

store_reg (label Str_m)	6																							
store_reg * (label Str_o)	10																							
store_modify *	8																							
adgen_s_z	5																							
adgen_s_zx, adgen_s_zy	6																							
IRQ_restore	4																							
flags_gen_nz	3																							
reg_fetch	5																							
op_fetch	5																							
adgen_z	6																							
adgen_zx, adgen_zy	7																							
adgen_zi	14																							
adgen_zxi, adgen_ziy	15																							
adgen_x, adgen_y	10																							
adgen_a	7																							
op_decode **	9																							
op_decode_single **	13																							
individual cycles																								
ns if page crossed																								
page cross add																								
65C02 cycles																								
65C02 ns @ 2MHz negative=AVR slower positive=AVR faster																								
AVR ns @ 16MHz																								
Total AVR cyles used																								
Instruction (condition f=#)																								
BBS zp bit=1	31	1937,5	562,5	5	1	1062,5	18	13																
JMP abs	20	1250,0	250,0	3		250,0	7	13																
JMP (abs)	32	2000,0	1000,0	6		1000,0	19	13																
JMP (abs,x)	33	2062,5	937,5	6		937,5	20	13																
JSR abs	35	2187,5	812,5	6		812,5	22	13																
RTS	26	1625,0	1375,0	6		1375,0	13	13																
RTI	39	2437,5	562,5	6		562,5	26		9									4						
RTI after WAI	44	2750,0	250,0	6		250,0	27	13										4						
BRK	49	3062,5	437,5	7		437,5	36	13																
IRQ	55	3437,5	62,5	7		62,5	42	13																
NMI	56	3500,0	0,0	7		0,0	43	13																
ADC abs D=0	33	2062,5	-62,5	4		-62,5	8	13		7					5									
ADC abs,x D=0	36	2250,0	-250,0	4	1	250,0	8	13			10				5									
ADC abs,y D=0	36	2250,0	-250,0	4	1	250,0	8	13			10				5									
ADC (zp) D=0	40	2500,0	0,0	5		0,0	8	13				14			5									
ADC (zp,x) D=0	41	2562,5	437,5	6		437,5	8	13				15			5									
ADC (zp),y D=0	41	2562,5	-62,5	5	1	437,5	8	13				15			5									
ADC zp,x D=0	28	1750,0	250,0	4		250,0	8	13					7											
ADC zp D=0	27	1687,5	-187,5	3		-187,5	8	13						6										
ADC # D=0	21	1312,5	-312,5	2		-312,5	8	13																
ADC abs D=1	43	2687,5	-187,5	5		-187,5	15	13		7					5		3							
ADC abs,x D=1	46	2875,0	-375,0	5	1	125,0	15	13			10				5		3							
ADC abs,y D=1	46	2875,0	-375,0	5	1	125,0	15	13			10				5		3							
ADC (zp) D=1	50	3125,0	-125,0	6		-125,0	15	13				14			5		3							
ADC (zp,x) D=1	51	3187,5	312,5	7		312,5	15	13				15			5		3							
ADC (zp),y D=1	51	3187,5	-187,5	6	1	312,5	15	13				15			5		3							

