

Instruction (condition f=#)	Total AVR cycles used	AVR ns @ 16MHz	NMOS 6502 ns @ 2MHz negative=AVR slower positive=AVR faster	NMOS 6502 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_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)	
								15	10	7	11	15	15	7	6	5	5	3	4	6	5	8	10	6	
ASL a	13	812,5	187,5	2		187,5	3	10																	
LSR a	13	812,5	187,5	2		187,5	3	10																	
ROL a	14	875,0	125,0	2		125,0	4	10																	
ROR a	14	875,0	125,0	2		125,0	4	10																	
CLC	12	750,0	250,0	2		250,0	2	10																	
CLD	12	750,0	250,0	2		250,0	2	10																	
CLI **-/ +6	14	875,0	125,0	2		125,0	4	10																	
CLV	12	750,0	250,0	2		250,0	2	10																	
SEC	12	750,0	250,0	2		250,0	2	10																	
SED	12	750,0	250,0	2		250,0	2	10																	
SEI ** +3/-	14	875,0	125,0	2		125,0	4	10																	
DEX	14	875,0	125,0	2		125,0	4	10																	
DEY	14	875,0	125,0	2		125,0	4	10																	
INX	14	875,0	125,0	2		125,0	4	10																	
INY	14	875,0	125,0	2		125,0	4	10																	
NOP	16	1000,0	0,0	2		0,0	6	10																	
PHA	20	1250,0	250,0	3		250,0	10	10																	
PHP	26	1625,0	-125,0	3		-125,0	16	10																	
PLA	21	1312,5	687,5	4		687,5	8	10									3								
PLP ** +5/+7	28	1750,0	250,0	4		250,0	14	10											4						
TAX	15	937,5	62,5	2		62,5	2	10									3								
TAY	15	937,5	62,5	2		62,5	2	10									3								
TSX	15	937,5	62,5	2		62,5	2	10									3								
TXA	15	937,5	62,5	2		62,5	2	10									3								
TXS	12	750,0	250,0	2		250,0	2	10																	
TYA	15	937,5	62,5	2		62,5	2	10									3								

Instruction (condition f=#)	Total AVR cycles used	AVR ns @ 16MHz	NMOS 6502 ns @ 2MHz negative=AVR slower positive=AVR faster	NMOS 6502 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_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)
								15	10	7	11	15	15	7	6	5	5	3	4	6	5	8	10	6
BCC C=1	18	1125,0	-125,0	2		-125,0	3 15																	
BCC C=0	23	1437,5	62,5	3	1	562,5	8 15																	
BCS C=0	18	1125,0	-125,0	2		-125,0	3 15																	
BCS C=1	23	1437,5	62,5	3	1	562,5	8 15																	
BEQ Z=0	18	1125,0	-125,0	2		-125,0	3 15																	
BEQ Z=1	23	1437,5	62,5	3	1	562,5	8 15																	
BMI N=0	18	1125,0	-125,0	2		-125,0	3 15																	
BMI N=1	23	1437,5	62,5	3	1	562,5	8 15																	
BNE Z=1	18	1125,0	-125,0	2		-125,0	3 15																	
BNE Z=0	23	1437,5	62,5	3	1	562,5	8 15																	
BPL N=1	18	1125,0	-125,0	2		-125,0	3 15																	
BPL N=0	23	1437,5	62,5	3	1	562,5	8 15																	
BVC V=1	18	1125,0	-125,0	2		-125,0	3 15																	
BVC V=0	23	1437,5	62,5	3	1	562,5	8 15																	
BVS V=0	18	1125,0	-125,0	2		-125,0	3 15																	
BVS V=1	23	1437,5	62,5	3	1	562,5	8 15																	
JMP abs	22	1375,0	125,0	3		125,0	7 15																	
JMP (abs)	34	2125,0	375,0	5		375,0	19 15																	
JSR abs	37	2312,5	687,5	6		687,5	22 15																	
RTS	29	1812,5	1187,5	6		1187,5	14 15																	
RTI **+5/+7	37	2312,5	687,5	6		687,5	18 15												4					
BRK **+3/-	50	3125,0	375,0	7		375,0	35 15																	
IRQ **+3/-	57	3562,5	-62,5	7		-62,5	42 15																	
NMI **+3/-	58	3625,0	-125,0	7		-125,0	43 15																	
ADC abs D=0	35	2187,5	-187,5	4		-187,5	8 15			7						5								
ADC abs,x D=0	39	2437,5	-437,5	4	1	62,5	8 15				11					5								

