The 6502 Instruction Set

Load and Store Group

LDA	Load Accumulator	N,Z
LDX	Load X Register	N,Z
LDY	Load Y Register	N,Z
STA	Store Accumulator	
STX	Store X Register	
STY	Store Y Register	

Arithmetic Group

ADC	Add with Carry	N,V,Z,C
SBC	Subtract with Carry	N,V,Z,C

Increment and Decrement Group

INC	Increment a memory location	N,Z
INX	Increment the X register	N,Z
INY	Increment the Y register	N,Z
DEC	Decrement a memory location	N,Z
DEX	Decrement the X register	N,Z
DEY	Decrement the Y register	N,Z

Register Transfer Group

TAX	Transfer accumulator to X	N,Z
TAY	Transfer accumulator to Y	N,Z
TXA	Transfer X to accumulator	N,Z
TYA	Transfer Y to accumulator	N,Z

Logical Group

AND	Logical AND	N,Z
EOR	Exclusive OR	N,Z
ORA	Logical Inclusive OR	N,Z

Compare and Bit Test Group

CMP	Compare accumulator	N,Z,C
CPX	Compare X register	N,Z,C
CPY	Compare Y register	N,Z,C
BIT	Bit Test	N,V,Z

Shift and Rotate Group

ASL	Arithmetic Shift Left	N,Z,C
LSR	Logical Shift Right	N,Z,C
ROL	Rotate Left	N,Z,C
ROR	Rotate Right	N,Z,C

Jump and Branch Group

JMP	Jump to another location	
BCC	Branch if carry flag clear	
BCS	Branch if carry flag set	
BEQ	Branch if zero flag set	
BMI	Branch if negative flag set	
BNE	Branch if zero flag clear	
BPL	Branch if negative flag clear	
BVC	Branch if overflow flag clear	
BVS	Branch if overflow flag set	

Stack Group

TSX	Transfer stack pointer to X	N,Z
TXS	Transfer X to stack pointer	
PHA	Push accumulator on stack	
PHP	Push processor status on stack	
PLA	Pull accumulator from stack	N,Z
PLP	Pull processor status from stack	All

Status Flag Change Group

CLC	Clear carry flag	C
CLD	Clear decimal mode flag	D
CLI	Clear interrupt disable flag	I
CLV	Clear overflow flag	V
SEC	Set carry flag	C
SED	Set decimal mode flag	D
SEI	Set interrupt disable flag	I

Subroutine and Interrupt Group

JSR	Jump to a subroutine	
RTS	Return from subroutine	
BRK	Force an interrupt	B
RTI	Return from Interrupt	All
NOP	No Operation	

Load and Store Group

LDA	Load Accumulator	Absolute	LDA $aaaa	$AD	3	N,Z
		Zero Page	LDA $aa	$A5	2	
		Immediate	LDA #$aa	$A9	2	
		Absolute Indexed, X	LDA $aaaa,X	$BD	3	
		Absolute Indexed, Y	LDA $aaaa,Y	$B9	3	
		Zero Page Indexed, X	LDA $aa,X	$B5	2	
		Indexed Indirect	LDA ($aa,X)	$A1	2	
		Indirect Indexed	LDA ($aa),Y	$B1	2	
LDX	Load X Register	Absolute	LDX $aaaa	$AE	3	N,Z
		Zero Page	LDX $aa	$A6	2	
		Immediate	LDX #$aa	$A2	2	
		Absolute Indexed, Y	LDX $aaaa,Y	$BE	3	
		Zero Page Indexed, Y	LDX $aa,Y	$B6	2	
LDY	Load Y Register	Absolute	LDY $aaaa	$AC	3	N,Z
		Zero Page	LDY $aa	$A4	2	
		Immediate	LDY #$aa	$A0	2	
		Absolute Indexed, X	LDY $aaaa,X	$BC	3	
		Zero Page Indexed, X	LDY $aa,X	$B4	2	
STA	Store Accumulator	Absolute	STA $aaaa	$8D	3	none
		Zero Page	STA $aa	$85	2	
		Absolute Indexed, X	STA $aaaa,X	$9D	3	
		Absolute Indexed, Y	STA $aaaa,Y	$99	3	
		Zero Page Indexed, X	STA $aa,X	$95	2	
		Indexed Indirect	STA ($aa,X)	$81	2	
		Indirect Indexed	STA ($aa),Y	$91	2	
STX	Store X Register	Absolute	STX $aaaa	$8E	3	none
		Zero Page	STX $aa	$86	2	
		Zero Page Indexed, Y	STX $aa,Y	$96	2	
STY	Store Y Register	Absolute	STY $aaaa	$8C	3	none
		Zero Page	STY $aa	$84	2	
		Zero Page Indexed, X	STY $aa,X	$94	2	

