A. CPU INSTRUCTION SET SUMMARY

This appendix summarizes the CPU instruction set.

Table A-1 is a matrix of CPU instructions and addressing modes arranged by operation code.

Table A-2 lists the CPU instruction mnemonics and titles by mnemonic code in Table A-1.

Table A-3 lists the CPU instruction operation codes, mnemonics and addressing modes in op code order.

Table A-4 summarizes the operation of the MCU CPU instructions as referenced to the R6502 CPU.

Table A-5 summarizes the differences in operation between the MCU CPU and the R6502 CPU.

Table A-6 summarizes the threaded code instructions.

Table A-1. CPU Instruction Set Operation Code Matrix

	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	BRK Imp 1 7	ORA (Ind) 2 5	MPY Imp 1 6	TIP Imp 1 2		ORA Zp 2 3	ASL Zp 2 5	RMB0 Zp 2 5	PHP Imp 1 3	ORA Imm 2 2	ASL Acc 1 2	JSB0 (FFE0) 1 6	JPI Imp 3 5	ORA Abs 3 4	ASL Abs 3 6	BBR0 Zp 3 5[b]	0
1	BPL Rel 2 2[b]	ORA (Ind),X 2 5[a]	MPA Imp 1 6	LAB Acc 1 3		ORA Zp,X 2 4	ASL Zp,X 2 6	RMB1 Zp 2 5	CLC Imp 1 2	ORA Abs,Y 3 4[a]	NEG Acc 1 2	JSB1 (FFE2) 1 6		ORA Abs,X 3 4[a]	ASL Abs,X 3 7	BBR1 Zp 3 5[b]	1
2	JSR Abs 3 5	AND (Ind) 2 5	PSH Imp 1 5	PHW Imp 1 4	BIT Zp 2 3	AND Zp 2 3	ROL Zp 2 5	RMB2 Zp 2 5	PLP Imp 1 4	AND Imm 2 2	ROL Acc 1 2	JSB2 (FFE4) 1 6	BIT Abs 3 4	AND Abs 3 4	ROL Abs 3 6	BBR2 Zp 3 5[b]	2
3	BMI Rel 2 2[b]	AND (Ind),X 2 5[a]	PUL Imp 1 6	PLW Imp 1 5		AND Zp,X 2 4	ROL Zp,X 2 6	RMB3 Zp 2 5	SEC Imp 1 2	AND Abs,Y 3 4[a]	ASR Acc 1 2	JSB3 (FFE6) 1 6		AND Abs,X 3 4[a]	ROL Abs,X 3 7	BBR3 Zp 3 5[b]	3
4	RTI Imp 1 6	EOR (Ind) 2 5	RND Imp 1 2			EOR Zp 2 3	LSR Zp 2 5	RMB4 Zp 2 5	PHA Imp 1 3	EOR Imm 2 2	LSR Acc 1 2	JSB4 (FFE8) 1 6	JMP Abs 3 3	EOR Abs 3 4	LSR Abs 3 6	BBR4 Zp 3 5[b]	4
5	BVC Rel 2 2[b]	EOR (Ind),X 2 5[a]	CLW Imp 1 2			EOR Zp,X 2 4	LSR Zp,X 2 6	RMB5 Zp 2 5	CLI Imp 1 2	EOR Abs,Y 3 4[a]	PHY Imp 1 3	JSB5 (FFEA) 1 6		EOR Abs,X 3 4[a]	LSR Abs,X 3 7	BBR5 Zp 3 5[b]	5
6	RTS Imp 1 5	ADC (Ind) 2 5[c]	TAW Imp 1 2		ADD Zp 2 3[c]	ADC Zp 2 3[c]	ROR Zp 2 5	RMB6 Zp 2 5	PLA Imp 1 4	ADC Acc 2 2[c]	ROR Acc 1 2	JSB6 (FFEC) 1 6	JMP (Abs) 3 5	ADC Abs 3 4[c]	ROR Abs 3 6	BBR6 Zp 3 5[b]	6
