Instruction Reference

Click on any of following links to go straight to the information for that instruction.

ADC	AND	ASL	BCC	BCS	BEQ	BIT	BMI	BNE	BPL	BRK	BVC	BVS	CLC
CLD	CLI	CLV	CMP	CPX	CPY	DEC	DEX	DEY	EOR	INC	INX	INY	JMP
JSR	LDA	LDX	LDY	LSR	NOP	ORA	PHA	PHP	PLA	PLP	ROL	ROR	RTI
RTS	SBC	SEC	SED	SEI	STA	STX	STY	TAX	TAY	TSX	TXA	TXS	TYA

ADC - Add with Carry

A,Z,C,N = A+M+C

This instruction adds the contents of a memory location to the accumulator together with the carry bit. If overflow occurs the carry bit is set, this enables multiple byte addition to be performed.

Processor Status after use:

C	Carry Flag	Set if overflow in bit 7
Z	Zero Flag	Set if A = 0
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Set if sign bit is incorrect
N	Negative Flag	Set if bit 7 set

Addressing Mode	Opcode	Bytes	Cycles
Immediate	$69	2	2
Zero Page	$65	2	3
Zero Page,X	$75	2	4
Absolute	$6D	3	4
Absolute,X	$7D	3	4 (+1 if page crossed)
Absolute,Y	$79	3	4 (+1 if page crossed)
(Indirect,X)	$61	2	6
(Indirect),Y	$71	2	5 (+1 if page crossed)

See also: SBC

AND - Logical AND

A,Z,N = A&M

A logical AND is performed, bit by bit, on the accumulator contents using the contents of a byte of memory.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Set if A = 0
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Set if bit 7 set

Addressing Mode	Opcode	Bytes	Cycles
Immediate	$29	2	2
Zero Page	$25	2	3
Zero Page,X	$35	2	4
Absolute	$2D	3	4
Absolute,X	$3D	3	4 (+1 if page crossed)
Absolute,Y	$39	3	4 (+1 if page crossed)
(Indirect,X)	$21	2	6
(Indirect),Y	$31	2	5 (+1 if page crossed)

See also: EOR, ORA

ASL - Arithmetic Shift Left

A,Z,C,N = M*2 or M,Z,C,N = M*2

This operation shifts all the bits of the accumulator or memory contents one bit left. Bit 0 is set to 0 and bit 7 is placed in the carry flag. The effect of this operation is to multiply the memory contents by 2 (ignoring 2's complement considerations), setting the carry if the result will not fit in 8 bits.

Processor Status after use:

C	Carry Flag	Set to contents of old bit 7
Z	Zero Flag	Set if A = 0
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Set if bit 7 of the result is set

Addressing Mode	Opcode	Bytes	Cycles
Accumulator	$0A	1	2
Zero Page	$06	2	5
Zero Page,X	$16	2	6
Absolute	$0E	3	6
	$1E	3	

Absolute,X			7

See also: LSR, ROL, ROR

BCC - Branch if Carry Clear

If the carry flag is clear then add the relative displacement to the program counter to cause a branch to a new location.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Not affected
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Not affected

Addressing Mode	Opcode	Bytes	Cycles
Relative	$90	2	2 (+1 if branch succeeds +2 if to a new page)

See also: BCS

BCS - Branch if Carry Set

If the carry flag is set then add the relative displacement to the program counter to cause a branch to a new location.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Not affected
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Not affected

Addressing Mode	Opcode	Bytes	Cycles
Relative	$B0	2	2 (+1 if branch succeeds +2 if to a new page)

See also: BCC

BEQ - Branch if Equal

If the zero flag is set then add the relative displacement to the program counter to cause a branch to a new location.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Not affected
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Not affected

Addressing Mode	Opcode	Bytes	Cycles
Relative	$F0	2	2 (+1 if branch succeeds +2 if to a new page)

See also: BNE

BIT - Bit Test

A & M, N = M7, V = M6

This instructions is used to test if one or more bits are set in a target memory location. The mask pattern in A is ANDed with the value in memory to set or clear the zero flag, but the result is not kept. Bits 7 and 6 of the value from memory are copied into the N and V flags.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Set if the result if the AND is zero
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Set to bit 6 of the memory value
N	Negative Flag	Set to bit 7 of the memory value

