6502 / 65c02 / 6280 / 65816 Cheatsheet

		Implied	Relative	Accumulator	Immediate	Zero Page	Zero Page,X	Zero PG,Y	Absolute	Absolute,X	Absolute,Y	Abs,X Indir	Indirect	(Indirect,X)	(Indirect),Y	C Z I D B V N	Absolute Long	Abs Indir Long	Direct Pg Indirect	Direct Pg Ind Lng	Abs Long,X	(Long Indirect),Y	Stack Relative	SR Indirect Indexed
		no params	jr	works on A	#nn &nn	$nn (&00nn)	$nn,X (&00nn+X)	$nn,Y (&00nn+Y)	$0100 (&0100)	$0100,X (&0100+X)	$0100,Y (&0100+Y)	($nnnn,X) ((&nnnn+X))	($nnnn) ((&nnnn))	($nn,X) ((&00nn+X))	($nn),Y ((&00nn)+Y)		$100000 ($100000)	[$1000] ([$1000])	($dpnn)	($dpdpnn)	$010000,X (&010000+X)	[$nn],Y ((&dpdpnn)+Y)	$ss,S ($ss+S)	($ss,S)+Y (($ss,S)+Y)
ADC	Add with Carry				$69 2 2	$65 2 3	$75 2 4		$6D 3 4	$7D 3 4/5	$79 3 4/5		*$72 3*	$61 2 6	$71 2 5/6	Ovf? z - - - - ?	$6F 3 4		$72 2 5	$67 2 6	$7F 4 5	$77 2 6	$63 2 4	$73 2 7
AND	Logical AND				$29 2 2	$25 2 3	$35 2 4		$2D 3 4	$3D 3 4/5	$39 3 4/5		*$32 3*	$21 2 6	$31 2 5/6	- z - - - - ?	$2F 4 5		$32 2 5	$27 2 4	$3F 4 5	$37 2 6	$23 2 4	$33 2 7
ASL	Arithmetic Shift Left			$0A 1 2		$06 2 5	$16 2 6		$0E 3 6	$1E 3 7						? z - - - - ?									
BCC	Branch if Carry Clear C=1 (Aka BLT)		$90 2 2/3/4													- - - - - - -									
BCS	Branch if Carry Set C=0 (Aka BGE)		$B0 2 2/3/4													- - - - - - -									
BEQ	Branch if Equal to Zero (Z = 1 JP Z,)		$F0 2 2/3/4													- - - - - - -									
BIT	Bit Test (And A with mem loc)				$89 2	$24 2 3	$34 2		$2C 3 4	$3C 3						- z - - - 6 7									
BMI	Branch if Minus (S = 1)		$30 2 2/3/4													- - - - - - -									
BNE	Branch if Not Equal to Zero (Z = 0 JP NZ,)		$D0 2 2/3/4													- - - - - - -									
BPL	Branch if Plus (S = 0)		$10 2 2/3/4													- - - - - - -									
BRK	Break	$00 1 7														- - - - =1 - -									
BVC	Branch if Overflow Clear		$50 2 2/3/4													- - - - - - -									
BVS	Branch if Overflow Set		$70 2 2/3/4													- - - - - - -									
CLC	Clear Carry Flag															=0 - - - - - -									
CLD	Clear Decimal Mode	$D8 1 2														- - - =0 - - -									
CLI	Clear Interrupt Mask (EI)	$58 1 2														- - =0 - - - -									
CLV	Clear Overflow Flag	$B8 1 2														- - - - - - =0 -									
CMP	Compare Accumulator to Memory				$C9 2 2	$C5 2 3	$D5 2 4		$CD 3 4	$DD 3 4/5	$D9 3 4/5		*$D2 3*	$C1 2 6	$D1 2 5/6	> = - - - - ?	$CF 4 5		$D2 2 5	$C7 2 6	$DF 4 5	$D7 2 6	$C3 2 4	$D3 2 7
CPX	Compare with Index Register X				$E0 2 2	$E4 2 3			$EC 3 4							> = - - - - ?									
CPY	Compare with Index Register Y				$C0 2 2	$C4 2 3			$CC 3 4							> = - - - - ?									
DEC	Decrement (Aka DEA)			$3A		$C6 2 5	$D6 2 6		$CE 3 6	$DE 3 7						- z - - - - ?									
DEX	Decrement Index Register X	$CA 1 2														- z - - - - ?									
DEY	Decrement Index Register Y	$88 1 2														- z - - - - ?									
EOR	Logical Exclusive-OR (XOR)				$49 2 2	$45 2 3	$55 2 4		$4D 3 4	$5D 3 4/5	$59 3/4/5		*$52 3*	$41 2 6	$51 2 5/6	- z - - - - ?	$4F 4 5		$52 2 5	$47 2 6	$5F 4 5	$57 2 6	$43 2 4	$53 2 7
INC	Increment (Aka INA)			$1A		$E6 2 5	$F6 2 6		$EE 3 6	$FE 3 7						- z - - - - ?									