7	BVS Rel 2 2[b]	ADC (Ind),X 2 5[a,c]	TWA Imp 1 2		ADD Zp,X 2 4[c]	ADC Zp,X 2 4[c]	ROR Zp,X 2 6	RMB7 Zp 2 5	SEI Imp 1 2	ADC Abs,Y 3 4[a,c]	PLY Imp 1 4	JSB7 (FFEE) 1 6	JMP (Abs,X) 3 6	ADC Abs,X 3 4[a,c]	ROR Abs,X 3 7	BBR7 Zp 3 5[b]	7
8	BRA Rel 2 3[a]	STA (Ind) 2 5			STY Zp 2 3	STA Zp 2 3	STX Zp 2 3	SMB0 Zp 2 5	DEY Imp 1 2	ADD Imm 2 2[c]	TXA Imp 1 2	NXT Imp 1 4	STY Abs 3 4	STA Abs 3 4	STX Abs 3 4	BBS0 Zp 3 5[b]	8
9	BCC Rel 2 2[b]	STA (Ind),X 2 6			STY Zp,X 2 4	STA Zp,X 2 4	STX Zp,Y 2 4	SMB1 Zp 2 5	TYA Imp 1 2	STA Abs,Y 3 5	TXS Imp 1 2	LII Imp 1 5		STA Abs,X 3 5		BBS1 Zp 3 5[b]	9
A	LDY Imm 2 2	LDA (Ind) 2 5	LDX Imm 2 2		LDY Zp 2 3	LDA Zp 2 3	LDX Zp 2 3	SMB2 Zp 2 5	TAY Imp 1 2	LDA Imm 2 2	TAX Imp 1 2	LAN Imp 1 3	LDY Abs 3 4	LDA Abs 3 4	LDX Abs 3 4	BBS2 Zp 3 5[b]	A
B	BCS Rel 2 2[b]	LDA (Ind),X 2 5[a]	STI Zp 3 4		LDY Zp,X 2 4	LDA Zp,X 2 4	LDX Zp,Y 2 4	SMB3 Zp 2 5	CLV Imp 1 2	LDA Abs,Y 3 4[a]	TSX Imp 1 2	INI Imp 1 3	LDY Abs,X 3 4[a]	LDA Abs,X 3 4[a]	LDX Abs,Y 3 4[a]	BBS3 Zp 3 5[b]	B
C	CPY Imm 2 2	CMP (Ind) 2 5	RBA Abs 4 7		CPY Zp 2 3	CMP Zp 2 3	DEC Zp 2 5	SMB4 Zp 2 5	INY Imp 1 2	CMP Imm 2 2	DEX Imp 1 2	PHI Imp 1 4	CPY Abs 3 4	CMP Abs 3 4	DEC Abs 3 6	BBS4 Zp 3 5[b]	C
D	BNE Rel 2 2[b]	CMP (Ind),X 2 5[a]	SBA Abs 4 7		EXC Zp,X 2 5	CMP Zp,X 2 4	DEC Zp,X 2 6	SMB5 Zp 2 5	CLD Imp 1 2	CMP Abs,Y 3 4[a]	PHX Imp 1 3	PLI Imp 1 6		CMP Abs,X 3 4[a]	DEC Abs,X 3 7	BBS5 Zp 3 5[b]	D
E	CPX Imm 2 2	SBC (Ind) 2 5[c]	BAR Abs 5 7[b]		CPX Zp 2 3	SBC Zp 2 3[c]	INC Zp 2 5	SMB6 Zp 2 5	INX Imp 1 2	SBC Imm 2 2[c]	NOP Imp 1 2	LAI Imp 1 3	CPX Abs 3 4	SBC Abs 3 4[c]	INC Abs 3 6	BBS6 Zp 3 5[b]	E
F	BEQ Rel 2 2[b]	SBC (Ind),X 2 5[a,c]	BAS Abs 5 7[b]			SBC Zp,X 2 4[c]	INC Zp,X 2 6	SMB7 Zp 2 5	SED Imp 1 2	SBC Abs,Y 3 4[a,c]	PLX Imp 1 4	PIA Imp 1 6		SBC Abs,X 3 4[a,c]	INC Abs,X 3 7	BBS7 Zp 3 5[b]	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

0

0 | BRK
Imp
1 7 | - Op Code
- Addressing Mode
- No. of Instruction Bytes; No. of Machine Cycles

a. Add 1 to N if page boundary is crossed.

b. Add 1 to N if branch occurs to same page.
Add 2 to N if branch occurs to different page.

c . Add 1 to N if in decimal mode.

