X	ADC zp,X	ADC zp, X	2	2
76	2	ROR zp,X	ROR zp,X	ROR zp,X	ROR zp,X	ROR zp,X	ROR zp,X	ROR dp,X	ROR bp,X	ROR zp,X	ROR zp, X	2	2
77		-	RMB 7,zp	-	-	RMB 7,zp	RMB 7,zp	* ADC (dp),Y	RMB 7,bp	RMB 7,zp	BBC 3, zp, rel	2	3
78	1	SEI	SEI	SEI	SEI	SEI	SEI	SEI	SEI	SEI	SEI	1	1
79	3	ADC abs,Y	ADC abs,Y	ADC abs,Y	ADC abs,Y	ADC abs,Y	ADC abs,Y	ADC abs,Y	ADC abs,Y	ADC abs,Y	ADC abs, Y	3	3
7A	1	-	-	PLY	PLY	PLY	PLY	PLY	PLY	PLY	-	1	
7B		-	-	-	-	-	-	TDC	* TBA	-	CLB 3, A	1	1
7C	3	-	JMP (abs,X)	JMP (abs,X)	JMP (abs,X)	JMP (abs,X)	JMP (abs,X)	JMP (abs,X)	JMP (abs,X)	JMP (abs,X)	-	3	
7D	3	ADC abs,X	ADC abs,X	ADC abs,X	ADC abs,X	ADC abs,X	ADC abs,X	ADC abs,X	ADC abs,X	ADC abs,X	ADC abs, X	3	3
7E	3	ROR abs,X	ROR abs,X	ROR abs,X	ROR abs,X	ROR abs,X	ROR abs,X	ROR abs,X	ROR abs,X	ROR abs,X	ROR abs, X	3	3
7F	3	-	BBR 7,zp,rel	BBR 7,zp,rel	-	BBR 7,zp,rel	BBR 7,zp,rel	* ADC long,X	BBR 7,bp,rel	BBR 7,zp,rel	CLB 3, zp	4	2
80	2	-	-	BRA rel	BRA rel	BRA rel	BRA rel	BRA rel	BRA rel (BRU)	BRA rel	BRA rel	2	2
81	2	STA (zp,X)	STA (zp,X)	STA (zp,X)	STA (zp,X)	STA (zp,X)	STA (zp,X)	STA (dp,X)	STA (bp,X)	STA (zp,X)	STA (zp, X)	2	2
82		-	-	-	-	-	-	BRL label	* STA (d,SP)	CLX	RRF zp	3	2
83		-	-	-	-	-	-	STA sr,S	* BRA rel16 (BRU)	TST #n, zp	BBS 4, A, rel	2	2
84	2	STY zp	STY zp	STY zp	STY zp	STY zp	STY zp	STY dp	STY bp	STY zp	STY zp	2	2
85	2	STA zp	STA zp	STA zp	STA zp	STA zp	STA zp	STA dp	STA bp	STA zp	STA zp	2	2
86	2	STX zp	STX zp	STX zp	STX zp	STX zp	STX zp	STX dp	STX bp	STX zp	STX zp	2	2
87		-	SMB 0,zp	-	-	SMB 0,zp	SMB 0,zp	* STA (dp)	SMB 0,bp	SMB 0,zp	BBS 4, zp, rel	2	3
88	1	DEY	DEY	DEY	DEY	DEY	DEY	DEY	DEY	DEY	DEY	1	1
89	2	-	-	BIT #n	BIT #n	BIT #n	BIT #n	* BIT #const	BIT #n	BIT #n	-	2(3)	
8A	1	TXA	TXA	TXA	TXA	TXA	TXA	TXA	TXA	TXA	TXA	1	1
8B		-	-	-	-	-	-	PHB	* STY abs,X	-	SEB 4, A	1	1
8C	3	STY abs	STY abs	STY abs	STY abs	STY abs	STY abs	STY abs	STY abs	STY abs	STY abs	3	3
8D	3	STA abs	STA abs	STA abs	STA abs	STA abs	STA abs	STA abs	STA abs	STA abs	STA abs	3	3
8E	3	STX abs	STX abs	STX abs	STX abs	STX abs	STX abs	STX abs	STX abs	STX abs	STX abs	3	3
8F		-	BBS 0,zp,rel	-	-	BBS 0,zp,rel	BBS 0,zp,rel	* STA long	BBS 0,bp,rel	BBS 0,zp,rel	SEB 4, zp	4	2
90	2	BCC rel	BCC rel	BCC rel	BCC rel	BCC rel	BCC rel	BCC rel	BCC rel	BCC rel	BCC rel	2	2
91	2	STA (zp),Y	STA (zp),Y	STA (zp),Y	STA (zp),Y	STA (zp),Y	STA (zp),Y	STA (dp),Y	STA (bp),Y	STA (zp),Y	STA (zp), Y	2	2
92	2	-	-	STA (zp)	STA (zp)	STA (zp)	STA (zp)	STA (dp)	STA (bp)	STA (zp)	-	2	
93		-	-	-	-	-	-	STA (sr,S),Y	* BCC rel16	TST #n, abs	BBC 4, A, rel	2	2
94	2	STY zp	STY zp	STY zp	STY zp	STY zp	STY zp	STY dp,X	STY bp	STY zp	STY zp, X	2	2
95	2	STA zp,X	STA zp,X	STA zp,X	STA zp,X	STA zp,X	STA zp,X	STA dpX	STA bp,X	STA zp,X	STA zp, X	2	2
96	2	STX zp,Y	STX zp,Y	STX zp,Y	STX zp,Y	STX zp,Y	STX zp,Y	STX dp,Y	STX bp,Y	STX zp,Y	STX zp, Y	2	2
97		-	SMB 1,zp	-	-	SMB 1,zp	SMB 1,zp	* STA (dp),Y	SMB 1,bp	SMB 1,zp	BBC 4, zp, rel	2	3
98	1	TYA	TYA	TYA	TYA	TYA	TYA	TYA	TYA	TYA	TYA	1	1
99	3	STA abs,Y	STA abs,Y	STA abs,Y	STA abs,Y	STA abs,Y	STA abs,Y	STA abs,Y	STA abs,Y	STA abs,Y	STA abs, Y	3	3
9A	1	TXS	TXS	TXS	TXS	TXS	TXS	TXS	TXS	TXS	TXS	1	1
9B		-	-	-	-	-	-	TXY	* STX abs,Y	-	CLB 4, A	1	1
9C	3	-	-	STZ abs	STZ abs	STZ abs	STZ abs	STZ abs	STZ abs	STZ abs	-	3	
9D	3	STA abs,X	STA abs,X	STA abs,X	STA abs,X	STA abs,X	STA abs,X	STA abs,X	STA abs,X	STA abs,X	STA abs, X	3	3
9E	3	-	-	STZ abs,X	STZ abs,X	STZ abs,X	STZ abs,X	STZ abs,X	STZ abs,X	STZ abs,X	-	3	
9F	3	-	BBS 1,zp,rel	-	-	BBS 1,zp,rel	BBS 1,zp,rel	* STA long,X	BBS 1,bp,rel	BBS 1,zp,rel	CLB 4, zp	4	2
A0	2	LDY #n	LDY #n	LDY #n	LDY #n	LDY #n	LDY #n	* LDY #const	LDY #n	LDY #n	LDY #n	2(3)	2
A1	2	LDA (zp,X)	LDA (zp,X)	LDA (zp,X)	LDA (zp,X)	LDA (zp,X)	LDA (zp,X)	LDA (dp,X)	LDA (bp,X)	LDA (zp,X)	LDA (zp, X)	2	2
A2	2	LDX #n	LDX #n	LDX #n	LDX #n	LDX #n	LDX #n	* LDX #const	LDX #n	LDX #n	LDX #n	2(3)	2
A3		-	-	-	-	-	-	LDA sr,S	* LDZ #n	TST #n, zp,X	BBS 5, A, rel	2	2
A4	2	LDY zp	LDY zp	LDY zp	LDY zp	LDY zp	LDY zp	LDY dp	LDY bp	LDY zp	LDY zp	2	2
A5	2	LDA zp	LDA zp	LDA zp	LDA zp	LDA zp	LDA zp	LDA dp	LDA bp	LDA zp	LDA zp	2	2
A6	2	LDX zp	LDX zp	LDX zp	LDX zp	LDX zp	LDX zp	LDX dp	LDX bp	LDX zp	LDX zp	2	2
A7		-	SMB 2,zp	-	-	SMB 2,zp	SMB 2,zp	* LDA (dp)	SMB 2,bp	SMB 2,zp	BBS 5, zp, rel	2	3
A8	1	TAY	TAY	TAY	TAY	TAY	TAY	TAY	TAY	TAY	TAY	1	1
A9	2	LDA #n	LDA #n	LDA #n	LDA #n	LDA #n	LDA #n	LDA #const	LDA #n	LDA #n	LDA #n	2(3)	2
AA	1	TAX	TAX	TAX	TAX	TAX	TAX	TAX	TAX	TAX	TAX	1	1
AB		-	-	-	-	-	-	PLB	* LDZ abs	-	SEB 5, A	1	1
AC	3	LDY abs	LDY abs	LDY abs	LDY abs	LDY abs	LDY abs	LDY abs	LDY abs	LDY abs	LDY abs	3	3
AD	3	LDA abs	LDA abs	LDA abs	LDA abs	LDA abs	LDA abs	LDA abs	LDA abs	LDA abs	LDA abs	3	3
AE	3	LDX abs	LDX abs	LDX abs	LDX abs	LDX abs	LDX abs	LDX abs	LDX abs	LDX abs	LDX abs	3	3
AF	3	-	BBS 2,zp,rel	-	-	BBS 2,zp,rel	BBS 2,zp,rel	* LDA long	BBS 2,bp,rel	BBS 2,zp,rel	SEB 5, zp	4	2
B0	2	BCS rel	BCS rel	BCS rel	BCS rel	BCS rel	BCS rel	BCS rel	BCS rel	BCS rel	BCS rel	2	2