Addressing Mode	Opcode	Bytes	Cycles
Zero Page	$24	2	3
Absolute	$2C	3	4

BMI - Branch if Minus

If the negative flag is set then add the relative displacement to the program counter to cause a branch to a new location.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Not affected
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Not affected

Addressing Mode	Opcode	Bytes	Cycles
Relative	$30	2	2 (+1 if branch succeeds +2 if to a new page)

See also: BPL

BNE - Branch if Not Equal

If the zero flag is clear then add the relative displacement to the program counter to cause a branch to a new location.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Not affected
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Not affected

Addressing Mode	Opcode	Bytes	Cycles
Relative	$D0	2	2 (+1 if branch succeeds +2 if to a new page)

See also: BEQ

BPL - Branch if Positive

If the negative flag is clear then add the relative displacement to the program counter to cause a branch to a new location.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Not affected
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Not affected

Addressing Mode	Opcode	Bytes	Cycles
Relative	$10	2	2 (+1 if branch succeeds +2 if to a new page)

See also: BMI

BRK - Force Interrupt

The BRK instruction forces the generation of an interrupt request. The program counter and processor status are pushed on the stack then the IRQ interrupt vector at $FFFE/F is loaded into the PC and the break flag in the status set to one.

C	Carry Flag	Not affected
Z	Zero Flag	Not affected
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Set to 1
V	Overflow Flag	Not affected
N	Negative Flag	Not affected

Addressing Mode	Opcode	Bytes	Cycles
Implied	$00	1	7

The interpretation of a BRK depends on the operating system. On the BBC Microcomputer it is used by language ROMs to signal run time errors but it could be used for other purposes (e.g. calling operating system functions, etc.).

BVC - Branch if Overflow Clear

If the overflow flag is clear then add the relative displacement to the program counter to cause a branch to a new location.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Not affected
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Not affected

Addressing Mode	Opcode	Bytes	Cycles
Relative	$50	2	2 (+1 if branch succeeds +2 if to a new page)

See also: BVS

BVS - Branch if Overflow Set

If the overflow flag is set then add the relative displacement to the program counter to cause a branch to a new location.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Not affected
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected

B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Not affected

Addressing Mode	Opcode	Bytes	Cycles
Relative	$70	2	2 (+1 if branch succeeds +2 if to a new page)

See also: BVC

CLC - Clear Carry Flag

C = 0

Set the carry flag to zero.

C	Carry Flag	Set to 0
Z	Zero Flag	Not affected
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Not affected

Addressing Mode	Opcode	Bytes	Cycles
Implied	$18	1	2

See also: SEC

CLD - Clear Decimal Mode

D = 0

Sets the decimal mode flag to zero.

C	Carry Flag	Not affected
Z	Zero Flag	Not affected
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Set to 0
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Not affected

Addressing Mode	Opcode	Bytes	Cycles
Implied	$D8	1	2

NB:
The state of the decimal flag is uncertain when the CPU is powered up and it is not reset when an interrupt is generated. In both cases you should include an explicit CLD to ensure that the flag is cleared before performing addition or subtraction.

See also: SED

CLI - Clear Interrupt Disable

I = 0

Clears the interrupt disable flag allowing normal interrupt requests to be serviced.

C	Carry Flag	Not affected
Z	Zero Flag	Not affected
I	Interrupt Disable	Set to 0
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Not affected

Addressing Mode	Opcode	Bytes	Cycles
Implied	$58	1	2

See also: SEI

CLV - Clear Overflow Flag

V = 0

Clears the overflow flag.

C	Carry Flag	Not affected
Z	Zero Flag	Not affected
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Set to 0
N	Negative Flag	Not affected

Addressing Mode	Opcode	Bytes	Cycles
Implied	$B8	1	2

CMP - Compare

Z,C,N = A-M

This instruction compares the contents of the accumulator with another memory held value and sets the zero and carry flags as appropriate.