Table A-2. CPU Instruction Set by Mnemonic

Mnemonic	Instruction	Mnemonic	Instruction
ADC	Add Memory to Accumulator with Carry	MPA*	Multiply and Accumulate
ADD*	Add Memory to Accumulator without Carry	MPY*	Multiply
AND	"AND" Memory with Accumulator		
ASL	Shift Left One Bit (Memory or Accumulator)	NEG*	Negate Accumulator
ASR*	Accumulator Shift Right One Bit, Sign Extend	NOP	No Operation
		NXT*	Next Instruction
BAR*	Branch On Bit(s) Reset		
BAS*	Branch On Bit(s) Set	ORA	"OR" Memory with Accumulator
BBR*	Branch On Bit Reset (8)		
BBS*	Branch On Bit Set (8)	PHA	Push Accumulator on Stack
BCC	Branch on Carry Clear	PHI*	Push I on Stack
BCS	Branch on Carry Set	PHP	Push Processor Status on Stack
BEQ	Branch on Equal	PHW*	Push W on Stack
BIT	Test Bits in Memory with Accumulator	PHX*	Push Index X on Stack
BMI	Branch on Minus	PHY*	Push Index Y on Stack
BNE	Branch on Not Zero	PIA*	Pull I from Stack, Load Accumulator
BPL	Branch on Plus	PLA	Pull Accumulator from Stack
BRA*	Branch Always	PLI*	Pull I from Stack
BRK	Break Command	PLP	Pull Processor Status from Stack
BVC	Branch on Overflow Clear	PLW	Pull W from Stack
BVS	Branch on Overflow Set	PLX*	Pull Index X from Stack
		PLY*	Pull Index Y from Stack
CLC	Clear Carry Flag	PSH*	Push A, X and Y on Stack
CLD	Clear Decimal Mode	PUL*	Pull Y, X and A from Stack
CLI	Clear Interrupt Disable Bit		
CLV	Clear Overflow Flag	RBA*	Reset Bit(s) in Memory
CLW*	Clear W Register and Overflow Flag	RMB*	Reset Memory Bit (8)
CMP	Compare Memory and Accumulator	RND*	Round
CPX	Compare Memory and Index X	ROL	Rotate Left One Bit (Memory or Accumulator)
CPY	Compare Memory and Index Y	ROR	Rotate Right One Bit (Memory or Accumulator)
		RTI	Return from Interrupt
DEC	Decrement Memory by One	RTS	Return from Subroutine
DEX	Decrement Index X by One		
DEY	Decrement Index Y by One	SBA*	Set Bit(s) in Memory
		SBC	Subtract Memory from Accumulator with Borrow
EOR	"Exclusive-Or" Memory with Accumulator	SEC	Set Carry Flag
EXC*	Exchange Accumulator and Memory	SED	Set Decimal Mode
		SEI	Set Interrupt Disable Status
INC	Increment Memory by One	SMB*	Set Memory Bit (8)
INI*	Increment I by One	STA	Store Accumulator in Memory
INX	Increment Index X by One	STI*	Store Immediate to Memory
INY	Increment Index Y by One	STX	Store Index X in Memory
		STY	Store Index Y in Memory
JMP*	Jump to New Location		
JPI*	Jump Indirect with Return in I	TAW*	Transfer Accumulator to W
JSB*	Jump to Subroutine (8)	TAX	Transfer Accumulator to Index X
JSR	Jump to New Location Saving Return Address	TAY	Transfer Accumulator to Index Y
		TIP*	Transfer I to Program Counter
LAB*	Load Absolute to Accumulator	TSX	Transfer Stack Pointer to Index X
LAI*	Load Accumlator Indirect through I	TWA*	Transfer W to Accumulator
LAN*	Load Accumulator Indirect and Increment I	TXA	Transfer Index X to Accumulator
LDA	Load Accumulator with Memory	TXS	Transfer Index X to Stack Pointer
LDX	Load Index X with Memory	TYA	Transfer Index Y to Accumulator
LDY	Load Index Y with Memory		
LII*	Load I Indirect through I		
LSR	Logical Shift Right One Bit (Memory or Accumulator)		

* = New instruction or addressing mode from R6502.