Instruction (condition f=#)	Total AVR cycles used	AVR ns @ 16MHz	NMOS 6502 ns @ 2MHz negative=AVR slower positive=AVR faster	NMOS 6502 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_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)
								15	10	7	11	15	15	7	6	5	5	3	4	6	5	8	10	6
ADC abs,y D=0	39	2437,5	-437,5	4	1	62,5	8 15			11					5									
ADC (zp,x) D=0	43	2687,5	312,5	6		312,5	8 15				15				5									
ADC (zp),y D=0	43	2687,5	-187,5	5	1	312,5	8 15					15			5									
ADC zp,x D=0	30	1875,0	125,0	4		125,0	8 15						7											
ADC zp D=0	29	1812,5	-312,5	3		-312,5	8 15							6										
ADC # D=0	23	1437,5	-437,5	2		-437,5	8 15																	
ADC abs D=1	42	2625,0	-625,0	4		-625,0	15 15			7					5									
ADC abs,x D=1	46	2875,0	-875,0	4	1	-375,0	15 15				11				5									
ADC abs,y D=1	46	2875,0	-875,0	4	1	-375,0	15 15				11				5									
ADC (zp,x) D=1	50	3125,0	-125,0	6		-125,0	15 15					15			5									
ADC (zp),y D=1	50	3125,0	-625,0	5	1	-125,0	15 15						15		5									
ADC zp,x D=1	37	2312,5	-312,5	4		-312,5	15 15							7										
ADC zp D=1	36	2250,0	-750,0	3		-750,0	15 15								6									
ADC # D=1	30	1875,0	-875,0	2		-875,0	15 15																	
SBC abs D=0	37	2312,5	-312,5	4		-312,5	10 15			7					5									
SBC abs,x D=0	41	2562,5	-562,5	4	1	-62,5	10 15				11				5									
SBC abs,y D=0	41	2562,5	-562,5	4	1	-62,5	10 15				11				5									
SBC (zp,x) D=0	45	2812,5	187,5	6		187,5	10 15					15			5									
SBC (zp),y D=0	45	2812,5	-312,5	5	1	187,5	10 15						15		5									
SBC zp,x D=0	32	2000,0	0,0	4		0,0	10 15							7										
SBC zp D=0	31	1937,5	-437,5	3		-437,5	10 15								6									
SBC # D=0	25	1562,5	-562,5	2		-562,5	10 15																	
SBC abs D=1	41	2562,5	-562,5	4		-562,5	14 15			7					5									
SBC abs,x D=1	45	2812,5	-812,5	4	1	-312,5	14 15				11				5									
SBC abs,y D=1	45	2812,5	-812,5	4	1	-312,5	14 15				11				5									
SBC (zp,x) D=1	49	3062,5	-62,5	6		-62,5	14 15					15			5									