Processor Status after use:

C	Carry Flag	Set if A >= M
Z	Zero Flag	Set if A = M
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected

B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Set if bit 7 of the result is set

Addressing Mode	Opcode	Bytes	Cycles
Immediate	$C9	2	2
Zero Page	$C5	2	3
Zero Page,X	$D5	2	4
Absolute	$CD	3	4
Absolute,X	$DD	3	4 (+1 if page crossed)
Absolute,Y	$D9	3	4 (+1 if page crossed)
(Indirect,X)	$C1	2	6
(Indirect),Y	$D1	2	5 (+1 if page crossed)

See also: CPX, CPY

CPX - Compare X Register

Z,C,N = X-M

This instruction compares the contents of the X register with another memory held value and sets the zero and carry flags as appropriate.

Processor Status after use:

C	Carry Flag	Set if X >= M
Z	Zero Flag	Set if X = M
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Set if bit 7 of the result is set

Addressing Mode	Opcode	Bytes	Cycles
Immediate	$E0	2	2
Zero Page	$E4	2	3
Absolute	$EC	3	4

See also: CMP, CPY

CPY - Compare Y Register

Z,C,N = Y-M

This instruction compares the contents of the Y register with another memory held value and sets the zero and carry flags as appropriate.

Processor Status after use:

C	Carry Flag	Set if Y >= M

Z	Zero Flag	Set if Y = M
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Set if bit 7 of the result is set

Addressing Mode	Opcode	Bytes	Cycles
Immediate	$C0	2	2
Zero Page	$C4	2	3
Absolute	$CC	3	4

See also: CMP, CPX

DEC - Decrement Memory

M,Z,N = M-1

Subtracts one from the value held at a specified memory location setting the zero and negative flags as appropriate.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Set if result is zero
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Set if bit 7 of the result is set

Addressing Mode	Opcode	Bytes	Cycles
Zero Page	$C6	2	5
Zero Page,X	$D6	2	6
Absolute	$CE	3	6
Absolute,X	$DE	3	7

See also: DEX, DEY

DEX - Decrement X Register

X,Z,N = X-1

Subtracts one from the X register setting the zero and negative flags as appropriate.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Set if X is zero
I	Interrupt Disable	Not affected

D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Set if bit 7 of X is set

Addressing Mode	Opcode	Bytes	Cycles
Implied	$CA	1	2

See also: DEC, DEY

DEY - Decrement Y Register

Y,Z,N = Y-1

Subtracts one from the Y register setting the zero and negative flags as appropriate.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Set if Y is zero
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Set if bit 7 of Y is set

Addressing Mode	Opcode	Bytes	Cycles
Implied	$88	1	2

See also: DEC, DEX

EOR - Exclusive OR

A,Z,N = A^M

An exclusive OR is performed, bit by bit, on the accumulator contents using the contents of a byte of memory.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Set if A = 0
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Set if bit 7 set

Addressing Mode	Opcode	Bytes	Cycles
Immediate	$49	2	2
	$45	2	

Zero Page			3
Zero Page,X	$55	2	4
Absolute	$4D	3	4
Absolute,X	$5D	3	4 (+1 if page crossed)
Absolute,Y	$59	3	4 (+1 if page crossed)
(Indirect,X)	$41	2	6
(Indirect),Y	$51	2	5 (+1 if page crossed)

See also: AND, ORA

INC - Increment Memory

M,Z,N = M+1

Adds one to the value held at a specified memory location setting the zero and negative flags as appropriate.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Set if result is zero
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Set if bit 7 of the result is set

Addressing Mode	Opcode	Bytes	Cycles
Zero Page	$E6	2	5
Zero Page,X	$F6	2	6
Absolute	$EE	3	6
Absolute,X	$FE	3	7

See also: INX, INY

INX - Increment X Register

X,Z,N = X+1

Adds one to the X register setting the zero and negative flags as appropriate.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Set if X is zero
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Set if bit 7 of X is set

	Opcode	Bytes

Addressing Mode			Cycles
Implied	$E8	1	2

See also: INC, INY

INY - Increment Y Register

Y,Z,N = Y+1

Adds one to the Y register setting the zero and negative flags as appropriate.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Set if Y is zero
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Set if bit 7 of Y is set

Addressing Mode	Opcode	Bytes	Cycles
Implied	$C8	1	2

See also: INC, INX

JMP - Jump

Sets the program counter to the address specified by the operand.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Not affected
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Not affected

Addressing Mode	Opcode	Bytes	Cycles
Absolute	$4C	3	3
Indirect	$6C	3	5

NB:
An original 6502 has does not correctly fetch the target address if the indirect vector falls on a page boundary (e.g. $xxFF where xx is any value from $00 to $FF). In this case fetches the LSB from $xxFF as expected but takes the MSB from $xx00. This is fixed in some later chips like the 65SC02 so for compatibility always ensure the indirect vector is not at the end of the page.