INX	Increment Index Register X	$E8 1 2														- z - - - - ?									
INY	Increment Index Register Y	$C8 1 2														- z - - - - ?									
JMP	Jump to New Location (or **JML** for long)								$4C 3 3			$7C 3	$6C 3 5			- - - - - - -	$5C 4 4	$DC 3 6							
JSR	Jump to Subroutine (or **JSL** for long)								$20 3 6	$FC 3 8						- - - - - - -	$22 4 8								
LDA	Load Accumulator				$A9 2 2	$A5 2 3	$B5 2 4		$AD 3 4	$BD 3 4/5	$B9 3 4/5		*$B2 3*	$A1 2 6	$B1 2 5	- z - - - - ?	$AF 4 5		$B2 2 5	$A7 2 6	$BF 4 5	$B7 2 6	$A3 2 4	#B3 2 7
LDX	Load Index Register X				$A2 2 2	$A6 2 3		$B6 2 4	$AE 3 4		$BE 3 4/5					- z - - - - ?									
LDY	Load Index Register Y				$A0 2 2	$A4 2 3	$B4 2 4		$AC 3 4	$BC 3 4/5						- z - - - - ?									
LSR	Logical Shift Right (BitShift Right topbit 0)			$4A 1 2		$46 2 5	$56 2 6		$4E 3 6	$5E 3 7						- z - - - - ?									
NOP	No Operation	$EA 1 2														- - - - - - -									
ORA	Logical (Inclusive) OR				$09 2 2	$05 2 3	$15 2 4		$0D 3 4	$1D 3 4/5	$19 3 4/5		*$12 3*	$01 2 6	$11 2 5/6	- z - - - - ?	$0F 4 5		$12 2 5	$07 2 6	$1F 4 5	$07 2 6	$03 2 4	$13 2 7
PHA	Push Accumulator onto Stack (PUSH A)	$48 1 3														- - - - - - -									
PHP	Push Processor Status (PUSH F)	$08 1 3														- - - - - - -									
PLA	Pull Accumulator from Stack (POP A)	$68 1 4														- z - - - - ?									
PLP	Pull Processor Status (POP F)	$28 1 4														S S S S S S S									
ROL	Rotate Left through Carry (RLCA)			$2A 1 2		$26 2 5	$36 2 6		$2E 3 6	$3E 3 7						old7 z - - - - ?									
ROR	Rotate Right through Carry			$6A 1 2		$66 2 5	$76 2 6		$6E 3 6	$7E 3 7						old0 z - - - - ?									
RTI	Return from Interrupt (RETI)	$40 1 6														S S S S S S S									
RTS	Return from Subroutine (RET) (or **RTL** for long)	$60 1 6 $6B 1 6														- - - - - - -									
SBC	Subtract with Carry				$E9 2 2	$E5 2 3			$ED 3 4	$FD 3 4/5	$F9 3 4/5		$F2 3	$E1 2 6	$F1 2 5/6	Ovf? z - - - - ?	$EF 4 5		$F2 2 5	$E7 2 6	$FF 4 5	$F7 2 6	$E3 2 4	$F3 2 7
SEC	Set Carry (SCF)	$38 1 2														1 - - - - - -									
SED	Set Decimal Flag	$F8 1 2														- - - 1 - - -									
SEI	Set Interrupt Mask (Disable Interrupts)	$78 1 2														- - 1 - - - -									
STA	Store Accumulator					$85 2 3	$95 2 4		$8D 3 4	$9D 3 5	$99 3 5		$92 3	$81 2 6	$91 2 6	- - - - - - -	$8F 4 5		$92 2 5	$87 2 6	$9F 4 5	$97 2 6	$83 2 4	$93 $2 7
STX	Store Index Register X					$86 2 3		$96 2 4	$8E 3 4							- - - - - - -									
STY	Store Index Register Y					$84 2 3	$94 2 4		$8C 3 4							- - - - - - -									
TAX	Transfer Accumulator to Index Register X	$AA 1 2														- z - - - - ?									
TAY	Transfer Accumulator to Index Register Y	$A8 1 2														- z - - - - ?									
TSX	Transfer Stack Pointer to X (LD X,SP)	$BA 1 2														- z - - - - ?									
TXA	Transfer Index Register X to Accumulator	$8A 1 2														- z - - - - ?									
TXS	Transfer X to Stack Pointer (LD SP,X)	$9A 1 2														- - - - - - -									
TYA	Transfer Index Register Y to Accumulator	$98 1 2														- z - - - - ?									