Arithmetic Group

ADC	Add with Carry	Absolute	ADC $aaaa	$6D	3	N,V,Z,C
		Zero Page	ADC $aa	$65	2	
		Immediate	ADC #$aa	$69	2	
		Absolute Indexed, X	ADC $aaaa,X	$7D	3	
		Absolute Indexed, Y	ADC $aaaa,Y	$79	3	
		Zero Page Indexed, X	ADC $aa,X	$75	2	
		Indexed Indirect	ADC ($aa,X)	$61	2	
		Indirect Indexed	ADC ($aa),Y	$71	2	
SBC	Subtract with Carry	Absolute	SBC $aaaa	$ED	3	N,V,Z,C
		Zero Page	SBC $aa	$E5	2	
		Immediate	SBC #$aa	$E9	2	
		Absolute Indexed, X	SBC $aaaa,X	$FD	3	
		Absolute Indexed, Y	SBC $aaaa,Y	$F9	3	
		Zero Page Indexed, X	SBC $aa,X	$F5	2	
		Indexed Indirect	SBC ($aa,X)	$E1	2	
		Indirect Indexed	SBC ($aa),Y	$F1	2	

Increment and Decrement Group

INC	Increment a memory location	Absolute	INC $aaaa	$EE	3	N,Z
		Zero Page	INC $aa	$E6	2	
		Absolute Indexed, X	INC $aaaa,X	$FE	3	
		Zero Page Indexed, X	INC $aa,X	$F6	2	
INX	Increment the X register	Implied	INX	$E8	1	N,Z
INY	Increment the Y register	Implied	INY	$C8	1	N,Z
DEC	Decrement a memory location	Absolute	DEC $aaaa	$CE	3	N,Z
		Zero Page	DEC $aa	$C6	2	
		Absolute Indexed, X	DEC $aaaa,X	$DE	3	
		Zero Page Indexed, X	DEC $aa,X	$D6	2	
DEX	Decrement the X register	Implied	DEX	$CA	1	N,Z
DEY	Decrement the Y register	Implied	DEY	$88	1	N,Z

Register Transfer Group

TAX	Transfer accumulator to X	Implied	TAX	$AA	1	N,Z
TAY	Transfer accumulator to Y	Implied	TAY	$A8	1	N,Z
TXA	Transfer X to accumulator	Implied	TXA	$8A	1	N,Z
TYA	Transfer Y to accumulator	Implied	TYA	$98	1	N,Z

Logical Group

AND	Logical AND	Absolute	AND $aaaa	$2D	3	N,Z
		Zero Page	AND $aa	$25	2	
		Immediate	AND #$aa	$29	2	
		Absolute Indexed, X	AND $aaaa,X	$3D	3	
		Absolute Indexed, Y	AND $aaaa,Y	$39	3	
		Zero Page Indexed, X	AND $aa,X	$35	2	
		Indexed Indirect	AND ($aa,X)	$21	2	
		Indirect Indexed	AND ($aa),Y	$31	2	
EOR	Exclusive OR	Absolute	EOR $aaaa	$4D	3	N,Z
		Zero Page	EOR $aa	$45	2	
		Immediate	EOR #$aa	$49	2	
		Absolute Indexed, X	EOR $aaaa,X	$5D	3	
		Absolute Indexed, Y	EOR $aaaa,Y	$59	3	
		Zero Page Indexed, X	EOR $aa,X	$55	2	
		Indexed Indirect	EOR ($aa,X)	$41	2	
		Indirect Indexed	EOR ($aa),Y	$51	2	

ORA	Logical Inclusive OR	Absolute	ORA $aaaa	$0D	3	N,Z
		Zero Page	ORA $aa	$05	2	
		Immediate	ORA #$aa	$09	2	
		Absolute Indexed, X	ORA $aaaa,X	$1D	3	
		Absolute Indexed, Y	ORA $aaaa,Y	$19	3	
		Zero Page Indexed, X	ORA $aa,X	$15	2	
		Indexed Indirect	ORA ($aa,X)	$01	2	
		Indirect Indexed	ORA ($aa),Y	$11	2	