store_reg (label Str_m)	6																				
store_reg * (label Str_o)	10																				
store_modify *	8																				
adgen_s_z	5																				
adgen_s_zx, adgen_s_zy	6																				
IRQ_restore	4																				
flags_gen_nz	3																				
reg_fetch	5																				
op_fetch	5																				
adgen_z	6																				
adgen_zx, adgen_zy	7																				
adgen_zi	14																				
adgen_zxi, adgen_ziy	15																				
adgen_x, adgen_y	10																				
adgen_a	7																				
op_decode_single **	9																				
op_decode **	13																				
individual cycles																					
ns if page crossed																					
page cross add																					
65C02 cycles																					
65C02 ns @ 2MHz negative=AVR slower positive=AVR faster																					
AVR ns @ 16MHz																					
Total AVR cyles used																					
Instruction (condition f=#)																					
ADC zp,x D=1	38	2375,0	125,0	5		125,0	15	13						7			3				
ADC zp D=1	37	2312,5	-312,5	4		-312,5	15	13							6			3			
ADC # D=1	31	1937,5	-437,5	3		-437,5	15	13									3				
SBC abs D=0	35	2187,5	-187,5	4		-187,5	10	13		7							5				
SBC abs,x D=0	38	2375,0	-375,0	4	1	125,0	10	13			10						5				
SBC abs,y D=0	38	2375,0	-375,0	4	1	125,0	10	13			10						5				
SBC (zp) D=0	42	2625,0	-125,0	5		-125,0	10	13				14					5				
SBC (zp,x) D=0	43	2687,5	312,5	6		312,5	10	13				15					5				
SBC (zp),y D=0	43	2687,5	-187,5	5	1	312,5	10	13				15					5				
SBC zp,x D=0	30	1875,0	125,0	4		125,0	10	13						7							
SBC zp D=0	29	1812,5	-312,5	3		-312,5	10	13							6						
SBC # D=0	23	1437,5	-437,5	2		-437,5	10	13													
SBC abs D=1	42	2625,0	-125,0	5		-125,0	14	13		7							5		3		
SBC abs,x D=1	45	2812,5	-312,5	5	1	187,5	14	13			10						5		3		
SBC abs,y D=1	45	2812,5	-312,5	5	1	187,5	14	13			10						5		3		
SBC (zp) D=1	49	3062,5	-62,5	6		-62,5	14	13					14				5		3		
SBC (zp,x) D=1	50	3125,0	375,0	7		375,0	14	13				15					5		3		
SBC (zp),y D=1	50	3125,0	-125,0	6	1	375,0	14	13				15					5		3		
SBC zp,x D=1	37	2312,5	187,5	5		187,5	14	13							7				3		
SBC zp D=1	36	2250,0	-250,0	4		-250,0	14	13								6			3		
SBC # D=1	30	1875,0	-375,0	3		-375,0	14	13											3		
AND abs	29	1812,5	187,5	4		187,5	4	13		7							5				
AND abs,x	32	2000,0	0,0	4	1	500,0	4	13			10						5				
AND abs,y	32	2000,0	0,0	4	1	500,0	4	13			10						5				
AND (zp)	36	2250,0	250,0	5		250,0	4	13					14				5				
AND (zp,x)	37	2312,5	687,5	6		687,5	4	13				15					5				

Instruction (condition f=#)	Total AVR cycles used	AVR ns @ 16MHz	65C02 ns @ 2MHz negative=AVR slower positive=AVR faster	65C02 cycles	page cross add	ns if page crossed	individual cycles	op_decode **	op_decode_single **	adgen_a	adgen_x, adgen_y	adgen_zxi, adgen_ziy	adgen_zi	adgen_zx, adgen_zy	adgen_z	op_fetch	reg_fetch	flags_gen_nz	IRQ_restore	adgen_s_zx, adgen_s_zy	adgen_s_z	store_modify *	store_reg * (label STR_o)	store_reg (label STR_m)
								13	9	7	10	15	14	7	6	5	5	3	4	6	5	8	10	6
AND (zp),y	37	2312,5	187,5	5	1	687,5	4 13					15				5								
AND zp,x	24	1500,0	500,0	4		500,0	4 13							7										
AND zp	23	1437,5	62,5	3		62,5	4 13								6									
AND #	17	1062,5	-62,5	2		-62,5	4 13																	
EOR abs	29	1812,5	187,5	4		187,5	4 13			7						5								
EOR abs,x	32	2000,0	0,0	4	1	500,0	4 13				10					5								
EOR abs,y	32	2000,0	0,0	4	1	500,0	4 13				10					5								
EOR (zp)	36	2250,0	250,0	5		250,0	4 13						14			5								
EOR (zp,x)	37	2312,5	687,5	6		687,5	4 13					15				5								
EOR (zp),y	37	2312,5	187,5	5	1	687,5	4 13					15				5								
EOR zp,x	24	1500,0	500,0	4		500,0	4 13							7										
EOR zp	23	1437,5	62,5	3		62,5	4 13								6									
EOR #	17	1062,5	-62,5	2		-62,5	4 13																	
ORA abs	29	1812,5	187,5	4		187,5	4 13			7						5								
ORA abs,x	32	2000,0	0,0	4	1	500,0	4 13				10					5								
ORA abs,y	32	2000,0	0,0	4	1	500,0	4 13				10					5								
ORA (zp)	36	2250,0	250,0	5		250,0	4 13						14			5								
ORA (zp,x)	37	2312,5	687,5	6		687,5	4 13					15				5								
ORA (zp),y	37	2312,5	187,5	5	1	687,5	4 13					15				5								
ORA zp,x	24	1500,0	500,0	4		500,0	4 13							7										
ORA zp	23	1437,5	62,5	3		62,5	4 13								6									
ORA #	17	1062,5	-62,5	2		-62,5	4 13																	
CMP abs	29	1812,5	187,5	4		187,5	4 13			7						5								
CMP abs,x	32	2000,0	0,0	4	1	500,0	4 13				10					5								
CMP abs,y	32	2000,0	0,0	4	1	500,0	4 13				10					5								
CMP (zp)	36	2250,0	250,0	5		250,0	4 13						14			5								