Instruction (condition f=#)	Total AVR cycles used	AVR ns @ 16MHz	NMOS 6502 ns @ 2MHz negative=AVR slower positive=AVR faster	NMOS 6502 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_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)
								15	10	7	11	15	15	7	6	5	5	3	4	6	5	8	10	6
SBC (zp),y D=1	49	3062,5	-562,5	5	1	-62,5	14 15						15			5								
SBC zp,x D=1	36	2250,0	-250,0	4		-250,0	14 15							7										
SBC zp D=1	35	2187,5	-687,5	3		-687,5	14 15								6									
SBC # D=1	29	1812,5	-812,5	2		-812,5	14 15																	
AND abs	31	1937,5	62,5	4		62,5	4 15			7						5								
AND abs,x	35	2187,5	-187,5	4	1	312,5	4 15				11					5								
AND abs,y	35	2187,5	-187,5	4	1	312,5	4 15				11					5								
AND (zp,x)	39	2437,5	562,5	6		562,5	4 15					15				5								
AND (zp),y	39	2437,5	62,5	5	1	562,5	4 15						15			5								
AND zp,x	26	1625,0	375,0	4		375,0	4 15							7										
AND zp	25	1562,5	-62,5	3		-62,5	4 15								6									
AND #	19	1187,5	-187,5	2		-187,5	4 15																	
EOR abs	31	1937,5	62,5	4		62,5	4 15			7						5								
EOR abs,x	35	2187,5	-187,5	4	1	312,5	4 15				11					5								
EOR abs,y	35	2187,5	-187,5	4	1	312,5	4 15				11					5								
EOR (zp,x)	39	2437,5	562,5	6		562,5	4 15					15				5								
EOR (zp),y	39	2437,5	62,5	5	1	562,5	4 15						15			5								
EOR zp,x	26	1625,0	375,0	4		375,0	4 15							7										
EOR zp	25	1562,5	-62,5	3		-62,5	4 15								6									
EOR #	19	1187,5	-187,5	2		-187,5	4 15																	
ORA abs	31	1937,5	62,5	4		62,5	4 15			7						5								
ORA abs,x	35	2187,5	-187,5	4	1	312,5	4 15				11					5								
ORA abs,y	35	2187,5	-187,5	4	1	312,5	4 15				11					5								
ORA (zp,x)	39	2437,5	562,5	6		562,5	4 15					15				5								
ORA (zp),y	39	2437,5	62,5	5	1	562,5	4 15						15			5								
ORA zp,x	26	1625,0	375,0	4		375,0	4 15							7										

Instruction (condition f=#)	Total AVR cycles used	AVR ns @ 16MHz	NMOS 6502 ns @ 2MHz negative=AVR slower positive=AVR faster	NMOS 6502 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_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)	
								15	10	7	11	15	15	7	6	5	5	3	4	6	5	8	10	6	
ORA zp	25	1562,5	-62,5	3		-62,5	4 15								6										
ORA #	19	1187,5	-187,5	2		-187,5	4 15																		
CMP abs	31	1937,5	62,5	4		62,5	4 15		7							5									
CMP abs,x	35	2187,5	-187,5	4	1	312,5	4 15			11						5									
CMP abs,y	35	2187,5	-187,5	4	1	312,5	4 15			11						5									
CMP (zp,x)	39	2437,5	562,5	6		562,5	4 15				15					5									
CMP (zp),y	39	2437,5	62,5	5	1	562,5	4 15					15				5									
CMP zp,x	26	1625,0	375,0	4		375,0	4 15						7												
CMP zp	25	1562,5	-62,5	3		-62,5	4 15								6										
CMP #	19	1187,5	-187,5	2		-187,5	4 15																		
BIT abs	36	2250,0	-250,0	4		-250,0	9 15		7							5									
BIT zp	30	1875,0	-375,0	3		-375,0	9 15								6										
CPX abs	31	1937,5	62,5	4		62,5	4 15		7							5									
CPX zp	25	1562,5	-62,5	3		-62,5	4 15								6										
CPX #	19	1187,5	-187,5	2		-187,5	4 15																		
CPY abs	31	1937,5	62,5	4		62,5	4 15		7							5									
CPY zp	25	1562,5	-62,5	3		-62,5	4 15								6										
CPY #	19	1187,5	-187,5	2		-187,5	4 15																		
LDA abs	31	1937,5	62,5	4		62,5	1 15		7							5		3							
LDA abs,x	35	2187,5	-187,5	4	1	312,5	1 15			11						5		3							
LDA abs,y	35	2187,5	-187,5	4	1	312,5	1 15			11						5		3							
LDA (zp,x)	39	2437,5	562,5	6		562,5	1 15				15					5		3							
LDA (zp),y	39	2437,5	62,5	5	1	562,5	1 15					15				5		3							
LDA zp,x	26	1625,0	375,0	4		375,0	1 15						7					3							
LDA zp	25	1562,5	-62,5	3		-62,5	1 15								6			3							
LDA #	19	1187,5	-187,5	2		-187,5	1 15											3							