Table A-3. CPU Instruction Set by Operation Code

Op Code	Mnemonic	Addressing Mode	Op Code	Mnemonic	Addressing Mode
00	BRK	Implied	20	JSR	Absolute
01	ORA	(Indirect)	21	AND	(Indirect)
02	MPY	Implied	22	PSH	Implied
03	TIP	Implied	23	PHW	Implied
04	Not Used		24	BIT	Zero Page
05	ORA	Zero Page	25	AND	Zero Page
06	ASL	Zero Page	26	ROL	Zero Page
07	RMB 0	Zero Page	27	RMB 2	Zero Page
08	PHP	Implied	28	PLP	Implied
09	ORA	Immediate	29	AND	Immediate
0A	ASL	Accumulator	2A	ROL	Accumulator
0B	JSB 0	(FFE0)	2B	JSB 2	(FFE4)
0C	JPI	Implied	2C	BIT	Absolute
0D	ORA	Absolute	2D	AND	Absolute
0E	ASL	Absolute	2E	ROL	Absolute
0F	BBR 0	Zero Page	2F	BBR 2	Zero Page
10	BPL	Relative	30	BMI	Relative
11	ORA	(Indirect),X	31	AND	(Indirect),X
12	MPA	Implied	32	PUL	Implied
13	LAB	Accumulator	33	PLW	Implied
14	Not Used		34	Not Used	
15	ORA	Zero Page,X	35	AND	Zero Page,X
16	ASL	Zero Page,X	36	ROL	Zero Page,X
17	RMB 1	Zero Page	37	RMB 3	Zero Page
18	CLC	Implied	38	SEC	Implied
19	ORA	Absolute,Y	39	AND	Absolute,Y
1A	NEG	Accumulator	3A	ASR	Accumulator
1B	JSB 1	(FFE2)	3B	JSB 3	(FFE6)
1C	Not Used		3C	Not Used	
1D	ORA	Absolute,X	3D	AND	Absolute,X
1E	ASL	Absolute,X	3E	ROL	Absolute,X
1F	BBR 1	Zero Page	3F	BBR 3	Zero Page

Table A-3. CPU Instruction Set by Operation Code (Cont'd)

Op Code	Mnemonic	Addressing Mode	Op Code	Mnemonic	Addressing Mode
40	RTI	Implied	60	RTS	Implied
41	EOR	(Indirect)	61	ADC	(Indirect)
42	RND	Implied	62	TAW	Implied
43	Not Used		63	Not Used	
44	Not Used		64	ADD	Zero Page
45	EOR	Zero Page	65	ADC	Zero Page
46	LSR	Zero Page	66	ROR	Zero Page
47	RMB 4	Zero Page	67	RMB 6	Zero Page
48	PHA	Implied	68	PLA	Implied
49	EOR	Immediate	69	ADC	Immediate
4A	LSR	Accumulator	6A	ROR	Accumulator
4B	JSB 4	(FFE8)	6B	JSB 6	(FFEC)
4C	JMP	Absolute	6C	JMP	(Absolute)
4D	EOR	Absolute	6D	ADC	Absolute
4E	LSR	Absolute	6E	ROR	Absolute
4F	BBR 4	Zero Page	6F	BBR 6	Zero Page
50	BVC	Relative	70	BVS	Relative
51	EOR	(Indirect),X	71	ADC	(Indirect),X
51	CLW	Implied	72	TWA	Implied
53	Not Used		73	Not Used	
54	Not Used		74	ADD	Zero Page,X
55	EOR	Zero Page,X	75	ADC	Zero Page,X
56	LSR	Zero Page,X	76	ROR	Zero Page,X
57	RMB 5	Zero Page	77	RMB 7	Zero Page
58	CLI	Implied	78	SEI	Implied
59	EOR	Absolute,Y	79	ADC	Absolute,Y
5A	PHY	Implied	7A	PLY	Implied
5B	JSB 5	(FFEA)	7B	JSB 7	(FFEE)
5C	Not Used		7C	JMP	Absolute,X
5D	EOR	Absolute,X	7D	ADC	Absolute,X
5E	LSR	Absolute,X	7E	ROR	Absolute,X
5F	BBR 5	Zero Page	7F	BBR 7	Zero Page

Table A-3. CPU Instruction Set by Operation Code (Cont'd)