BRA	Branch Relative Always (JR) (**BRL** for long)		$80 2 $82 3 4													- - - - - - -									
COP	Coprocessor Enable	$02 2 7														- - I D - - -									
MVN	Block Move Next (LDIR) MVN M,N A bytes from MX->NY (Alters DBR)				$54 3 ?											- - - - - - -									
MVP	Block Move Previous (LDDR) MVP M,N A bytes from MX→NY (Alters DBR)				$44 3 ?											- - - - - - -									
PEA	Push Effective Absolute address							$F4 3 5								- - - - - - -									
PEI	Push Effective Indirect Address															- - - - - - -			$D4 2 6						
PER	Push effective PC Relative Indirect Address		$62 3 6													- - - - - - -									
PHB	Push 8 bit Data Bank Reg (DBR)	$8B 1 3														- - - - - - -									
PHD	Push 16bit Direct Page Register	$0B 1 4														- - - - - - -									
PHK	Push 8 bit Program Bank Register (PBR)	$4B 1 3														- - - - - - -									
PHX	Push X	$DA 1														- - - - - - -									
PHY	Push Y	$5A 1														- - - - - - -									
PLB	Pull 8 bit Data Bank Reg (DBR)	$AB 1 4														- z - - - - ?									
PLD	Pull 16bit Direct Page Register	$2B 1 5														- z - - - - ?									
PLX	Pull X	$FA 1														- z - - - - ?									
PLY	Pull Y	$7A 1														- z - - - - ?									
REP	Reset Status Bits				$C2 2 3											? ? ? ? ? ? ? ?									
SEP	Set Status Bit				$E2 2 3											? ? ? ? ? ? ? ?									
STP	Stop processor until next RST	$DB 1														- - - - - - -									
STZ	Store Zero to address					$64 2	$74 2		$9C 3	$9E 3						- - - - - - -									
TCD	Transer Accumulator to the Direct page register (aka TAD)	$5B 1 2														- z - - - - ?									
TDC	Transfer Direct Page register to the Accumulator (aka TDA)	$7B 1 2														- z - - - - ?									
TCS	Transfer Accumulator to SP (aka TAS)	$1B 1 2														- z - - - - ?									
TRB	Test and Reset Bits with A					$14 2			$1C 3							- z - - - - -									
TSB	Test and Set Bits with A					$04 2			$0C 3							- z - - - - -									
TSC	Transfer SP to Accumulator (aka TSA)	$3B 1 2														- z - - - - -									
TXY	Transfer X to Y	$9B 1 2														- z - - - - ?									
TYX	Transfer Y to X	$BB 1 2														- z - - - - ?									
WAI	Wait until any interrupt	$CB 1														- - - - - - -									
WDM	Reseverd for future use!	$42 2														- - - - - - -									
XBA	Exchange A and B (aka SWA)	$EB 1 3														- z - - - - ?									
XCE	Exchange Carry (C) and Emu bits (E)	$FB 1 2														E - - - M B - -									
BBR	Branch if bit n is Reset (also some 65c02)					$0f-$7F 2										- - - - - - -									
BBS	Branch if bit n is Reset (also some 65c02)					$8f-$FF 2										- - - D - - -									
BSR	Branch to subroutine (Call Relative)	$44 2 8																						
CLX	Clear X	$82 1 2																						
CLY	Clear Y	$C2 1 2																						
CSH	Change Speed High (7.16 MHz)	$D4 1 3																						
CSL	Change Speed Low (1.78 MHz)	$54 1 3																						
RMB	Reset Memory Bit n (also some 65c02)					$07-$77										- - - - - - -									
SAX	Swap A and X	$22 1 3																						
SAY	Swap A and Y	$42 1 3																						
SET	Set T flag	$F4 1 2																						
SMB	Set Memory Bit n (also some 65c02)					$87-$F7										- - - - - - -									
ST0	ST0 - Store (HuC6270) VDC No. 0				$03 2 5																			
ST1	ST1 - Store (HuC6270) VDC No. 1				$13 2 5																			
ST2	ST2 - Store (HuC6270) VDC No. 2				$23 2 5																			
SXY	Swap X and Y registers	$02 1 3																						
TAI	Transfer Alternate Increment				$F3 7 17+																			
TAM	Tranfer Accumulator to MPR				$53 2 5																			
TIA	Transfer Increment Alternate				$E3 7 17+																			
TII	Transfer Increment Increment				$73 7 17+																			
TIN	Transfer Increment				$D3 7 17+																			
TMA	Tranfer MPR to Accumulator				$43 2 4																			
TST	Test Bits at n2 witn n1					$83 3 7	$A3 3 7	$93 4 8	$B3 4 8															
	http://www.LearnAsm.net	>=6502	65C02, 65816, 6280		65C02,6280 NOT 65816			6280	6280 + 65816		65816		16 bit in 65816 M=0 / X=0											