JSR - Jump to Subroutine

The JSR instruction pushes the address (minus one) of the return point on to the stack and then sets the program counter to the target memory address.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Not affected
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Not affected

Addressing Mode	Opcode	Bytes	Cycles
Absolute	$20	3	6

See also: RTS

LDA - Load Accumulator

A,Z,N = M

Loads a byte of memory into the accumulator setting the zero and negative flags as appropriate.

C	Carry Flag	Not affected
Z	Zero Flag	Set if A = 0
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Set if bit 7 of A is set

Addressing Mode	Opcode	Bytes	Cycles
Immediate	$A9	2	2
Zero Page	$A5	2	3
Zero Page,X	$B5	2	4
Absolute	$AD	3	4
Absolute,X	$BD	3	4 (+1 if page crossed)
Absolute,Y	$B9	3	4 (+1 if page crossed)
(Indirect,X)	$A1	2	6
(Indirect),Y	$B1	2	5 (+1 if page crossed)

See also: LDX, LDY

LDX - Load X Register

X,Z,N = M

Loads a byte of memory into the X register setting the zero and negative flags as appropriate.

C	Carry Flag	Not affected
Z	Zero Flag	Set if X = 0
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Set if bit 7 of X is set

Addressing Mode	Opcode	Bytes	Cycles
Immediate	$A2	2	2
Zero Page	$A6	2	3
Zero Page,Y	$B6	2	4
Absolute	$AE	3	4
Absolute,Y	$BE	3	4 (+1 if page crossed)

See also: LDA, LDY

LDY - Load Y Register

Y,Z,N = M

Loads a byte of memory into the Y register setting the zero and negative flags as appropriate.

C	Carry Flag	Not affected
Z	Zero Flag	Set if Y = 0
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Set if bit 7 of Y is set

Addressing Mode	Opcode	Bytes	Cycles
Immediate	$A0	2	2
Zero Page	$A4	2	3
Zero Page,X	$B4	2	4
Absolute	$AC	3	4
Absolute,X	$BC	3	4 (+1 if page crossed)

See also: LDA, LDX

LSR - Logical Shift Right

A,C,Z,N = A/2 or M,C,Z,N = M/2

Each of the bits in A or M is shift one place to the right. The bit that was in bit 0 is shifted into the carry flag. Bit 7 is set to zero.

Processor Status after use:

C	Carry Flag	Set to contents of old bit 0

Z	Zero Flag	Set if result = 0
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Set if bit 7 of the result is set

Addressing Mode	Opcode	Bytes	Cycles
Accumulator	$4A	1	2
Zero Page	$46	2	5
Zero Page,X	$56	2	6
Absolute	$4E	3	6
Absolute,X	$5E	3	7

See also: ASL, ROL, ROR

NOP - No Operation

The NOP instruction causes no changes to the processor other than the normal incrementing of the program counter to the next instruction.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Not affected
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Not affected

Addressing Mode	Opcode	Bytes	Cycles
Implied	$EA	1	2

ORA - Logical Inclusive OR

A,Z,N = A|M

An inclusive OR is performed, bit by bit, on the accumulator contents using the contents of a byte of memory.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Set if A = 0
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Set if bit 7 set

Addressing Mode	Opcode	Bytes	Cycles
Immediate	$09	2	2
Zero Page	$05	2	3
Zero Page,X	$15	2	4
Absolute	$0D	3	4
Absolute,X	$1D	3	4 (+1 if page crossed)
Absolute,Y	$19	3	4 (+1 if page crossed)
(Indirect,X)	$01	2	6
(Indirect),Y	$11	2	5 (+1 if page crossed)

See also: AND, EOR

PHA - Push Accumulator

Pushes a copy of the accumulator on to the stack.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Not affected
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Not affected

Addressing Mode	Opcode	Bytes	Cycles
Implied	$48	1	3

See also: PLA

PHP - Push Processor Status

Pushes a copy of the status flags on to the stack.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Not affected
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Not affected

Addressing Mode	Opcode	Bytes	Cycles
Implied	$08	1	3

See also: PLP

PLA - Pull Accumulator

Pulls an 8 bit value from the stack and into the accumulator. The zero and negative flags are set as appropriate.

