The Instruction Set

The 6502 has a relatively basic set of instructions, many having similar functions (e.g. memory access, arithmetic, etc.). The following sections list the complete set of 56 instructions in functional groups.

Load/Store Operations

These instructions transfer a single byte between memory and one of the registers. Load operations set the negative (N) and zero (Z) flags depending on the value of transferred. Store operations do not affect the flag settings.

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	

Register Transfers

The contents of the X and Y registers can be moved to or from the accumulator, setting the negative (N) and zero (Z) flags as appropriate.

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

Stack Operations

The 6502 microprocessor supports a 256 byte stack fixed between memory locations $0100 and $01FF. A special 8-bit register, S, is used to keep track of the next free byte of stack space. Pushing a byte on to the stack causes the value to be stored at the current free location (e.g. $0100,S) and then the stack pointer is post decremented. Pull operations reverse this procedure.

The stack register can only be accessed by transferring its value to or from the X register. Its value is automatically modified by push/pull instructions, subroutine calls and returns, interrupts and returns from interrupts.

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

Logical

The following instructions perform logical operations on the contents of the accumulator and another value held in memory. The BIT instruction performs a logical AND to test the presence of bits in the memory value to set the flags but does not keep the result.

AND	Logical AND	N,Z
EOR	Exclusive OR	N,Z
ORA	Logical Inclusive OR	N,Z
BIT	Bit Test	N,V,Z

Arithmetic

The arithmetic operations perform addition and subtraction on the contents of the accumulator. The compare operations allow the comparison of the accumulator and X or Y with memory values.

ADC	Add with Carry	N,V,Z,C
SBC	Subtract with Carry	N,V,Z,C
CMP	Compare accumulator	N,Z,C
CPX	Compare X register	N,Z,C

CPY	Compare Y register	N,Z,C

Increments & Decrements

Increment or decrement a memory location or one of the X or Y registers by one setting the negative (N) and zero (Z) flags as appropriate,

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

Shifts

Shift instructions cause the bits within either a memory location or the accumulator to be shifted by one bit position. The rotate instructions use the contents if the carry flag (C) to fill the vacant position generated by the shift and to catch the overflowing bit. The arithmetic and logical shifts shift in an appropriate 0 or 1 bit as appropriate but catch the overflow bit in the carry flag (C).

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

Jumps & Calls

The following instructions modify the program counter causing a break to normal sequential execution. The JSR instruction pushes the old PC onto the stack before changing it to the new location allowing a subsequent RTS to return execution to the instruction after the call.

JMP	Jump to another location	
JSR	Jump to a subroutine	

RTS	Return from subroutine	

Branches

Branch instructions break the normal sequential flow of execution by changing the program counter if a specified condition is met. All the conditions are based on examining a single bit within the processor status.

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	

Branch instructions use relative address to identify the target instruction if they are executed. As relative addresses are stored using a signed 8 bit byte the target instruction must be within 126 bytes before the branch or 128 bytes after the branch.

Status Flag Changes

The following instructions change the values of specific status flags.

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

System Functions

The remaining instructions perform useful but rarely used functions.

BRK	Force an interrupt	B
NOP	No Operation	
RTI	Return from Interrupt	All