Instruction (condition f=#)	Total AVR cycles used	AVR ns @ 16MHz	65C02 ns @ 2MHz negative=AVR slower positive=AVR faster	65C02 cycles	page cross add	ns if page crossed	individual cycles	op_decode **	op_decode_single **	adgen_a	adgen_x, adgen_y	adgen_zxi, adgen_ziy	adgen_zi	adgen_zx, adgen_zy	adgen_z	op_fetch	reg_fetch	flags_gen_nz	IRQ_restore	adgen_s_zx, adgen_s_zy	adgen_s_z	store_modify *	store_reg * (label STR_o)	store_reg (label STR_m)
								13	9	7	10	15	14	7	6	5	5	3	4	6	5	8	10	6
CMP (zp,x)	37	2312,5	687,5	6		687,5	4 13					15			5									
CMP (zp),y	37	2312,5	187,5	5	1	687,5	4 13					15			5									
CMP zp,x	24	1500,0	500,0	4		500,0	4 13							7										
CMP zp	23	1437,5	62,5	3		62,5	4 13								6									
CMP #	17	1062,5	-62,5	2		-62,5	4 13																	
BIT abs	34	2125,0	-125,0	4		-125,0	9 13			7						5								
BIT abs,x	37	2312,5	-312,5	4	1	187,5	9 13				10					5								
BIT zp,x	29	1812,5	187,5	4		187,5	9 13							7										
BIT zp	28	1750,0	-250,0	3		-250,0	9 13								6									
BIT #	18	1125,0	-125,0	2		-125,0	5 13																	
CPX abs	29	1812,5	187,5	4		187,5	4 13			7						5								
CPX zp	23	1437,5	62,5	3		62,5	4 13								6									
CPX #	17	1062,5	-62,5	2		-62,5	4 13																	
CPY abs	29	1812,5	187,5	4		187,5	4 13			7						5								
CPY zp	23	1437,5	62,5	3		62,5	4 13								6									
CPY #	17	1062,5	-62,5	2		-62,5	4 13																	
LDA abs	29	1812,5	187,5	4		187,5	1 13			7						5		3						
LDA abs,x	32	2000,0	0,0	4	1	500,0	1 13				10					5		3						
LDA abs,y	32	2000,0	0,0	4	1	500,0	1 13				10					5		3						
LDA (zp)	36	2250,0	250,0	5		250,0	1 13						14			5		3						
LDA (zp,x)	37	2312,5	687,5	6		687,5	1 13					15				5		3						
LDA (zp),y	37	2312,5	187,5	5	1	687,5	1 13					15				5		3						
LDA zp,x	24	1500,0	500,0	4		500,0	1 13							7				3						
LDA zp	23	1437,5	62,5	3		62,5	1 13								6			3						
LDA #	17	1062,5	-62,5	2		-62,5	1 13											3						
LDX abs	29	1812,5	187,5	4		187,5	1 13			7						5		3						

