	Instruction (condition f=#)	Total AVR cyles 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)
									13	9	7	10	15	15	7	6	5	5	3	4	6	5	8	10	6
ASL a		12	750,0	250,0	2		250,0	3		9															
LSR a		12	750,0	250,0	2		250,0	3		9															
ROL a		13	812,5	187,5	2		187,5	4		9															
ROR a		13	812,5	187,5	2		187,5	4		9															
CLC		11	687,5	312,5	2		312,5	2		9															
CLD		11	687,5	312,5	2		312,5	2		9															
CLI		13	812,5	187,5	2		187,5	4		9															
CLV		11	687,5	312,5	2		312,5	2		9															
SEC		11	687,5	312,5	2		312,5	2		9															
SED		11	687,5	312,5	2		312,5	2		9															
SEI		13	812,5	187,5	2		187,5	4		9															
DEX		13	812,5	187,5	2		187,5	4		9															
DEY		13	812,5	187,5	2		187,5	4		9															
INX		13	812,5	187,5	2		187,5	4		9															
INY		13	812,5	187,5	2		187,5	4		9															
NOP		16	1000,0	0,0	2		0,0	7		9															
PHA		19	1187,5	312,5	3		312,5	10		9															
PHP		25	1562,5	-62,5	3		-62,5	16		9															
PLA		20	1250,0	750,0	4		750,0	8		9									3						
PLP		27	1687,5	312,5	4		312,5	14		9										4					
TAX		14	875,0		2		125,0	2		9									3						
TAY		14	875,0	125,0	2		125,0	2		9									3						
TSX		14	875,0	125,0	2		125,0	2		9									3						
TXA		14	875,0	125,0	2		125,0	2		9									3						
TXS		11	687,5	312,5	2		312,5	2		9															
TYA		14	875,0	125,0	2		125,0	2		9	-				-				3						

(condition f=#)	Instruction	Total AVR cyles 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)
									13	9	7	10	15	15	7	6	5	5	3	4	6	5	8	10	6
BCC C=1		16	1000,0	0,0	2		0,0	3	13																
BCC C=0		21	1312,5	187,5	3		687,5	8	13																
BCS C=0		16	1000,0	0,0	2		0,0	3	13																
BCS C=1		21	1312,5	187,5	3		687,5	8	13																
BEQ Z=0		16	1000,0	0,0	2		0,0	3	13																
BEQ Z=1		21	1312,5	187,5	3		687,5	8	13																
BMI N=0		16	1000,0	0,0	2		0,0	3	13																
BMI N=1		21	1312,5	187,5	3	1	687,5	8	13																
BNE Z=1		16	1000,0	0,0	2		0,0	3	13																
BNE Z=0		21	1312,5	187,5	3		687,5	8	13																
BPL N=1		16	1000,0	0,0	2		0,0	3	13																
BPL N=0		21	1312,5	187,5	3	1	687,5	8	13																
BVC V=1		16	1000,0	0,0	2		0,0	3	13																
BVC V=0		21	1312,5	187,5	3	1	687,5	8	13																
BVS V=0		16	1000,0	0,0	2		0,0	3	13																
BVS V=1		21	1312,5	187,5	3	1	687,5	8	13																
JMP abs		20	1250,0	250,0	3		250,0	7	13																
JMP (abs)		32	2000,0	500,0	5		500,0	19	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		35	2187,5	812,5	6		812,5	18	13											4					
BRK		48	3000,0	500,0	7		500,0	35	13																
IRQ		54	3375,0	125,0	7		125,0	41	13																
NMI		55	3437,5	62,5	7		62,5	42	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								

Instruction (condition f=#)	Total AVR cyles 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)
								13	9	7			15	7	6	5	5	3	4	6	5	8	10	6
ADC abs,y D=0	36	2250,0	-250,0	4	1	250,0	8	13			10					5								
ADC (zp,x) D=0	41	2562,5	437,5	6		437,5	8	13				15				5							<u> </u>	
ADC (zp),y D=0	41	2562,5	-62,5	5		437,5	8	13					15			5							<u> </u>	
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	40	2500,0	-500,0	4		-500,0	15	13		7						5								
ADC abs,x D=1	43	2687,5	-687,5	4	1	-187,5	15	13			10					5								
ADC abs,y D=1	43	2687,5	-687,5	4	1	-187,5	15	13			10					5								
ADC (zp,x) D=1	48	3000,0	0,0	6		0,0	15	13				15				5								
ADC (zp),y D=1	48	3000,0	-500,0	5	1	0,0	15	13					15			5								
ADC zp,x D=1	35	2187,5	-187,5	4		-187,5	15	13						7										
ADC zp D=1	34	2125,0	-625,0	3		-625,0	15	13							6									
ADC # D=1	28	1750,0	-750,0	2		-750,0	15	13																
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,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	39	2437,5	-437,5	4		-437,5	14	13		7						5								
SBC abs,x D=1	42	2625,0	-625,0	4	1	-125,0	14	13			10					5								
SBC abs,y D=1	42	2625,0	-625,0	4	1	-125,0	14	13			10					5								
SBC (zp,x) D=1	47	2937,5	62,5	6		62,5	14	13				15				5								

Instruction (condition f=#)	Total AVR cyles 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)
								13	9	7	10	15	15	7	6	5	5	3	4	6	5	8	10	6
SBC (zp),y D=1	47	2937,5	-437,5	5		62,5	14	13					15			5								
SBC zp,x D=1	34	2125,0	-125,0	4		-125,0	14	13						7										
SBC zp D=1	33	2062,5	-562,5	3		-562,5	14	13							6									
SBC # D=1	27	1687,5	-687,5	2		-687,5	14	13																
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,x)	37	2312,5	687,5	6		687,5	4	13				15				5								
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,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,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										

Instruction (condition f=#)	Total AVR cyles 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)
								13	9	7	10	15	15	7	6	5	5	3	4	6	5	8	10	6
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,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 zp	28	1750,0	-250,0	3		-250,0	9	13							6									
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,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						

Instruction (condition f=#)	Total AVR cyles 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)
								13	9	7		15	15	7	6		5	3	4	6	5	8	10	6
LDX abs	29	1812,5	187,5	4		187,5	1	13		7						5		3						
LDX abs,y	32	2000,0	0,0	4	1	500,0	1	13			10					5		3						
LDX zp,y	24	1500,0	500,0	4		500,0	1	13						7				3						
LDX zp	23	1437,5	62,5	3		62,5	1	13							6			3						
LDX #	17	1062,5	-62,5	2		-62,5	1	13										3						
LDY abs	29	1812,5	187,5	4		187,5	1	13		7						5		3						
LDY abs,x	32	2000,0	0,0	4	1	500,0	1	13			10					5		3						
LDY zp,x	24	1500,0	500,0	4		500,0	1	13						7				3						
LDY zp	23	1437,5	62,5	3		62,5	1	13							6			3						
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,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
ASL abs	36	2250,0	750,0	6		750,0	3	13		7						5						8		
ASL abs,x	39	2437,5	1062,5	7		1062,5	3	13			10					5						8		
ASL zp,x	31	1937,5	1062,5	6		1062,5	3	13						7								8		

Instruction (condition f=#)	Total AVR cyles 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)
								13	9	7	10	15	15	7	6	5	5	3	4	6	5	8	10	6
ASL zp	30	1875,0		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	1062,5	7		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	1000,0	7		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	1000,0	7		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		
average			200,6			305,1		**	**													*	*	
					-	cle if IRQ_DIS	_				disab	led)				add write	•				vP is (	defin	ed	