Compare and Bit Test Group

CMP	Compare accumulator	Absolute	CMP $aaaa	$CD	3	N,Z,C
		Zero Page	CMP $aa	$C5	2	
		Immediate	CMP #$aa	$C9	2	
		Absolute Indexed, X	CMP $aaaa,X	$DD	3	
		Absolute Indexed, Y	CMP $aaaa,Y	$D9	3	
		Zero Page Indexed, X	CMP $aa,X	$D5	2	
		Indexed Indirect	CMP ($aa,X)	$C1	2	
		Indirect Indexed	CMP ($aa),Y	$D1	2	
CPX	Compare X register	Absolute	CPX $aaaa	$EC	3	N,Z,C
		Zero Page	CPX $aa	$E4	2	
		Immediate	CPX #$aa	$E0	2	
CPY	Compare Y register	Absolute	CPY $aaaa	$CC	3	N,Z,C
		Zero Page	CPY $aa	$C4	2	
		Immediate	CPY #$aa	$C0	2	
BIT	Bit Test	Absolute	BIT $aaaa	$2C	3	N,V,Z
		Zero Page	BIT $aa	$24	2	

Shift and Rotate Group

ASL	Arithmetic Shift Left	Absolute	ASL $aaaa	$0E	3	N,Z,C
		Zero Page	ASL $aa	$06	2	
		Accumulator	ASL A	$0A	1	
		Absolute Indexed, X	ASL $aaaa,X	$1E	3	
		Zero Page Indexed, X	ASL $aa,X	$16	2	
LSR	Logical Shift Right	Absolute	LSR $aaaa	$4E	3	N,Z,C
		Zero Page	LSR $aa	$46	2	
		Accumulator	LSR A	$4A	1	
		Absolute Indexed, X	LSR $aaaa,X	$5E	3	
		Zero Page Indexed, X	LSR $aa,X	$56	2	

ROL	Rotate Left	Absolute	ROL $aaaa	$2E	3	N,Z,C
		Zero Page	ROL $aa	$26	2	
		Accumulator	ROL A	$2A	1	
		Absolute Indexed, X	ROL $aaaa,X	$3E	3	
		Zero Page Indexed, X	ROL $aa,X	$36	2	
ROR	Rotate Right	Absolute	ROR $aaaa	$6E	3	N,Z,C
		Zero Page	ROR $aa	$66	2	
		Accumulator	ROR A	$6A	1	
		Absolute Indexed, X	ROR $aaaa,X	$7E	3	
		Zero Page Indexed, X	ROR $aa,X	$76	2	

Jump and Branch Group

JMP	Jump to another location	Absolute	JMP $aaaa	$4C	3	none
		Indirect	JMP ($aaaa)	$6C	3	
BCC	Branch if carry flag clear	Relative	BCC aa	$90	2	none
BCS	Branch if carry flag set	Relative	BCS aa	$B0	2	none
BEQ	Branch if zero flag set	Relative	BEQ aa	$F0	2	none
BMI	Branch if negative flag set	Relative	BMI aa	$30	2	none
BNE	Branch if zero flag clear	Relative	BNE aa	$D0	2	none
BPL	Branch if negative flag clear	Relative	BPL aa	$10	2	none
BVC	Branch if overflow flag clear	Relative	BVC aa	$50	2	none
BVS	Branch if overflow flag set	Relative	BVS aa	$70	2	none

Stack Group

TSX	Transfer stack pointer to X	Implied	TSX	$BA	1	N,Z
TXS	Transfer X to stack pointer	Implied	TXS	$9A	1	none
PHA	Push accumulator on stack	Implied	PHA	$48	1	none
PHP	Push processor status on stack	Implied	PHP	$08	1	none
PLA	Pull accumulator from stack	Implied	PLA	$68	1	N,Z
PLP	Pull processor status from stack	Implied	PLP	$28	1	All

Status Flag Change Group

CLC	Clear carry flag	Implied	CLC	$18	1	C
CLD	Clear decimal mode flag	Implied	CLD	$D8	1	D
CLI	Clear interrupt disable flag	Implied	CLI	$58	1	I
CLV	Clear overflow flag	Implied	CLV	$B8	1	V
SEC	Set carry flag	Implied	SEC	$38	1	C
SED	Set decimal mode flag	Implied	SED	$F8	1	D
SEI	Set interrupt disable flag	Implied	SEI	$78	1	I

Subroutine and Interrupt Group

JSR	Jump to a subroutine	Absolute	JSR $aaaa	$20	3	none
RTS	Return from subroutine	Implied	RTS	$60	1	none
BRK	Force an interrupt	Implied	BRK	$00	1	B, I
RTI	Return from Interrupt	Implied	RTI	$40	1	All
NOP	No Operation	Implied	NOP	$EA	1	none