Op Code	Mnemonic	Addressing Mode	Op Code	Mnemonic	Addressing Mode
80	BRA	Relative	A0	LDY	Immediate
81	STA	(Indirect)	A1	LDA	(Indirect)
82	Not Used		A2	LDX	Immediate
83	Not Used		A3	Not Used	
84	STY	Zero Page	A4	LDY	Zero Page
85	STA	Zero Page	A5	LDA	Zero Page
86	STX	Zero Page	A6	LDX	Zero Page
87	SMB 0	Zero Page	A7	SMB 2	Zero Page
88	DEY	Implied	A8	TAY	(Indirect)
89	ADD	Immediate	A9	LDA	Immediate
8A	TXA	Implied	AA	TAX	Implied
8B	NXT	Implied	AB	LAN	Implied
8C	STY	Absolute	AC	LDY	Absolute
8D	STA	Absolute	AD	LDA	Absolute
8E	STX	Absolute	AE	LDX	Absolute
8F	BBS 0	Zero Page	AF	BBS 2	Zero Page
90	BCC	Relative	B0	BCS	Relative
91	STA	(Indirect),X	B1	LDA	(Indirect),X
92	Not Used		B2	STI	Zero Page
93	Not Used		B3	Not Used	
94	STY	Zero Page,X	B4	LDY	Zero Page,X
95	STA	Zero Page,X	B5	LDA	Zero Page,X
96	STX	Zero Page,Y	B6	LDX	Zero Page,Y
97	SMB 1	Zero Page	B7	SMB 3	Zero Page
98	TYA	Implied	B8	CLV	Implied
99	STA	Absolute,Y	B9	LDA	Absolute,Y
9A	TXS	Implied	BA	TSX	Implied
9B	LII	Implied	BB	INI	Implied
9C	Not Used		BC	LDY	Absolute,X
9D	STA	Absolute,X	BD	LDA	Absolute,X
9E	Not Used		BE	LDX	Absolute,Y
9F	BBS 1	Zero Page	BF	BBS 3	Zero Page

Table A-3. CPU Instruction Set by Operation Code (Cont'd)

Op Code	Mnemonic	Addressing Mode	Op Code	Mnemonic	Addressing Mode
C0	CPY	Immediate	E0	CPX	Immediate
C1	CMP	(Indirect)	E1	SBC	(Indirect)
C2	RBA	Absolute	E2	BAR	Immediate
C3	Not Used		E3	Not Used	
C4	CPY	Zero Page	E4	CPX	Zero Page
C5	CMP	Zero Page	E5	SBC	Zero Page
C6	DEC	Zero Page	E6	INC	Zero Page
C7	SMB 4	Zero Page	E7	SMB 6	Zero Page
C8	INY	Implied	E8	INX	Implied
C9	CMP	Immediate	E9	SBC	Immediate
CA	DEX	Implied	EA	NOP	Implied
CB	PHI	Implied	EB	LAI	Implied
CC	CPY	Absolute	EC	CPX	Absolute
CD	CMP	Absolute	ED	SBC	Absolute
CE	DEC	Absolute	EE	INC	Absolute
CF	BBS 4	Zero Page	EF	BBS 6	Zero Page
D0	BNE	Relative	F0	BEQ	Relative
D1	CMP	(Indirect),X	F1	SBC	(Indirect),X
D2	SBA	Absolute	F2	BAS	Absolute
D3	Not Used		F3	Not Used	
D4	EXC	Zero Page,X	F4	Not Used	
D5	CMP	Zero Page,X	F5	SBC	Zero Page,X
D6	DEC	Zero Page,X	F6	INC	Zero Page,X
D7	SMB 5	Zero Page	F7	SMB 7	Zero Page
D8	CLD	Implied	F8	SED	Implied
D9	CMP	Absolute,Y	F9	SBC	Absolute,Y
DA	PHX	Implied	FA	PLX	Implied
DB	PLI	Implied	FB	PIA	Implied
DC	Not Used		FC	Not Used	
DD	CMP	Absolute,X	FD	SBC	Absolute,X
DE	DEC	Absolute,X	FE	INC	Absolute,X
DF	BBS 5	Zero Page	FF	BBS 7	Zero Page

Table A-4. New CPU Instructions Since R6502 CPU

Nine Basic Instructions

Mnemonic	Operation	Addressing Mode	No. Bytes	No. Cycles
SMB	Set Memory Bit (8)	ZP	2	5
RMB	Reset Memory Bit (8)	ZP	2	5
BBS	Branch on Bit Set (8)	ZP	3	5, 6, 7
BBR	Branch on Bit Reset (8)	ZP	3	5, 6, 7
BRA	Branch Always	ZP	2	2, 3, 4
PHX	Push X	Implied	1	3
PHY	Push Y	Implied	1	3
PLX	Pull X	Implied	1	4
PLY	Pull Y	Implied	1	4

