

## 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.

<a href="#">LDA</a>	Load Accumulator	<a href="#">N,Z</a>
<a href="#">LDX</a>	Load X Register	<a href="#">N,Z</a>
<a href="#">LDY</a>	Load Y Register	<a href="#">N,Z</a>
<a href="#">STA</a>	Store Accumulator	
<a href="#">STX</a>	Store X Register	
<a href="#">STY</a>	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.

<a href="#">TAX</a>	Transfer accumulator to X	<a href="#">N,Z</a>
<a href="#">TAY</a>	Transfer accumulator to Y	<a href="#">N,Z</a>
<a href="#">TXA</a>	Transfer X to accumulator	<a href="#">N,Z</a>
<a href="#">TYA</a>	Transfer Y to accumulator	<a href="#">N,Z</a>

### 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.

<a href="#">TSX</a>	Transfer stack pointer to X	<a href="#">N,Z</a>
<a href="#">TXS</a>	Transfer X to stack pointer	
<a href="#">PHA</a>	Push accumulator on stack	
<a href="#">PHP</a>	Push processor status on stack	
<a href="#">PLA</a>	Pull accumulator from stack	<a href="#">N,Z</a>
<a href="#">PLP</a>	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.

<a href="#">AND</a>	Logical AND	<a href="#">N,Z</a>
<a href="#">EOR</a>	Exclusive OR	<a href="#">N,Z</a>
<a href="#">ORA</a>	Logical Inclusive OR	<a href="#">N,Z</a>
<a href="#">BIT</a>	Bit Test	<a href="#">N,V,Z</a>

## 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.

<a href="#">ADC</a>	Add with Carry	<a href="#">N,V,Z,C</a>
<a href="#">SBC</a>	Subtract with Carry	<a href="#">N,V,Z,C</a>
<a href="#">CMP</a>	Compare accumulator	<a href="#">N,Z,C</a>
<a href="#">CPX</a>	Compare X register	<a href="#">N,Z,C</a>
<a href="#">CPY</a>	Compare Y register	<a href="#">N,Z,C</a>

## 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,

<a href="#">INC</a>	Increment a memory location	<a href="#">N,Z</a>
<a href="#">INX</a>	Increment the X register	<a href="#">N,Z</a>
<a href="#">INY</a>	Increment the Y register	<a href="#">N,Z</a>
<a href="#">DEC</a>	Decrement a memory location	<a href="#">N,Z</a>
<a href="#">DEX</a>	Decrement the X register	<a href="#">N,Z</a>
<a href="#">DEY</a>	Decrement the Y register	<a href="#">N,Z</a>

## 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 of 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](#)).

<a href="#">ASL</a>	Arithmetic Shift Left	<a href="#">N,Z,C</a>
<a href="#">LSR</a>	Logical Shift Right	<a href="#">N,Z,C</a>
<a href="#">ROL</a>	Rotate Left	<a href="#">N,Z,C</a>
<a href="#">ROR</a>	Rotate Right	<a href="#">N,Z,C</a>

## 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.

<a href="#">JMP</a>	Jump to another location	
<a href="#">JSR</a>	Jump to a subroutine	
<a href="#">RTS</a>	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.

<a href="#">BCC</a>	Branch if carry flag clear	
<a href="#">BCS</a>	Branch if carry flag set	
<a href="#">BEQ</a>	Branch if zero flag set	
<a href="#">BMI</a>	Branch if negative flag set	
<a href="#">BNE</a>	Branch if zero flag clear	
<a href="#">BPL</a>	Branch if negative flag clear	
<a href="#">BVC</a>	Branch if overflow flag clear	
<a href="#">BVS</a>	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.

<a href="#">CLC</a>	Clear carry flag	<a href="#">C</a>
<a href="#">CLD</a>	Clear decimal mode flag	<a href="#">D</a>
<a href="#">CLI</a>	Clear interrupt disable flag	<a href="#">I</a>
<a href="#">CLV</a>	Clear overflow flag	<a href="#">V</a>
<a href="#">SEC</a>	Set carry flag	<a href="#">C</a>
<a href="#">SED</a>	Set decimal mode flag	<a href="#">D</a>
<a href="#">SEI</a>	Set interrupt disable flag	<a href="#">I</a>

## System Functions

The remaining instructions perform useful but rarely used functions.

<a href="#">BRK</a>	Force an interrupt	<a href="#">B</a>
<a href="#">NOP</a>	No Operation	
<a href="#">RTI</a>	Return from Interrupt	All

[<< Back](#)
[Home](#)
[Contents](#)
[Next >>](#)

This page was last updated on 2nd January 2002