Opcode	Aka ...	Opcode	Addr	Bytes	Cycles	Flags	Description
KIL		$02	imp	1	?	None	Perform "STP"
		$12	imp	1	?		
		$22	imp	1	?		
		$32	imp	1	?		
		$42	imp	1	?		
		$52	imp	1	?		
		$62	imp	1	?		
		$72	imp	1	?		
		$92	imp	1	?		
		$B2	imp	1	?		
		$D2	imp	1	?		
		$F2	imp	1	?		
LAX		$A7	zp	2	3	NZ	A, X <- MEM(adr)
		$B7	zp,y	2	4		
		$AF	abs	3	4		
		$BF	abs,y	3	4*		
		$A3	(ind,X)	2	6		
		$B3	(ind),y	2	5*		
LDD	NOP	$80	imm	2	2	None	Load and Discard - KimKlone mnemonic. Multi-cycle NOP
		$82	imm	2	2		
		$89	imm	2	2		
		$C2	imm	2	2		
		$E2	imm	2	2		
		$04	zp	2	3		
		$44	zp	2	3		
		$64	zp	2	3		
		$14	zp,x	2	4		
		$34	zp,x	2	4		
		$54	zp,x	2	4		
		$74	zp,x	2	4		
		$D4	zp,x	2	4		
		$F4	zp,x	2	4		
		$0C	abs	3	4		
		$1C	abs,x	3	4*		
		$3C	abs,x	3	4*		
		$5C	abs,x	3	4*		
		$7C	abs,x	3	4*		
		$DC	abs,x	3	4*		
		$FC	abs,x	3	4*		
NOP		$1A	imp	1	2	None	No Operation
		$3A	imp	1	2		
		$5A	imp	1	2		
		$7A	imp	1	2		
		$DA	imp	1	2		
		$FA	imp	1	2		
RLA		$27	zp	2	5	NZC	MEM(Adr) <- ROL(adr), A <- A AND (adr)
		$37	zp,x	2	6		
		$2F	abs	3	6		
		$3F	abs,x	3	7		
		$3B	abs,y	3	7		
		$23	(ind,x)	2	8		
		$33	(ind),y	2	8		
RRA		$67	zp	2	5	NZCV	MEM(adr) <- ROR (adr), A <- A ADC (adr)
		$77	zp,x	2	6		
		$6F	abs	3	6		
		$7F	abs,x	3	7		
		$7B	abs,y	3	7		
		$63	(ind,x)	2	8		
		$73	(ind),y	2	8		
SLO		$07	zp	2	5	NZC	MEM(adr) <- ASL(adr), A <- A ORA(adr)
		$17	zp,x	2	6		
		$0F	abs	3	6		
		$1F	abs,x	3	7		
		$1B	abs,y	3	7		
		$03	(ind,x)	2	8		
		$13	(ind),y	2	8		
SRE		$47	zp	2	5	NZC	MEM(adr) <- LSR (adr), A <- A EOR (adr)
		$57	zp,x	2	6		
		$4F	abs	3	6		
		$5F	abs,x	3	7		
		$5B	abs,y	3	7		
		$43	(ind,x)	2	8		

		$53	(ind),y	2	8		
DCP		$C7	zp	2	5	NZC	MEM(adr) <- DEC (adr), A <- A CMP (adr)
		$D7	zp,x	2	6		
		$CF	abs	3	6		
		$DF	abs,x	3	7		
		$DB	abs,y	3	7		
		$C3	(ind,x)	2	8		
		$D3	(ind),y	2	8		
ISC	(ISB)	$E7	zp	2	5	NZCV	MEM(adr) <- INC (adr), A <- A SBC (adr)
		$F7	zp,x	2	6		
		$EF	abs	3	6		
		$FF	abs,x	3	7		
		$FB	abs,y	3	7		
		$E3	(ind,x)	2	8		
		$F3	(ind),y	2	8		
SBC		$EB	imm	2	2	NZCV	A <- A SBC (imm)
SAX	(AAX)	$87	zpg	2	3	None	MEM(Adr) <- A&X
		$97	zpg,y	2	4		
		$83	(ind,X)	2	6		
		$8F	Abs	3	4		
AXS	(SBX)	$CB	imm	2	2	NZC	X <- A&X SBC1 Imm; where SBC1 means SBC w/ C = 1. Always in binary mode
ANC		$0B	Imm	2	2	NZC	A <- A AND Imm, Carry <- A.7; Requires special flag handling
		$2B	Imm	2	2		
*LAS	(LAR)	$BB	abs,y	3	4*	NZ	A,X,SP <- MEM(adr) & SP Visual: correct values loaded to A, X, SP. Then reset and A0.3 and A4.7 are incremented Vic20: As documented VICE: "Stable"

"Unstables" - Magic Constant

*ALR	(ASR)	$4B	imm	2	2	NZC	A <- A AND (CONST | Imm), A <- LSR A; where CONST= VICE($00), VIC20($00), Visual($FF), go with "$00" Vice: claims stable
*ARR		$6B	imm	2	2	NZCV	A <- A & (CONST | Imm), ROR A; where CONST= VIC20($00), Visual($FF), go with "$00" Special Flags Handling (Binary mode): C <- A.6, V <- A.6 XOR A.5 Vice: claims stable
*LXA	(ATX)	$AB	imm	2	2	NZ	A,X <- (CONST | A) & B : where CONST= VICE(Unstable), VIC20($EE - Stable), Visual($00), go with "$FF" Vice: CONST is chip- and/or temperature dependent (common values may be $00, $ff, …) Shows as stable on VIC20 in thousands of operations
*XAA	(ANE)	$8B	imm	2	2	None	(CONST | A) & X & imm -> A; where CONST = VICE(Unstable), C64($FF - stable?), VIC20($FF - Stable), Visual($00), go with "$FF" See: http://visual6502.org/wiki/index.php?title=6502_Opcode_8B_%28XAA,_ANE%29 Vice: CONST is chip- and/or temperature dependent (common values may be $00, $ff, …) Shows as stable on VIC20 in thousands of operations

"Unstables" - & H + 1

*AHX	(AXA)	$9F	abs,Y	3	5	None	MEM(adr) <- A&X AND hi(adr) + 1
		$93	(ind),Y	2	6		**& H + 1 Group - see below**
*SHX	(SXA)	$9E	abs,y	3	5	None	MEM(adr) <- X AND hi(adr) + 1 **& H + 1 Group - see below**
*SHY	(SYA)	$9C	abs,x	3	5	None	MEM(adr) <- Y AND hi(addr) + 1 **& H + 1 Group - see below**
*TAS	(XAS)	$9B	abs,y	3	5	None	SP <- A&X, MEM(Adr) <- A&X AND hi(adr) + 1 **& H + 1 Group - see below**

VICE: & hi(adr) may be dropped - related to RDY and timing of VIC bus access
VICE: page boundary crossing may not work as expected (the page where the value is stored may be equal to the value stored).
Page boundary crossig problem seems to occur when crossing above the lower nibble pages (i.e. $0f to $10) but not below.