B1	2	LDA (zp),Y	LDA (zp),Y	LDA (zp),Y	LDA (zp),Y	LDA (zp),Y	LDA (zp),Y	LDA (dp),Y	LDA (bp),Y	LDA (zp),Y	LDA (zp), Y	2	2
B2		-	-	LDA (zp)	LDA (zp)	LDA (zp)	LDA (zp)	LDA (dp)	LDA (bp)	LDA (zp)	JMP (zp)	2	2
B3		-	-	-	-	-	-	LDA (sr,S),Y	* BCS rel16	TST #n, abs,X	BBC 5, A, rel	2	2
B4	2	LDY zp	LDY zp	LDY zp	LDY zp	LDY zp	LDY zp	LDY dp,X	LDY bp	LDY zp	LDY zp, X	2	2
B5	2	LDA zp,X	LDA zp,X	LDA zp,X	LDA zp,X	LDA zp,X	LDA zp,X	LDA dp,X	LDA bp,X	LDA zp,X	LDA zp, X	2	2
B6	2	LDX zp,Y	LDX zp,Y	LDX zp,Y	LDX zp,Y	LDX zp,Y	LDX zp,Y	LDX dp,Y	LDX bp,Y	LDX zp,Y	LDX zp, Y	2	2
B7		-	SMB 3,zp	-	-	SMB 3,zp	SMB 3,zp	* LDA (dp),Y	SMB 3,bp	SMB 3,zp	BBC 5, zp, rel	2	3
B8	1	CLV	CLV	CLV	CLV	CLV	CLV	CLV	CLV	CLV	CLV	1	1
B9	3	LDA abs,Y	LDA abs,Y	LDA abs,Y	LDA abs,Y	LDA abs,Y	LDA abs,Y	LDA abs,Y	LDA abs,Y	LDA abs,Y	LDA abs, Y	3	3
BA	1	TSX	TSX	TSX	TSX	TSX	TSX	TSX	TSX	TSX	TSX	1	1
BB		-	-	-	-	-	-	TYX	* LDZ abs,X	-	CLB 5, A	1	1
BC	3	LDY abs,X	LDY abs,X	LDY abs,X	LDY abs,X	LDY abs,X	LDY abs,X	LDY abs,X	LDY abs,X	LDY abs,X	LDY abs, X	3	3
BD	3	LDA abs,X	LDA abs,X	LDA abs,X	LDA abs,X	LDA abs,X	LDA abs,X	LDA abs,X	LDA abs,X	LDA abs,X	LDA abs, X	3	3
BE	3	LDX abs,Y	LDX abs,Y	LDX abs,Y	LDX abs,Y	LDX abs,Y	LDX abs,Y	LDX abs,Y	LDX abs,Y	LDX abs,Y	LDX abs, Y	3	3
BF		-	BBS 3,zp,rel	-	-	BBS 3,zp,rel	BBS 3,zp,rel	* LDA long,X	BBS 3,bp,rel	BBS 3,zp,rel	CLB 5, zp	4	2
C0	2	CPY #n	CPY #n	CPY #n	CPY #n	CPY #n	CPY #n	* CPY #const	CPY #n	CPY #n	CPY #n	2(3)	2
C1	2	CMP (zp,X)	CMP (zp,X)	CMP (zp,X)	CMP (zp,X)	CMP (zp,X)	CMP (zp,X)	CMP (dp,X)	CMP (bp,X)	CMP (zp,X)	CMP (zp, X)	2	2
C2		-	-	-	-	-	-	REP #const	* CPZ #n	CLY	WIT	2	1
C3		-	-	-	-	-	-	CMP sr,S	* DEW bp	TDD s,d,l	BBS 6, A, rel	2	2
C4	2	CPY zp	CPY zp	CPY zp	CPY zp	CPY zp	CPY zp	CPY dp	CPY bp	CPY zp	CPY zp	2	2
C5	2	CMP zp	CMP zp	CMP zp	CMP zp	CMP zp	CMP zp	CMP dp	CMP bp	CMP zp	CMP zp	2	2
C6	2	DEC zp	DEC zp	DEC zp	DEC zp	DEC zp	DEC zp	DEC dp	DEC bp	DEC zp	DEC zp	2	2
C7		-	SMB 4,zp	-	-	SMB 4,zp	SMB 4,zp	* CMP (dp)	SMB 4,bp	SMB 4,zp	BBS 6, zp, rel	2	3
C8	1	INY	INY	INY	INY	INY	INY	INY	INY	INY	INY	1	1
C9	2	CMP #n	CMP #n	CMP #n	CMP #n	CMP #n	CMP #n	* CMP #const	CMP #n	CMP #n	CMP #n	2(3)	2
CA	1	DEX	DEX	DEX	DEX	DEX	DEX	DEX	DEX	DEX	DEX	1	1
CB		-		-	-		WAI	WAI	* ASW abs	-	SEB 6, A	1	1
CC	3	CPY abs	CPY abs	CPY abs	CPY abs	CPY abs	CPY abs	CPY abs	CPY abs	CPY abs	CPY abs	3	3
CD	3	CMP abs	CMP abs	CMP abs	CMP abs	CMP abs	CMP abs	CMP abs	CMP abs	CMP abs	CMP $ hhll	3	3
CE	3	DEC abs	DEC abs	DEC abs	DEC abs	DEC abs	DEC abs	DEC abs	DEC abs	DEC abs	DEC abs	3	3
CF		-	BBS 4,zp,rel		-	BBS 4,zp,rel	BBS 4,zp,rel	* CMP long	BBS 4,bp,rel	BBS 4,zp,rel	SEB 6, zp	4	2
D0	2	BNE rel	BNE rel	BNE rel	BNE rel	BNE rel	BNE rel	BNE rel	BNE rel	BNE rel	BNE rel	2	2
D1	2	CMP (zp),Y	CMP (zp),Y	CMP (zp),Y	CMP (zp),Y	CMP (zp),Y	CMP (zp),Y	CMP (dp),Y	CMP (bp),Y	CMP (zp),Y	CMP (zp), Y	2	2
D2	2	-	-	CMP (zp)	CMP (zp)	CMP (zp)	CMP (zp)	CMP (dp)	CMP (bp)	CMP (zp)	-	2	
D3		-	-	-	-	-	-	CMP (sr,S),Y	* BNE rel16	TIN s,d,l	BBC 6, A, rel	2	2
D4		-	-	-	-	-	-	PEI (dp)	* CPZ bp	CSH	-	2	
D5	2	CMP zp,X	CMP zp,X	CMP zp,X	CMP zp,X	CMP zp,X	CMP zp,X	CMP dp,X	CMP bp,X	CMP zp,X	CMP zp, X	2	2
D6	2	DEC zp,X	DEC zp,X	DEC zp,X	DEC zp,X	DEC zp,X	DEC zp,X	DEC dp,X	DEC bp,X	DEC zp,X	DEC zp, X	2	2
D7		-	SMB 5,zp	-	-	SMB 5,zp	SMB 5,zp	* CMP (dp),Y	SMB 5,bp	SMB 5,zp	BBC 6, zp, rel	2	3
D8	1	CLD	CLD	CLD	CLD	CLD	CLD	CLD	CLD	CLD	CLD	1	1
D9	3	CMP abs,Y	CMP abs,Y	CMP abs,Y	CMP abs,Y	CMP abs,Y	CMP abs,Y	CMP abs,Y	CMP abs,Y	CMP abs,Y	CMP abs, Y	3	3
DA	1	-	-	PHX	PHX	PHX	PHX	PHX	PHX	PHX	-	1	
DB		-		-	-		STP	STP	* PHZ	-	CLB 6, A	1	1
DC		-	-	-	-	-	-	JMP [abs]	* CPZ abs	-	-	3	
DD	3	CMP abs,X	CMP abs,X	CMP abs,X	CMP abs,X	CMP abs,X	CMP abs,X	CMP abs,X	CMP abs,X	CMP abs,X	CMP abs, X	3	3
DE	3	DEC abs,X	DEC abs,X	DEC abs,X	DEC abs,X	DEC abs,X	DEC abs,X	DEC abs,X	DEC abs,X	DEC abs,X	DEC abs, X	3	3
DF		-	BBS 5,zp,rel	-	-	BBS 5,zp,rel	BBS 5,zp,rel	* CMP long,X	BBS 5,bp,rel	BBS 5,zp,rel	CLB 6, zp	4	2
E0	2	CPX #n	CPX #n	CPX #n	CPX #n	CPX #n	CPX #n	* CPX #const	CPX #n	CPX #n	CPX #n	2(3)	2
E1	2	SBC (zp,X)	SBC (zp,X)	SBC (zp,X)	SBC (zp,X)	SBC (zp,X)	SBC (zp,X)	SBC (dp,X)	SBC (bp,X)	SBC (zp,X)	SBC (zp, X)	2	2
E2		-	-	-	-	-	-	SEP	* LDA (d,SP),Y	-	DIV zp, X	2	2
E3		-	-	-	-	-	-	SBC sr,S	* INW bp	TIA s,d,l	BBS 7, A, rel	2	2
E4	2	CPX zp	CPX zp	CPX zp	CPX zp	CPX zp	CPX zp	CPX dp	CPX bp	CPX zp	CPX zp	2	2
E5	2	SBC zp	SBC zp	SBC zp	SBC zp	SBC zp	SBC zp	SBC dp	SBC bp	SBC zp	SBC zp	2	2
E6	2	INC zp	INC zp	INC zp	INC zp	INC zp	INC zp	INC dp	INC bp	INC zp	INC zp	2	2
E7		-	SMB 6,zp		-	SMB 6,zp	SMB 6,zp	* SBC (dp)	SMB 6,bp	SMB 6,zp	BBS 7, zp, rel	2	3
E8	1	INX	INX	INX	INX	INX	INX	INX	INX	INX	INX	1	1
E9	2	SBC #n	SBC #n	SBC #n	SBC #n	SBC #n	SBC #n	* SBC #const	SBC #n	SBC #n	SBC #n	2(3)	2
EA	1	NOP	NOP	NOP	NOP	NOP	NOP	NOP	NOP	NOP	NOP	1	1
EB	2	SBC #n	SBC #n	SBC #n	SBC #n	SBC #n	SBC #n	* XBA	SBC #n	SBC #n	SEB 7, A	1	1
EC	3	CPX abs	CPX abs	CPX abs	CPX abs	CPX abs	CPX abs	CPX abs	CPX abs	CPX abs	CPX abs	3	3