store_reg (label Str_m)	6															
store_reg * (label Str_o)	10															
store_modify *	8															
adgen_s_z	5															
adgen_s_zx, adgen_s_zy	6															
IRQ_restore	4															
flags_gen_nz	3															
reg_fetch	5															
op_fetch	5															
adgen_z	6															
adgen_zx, adgen_zy	7															
adgen_zi	14															
adgen_zxi, adgen_ziy	15															
adgen_x, adgen_y	10															
adgen_a	7															
op_decode_single **	9															
op_decode **	13															
individual cycles																
ns if page crossed																
page cross add																
65C02 cycles																
65C02 ns @ 2MHz negative=AVR slower positive=AVR faster																
AVR ns @ 16MHz																
Total AVR cyles used																
Instruction (condition f=#)																
LDX abs,y	32	2000,0	0,0	4	1	500,0	1	13			10					
LDX zp,y	24	1500,0	500,0	4		500,0	1	13				7				
LDX zp	23	1437,5	62,5	3		62,5	1	13					6			
LDX #	17	1062,5	-62,5	2		-62,5	1	13								
LDY abs	29	1812,5	187,5	4		187,5	1	13		7				5		
LDY abs,x	32	2000,0	0,0	4	1	500,0	1	13			10			5		
LDY zp,x	24	1500,0	500,0	4		500,0	1	13				7				
LDY zp	23	1437,5	62,5	3		62,5	1	13					6			
LDY #	17	1062,5	-62,5	2		-62,5	1	13							3	
STA abs	30	1875,0	125,0	4		125,0	0	13		7						10
STA abs,x	33	2062,5	437,5	5		437,5	0	13			10					10
STA abs,y	33	2062,5	437,5	5		437,5	0	13			10					10
STA (zp)	37	2312,5	187,5	5		187,5	0	13				14				10
STA (zp,x)	38	2375,0	625,0	6		625,0	0	13				15				10
STA (zp),y	38	2375,0	625,0	6		625,0	0	13				15				10
STA zp,x	25	1562,5	437,5	4		437,5	0	13							6	6
STA zp	24	1500,0	0,0	3		0,0	0	13							5	6
STX abs	30	1875,0	125,0	4		125,0	0	13		7						10
STX zp,y	25	1562,5	437,5	4		437,5	0	13							6	6
STX zp	24	1500,0	0,0	3		0,0	0	13							5	6
STY abs	30	1875,0	125,0	4		125,0	0	13		7						10
STY zp,x	25	1562,5	437,5	4		437,5	0	13							6	6
STY zp	24	1500,0	0,0	3		0,0	0	13							5	6
STZ abs	30	1875,0	125,0	4		125,0	0	13		7						10
STZ abs,x	33	2062,5	437,5	5		437,5	0	13			10					10
STZ zp,x	25	1562,5	437,5	4		437,5	0	13							6	6

Instruction (condition f=#)	Total AVR cycles used	AVR ns @ 16MHz	65C02 ns @ 2MHz negative=AVR slower positive=AVR faster	65C02 cycles	page cross add	ns if page crossed	individual cycles	op_decode **	op_decode_single **	adgen_a	adgen_x, adgen_y	adgen_zxi, adgen_ziy	adgen_zi	adgen_zx, adgen_zy	adgen_z	op_fetch	reg_fetch	flags_gen_nz	IRQ_restore	adgen_s_zx, adgen_s_zy	adgen_s_z	store_modify *	store_reg * (label STR_o)	store_reg (label STR_m)
								13	9	7	10	15	14	7	6	5	5	3	4	6	5	8	10	6
STZ zp	24	1500,0	0,0	3		0,0	0	13													5			6
ASL abs	36	2250,0	750,0	6		750,0	3	13		7						5						8		
ASL abs,x	39	2437,5	562,5	6	1	1062,5	3	13			10					5						8		
ASL zp,x	31	1937,5	1062,5	6		1062,5	3	13						7								8		
ASL zp	30	1875,0	625,0	5		625,0	3	13							6							8		
LSR abs	36	2250,0	750,0	6		750,0	3	13		7						5						8		
LSR abs,x	39	2437,5	562,5	6	1	1062,5	3	13			10					5						8		
LSR zp,x	31	1937,5	1062,5	6		1062,5	3	13						7								8		
LSR zp	30	1875,0	625,0	5		625,0	3	13							6							8		
ROL abs	37	2312,5	687,5	6		687,5	4	13		7						5						8		
ROL abs,x	40	2500,0	500,0	6	1	1000,0	4	13			10					5						8		
ROL zp,x	32	2000,0	1000,0	6		1000,0	4	13						7								8		
ROL zp	31	1937,5	562,5	5		562,5	4	13							6							8		
ROR abs	37	2312,5	687,5	6		687,5	4	13		7						5						8		
ROR abs,x	40	2500,0	500,0	6	1	1000,0	4	13			10					5						8		
ROR zp,x	32	2000,0	1000,0	6		1000,0	4	13						7								8		
ROR zp	31	1937,5	562,5	5		562,5	4	13							6							8		
DEC abs	37	2312,5	687,5	6		687,5	4	13		7						5						8		
DEC abs,x	40	2500,0	1000,0	7		1000,0	4	13			10					5						8		
DEC zp,x	32	2000,0	1000,0	6		1000,0	4	13						7								8		
DEC zp	31	1937,5	562,5	5		562,5	4	13							6							8		
INC abs	37	2312,5	687,5	6		687,5	4	13		7						5						8		
INC abs,x	40	2500,0	1000,0	7		1000,0	4	13			10					5						8		
INC zp,x	32	2000,0	1000,0	6		1000,0	4	13						7								8		
INC zp	31	1937,5	562,5	5		562,5	4	13							6							8		
RMB zp	27	1687,5	812,5	5		812,5	14	13																

[illegible]