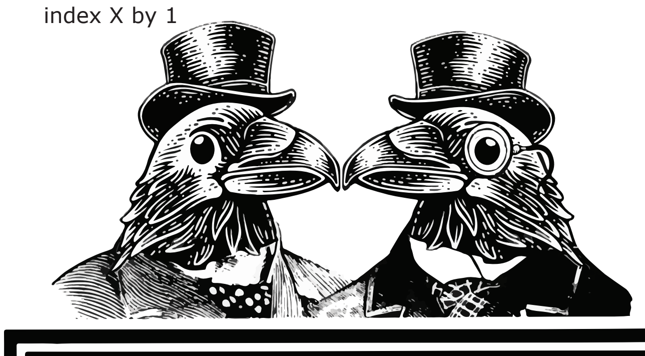
6502 & 65816 Instructions

Name and Description	Addressing Modes	Op-Codes	Status NVMXDIZC
ADC Add memory to accumulator with carry	ADC (ZP,X)	61 -	NV----ZC
	ADC SR, S	**63 -**	**NV----ZC**
	ADC ZP	65 -	NV----ZC
	ADC [ZP]	**67 -**	**NV----ZC**
	ADC #Imm	69 -	NV----ZC
	ADC Addr	6D --	NV----ZC
	ADC LongAddr	**6F ---**	**NV----ZC**
	ADC (ZP),Y	71 --	NV----ZC
	• ADC (ZP)	72 -	NV----ZC
	ADC (SR, S), Y	**73 -**	**NV----ZC**
	ADC ZP,X	75 -	NV----ZC
	ADC [ZP],Y	**77 -**	**NV----ZC**
	ADC Addr,Y	79 --	NV----ZC
	ADC Addr,X	7D --	NV----ZC
	ADC LongAddr,X	**7F ---**	**NV----ZC**
AND "AND" memory with accumulator	AND (ZP,X)	21 -	N-----Z-
	AND SR, S	**23 -**	**N-----Z-**
	AND ZP	25 -	N-----Z-
	AND [ZP]	**27 -**	**N-----Z-**
	AND #Imm	29 -	N-----Z-
	AND Addr	2D --	N-----Z-
	AND LongAddr	**2F ---**	**N-----Z-**
	AND (ZP),Y	31 --	N-----Z-
	• AND (ZP)	32 -	N-----Z-
	AND (SR, S), Y	**33 -**	**N-----Z-**
	AND ZP,X	35 -	N-----Z-
	AND [ZP],Y	**37 -**	**N-----Z-**
	AND Addr,Y	39 --	N-----Z-
	AND Addr,X	3D --	N-----Z-
	AND LongAddr,X	**3F ---**	**N-----Z-**
ASL Shift left one bit (Memory or Accumulator)	ASL ZP	06 -	N-----ZC
	ASL A	0A	N-----ZC
	ASL Addr	0E --	N-----ZC
	ASL ZP,X	16 -	N-----ZC
	ASL Addr,X	1E --	N-----ZC
BCC Branch if carry clear	BCC Rel	90 -	--------
BCS Branch if carry set	BCS Rel	B0 -	--------
BEQ Branch if equal	BEQ Rel	F0 -	--------
BIT Test bits	BIT ZP	24 -	NV----Z-
	BIT Addr	2C --	NV----Z-
	• BIT ZP,X	34 -	NV----Z-
	• BIT Addr,X	3C --	NV----Z-
	• BIT #Imm	89 -	NV----Z-
BMI Branch if minus	BMI Rel	30 -	--------
BNE Branch if not equal	BNE Rel	D0 -	--------
BPL Branch if plus	BPL Rel	10 -	--------
BRA Branch always	• BRA Rel	80 -	--------
BRK Break	BRK	00	----DI--
BRL Branch Long Always	**BRL Rel**	**82 -**	**--------**
BVC Branch if overflow clear	BVC Rel	50 -	--------
BVS Branch if overflow set	BVS Rel	70 -	--------
CLC Clear carry	CLC Rel	18 -	-------C

Name and Description	Addressing Modes	Op-Codes	Status NVMXDIZC
CLD Clear decimal mode flag	CLD	D8	----D---
CLI Clear interrupt disable flag	CLI	58	-----I--
CLV Clear overflow flag	CLV	B8	-V------
CMP Compare accumulator and memory	CMP (ZP,X)	C1 -	N-----ZC
	CMP SR, S	**C3 -**	**N-----ZC**
	CMP ZP	C5 -	N-----ZC
	CMP [ZP]	**C7 -**	**N-----ZC**
	CMP #Imm	C9 -	N-----ZC
	CMP Addr	CD --	N-----ZC
	CMP LongAddr	**CF ---**	**N-----ZC**
	CMP (ZP),Y	D1 --	N-----ZC
	• CMP (ZP)	D2 -	N-----ZC
	CMP (SR, S), Y	**D3 -**	**N-----ZC**
	CMP ZP,X	D5 -	N-----ZC
	CMP [ZP],Y	**D7 -**	**N-----ZC**
	CMP Addr,Y	D9 --	N-----ZC
	CMP Addr,X	DD --	N-----ZC
	CMP LongAddr,X	**DF ---**	**N-----ZC**
COP Co-Processor Enable	**COP IMM**	**02**	**----DI--**
CPX Compare memory with index X	CPX #Imm	E0 -	N-----ZC
	CPX ZP	E4 -	N-----ZC
	CPX Addr	EC --	N-----ZC
CPY Compare memory with index Y	CPY #Imm	C0 -	N-----ZC
	CPY ZP	C4 -	N-----ZC
	CPY Addr	CC --	N-----ZC
DEC Decrement memory by 1	• DEC A	3A	N-----Z-
	DEC ZP	C6 -	N-----Z-
	DEC Addr	CE --	N-----Z-
	DEC ZP,X	D6 -	N-----Z-
	DEC Addr,X	DE --	N-----Z-
DEX Decrement index X by 1	DEX	CA	N-----Z-
DEY Decrement index Y by 1	DEY	88	N-----Z-
EOR "Exclusive OR" accumulator with memory	EOR (ZP,X)	41 -	N-----Z-
	EOR SR, S	**43 -**	**N-----Z-**
	EOR ZP	45 -	N-----Z-
	EOR [ZP]	**47 -**	**N-----Z-**
	EOR #Imm	49 -	N-----Z-
	EOR Addr	4D --	N-----Z-
	EOR LongAddr	**4F ---**	**N-----Z-**
	EOR (ZP),Y	51 --	N-----Z-
	• EOR (ZP)	52 -	N-----Z-
	EOR (SR, S), Y	**53 -**	**N-----Z-**
	EOR ZP,X	55 -	N-----Z-
	EOR [ZP],Y	**57 -**	**N-----Z-**
	EOR Addr,Y	59 --	N-----Z-
	EOR Addr,X	5D --	N-----Z-
	EOR LongAddr,X	**5F ---**	**N-----Z-**
INC Increment memory by 1	• INC A	1A	N-----Z-
	INC ZP	E6 -	N-----Z-
	INC Addr	EE --	N-----Z-
	INC ZP,X	F6 -	N-----Z-
	INC Addr,X	FE --	N-----Z-
INX Increment index X by 1	INX	E8	N-----Z-