ED	3	SBC abs	SBC abs	SBC abs	SBC abs	SBC abs	SBC abs	SBC abs	SBC abs	SBC abs	SBC abs	3	3
EE	3	INC abs	INC abs	INC abs	INC abs	INC abs	INC abs	INC abs	INC abs	INC abs	INC abs	3	3
EF		-	BBS 6,zp,rel	-	-	BBS 6,zp,rel	BBS 6,zp,rel	* SBC long	BBS 6,bp,rel	BBS 6,zp,rel	SEB 7, zp	4	2
F0	2	BEQ rel	BEQ rel	BEQ rel	BEQ rel	BEQ rel	BEQ rel	BEQ rel	BEQ rel	BEQ rel	BEQ rel	2	2
F1	2	SBC (zp),Y	SBC (zp),Y	SBC (zp),Y	SBC (zp),Y	SBC (zp),Y	SBC (zp),Y	SBC (dp),Y	SBC (bp),Y	SBC (zp),Y	SBC (zp), Y	2	2
F2	2	-	-	SBC (zp)	SBC (zp)	SBC (zp)	SBC (zp)	SBC (dp)	SBC (bp)	SBC (zp)	-	2	
F3		-	-	-	-	-	-	SBC (sr,S),Y	* BEQ rel16	TAI s,d,l	BBC 7, A, rel	2	2
F4		-	-	-	-	-	-	PEA abs	* PHW #n16	SET	-	3	
F5	2	SBC zp,X	SBC zp,X	SBC zp,X	SBC zp,X	SBC zp,X	SBC zp,X	SBC dp,X	SBC bp,X	SBC zp,X	SBC zp, X	2	2
F6	2	INC zp,X	INC zp,X	INC zp,X	INC zp,X	INC zp,X	INC zp,X	INC dp,X	INC bp,X	INC zp,X	INC zp, X	2	2
F7		-	SMB 7,zp	-	-	SMB 7,zp	SMB 7,zp	* SBC (dp),Y	SMB 7,bp	SMB 7,zp	BBC 7, zp, rel	2	3
F8	1	SED	SED	SED	SED	SED	SED	SED	SED	SED	SED	1	1
F9	3	SBC abs,Y	SBC abs,Y	SBC abs,Y	SBC abs,Y	SBC abs,Y	SBC abs,Y	SBC abs,Y	SBC abs,Y	SBC abs,Y	SBC abs, Y	3	3
FA	1	-	-	PLX	PLX	PLX	PLX	PLX	PLX	PLX	-	1	
FB		-	-	-	-	-	-	XCE	* PLZ	-	CLB 7, A	1	1
FC		-	-	-	-	-	-	JSR (abs,X))	* PHW abs	-	-	3	
FD	3	SBC abs,X	SBC abs,X	SBC abs,X	SBC abs,X	SBC abs,X	SBC abs,X	SBC abs,X	SBC abs,X	SBC abs,X	SBC abs, X	3	3
FE	3	INC abs,X	INC abs,X	INC abs,X	INC abs,X	INC abs,X	INC abs,X	INC abs,X	INC abs,X	INC abs,X	INC abs, X	3	3
FF		-	BBS 7,zp,rel	-	-	BBS 7,zp,rel	BBS 7,zp,rel	* SBC long,X	BBS 7,bp,rel	BBS 7,zp,rel	CLB 7, zp	4	2

OP	Bedeutung	65C02	65C816
ADC	Add With Carry [Flags affected: n,v,z,c]		
AND	AND Accumulator With Memory [Flags affected: n,z]		
ASL	Accumulator or Memory Shift Left [Flags affected: n,z,c]		
BCC	Branch if Carry Clear [Flags affected: none][Alias: BLT]		
BCS	Branch if Carry Set [Flags affected: none][Alias: BGE]		
BEQ	Branch if Equal [Flags affected: none]		
BGE	Branch Greater or Equal [Flags affected: none][Alias for BCS]		
BIT	Test Bits [Flags affected: z (immediate mode) n,v,z (non-immediate modes)]		
BLT	Branch Less Than [Flags affected: none][Alias for BCC]		
BMI	Branch if Minus [Flags affected: none]		
BNE	Branch if Not Equal [Flags affected: none]		
BPL	Branch if Plus [Flags affected: none]		
BRA	Branch Always [Flags affected: none]		
BRK	Break [Flags affected: b,i -(6502 b,d,i (65C02/65816 Emulation) d,i (65816 Native)]		
BRL	Branch Long Always [Flags affected: none]		65C816
BVC	Branch if Overflow Clear [Flags affected: none]		
BVS	Branch if Overflow Set [Flags affected: none]		
CLC	Clear Carry [Flags affected: c]		
CLD	Clear Decimal Mode Flag [Flags affected: d]		
CLI	Clear Interrupt Disable Flag [Flags affected: i]		
CLV	Clear Overflow Flag [Flags affected: v]		
CMP	Compare Accumulator With Memory [Flags affected: n,z,c]		
COP	Co-Processor Enable [Flags affected: d,i]		65C816
CPX	Compare Index Register X with Memory [Flags affected: n,z,c]		
CPY	Compare Index Register Y with Memory [Flags affected: n,z,c]		
DEC	Decrement [Flags affected: n,z]		
DEX	Decrement Index Register X [Flags affected: n,z]		
DEY	Decrement Index Register Y [Flags affected: n,z]		
EOR	Exclusive-OR Accumulator with Memory [Flags affected: n,z]		
INC	Increment [Flags affected: n,z]		
INX	Increment Index Register X [Flags affected: n,z]		
INY	Increment Index Register Y [Flags affected: n,z]		
JML	Jump Long [Flags affected: none]		65C816
JMP	Jump [Flags affected: none][Alias: JML for all Long addressing modes]		
JSR	Jump to Subroutine [Flags affected: none][Alias: JSL for Absolute Long]		
JSR	Jump to Subroutine Absolute Long [Flags affected: none]		65C816
LDA	Load Accumulator from Memory [Flags affected: n,z]		
LDX	Load Index Register X from Memory [Flags affected: n,z]		
LDY	Load Index Register Y from Memory [Flags affected: n,z]		
LSR	Logical Shift Memory or Accumulator Right [Flags affected: n,z,c]		
MVN	Block Move Negative [Flags affected: none][Registers: X,Y,C]		65C816
MVP	Block Move Positive [Flags affected: none][Registers: X,Y,C]		65C816
NOP	No Operation [Flags affected: none]		
ORA	OR Accumulator with Memory [Flags affected: n,z]		
PEA	Push Effective Absolute Address [Flags affected: none]		65C816
PEI	Push Effective Indirect Address [Flags affected: none]		65C816
PER	Push Effective PC Relative Indirect Address [Flags affected: none]		65C816
PHA	Push Accumulator [Flags affected: none]		
PHB	Push Data Bank Register [Flags affected: none]		
PHD	Push Direct Page Register [Flags affected: none]		
PHK	Push Program Bank Register [Flags affected: none]		
PHP	Push Processor Status Register [Flags affected: none]		
PHX	Push Index Register X [Flags affected: none]		
PHY	Push Index Register Y [Flags affected: none]		
PLA	Pull Accumulator [Flags affected: n,z]		
PLB	Pull Data Bank Register [Flags affected: n,z]		