Instruction (condition f=#)	Total AVR cycles used	AVR ns @ 16MHz	NMOS 6502 ns @ 2MHz negative=AVR slower positive=AVR faster	NMOS 6502 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_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)
								15	10	7	11	15	15	7	6	5	5	3	4	6	5	8	10	6
LDX abs	31	1937,5	62,5	4		62,5	1 15			7						5		3						
LDX abs,y	35	2187,5	-187,5	4	1	312,5	1 15				11					5		3						
LDX zp,y	26	1625,0	375,0	4		375,0	1 15							7				3						
LDX zp	25	1562,5	-62,5	3		-62,5	1 15								6			3						
LDX #	19	1187,5	-187,5	2		-187,5	1 15											3						
LDY abs	31	1937,5	62,5	4		62,5	1 15			7						5		3						
LDY abs,x	35	2187,5	-187,5	4	1	312,5	1 15				11					5		3						
LDY zp,x	26	1625,0	375,0	4		375,0	1 15							7				3						
LDY zp	25	1562,5	-62,5	3		-62,5	1 15								6			3						
LDY #	19	1187,5	-187,5	2		-187,5	1 15											3						
STA abs	32	2000,0	0,0	4		0,0	0 15			7														10
STA abs,x	36	2250,0	250,0	5		250,0	0 15				11													10
STA abs,y	36	2250,0	250,0	5		250,0	0 15				11													10
STA (zp,x)	40	2500,0	500,0	6		500,0	0 15					15												10
STA (zp),y	40	2500,0	500,0	6		500,0	0 15						15											10
STA zp,x	27	1687,5	312,5	4		312,5	0 15													6				6
STA zp	26	1625,0	-125,0	3		-125,0	0 15														5			6
STX abs	32	2000,0	0,0	4		0,0	0 15			7														10
STX zp,y	27	1687,5	312,5	4		312,5	0 15													6				6
STX zp	26	1625,0	-125,0	3		-125,0	0 15														5			6
STY abs	32	2000,0	0,0	4		0,0	0 15			7														10
STY zp,x	27	1687,5	312,5	4		312,5	0 15													6				6
STY zp	26	1625,0	-125,0	3		-125,0	0 15														5			6
ASL abs	38	2375,0	625,0	6		625,0	3 15			7						5							8	
ASL abs,x	42	2625,0	875,0	7		875,0	3 15				11					5							8	
ASL zp,x	33	2062,5	937,5	6		937,5	3 15							7									8	

Instruction (condition f=#)	Total AVR cycles used	AVR ns @ 16MHz	NMOS 6502 ns @ 2MHz negative=AVR slower positive=AVR faster	NMOS 6502 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_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)
								15	10	7	11	15	15	7	6	5	5	3	4	6	5	8	10	6
ASL zp	32	2000,0	500,0	5		500,0	3 15								6							8		
LSR abs	38	2375,0	625,0	6		625,0	3 15		7							5						8		
LSR abs,x	42	2625,0	875,0	7		875,0	3 15			11						5						8		
LSR zp,x	33	2062,5	937,5	6		937,5	3 15							7								8		
LSR zp	32	2000,0	500,0	5		500,0	3 15								6							8		
ROL abs	39	2437,5	562,5	6		562,5	4 15		7							5						8		
ROL abs,x	43	2687,5	812,5	7		812,5	4 15			11						5						8		
ROL zp,x	34	2125,0	875,0	6		875,0	4 15							7								8		
ROL zp	33	2062,5	437,5	5		437,5	4 15								6							8		
ROR abs	39	2437,5	562,5	6		562,5	4 15		7							5						8		
ROR abs,x	43	2687,5	812,5	7		812,5	4 15			11						5						8		
ROR zp,x	34	2125,0	875,0	6		875,0	4 15							7								8		
ROR zp	33	2062,5	437,5	5		437,5	4 15								6							8		
DEC abs	39	2437,5	562,5	6		562,5	4 15		7							5						8		
DEC abs,x	43	2687,5	812,5	7		812,5	4 15			11						5						8		
DEC zp,x	34	2125,0	875,0	6		875,0	4 15							7								8		
DEC zp	33	2062,5	437,5	5		437,5	4 15								6							8		
INC abs	39	2437,5	562,5	6		562,5	4 15		7							5						8		
INC abs,x	43	2687,5	812,5	7		812,5	4 15			11						5						8		
INC zp,x	34	2125,0	875,0	6		875,0	4 15							7								8		
INC zp	33	2062,5	437,5	5		437,5	4 15								6							8		
average			74,2			178,7																*	*	
** add AVR cycles if IRQ_DIS_REAL is defined , I=1/I=0 (force atomic instruction timing if interrupts disabled)							* add 2 cycles if ROMMAP is defined (write protected RAM)																	