Fifteen Filter Enhancement Instructions

Mnemonic	Operation	Addressing Mode	No. Bytes	No. Cycles
ASR	Shift A Right, Sign Extend	Accum	1	2
CLW	Clear W and V	Implied	1	2
EXC	Swap A, M	ZP, X	1	5
JSB	Jump to Subroutine (8)	(FFE_)	1	6
LAB	Load Absolute to Accumulator	Accum	1	2
MPA	Multiply and Accumulate	Implied	1	6
MPY	Multiply	Implied	1	6
PSH	Push A, X and Y on Stack	Implied	1	5
PUL	Pull Y, X and A from Stack	Implied	1	6
RND	Round	Implied	1	2
TAW	Transfer Accumulator to W	Implied	1	2
TWA	Transfer W toAccumulator	Implied	1	2
NEG	Negate Accumulator	Accum	1	2
PHW	Push WH, WL	Implied	1	4
PLW	Pull WL, WH	Implied	1	5

Table A-4. New CPU Instructions Since R6502 CPU (Cont'd)

Ten Direct Threaded Instructions

Mnemonic	Operation	Addressing Mode	No. Bytes	No. Cycles
NXT	(I) → PC, I + 2 → I	Implied	1	4
LII	(I) → I	Implied	1	5
LAI	(I) → A	Implied	1	3
INI	I + 1 → I	Implied	1	3
PHI	Push I	Implied	1	4
PLI	Pull I	Implied	1	6
JPI	PC + 1 → IL, PC + 2 → IH, (I) → PC, I + 2 → I	Implied	3	5
TIP	I → PC	Implied	1	2
PIA	Pull I, (I) → A, (I) → X, I +1 → I	Implied	1	6
LAN	(I) → A, I + 1 → I	Implied	1	3

Seven Controller Instructions

Mnemonic	Operation	Addressing Mode	No. Bytes	No. Cycles
BAR	Branch on Bit(s) Reset	ABS	5	7, 8, 9
BAS	Branch on Bit(s) Set	ABS	5	7, 8, 9
JMP	Jump	(ABS, X)	3	6
STI	Move IMM to Memory	ZP	3	4
RBA	Reset Bit(s) in Memory	ABS	4	7
SBA	Set Bit(s) in Memory	ABS	4	7
ADD	Add without Carry	IMM	2	2
ADD	Add without Carry	ZP	2	3
ADD	Add without Carry	ZP, X	2	4

Table A-5. CPU Instruction Enhancments Over R6502 CPU

Function	NMOS R6502 CPU	CMOS MCU CPU
Jump indirect, operand = XXFF.	Page address does not increment.	Page address increments.
Read/modify/write instructions at effective address.	One read cycle and two write cycles.	Two read cycles and one write cycle.
Decimal flag.	Indeterminate after reset.	Initialized to binary mode (D = 0) after reset.
Decimal ADD/SUB execution time.	Same execution time as binary.	One additional cycle for decimal correct.
Flags after decimal ADD/SUB.	N, V and Z flags are invalid.	N, V and Z flags are valid.
Interrupt coincident with BRK instruction.	Interrupt vector is loaded, BRK vector is ignored.	Interrupt is executed, then BRK is executed.
JSR instruction.	Stacked address points to last byte of JSR instruction.	Stacked address points to next op code. Instruction is one cycle shorter.
RTS instruction.	Return address is incremented before use.	Return address is ready for use. Instruction is one cycle shorter.
Indirect Addressing Opcodes changed.	(INDIRECT, X)	(INDIRECT)
Indirect Addressing Opcodes changed.	(INDIRECT), Y	(INDIRECT), X