PLD	Pull Direct Page Register [Flags affected: n,z]		
PLP	Pull Processor Status Register [Flags affected: n,z]		
PLX	Pull Index Register X [Flags affected: n,z]		
PLY	Pull Index Register Y [Flags affected: n,z]		
REP	Reset Processor Status Bits [Flags affected: all except b per operand]		65C816
ROL	Rotate Memory or Accumulator Left [Flags affected: n,z,c]		
ROR	Rotate Memory or Accumulator Right [Flags affected: n,z,c]		
RTI	Return from Interrupt [Flags affected: all except b]		
RTL	Return from Subroutine Long [Flags affected: none]		65C816
RTS	Return from Subroutine [Flags affected: none]		
SBC	Subtract with Borrow from Accumulator [Flags affected: n,v,z,c]		
SEC	Set Carry Flag [Flags affected: c]		
SED	Set Decimal Flag [Flags affected: d]		
SEI	Set Interrupt Disable Flag [Flags affected: i]		
SEP	Set Processor Status Bits [Flags affected: all except b per operand]		65C816
STA	Store Accumulator to Memory [Flags affected: none]		
STP	Stop Processor [Flags affected: none]	65C02	65C816
STX	Store Index Register X to Memory [Flags affected: none]		
STY	Store Index Register Y to Memory [Flags affected: none]		
STZ	Store Zero to Memory [Flags affected: none]		
TAX	Transfer Accumulator to Index Register X [Flags affected: n,z]		
TAY	Transfer Accumulator to Index Register Y [Flags affected: n,z]		
TCD	Transfer 16-bit Accumulator to Direct Page Register [Flags affected: n,z]		65C816
TCS	Transfer 16-bit Accumulator to Stack Pointer [Flags affected: none]		65C816
TDC	Transfer Direct Page Register to 16-bit Accumulator [Flags affected: n,z]		65C816
TRB	Test and Reset Memory Bits Against Accumulator [Flags affected: z]		
TSB	Test and Set Memory Bits Against Accumulator [Flags affected: z]		
TSC	Transfer Stack Pointer to 16-bit Accumulator [Flags affected: n,z]		65C816
TSX	Transfer Stack Pointer to Index Register X [Flags affected: n,z]		
TXA	Transfer Index Register X to Accumulator [Flags affected: n,z]		
TXS	Transfer Index Register X to Stack Pointer [Flags affected: none]		
TXY	Transfer Index Register X to Index Register Y [Flags affected: n,z]		
TYA	Transfer Index Register Y to Accumulator [Flags affected: n,z]		
TYX	Transfer Index Register Y to Index Register X [Flags affected: n,z]		
WAI	Wait for Interrupt [Flags affected: none]	65C02	65C816
WDM	Reserved for Future Expansion [Flags affected: none (subject to change)]		65C816
XBA	Exchange B and A 8-bit Accumulators [Flags affected: n,z]		66C816
XCE	Exchange Carry and Emulation Flags [Flags affected: m,b/x,c,e]		67C816

Offene Versioen			Basis
Winbond/Nuvoton	W55V96	Game/TV Conroller	65C816
Winbond/Nuvoton	W55U02	USB Controller	6502?
Sunplus Technolog	8502	Controller im TschenrechnerHP35S	

OP	Param	OPC	Addressing Mode	02	C02	816	Bytes	Zyklen
BRK		00	Stack/Interrupt	x	x	x	1	7(+1)
ORA	(dp,X)	01	DP Indexed Indirect,X	x	x	x	2	6
COP	const	02	Stack/Interrupt			x	1	7(+1)
ORA	sr,S	03	Stack Relative			x	2	4(+1)
TSB	dp	04	Direct Page		x	x	2	5(+1+2)
ORA	dp	05	Direct Page	x	x	x	2	3(+1+1)
ASL	dp	06	Direct Page	x	x	x	2	5(+1+2)
ORA	[dp]	07	DP Indirect Long			x	2	6(+1+1)
PHP		08	Stack (Push)	x	x	x	1	3
ORA	#const	09	Immediate	x	x	x	2(3)	2(+1)
ASL	A	0A	Accumulator	x	x	x	1	2
PHD		0B	Stack (Push)			x	1	4
TSB	addr	0C	Absolute		x	x	3	6(+2)
ORA	addr	0D	Absolute	x	x	x	3	4(+1)
ASL	addr	0E	Absolute	x	x	x	3	6(+2)
ORA	long	0F	Absolute Long			x	4	5(+1)
BPL	nearlabel	10	Program Counter Relative	x	x	x	2	2(+1+1)
ORA	(dp),Y	11	DP Indirect Indexed, Y	x	x	x	2	5(+1+1/+1)
ORA	(dp)	12	DP Indirect		x	x	2	5(+1+1)
ORA	(sr,S),Y	13	SR Indirect Indexed,Y			x	2	7(+1)
TRB	dp	14	Direct Page		x	x	2	5(+1+2)
ORA	dp,X	15	DP Indexed,X	x	x	x	2	4(+1+1)
ASL	dp,X	16	DP Indexed,X	x	x	x	2	6(+1+2)
ORA	[dp],Y	17	DP Indirect Long Indexed, Y			x	2	6(+1+1)
CLC		18	Implied	x	x	x	1	2
ORA	addr,Y	19	Absolute Indexed,Y	x	x	x	3	4(+1+1)
INC	A	1A	Accumulator			x	1	2
TCS		1B	Implied			x	1	2
TRB	addr	1C	Absolute		x	x	3	6(+1)
ORA	addr,X	1D	Absolute Indexed,X	x	x	x	3	4(+1+1)
ASL	addr,X	1E	Absolute Indexed,X	x	x	x	3	7(+2+1)
ORA	long,X	1F	Absolute Long Indexed,X			x	4	5(+1)
JSR	addr	20	Absolute	x	x	x	3	6
AND	(dp,X)	21	DP Indexed Indirect,X	x	x	x	2	6(+1+1)
JSR	long	22	Absolute Long			x	4	8
AND	sr,S	23	Stack Relative			x	2	4(+1)
BIT	dp	24	Direct Page	x	x	x	2	3(+1+1)
AND	dp	25	Direct Page	x	x	x	2	3(+1+1)
ROL	dp	26	Direct Page	x	x	x	2	5(+1+2)
AND	[dp]	27	DP Indirect Long			x	2	6(+1+1)
PLP		28	Stack (Pull)	x	x	x	1	4
AND	#const	29	Immediate	x	x	x	2(3)	2(+1)
ROL	A	2A	Accumulator	x	x	x	1	2
PLD		2B	Stack (Pull)			x	1	5
BIT	addr	2C	Absolute	x	x	x	3	4(+1)
AND	addr	2D	Absolute	x	x	x	3	4(+1)
ROL	addr	2E	Absolute	x	x	x	3	6(+2)
AND	long	2F	Absolute Long			x	4	5(+1)
BMI	nearlabel	30	Program Counter Relative	x	x	x	2	2(+1+1)
AND	(dp),Y	31	DP Indirect Indexed, Y	x	x	x	2	5(+1+1/+1)
AND	(dp)	32	DP Indirect		x	x	2	5(+1+1)
AND	(sr,S),Y	33	SR Indirect Indexed,Y			x	2	7(+1)
BIT	dp,X	34	DP Indexed,X			x	2	4(+1+1)
AND	dp,X	35	DP Indexed,X	x	x	x	2	4(+1+1)