C	Carry Flag	Not affected
Z	Zero Flag	Set if A = 0
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Set if bit 7 of A is set

Addressing Mode	Opcode	Bytes	Cycles
Implied	$68	1	4

See also: PHA

PLP - Pull Processor Status

Pulls an 8 bit value from the stack and into the processor flags. The flags will take on new states as determined by the value pulled.

Processor Status after use:

C	Carry Flag	Set from stack
Z	Zero Flag	Set from stack
I	Interrupt Disable	Set from stack
D	Decimal Mode Flag	Set from stack
B	Break Command	Set from stack
V	Overflow Flag	Set from stack
N	Negative Flag	Set from stack

Addressing Mode	Opcode	Bytes	Cycles
Implied	$28	1	4

See also: PHP

ROL - Rotate Left

Move each of the bits in either A or M one place to the left. Bit 0 is filled with the current value of the carry flag whilst the old bit 7 becomes the new carry flag value.

Processor Status after use:

C	Carry Flag	Set to contents of old bit 7
Z	Zero Flag	Set if A = 0
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected

B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Set if bit 7 of the result is set

Addressing Mode	Opcode	Bytes	Cycles
Accumulator	$2A	1	2
Zero Page	$26	2	5
Zero Page,X	$36	2	6
Absolute	$2E	3	6
Absolute,X	$3E	3	7

See also: ASL, LSR, ROR

ROR - Rotate Right

Move each of the bits in either A or M one place to the right. Bit 7 is filled with the current value of the carry flag whilst the old bit 0 becomes the new carry flag value.

Processor Status after use:

C	Carry Flag	Set to contents of old bit 0
Z	Zero Flag	Set if A = 0
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Set if bit 7 of the result is set

Addressing Mode	Opcode	Bytes	Cycles
Accumulator	$6A	1	2
Zero Page	$66	2	5
Zero Page,X	$76	2	6
Absolute	$6E	3	6
Absolute,X	$7E	3	7

See also ASL, LSR, ROL

RTI - Return from Interrupt

The RTI instruction is used at the end of an interrupt processing routine. It pulls the processor flags from the stack followed by the program counter.

Processor Status after use:

C	Carry Flag	Set from stack
Z	Zero Flag	Set from stack
I	Interrupt Disable	Set from stack
D	Decimal Mode Flag	Set from stack
B	Break Command	Set from stack
V	Overflow Flag	Set from stack

N	Negative Flag	Set from stack

Addressing Mode	Opcode	Bytes	Cycles
Implied	$40	1	6

RTS - Return from Subroutine

The RTS instruction is used at the end of a subroutine to return to the calling routine. It pulls the program counter (minus one) from the stack.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Not affected
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Not affected

Addressing Mode	Opcode	Bytes	Cycles
Implied	$60	1	6

See also: JSR

SBC - Subtract with Carry

A,Z,C,N = A-M-(1-C)

This instruction subtracts the contents of a memory location to the accumulator together with the not of the carry bit. If overflow occurs the carry bit is clear, this enables multiple byte subtraction to be performed.

Processor Status after use:

C	Carry Flag	Clear if overflow in bit 7
Z	Zero Flag	Set if A = 0
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Set if sign bit is incorrect
N	Negative Flag	Set if bit 7 set

Addressing Mode	Opcode	Bytes	Cycles
Immediate	$E9	2	2
Zero Page	$E5	2	3
Zero Page,X	$F5	2	4
Absolute	$ED	3	4
Absolute,X	$FD	3	4 (+1 if page crossed)

Absolute,Y	$F9	3	4 (+1 if page crossed)
(Indirect,X)	$E1	2	6
(Indirect),Y	$F1	2	5 (+1 if page crossed)

See also: ADC

SEC - Set Carry Flag

C = 1

Set the carry flag to one.