Name and Description	Addressing Modes	Op-Codes	Status NVMXDIZC
INY Increment index Y by 1	INY	C8	N-----Z-
JML Jump to new, long indirect location	**JML [Addr]**	**DC --**	**--------**
JMP Jump to new location	JMP Addr	4C --	--------
	JMP LongAddr	**5C ---**	**--------**
	JMP (Addr)	6C --	--------
	• **JMP (Addr,X)**	**7C --**	**--------**
JSL Jump subroutine long, saving return addr	**JSL LongAddr**	**22 ---**	**--------**
JSR Jump to subroutine, saving return addr	JSR Addr	20 --	--------
	JSR (Addr,X)	**FC --**	**--------**
LDA Load accumulator from memory	LDA (ZP,X)	A1 -	N-----Z-
	LDA SR, S	**A3 -**	**N-----Z-**
	LDA ZP	A5 -	N-----Z-
	LDA [ZP]	**A7 -**	**N-----Z-**
	LDA #Imm	A9 -	N-----Z-
	LDA Addr	AD --	N-----Z-
	LDA LongAddr	**AF ---**	**N-----Z-**
	LDA (ZP),Y	B1 --	N-----Z-
	• LDA (ZP)	B2 -	N-----Z-
	LDA (SR, S), Y	**B3 -**	**N-----Z-**
	LDA ZP,X	B5 -	N-----Z-
	LDA [ZP],Y	**B7 -**	**N-----Z-**
	LDA Addr,Y	B9 --	N-----Z-
	LDA Addr,X	BD --	N-----Z-
	LDA LongAddr,X	**BF ---**	**N-----Z-**
LDX Load index X from memory	LDX #Imm	A2 -	N-----Z-
	LDX ZP	A6 -	N-----Z-
	LDA Addr	AE --	N-----Z-
	LDX ZP,Y	B6 -	N-----Z-
	LDX Addr,Y	BE --	N-----Z-
LDY Load index Y from memory	LDY #Imm	A0 -	N-----Z-
	LDY ZP	A4 -	N-----Z-
	LDY Addr	AC --	N-----Z-
	LDY ZP,X	B4 -	N-----Z-
	LDY Addr,X	BC --	N-----Z-
LSR Shift right one bit (Memory or Accumulator)	LSR ZP	46 -	N-----ZC
	LSR A	4A -	N-----ZC
	LSR Addr	4E --	N-----ZC
	LSR ZP,X	56 -	N-----ZC
	LSR Addr,X	5E --	N-----ZC
MVN Move memory block in a negative direction	**MVN Src,Dest**	**54 --**	**--------**
MVP Move memory block in a positive direction	**MVP Sec,Dest**	**44 --**	**--------**
NOP No operation	NOP	EA	--------
ORA "OR" accumulator with memory	ORA (ZP,X)	01 -	N-----Z-
	ORA SR, S	**03 -**	**N-----Z-**
	ORA ZP	05 -	N-----Z-
	ORA [ZP]	**07 -**	**N-----Z-**
	ORA #Imm	09 -	N-----Z-
	ORA Addr	0D --	N-----Z-
	ORA LongAddr	**0F ---**	**N-----Z-**
	ORA (ZP),Y	11 --	N-----Z-
	• ORA (ZP)	12 -	N-----Z-
	ORA (SR, S), Y	**13 -**	**N-----Z-**
	ORA ZP,X	15 -	N-----Z-
	ORA [ZP],Y	**17 -**	**N-----Z-**
	ORA Addr,Y	19 --	N-----Z-
	ORA Addr,X	1D --	N-----Z-
	ORA LongAddr,X	**1F ---**	**N-----Z-**
PEA Push effective absolute address	**PEA Addr**	**F4 --**	**--------**
PEI Push effective indirect address	**PEI ZP**	**D4 -**	**--------**

CROW™ COUSINS
MICRO SOFTWARE

COPYRIGHT (C) 2019, CROW COUSINS

"APPLE" IS A REGISTERED TRADE MARK OF APPLE, INC.

Entries with bullets (•) are present in the 65C02 and the 65816
BOLDED entries are 65816-specific and not applicable to previous 6502 processors
The dashes following opcodes indicate the total length of the instruction. e.g. 2 dashes = a 3 byte instruction