ROL	dp,X	36	DP Indexed,X	x	x	x	2	6(+1+2)
AND	[dp],Y	37	DP Indirect Long Indexed, Y			x	2	6(+1+1)
SEC		38	Implied	x	x	x	1	2
AND	addr,Y	39	Absolute Indexed,Y	x	x	x	3	4(+1+1)
DEC	A	3A	Accumulator			x	1	2
TSC		3B	Implied			x	1	2
BIT	addr,X	3C	Absolute Indexed,X			x	3	4(+1+1)
AND	addr,X	3D	Absolute Indexed,X	x	x	x	3	4(+1+1)
ROL	addr,X	3E	Absolute Indexed,X	x	x	x	3	7(+2+1)
AND	long,X	3F	Absolute Long Indexed,X			x	4	5(+1)
RTI		40	Stack (RTI)	x	x	x	1	6(+1)
EOR	(dp,X)	41	DP Indexed Indirect,X	x	x	x	2	6(+1+1)
WDM		42	n/a			x	2	n/a16
EOR	sr,S	43	Stack Relative			x	2	4(+1)
MVN	srcbk,destbk	44	Block Move			x	3	7(*n)
EOR	dp	45	Direct Page	x	x	x	2	3(+1+1)
LSR	dp	46	Direct Page	x	x	x	2	5(+1+2)
EOR	[dp]	47	DP Indirect Long			x	2	6(+1+1)
PHA		48	Stack (Push)	x	x	x	1	3(+1)
EOR	#const	49	Immediate	x	x	x	2(3)	2(+1)
LSR	A	4A	Accumulator	x	x	x	1	2
PHK		4B	Stack (Push)			x	1	3
JMP	addr	4C	Absolute	x	x	x	3	3
EOR	addr	4D	Absolute	x	x	x	3	4(+1)
LSR	addr	4E	Absolute	x	x	x	3	6(+2)
EOR	long	4F	Absolute Long			x	4	5(+1)
BVC	nearlabel	50	Program Counter Relative	x	x	x	2	2(+1+1)
EOR	(dp),Y	51	DP Indirect Indexed, Y	x	x	x	2	5(+1+1/+1)
EOR	(dp)	52	DP Indirect		x	x	2	5(+1+1)
EOR	(sr,S),Y	53	SR Indirect Indexed,Y			x	2	7(+1)
MVN	srcbk,destbk	54	Block Move			x	3	7(*n)
EOR	dp,X	55	DP Indexed,X	x	x	x	2	4(+1+1)
LSR	dp,X	56	DP Indexed,X	x	x	x	2	6(+1+2)
EOR	[dp],Y	57	DP Indirect Long Indexed, Y			x	2	6(+1+1)
CLI		58	Implied	x	x	x	1	2
EOR	addr,Y	59	Absolute Indexed,Y	x	x	x	3	4(+1+1)
PHY		5A	Stack (Push)		x	x	1	3(+1)
TCD		5B	Implied			x	1	2
JMP	long	5C	Absolute Long			x	4	4
EOR	addr,X	5D	Absolute Indexed,X	x	x	x	3	4(+1+1)
LSR	addr,X	5E	Absolute Indexed,X	x	x	x	3	7(+2/+1)
EOR	long,X	5F	Absolute Long Indexed,X			x	4	5(+1)
RTS		60	Stack (RTS)	x	x	x	1	6
ADC	(dp,X)	61	DP Indexed Indirect,X	x	x	x	2	6(+1+1+1)
PER	label	62	Stack (PC Relative Long)			x	3	6
ADC	sr,S	63	Stack Relative			x	2	4(+1+1)
STZ	dp	64	Direct Page		x	x	2	3(+1+1)
ADC	dp	65	Direct Page	x	x	x	2	3(+1+1+1)
ROR	dp	66	Direct Page	x	x	x	2	5(+1+2)
ADC	[dp]	67	DP Indirect Long			x	2	6(+1+1+1)
PLA		68	Stack (Pull)	x	x	x	1	4(+1)
ADC	#const	69	Immediate	x	x	x	2(3)	2(+1+1)
ROR	A	6A	Accumulator	x	x	x	1	2
RTL		6B	Stack (RTL)			x	1	6
JMP	(addr)	6C	Absolute Indirect	x	x	x	3	5(+1)
ADC	addr	6D	Absolute	x	x	x	3	4(+1+1)
ROR	addr	6E	Absolute	x	x	x	3	6(+2)
ADC	long	6F	Absolute Long			x	4	5(+1+1)
BVS	nearlabel	70	Program Counter Relative	x	x	x	2	2(+1+1)
ADC	(dp),Y	71	DP Indirect Indexed, Y	x	x	x	2	5(+1+1/+1)
ADC	(dp)	72	DP Indirect		x	x	2	5(+1+1+1)
ADC	(sr,S),Y	73	SR Indirect Indexed,Y			x	2	7(+1+1)
STZ	dp,X	74	DP Indexed,X			x	2	4(+1+1)
ADC	dp,X	75	DP Indexed,X	x	x	x	2	4(+1+1+1)
ROR	dp,X	76	DP Indexed,X	x	x	x	2	6(+1+2)
ADC	[dp],Y	77	DP Indirect Long Indexed, Y			x	2	6(+1+1/+1)
SEI		78	Implied	x	x	x	1	2
ADC	addr,Y	79	Absolute Indexed,Y	x	x	x	3	4(+1+1/+1)
PLY		7A	Stack (Pull)			x	1	4(+1)
TDC		7B	Implied			x	1	2
JMP	(addr,X)	7C	Absolute Indexed Indirect			x	3	6
ADC	addr,X	7D	Absolute Indexed,X	x	x	x	3	4(+1+1/+1)
ROR	addr,X	7E	Absolute Indexed,X	x	x	x	3	7(+2/+1)
ADC	long,X	7F	Absolute Long Indexed,X			x	4	5(+1+1)
BRA	nearlabel	80	Program Counter Relative			x	2	3(+1)
STA	(dp,X)	81	DP Indexed Indirect,X	x	x	x	2	6(+1+1)
BRL	label	82	Program Counter Relative Long			x	3	4
STA	sr,S	83	Stack Relative			x	2	4(+1)
STY	dp	84	Direct Page	x	x	x	2	3(+1+1)
STA	dp	85	Direct Page	x	x	x	2	3(+1+1)
STX	dp	86	Direct Page	x	x	x	2	3(+1+1)
STA	[dp]	87	DP Indirect Long			x	2	6(+1+1)
DEY		88	Implied	x	x	x	1	2
BIT	#const	89	Immediate			x	2(3)	2(+1)
TXA		8A	Implied	x	x	x	1	2
PHB		8B	Stack (Push)			x	1	3
STY	addr	8C	Absolute	x	x	x	3	4(+1)
STA	addr	8D	Absolute	x	x	x	3	4(+1)
STX	addr	8E	Absolute	x	x	x	3	4(+1)
STA	long	8F	Absolute Long			x	4	5(+1)
BCC	nearlabel	90	Program Counter Relative	x	x	x	2	2(+1+1)
STA	(dp),Y	91	DP Indirect Indexed, Y	x	x	x	2	6(+1+1)
STA	(dp)	92	DP Indirect		x	x	2	5(+1+1)
STA	(sr,S),Y	93	SR Indirect Indexed,Y			x	2	7(+1)
STY	dp,X	94	DP Indexed,X	x	x	x	2	4(+1+1)
STA	dpX	95	DP Indexed,X	x	x	x	2	4(+1+1)
STX	dp,Y	96	DP Indexed,Y	x	x	x	2	4(+1+1)
STA	[dp],Y	97	DP Indirect Long Indexed, Y			x	2	6(+1+1)
TYA		98	Implied	x	x	x	1	2
STA	addr,Y	99	Absolute Indexed,Y	x	x	x	3	5(+1)
TXS		9A	Implied	x	x	x	1	2
TXY		9B	Implied			x	1	2
STZ	addr	9C	Absolute		x	x	3	4(+1)
STA	addr,X	9D	Absolute Indexed,X	x	x	x	3	5(+1)
STZ	addr,X	9E	Absolute Indexed,X		x	x	3	5(+1)
STA	long,X	9F	Absolute Long Indexed,X			x	4	5(+1+1)
LDY	#const	A0	Immediate	x	x	x	2(3)	2(+1)
LDA	(dp,X)	A1	DP Indexed Indirect,X	x	x	x	2	6(+1+1)
LDX	#const	A2	Immediate	x	x	x	2(3)	2(+1)
LDA	sr,S	A3	Stack Relative			x	2	4(+1)
LDY	dp	A4	Direct Page	x	x	x	2	3(+1+1)
LDA	dp	A5	Direct Page	x	x	x	2	3(+1+1)
LDX	dp	A6	Direct Page	x	x	x	2	3(+1+1)
LDA	[dp]	A7	DP Indirect Long			x	2	6(+1+1)
TAY		A8	Implied	x	x	x	1	2
LDA	#const	A9	Immediate	x	x	x	2(3)	2(+1)
TAX		AA	Implied	x	x	x	1	2
PLB		AB	Stack (Pull)			x	1	4
LDY	addr	AC	Absolute	x	x	x	3	4(+1)