C	Carry Flag	Set to 1
Z	Zero Flag	Not affected
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Not affected

Addressing Mode	Opcode	Bytes	Cycles
Implied	$38	1	2

See also: CLC

SED - Set Decimal Flag

D = 1

Set the decimal mode flag to one.

C	Carry Flag	Not affected
Z	Zero Flag	Not affected
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Set to 1
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Not affected

Addressing Mode	Opcode	Bytes	Cycles
Implied	$F8	1	2

See also: CLD

SEI - Set Interrupt Disable

I = 1

Set the interrupt disable flag to one.

C	Carry Flag	Not affected

Z	Zero Flag	Not affected
I	Interrupt Disable	Set to 1
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Not affected

Addressing Mode	Opcode	Bytes	Cycles
Implied	$78	1	2

See also: CLI

STA - Store Accumulator

M = A

Stores the contents of the accumulator into memory.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Not affected
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Not affected

Addressing Mode	Opcode	Bytes	Cycles
Zero Page	$85	2	3
Zero Page,X	$95	2	4
Absolute	$8D	3	4
Absolute,X	$9D	3	5
Absolute,Y	$99	3	5
(Indirect,X)	$81	2	6
(Indirect),Y	$91	2	6

See also: STX, STY

STX - Store X Register

M = X

Stores the contents of the X register into memory.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Not affected
I	Interrupt Disable	Not affected

D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Not affected

Addressing Mode	Opcode	Bytes	Cycles
Zero Page	$86	2	3
Zero Page,Y	$96	2	4
Absolute	$8E	3	4

See also: STA, STY

STY - Store Y Register

M = Y

Stores the contents of the Y register into memory.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Not affected
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Not affected

Addressing Mode	Opcode	Bytes	Cycles
Zero Page	$84	2	3
Zero Page,X	$94	2	4
Absolute	$8C	3	4

See also: STA, STX

TAX - Transfer Accumulator to X

X = A

Copies the current contents of the accumulator into the X register and sets the zero and negative flags as appropriate.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Set if X = 0
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected

N	Negative Flag	Set if bit 7 of X is set

Addressing Mode	Opcode	Bytes	Cycles
Implied	$AA	1	2

See also: TXA

TAY - Transfer Accumulator to Y

Y = A

Copies the current contents of the accumulator into the Y register and sets the zero and negative flags as appropriate.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Set if Y = 0
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Set if bit 7 of Y is set

Addressing Mode	Opcode	Bytes	Cycles
Implied	$A8	1	2

See also: TYA

TSX - Transfer Stack Pointer to X

X = S

Copies the current contents of the stack register into the X register and sets the zero and negative flags as appropriate.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Set if X = 0
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Set if bit 7 of X is set

Addressing Mode	Opcode	Bytes	Cycles
Implied	$BA	1	2

See also: TXS

TXA - Transfer X to Accumulator

A = X

Copies the current contents of the X register into the accumulator and sets the zero and negative flags as appropriate.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Set if A = 0
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Set if bit 7 of A is set

Addressing Mode	Opcode	Bytes	Cycles
Implied	$8A	1	2

See also: TAX

TXS - Transfer X to Stack Pointer

S = X

Copies the current contents of the X register into the stack register.

Processor Status after use:

C	Carry Flag	Not affected
Z	Zero Flag	Not affected
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Not affected

Addressing Mode	Opcode	Bytes	Cycles
Implied	$9A	1	2

See also: TSX

TYA - Transfer Y to Accumulator

A = Y

Copies the current contents of the Y register into the accumulator and sets the zero and negative flags as appropriate.

Processor Status after use:

C	Carry Flag	Not affected

Z	Zero Flag	Set if A = 0
I	Interrupt Disable	Not affected
D	Decimal Mode Flag	Not affected
B	Break Command	Not affected
V	Overflow Flag	Not affected
N	Negative Flag	Set if bit 7 of A is set

Addressing Mode	Opcode	Bytes	Cycles
Implied	$98	1	2

See also: TAY

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

This page was last updated on 17th February, 2008