LDA	addr	AD	Absolute	x	x	x	3	4(+1)
LDX	addr	AE	Absolute	x	x	x	3	4(+1)
LDA	long	AF	Absolute Long			x	4	5(+1)
BCS	nearlabel	B0	Program Counter Relative	x	x	x	2	2(+1+1)
LDA	(dp),Y	B1	DP Indirect Indexed, Y	x	x	x	2	5(+1+1/+1)
LDA	(dp)	B2	DP Indirect		x	x	2	5(+1+1)
LDA	(sr,S),Y	B3	SR Indirect Indexed,Y			x	2	7(+1)
LDY	dp,X	B4	DP Indexed,X	x	x	x	2	4(+1+1)
LDA	dp,X	B5	DP Indexed,X	x	x	x	2	4(+1+1)
LDX	dp,Y	B6	DP Indexed,Y	x	x	x	2	4(+1+1)
LDA	[dp],Y	B7	DP Indirect Long Indexed, Y			x	2	6(+1+1)
CLV		B8	Implied	x	x	x	1	2
LDA	addr,Y	B9	Absolute Indexed,Y	x	x	x	3	4(+1+1)
TSX		BA	Implied	x	x	x	1	2
TYX		BB	Implied			x	1	2
LDY	addr,X	BC	Absolute Indexed,X	x	x	x	3	43,1
LDA	addr,X	BD	Absolute Indexed,X	x	x	x	3	4(+1+1)
LDX	addr,Y	BE	Absolute Indexed,Y	x	x	x	3	43,1
LDA	long,X	BF	Absolute Long Indexed,X			x	4	5(+1)
CPY	#const	C0	Immediate	x	x	x	2(3)	2(+1)
CMP	(dp,X)	C1	DP Indexed Indirect,X	x	x	x	2	6(+1+1)
REP	#const	C2	Immediate			x	2	3
CMP	sr,S	C3	Stack Relative			x	2	4(+1)
CPY	dp	C4	Direct Page	x	x	x	2	3(+1+1)
CMP	dp	C5	Direct Page	x	x	x	2	3(+1+1)
DEC	dp	C6	Direct Page	x	x	x	2	5(+1+2)
CMP	[dp]	C7	DP Indirect Long			x	2	6(+1+1)
INY		C8	Implied	x	x	x	1	2
CMP	#const	C9	Immediate	x	x	x	2(3)	2(+1)
DEX		CA	Implied	x	x	x	1	2
WAI		CB	Implied			x	1	3(+1)5
CPY	addr	CC	Absolute	x	x	x	3	4(+1)
CMP	addr	CD	Absolute	x	x	x	3	4(+1)
DEC	addr	CE	Absolute	x	x	x	3	6(+2)
CMP	long	CF	Absolute Long			x	4	5(+1)
BNE	nearlabel	D0	Program Counter Relative	x	x	x	2	2(+1+1)
CMP	(dp),Y	D1	DP Indirect Indexed, Y	x	x	x	2	5(+1+1/+1)
CMP	(dp)	D2	DP Indirect		x	x	2	5(+1+1)
CMP	(sr,S),Y	D3	SR Indirect Indexed,Y			x	2	7(+1)
PEI	(dp)	D4	Stack (DP Indirect)			x	2	62
CMP	dp,X	D5	DP Indexed,X	x	x	x	2	4(+1+1)
DEC	dp,X	D6	DP Indexed,X	x	x	x	2	6(+1+2)
CMP	[dp],Y	D7	DP Indirect Long Indexed, Y			x	2	6(+1+1)
CLD		D8	Implied	x	x	x	1	2
CMP	addr,Y	D9	Absolute Indexed,Y	x	x	x	3	4(+1+1)
PHX		DA	Stack (Push)			x	1	3(+1)
STP		DB	Implied			x	1	3(+)4
JMP	[addr]	DC	Absolute Indirect Long			x	3	6
CMP	addr,X	DD	Absolute Indexed,X	x	x	x	3	4(+1+1)
DEC	addr,X	DE	Absolute Indexed,X	x	x	x	3	7(+2+1)
CMP	long,X	DF	Absolute Long Indexed,X			x	4	5(+1)
CPX	#const	E0	Immediate	x	x	x	2(3)	2(+1)
SBC	(dp,X)	E1	DP Indexed Indirect,X	x	x	x	2	6(+1+1+1)
SEP		E2	Immediate			x	2	3
SBC	sr,S	E3	Stack Relative			x	2	4(+1+1)
CPX	dp	E4	Direct Page	x	x	x	2	3(+1+1)
SBC	dp	E5	Direct Page	x	x	x	2	3(+1+1+1)
INC	dp	E6	Direct Page	x	x	x	2	5(+1+2)
SBC	[dp]	E7	DP Indirect Long			x	2	6(+1+1+1)
INX		E8	Implied	x	x	x	1	2
SBC	#const	E9	Immediate	x	x	x	2(3)	2(+1+1)
NOP		EA	Implied	x	x	x	1	2
XBA		EB	Implied			x	1	3
CPX	addr	EC	Absolute	x	x	x	3	4(+1)
SBC	addr	ED	Absolute	x	x	x	3	4(+1+1)
INC	addr	EE	Absolute	x	x	x	3	6(+2)
SBC	long	EF	Absolute Long			x	4	5(+1+1)
BEQ	nearlabel	F0	Program Counter Relative	x	x	x	2	2(+1+1)
SBC	(dp),Y	F1	DP Indirect Indexed, Y	x	x	x	2	5(+1+1/+1)
SBC	(dp)	F2	DP Indirect		x	x	2	5(+1+1/+1)
SBC	(sr,S),Y	F3	SR Indirect Indexed,Y			x	2	7(+1+1)
PEA	addr	F4	Stack (Absolute)			x	3	5
SBC	dp,X	F5	DP Indexed,X	x	x	x	2	4(+1+1/+1)
INC	dp,X	F6	DP Indexed,X	x	x	x	2	6(+1+2)
SBC	[dp],Y	F7	DP Indirect Long Indexed, Y			x	2	6(+1+1/+1)
SED		F8	Implied	x	x	x	1	2
SBC	addr,Y	F9	Absolute Indexed,Y	x	x	x	3	4(+1+1/+1)
PLX		FA	Stack (Pull)			x	1	4(+1)
XCE		FB	Implied			x	1	2
JSR	(addr,X))	FC	Absolute Indexed Indirect			x	3	8
SBC	addr,X	FD	Absolute Indexed,X	x	x	x	3	4(+1+1/+1)
INC	addr,X	FE	Absolute Indexed,X	x	x	x	3	7(+2/+1)
SBC	long,X	FF	Absolute Long Indexed,X			x	4	5(+1+1)

OP	OPC	Bytes	Zykl		Alt-OP
BRK	00	1	7		
ORA ($ zz, X)	01	2	6		
JSR ($ zz)	02	2	7		
BBS 0, A, $ rr	03	2	4	Branch when the designated bit in the Accumulator is “1”	
-	04				
ORA $ zz	05	2	3		
ASL $ zz	06	2	5		
BBS 0, $ zz, $ rr	07	3	5	Branch when the designated bit in memory is “1”	Wie BBS 0,zp,rel (8F) er C02
PHP	08	1	3		
ORA # $ nn	09	2	2		
ASL A	0A	1	2		
SEB 0, A	0B	1	2	Set designated bit in the Accumulator	
-	0C				
ORA $ hhll	0D	3	4		
ASL $ hhll	0E	3	6		
SEB 0, $ zz	0F	2	5	Set designated bit in memory	Wie SMB 0,zp (87) der C02 SMB
BPL $ rr	10	2	2		
ORA ($ zz), Y	11	2	6		
CLT	12	1	2	Clear T flag - T flag : X Modified Operation Mode Flag	
BBC 0, A, $ rr	13	2	4	Branch when the designated bit in the Accumulator is “0”	
-	14				
ORA $ zz, X	15	2	4		
ASL $ zz, X	16	2	6		
BBC 0, $ zz, $ rr	17	3	5	Branch when the designated bit in memory is “0”	Wie BBR 0,zp,rel (0F) der C02 BBR
CLC	18	1	2		
ORA $ hhll, Y	19	3	5		
DEC A	1A	1	2		
CLB 0, A	1B	1	2	Clear designated bit in the Accumulator	
-	1C				
ORA $ hhll, X	1D	3	5		
ASL $ hhll, X	1E	3	7		
CLB 0, $ zz	1F	2	5	Clear designated bit in memory	Wie RMB 0,zp (07) der C02 RMB
JSR $ hhll	20	3	6		
AND ($ zz, X)	21	2	6		
JSR $ FF sp	22	2	5	JSR special page ($FFxx)	
BBS 1, A, $ rr	23	2	4		
BIT $ zz	24	2	3		
AND $ zz	25	2	3		
ROL $ zz	26	2	5		
BBS 1, $ zz, $ rr	27	3	5		
PLP	28	1	4		
AND # $ nn	29	2	2		
ROL A	2A	1	2		
SEB 1, A	2B	1	2		
BIT $ hhll	2C	3	4		
AND $ hhll	2D	3	4		
ROL $ hhll	2E	3	6		
SEB 1, $ zz	2F	2	5		
BMI $ rr	30	2	2		
AND ($ zz), Y	31	2	6		
SET	32	1	2	Set T flag - T flag : X Modified Operation Mode Flag	
BBC 1, A, $ rr	33	2	4		
-	34				
AND $ zz, X	35	2	4		
ROL $ zz, X	36	2	6		
BBC 1, $ zz, $ rr	37	3	5		
SEC	38	1	2		
AND $ hhll, Y	39	3	5		
INC A	3A	1	2		
CLB 1, A	3B	1	2		
LDM # $ nn, $ zz	3C	3	4	Load immediate value into memory	
AND $ hhll, X	3D	3	5		
ROL $ hhll, X	3E	3	7		
CLB 1, $ zz	3F	2	5		
RTI	40	1	6		
EOR ($ zz, X)	41	2	6		
STP	42	1	2	Stops the oscillation of oscillator	
BBS 2, A, $ rr	43	2	4		
COM $ zz	44	2	5	Store one’s complement of memory contents to memory	
EOR $ zz	45	2	3		
LSR $ zz	46	2	5		
BBS 2, $ zz, $ rr	47	3	5		
PHA	48	1	3		
EOR # $ nn	49	2	2		
LSR A	4A	1	2		
SEB 2, A	4B	1	2		
JMP $ hhll	4C	3	3		
EOR $ hhll	4D	3	4		
LSR $ hhll	4E	3	6		
SEB 2, $ zz	4F	2	5		
BVC $ rr	50	2	2		
EOR ($ zz), Y	51	2	6		
-	52				
BBC 2, A, $ rr	53	2	4		
-	54				
EOR $ zz, X	55	2	4		
LSR $ zz, X	56	2	6		
BBC 2, $ zz, $ rr	57	3	5		
CLI	58	1	2		
EOR $ hhll, Y	59	3	5		
-	5A				
CLB 2, A	5B	1	2		
-	5C				
EOR $ hhll, X	5D	3	5		
LSR $ hhll, X	5E	3	7		
CLB 2, $ zz	5F	2	5		
RTS	60	1	6		
ADC ($ zz, X)	61	2	6		
MUL $ zz, X	62	2	15	Multiply Accumulator with memory and store high-order byte of result on Stack and low-order byte in Accumulator	
BBS 3, A, $ rr	63	2	4		
TST $ zz	64	2	3	Test whether memory content is “0” or not	
ADC $ zz	65	2	3		
ROR $ zz	66	2	5		
BBS 3, $ zz, $ rr	67	3	5		
PLA	68	1	4		
ADC # $ nn	69	2	2		
ROR A	6A	1	2		
SEB 3, A	6B	1	2		
JMP ($ hhll)	6C	3	5		
ADC $ hhll	6D	3	4		
ROR $ hhll	6E	3	6		
SEB 3, $ zz	6F	2	5		
BVS $ rr	70	2	2		
ADC ($ zz), Y	71	2	6		
-	72				
BBC 3, A, $ rr	73	2	4		
-	74				
ADC $ zz, X	75	2	4		
ROR $ zz, X	76	2	6		
BBC 3, $ zz, $ rr	77	3	5		
SEI	78	1	2		
ADC $ hhll, Y	79	3	5		
-	7A				
CLB 3, A	7B	1	2		
-	7C				
ADC $ hhll, X	7D	3	5		
ROR $ hhll, X	7E	3	7		
CLB 3, $ zz	7F	2	5		
BRA $ rr	80	2	4		
STA ($ zz, X)	81	2	7		
RRF $ zz	82	2	8	Rotate four bits right witout carry (memory)	
BBS 4, A, $ rr	83	2	4		
STY $ zz	84	2	4		
STA $ zz	85	2	4		
STX $ zz	86	2	4		
BBS 4, $ zz, $ rr	87	3	5		
DEY	88	1	2		
-	89				
TXA	8A	1	2		
SEB 4, A	8B	1	2		
STY $ hhll	8C	3	6		
STA $ hhll	8D	3	5		
STX $ hhll	8E	3	5		
SEB 4, $ zz	8F	2	5		
BCC $ rr	90	2	2		
STA ($ zz), Y	91	2	7		
-	92				
BBC 4, A, $ rr	93	2	4		
STY $ zz, X	94	2	5		
STA $ zz, X	95	2	5		
STX $ zz, Y	96	2	5		
BBC 4, $ zz, $ rr	97	3	5		
TYA	98	1	2		
STA $ hhll, Y	99	3	6		
TXS	9A	1	2		
CLB 4, A	9B	1	2		
-	9C				
STA $ hhll, X	9D	3	6		
-	9E				
CLB 4, $ zz	9F	2	5		
LDY # $ nn	A0	2	2		
LDA ($ zz, X)	A1	2	6		
LDX # $ nn	A2	2	2		
BBS 5, A, $ rr	A3	2	4		
LDY $ zz	A4	2	3		
LDA $ zz	A5	2	3		
LDX $ zz	A6	2	3		
BBS 5, $ zz, $ rr	A7	3	5		
TAY	A8	1	2		
LDA # $ nn	A9	2	2		
TAX	AA	1	2		
SEB 5, A	AB	1	2		
LDY $ hhll	AC	3	4		
LDA $ hhll	AD	3	4		
LDX $ hhll	AE	3	4		
SEB 5, $ zz	AF	2	5		
BCS $ rr	B0	2	2		
LDA ($ zz), Y	B1	2	6		
JMP ($ zz)	B2	2	4	Jmp ZP indirekt	
BBC 5, A, $ rr	B3	2	4		
LDY $ zz, X	B4	2	4		
LDA $ zz, X	B5	2	4		
LDX $ zz, Y	B6	2	4		
BBC 5, $ zz, $ rr	B7	3	5		
CLV	B8	1	2		
LDA $ hhll, Y	B9	3	5		
TSX	BA	1	2		
CLB 5, A	BB	1	2		
LDY $ hhll, X	BC	3	5		
LDA $ hhll, X	BD	3	5		
LDX $ hhll, Y	BE	3	5		
CLB 5, $ zz	BF	2	5		
CPY # $ nn	C0	2	2		
CMP ($ zz, X)	C1	2	6		
WIT	C2	1	2	Stops the internal clock	
BBS 6, A, $ rr	C3	2	4		
CPY $ zz	C4	2	3		
CMP $ zz	C5	2	3		
DEC $ zz	C6	2	5		
BBS 6, $ zz, $ rr	C7	3	5		
INY	C8	1	2		
CMP # $ nn	C9	2	2		
DEX	CA	1	2		
SEB 6, A	CB	1	2		
CPY $ hhll	CC	3	4		
CMP $ hhll	CD	3	4		
DEC $ hhll	CE	3	6		
SEB 6, $ zz	CF	2	5		
BNE $ rr	D0	2	2		
CMP ($ zz), Y	D1	2	6		
-	D2				
BBC 6, A, $ rr	D3	2	4		
-	D4				
CMP $ zz, X	D5	2	4		
DEC $ zz, X	D6	2	6		
BBC 6, $ zz, $ rr	D7	3	5		
CLD	D8	1	2		
CMP $ hhll, Y	D9	3	5		
-	DA				
CLB 6, A	DB	1	2		
-	DC				
CMP $ hhll, X	DD	3	5		
DEC $ hhll, X	DE	3	7		
CLB 6, $ zz	DF	2	5		
CPX # $ nn	E0	2	2		
SBC ($ zz, X)	E1	2	6		
DIV $ zz, X	E2	2	16	Divide Accumulator by memory - Quotient is stored in Accumulator and one’s complement of remainder is pushed onto stack	
BBS 7, A, $ rr	E3	2	4		
CPX $ zz	E4	2	3		
SBC $ zz	E5	2	3		
INC $ zz	E6	2	5		
BBS 7, $ zz, $ rr	E7	3	5		
INX	E8	1	2		
SBC # $ nn	E9	2	2		
NOP	EA	1	2		
SEB 7, A	EB	1	2		
CPX $ hhll	EC	3	4		
SBC $ hhll	ED	3	4		
INC $ hhll	EE	3	6		
SEB 7, $ zz	EF	2	5		
BEQ $ rr	F0	2	2		
SBC ($ zz), Y	F1	2	6		
-	F2				
BBC 7, A, $ rr	F3	2	4		
-	F4				
SBC $ zz, X	F5	2	4		
INC $ zz, X	F6	2	6		
BBC 7, $ zz, $ rr	F7	3	5		
SED	F8	1	2		
SBC $ hhll, Y	F9	3	5		
-	FA				
CLB 7, A	FB	1	2		
-	FC				
SBC $ hhll, X	FD	3	5		
INC $ hhll, X	FE	3	7		
CLB 7, $ zz	FF	2	5		

HuC6280	OPC	Bytes	Zykl		WDC65C02 mit zusatz fuer PC-Engine	
BRK	00	1				
ORA (zp,X)	01	2				
SXY	02	1	3	Swap X and Y Registers		
-	03	2				
TSB zp	04	2				
ORA zp	05	2				
ASL zp	06	2				
RMB 0,zp	07	2				
PHP	08	1				
ORA #n	09	2				
ASL A	0A	1				
-	0B					
TSB abs	0C	3				
ORA abs	0D	3				
ASL abs	0E	3				
BBR 0,zp,rel	0F	3				
BPL rel	10	2				
ORA (zp),Y	11	2				
ORA (zp)	12	2				
ST1	13	2	4	Store HuC6270 No. 1 (Low Data Register)		
TRB zp	14	2				
ORA zp,X	15	2				
ASL zp,X	16	2				
RMB 1,zp	17	3				
CLC	18	1				
ORA abs,Y	19	3				
INC A	1A	1				
-	1B					
TRB abs	1C	3				
ORA abs,X	1D	3				
ASL abs,X	1E	3				
BBR 1,zp,rel	1F	3				
JSR abs	20	3				
AND (zp,X)	21	2				
SAX	22	1	3	Swap Accumulator and X Register		
ST2	23	2	4	Store HuC6270 No. 2 (High Data Register)		
BIT zp	24	2				
AND zp	25	2				
ROL zp	26	2				
RMB 2,zp	27	3				
PLP	28	1				
AND #n	29	2				
ROL A	2A	1				
-	2B					
BIT abs	2C	3				
AND abs	2D	3				
ROL abs	2E	3				
BBR 2,zp,rel	2F	3				
BMI rel	30	2				
AND (zp),Y	31	2				
AND (zp)	32	2				
-	33					
BIT zp,X	34	2				
AND zp,X	35	2				
ROL zp,X	36	2				
RMB 3,zp	37	3				
SEC	38	1				
AND abs,Y	39	3				
DEC A	3A	1				
-	3B					
BIT abs,X	3C	3				
ORA abs,X	3D	3				
ASL abs,X	3E	3				
BBR 3,zp,rel	3F	3				
RTI	40	1				
EOR (zp,X)	41	2				
SAY	42	1	3	Swap Accumulator and Y Register		
TMA #n	53	2	5	Transfer MPRi to Accumulator (TMAi)	#n = Bitmaske der 8 mögichen Register	
-	44					
EOR zp	45	2				
LSR zp	46	2				
RMB 4,zp	47	3				
PHA	48	1				
EOR #n	49	2				
LSR A	4A	1				
-	4B					
JMP abs	4C	3				
EOR abs	4D	3				
LSR abs	4E	3				
BBR 4,zp,rel	4F	3				
BVC rel	50	2				
EOR (zp),Y	51	2				
EOR (zp)	52	2				
TAM #n	53	2	5	Transfer Accumulator to MPRi (TAMi)	#n = Bitmaske der 8 mögichen Register	
CSL	54	1		Change Speed Low		
EOR zp,X	55	2				
LSR zp,X	56	2				
RMB 5,zp	57	3				
CLI	58	1				
EOR abs,Y	59	3				
PHY	5A	1				
-	5B					
-	5C					
EOR abs,X	5D	3				
LSR abs,X	5E	3				
BBR 5,zp,rel	5F	3				
RTS	60	1				
ADC (zp,X)	61	2				
CLA	62	1	2	Clear Accumulator		
-	63					
STZ zp	64	2				
ADC zp	65	2				
ROR zp	66	2				
RMB 6,zp	67	3				
PLA	68	1				
ADC #n	69	2				
ROR A	6A	1				
-	6B					
JMP (abs)	6C	3				
ADC abs	6D	3				
ROR abs	6E	3				
BBR 6,zp,rel	6F	3				
BVS rel	70	2				
ADC (zp),Y	71	2				
ADC (zp)	72	2				
TII s,d,l	73	7	17+6n	Transfer Increment Increment	Blockmove	73 SL SH DL DH LL LH
STZ zp,X	74	2				
ADC zp,X	75	2				
ROR zp,X	76	2				
RMB 7,zp	77	3				
SEI	78	1				
ADC abs,Y	79	3				
PLY	7A	1				
-	7B					
JMP (abs,X)	7C	3				
ADC abs,X	7D	3				
ROR abs,X	7E	3				
BBR 7,zp,rel	7F	3				
BRA rel	80	2				
STA (zp,X)	81	2				
CLX	82	1	2	Clear X Register		
TST #n, zp	83	3	7	Test and Reset Memory Bits		
STY zp	84	2				
STA zp	85	2				
STX zp	86	2				
SMB 0,zp	87	3				
DEY	88	1				
BIT #n	89	2				
TXA	8A	1	2			
-	8B					
STY abs	8C	3				
STA abs	8D	3				
STX abs	8E	3				
BBS 0,zp,rel	8F	3				
BCC rel	90	2				
STA (zp),Y	91	2				
STA (zp)	92	2				
TST #n, abs	93	4	8	Test and Reset Memory Bits		
STY zp	94	2				
STA zp,X	95	2				
STX zp,Y	96	2				
SMB 1,zp	97	3				
TYA	98	1	2			
STA abs,Y	99	3				
TXS	9A	1	2			
-	9B					
STZ abs	9C	3				
STA abs,X	9D	3				
STZ abs,X	9E	3				
BBS 1,zp,rel	9F	3				
LDY #n	A0	2				
LDA (zp,X)	A1	2				
LDX #n	A2	2				
TST #n, zp,X	A3	3	7	Test and Reset Memory Bits		
LDY zp	A4	2				
LDA zp	A5	2				
LDX zp	A6	2				
SMB 2,zp	A7	3				
TAY	A8	1				
LDA #n	A9	2				
TAX	AA	1				
-	AB					
LDY abs	AC	3				
LDA abs	AD	3				
LDX abs	AE	3				
BBS 2,zp,rel	AF	3				
BCS rel	B0	2				
LDA (zp),Y	B1	2				
LDA (zp)	B2	2				
TST #n, abs,X	B3	4	8	Test and Reset Memory Bits		
LDY zp	B4	2				
LDA zp,X	B5	2				
LDX zp,Y	B6	2				
SMB 3,zp	B7	3				
CLV	B8	1				
LDA abs,Y	B9	3				
TSX	BA	1				
-	BB					
LDY abs,X	BC	3				
LDA abs,X	BD	3				
LDX abs,Y	BE	3				
BBS 3,zp,rel	BF	3				
CPY #n	C0	2				
CMP (zp,X)	C1	2				
CLY	C2	1	2	Clear Y Register		
TDD s,d,l	C3	7	17+6n	Transfer Decrement Decrement	Blockmove	C3 SL SH DL DH LL LH
CPY zp	C4	2				
CMP zp	C5	2				
DEC zp	C6	2				
SMB 4,zp	C7	3				
INY	C8	1				
CMP #n	C9	2				
DEX	CA	1				
-	CB					
CPY abs	CC	3				
CMP abs	CD	3				
DEC abs	CE	3				
BBS 4,zp,rel	CF	3				
BNE rel	D0	2				
CMP (zp),Y	D1	2				
CMP (zp)	D2	1				
TIN s,d,l	D3	7	17+6n	Transfer Increment None	Blockmove	D3 SL SH DL DH LL LH
CSH	D4	1		Change Speed High		
CMP zp,X	D5	2				
DEC zp,X	D6	2				
SMB 5,zp	D7	3				
CLD	D8	1				
CMP abs,Y	D9	3				
PHX	DA	1				
-	DB					
-	DC					
CMP abs,X	DD	3				
DEC abs,X	DE	3				
BBS 5,zp,rel	DF	3				
CPX #n	E0	2				
SBC (zp,X)	E1	2				
-	E2					
TIA s,d,l	E3	7	17+6n	Transfer Increment Alternate	Blockmove	E3 SL SH DL DH LL LH
CPX zp	E4	2				
SBC zp	E5	2				
INC zp	E6	2				
SMB 6,zp	E7	3				
INX	E8	1				
SBC #n	E9	2				
NOP	EA	1				
SBC #n	EB	2				
CPX abs	EC	3				
SBC abs	ED	3				
INC abs	EE	3				
BBS 6,zp,rel	EF	3				
BEQ rel	F0	2				
SBC (zp),Y	F1	2				
SBC (zp)	F2	2				
TAI s,d,l	F3	7	17+6n	Transfer Alternate Increment	Blockmove	F3 SL SH DL DH LL LH
SET	F4	1	2	Set T Flag	Prefixbyte	Wie Renesas 740 wird aber scheinbar imemr wieder zurueckgesetzt
SBC zp,X	F5	2				
INC zp,X	F6	2				
SMB 7,zp	F7	3				
SED	F8	1				
SBC abs,Y	F9	3				
PLX	FA	1				
-	FB					
-	FC					
SBC abs,X	FD	3				
INC abs,X	FE	3				
BBS 7,zp,rel	FF	3				