Table A-6. CPU Threaded Instructions

Mnemonic	Instruction	Operation	Description
NXT	Next Instruction	(I) → PC I+2 → I	The I register points to an address. The two bytes at that address are loaded into the Program Counter. The contents of the I register are incremented by 2.
LII	Load I Indirect through I	(I) → I	The I register points to an address. The two bytes at that address are loaded into the I register.
LAI	Load A Indirect through I	(I) → A	The I register points to an address. The byte at that address is loaded into the Accumulator.
INI	Increment I by One	I+1 → I	The I register is incremented by 1.
PHI	Push I on Stack	I → (stack) SP–2 → SP	The contents of the I register are pushed onto the stack, high byte first.
PLI	Pull I from Stack	(stack) → I SP+2 → SP	The two bytes pointed to by the Stack Pointer are loaded into the I register, low byte first.
JPI (Operand)	Jump Indirect with Return in I	PC+1 → I (I) → PC I+2 →I	The contents of the Program Counter (the address of the JPI instruction) +1 are loaded into the I register. I now points to the two-byte operand of the JPI instruction. This operand is used as an indirect pointer to the next execution address. I is incremented by 2 to point to the next opcode following the JPI instruction. This instruction functions as a JSR indirect with the return address in the I register.
TIP	Transfer I to Program Counter	I →PC	Transfer the contents of the I register to the Program Counter. This instruction functions as an RTS to the JPI instruction.
PIA	Pull I from Stack, Load A	(stack) → I SP+2 → SP (I) → A I+1 → I	Load the I register with the two bytes pointed to by the Stack Pointer, low byte first. Increment the Stack Pointer by 2. Load the byte pointed to by the I register into the Accumulator. Increment the I register by 1.
LAN	Load A Indirect and Increment I	(I) → A I+1 → I	Load the byte pointed to by the I register into the Accumulator. Increment the I register by 1.

This page is intentionally blank.

B. CPU INSTRUCTION SET DESCRIPTION

This appendix describes the CPU instructions. The instructions are listed in alphabetic order. The following is provided for each instruction:

- Standard mnemonic
- Title
- Symbolic operation
- Processor status bits affected
- Operation description

Also provided, in tabular form, for each available addressing mode are:

- Addressing mode
- Assembly language form
- Operation code (Op Code)
- Number of bytes in the instruction
- Number of cycles to execute the instruction

The reference number shown at the end of the description of each instruction refers to the section in Appendix C that provides additional information about the instruction.

The following notation applies to this appendix:

A	Accumulator
ADL	Effective address low
ADH	Effective address high
X,Y	Index Registers
M	Memory
M_b	Selector Zero Page Memory Bit
M_6	Memory Bit 6
M_7	Memory Bit 7
P	Processor Status Register
S	Stack Pointer
Ã	Change
	No Change
+	Add
^	Logical AND
-	Subtract
∨	Logical EXCLUSIVE OR
↑	Transfer from Stack
↓	Transfer to Stack
→	Transfer to
←	Transfer to
∨	Logical OR
PC	Program Counter
PCH	Program Counter High
PCL	Program Counter Low
Oper	Operand
#	Immediate Addressing Mode
M	Mask
Dest	Destination

ADC Add Memory to Accumulator with Carry

Operation: A + M + C → A, C

N	V	–	B	D	I	Z	C
Ã	Ã	–	–	–	–	Ã	Ã

ADC adds the value in memory and the C flag to the value in the accumulator and stores the result in the accumulator.

The C flag is set when the sum exceeds hexadecimal 255 (binary mode) or decimal 99 (decimal mode), otherwise C is reset. In the binary mode, the V flag is set when the sign (bit 7) changes due to the result exceeding +127 or –128, otherwise V is reset. In the decimal mode, V is reset. The N flag is set if bit 7 of the result is a 1, otherwise N is reset. The Z flag is set if the result is zero, otherwise Z is reset.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Immediate	ADC	#Oper	69	2	2[c]	C2.1
Zero Page	ADC	Oper	65	2	3[c]	C3.1
Zero Page,X	ADC	Oper,X	75	2	4[c]	C4.1
Absolute	ADC	Oper	6D	3	4[c]	C6.1
Absolute,X	ADC	Oper,X	7D	3	4[a,c]	C7.1
Absolute,Y	ADC	Oper,Y	79	3	4[a,c]	C8.1
(Indirect)	ADC	(Oper)	61	2	5[c]	C9.1
(Indirect),X	ADC	(Oper),X	71	2	5[a,c]	C10.1

a. Add 1 if a page boundary is crossed.

c. Add 1 if in decimal mode.

ADD Add Memory to Accumulator without Carry

Operation: A + M → A, C

N	V	–	B	D	I	Z	C
Ã	Ã	–	–	–	–	Ã	Ã

ADD adds the value in memory to the value in the accumulator and stores the result in the accumulator.

The C flag is set when the sum exceeds hexadecimal 255 (binary mode) or decimal 99 (decimal mode), otherwise C is reset. In the binary mode, the V flag is set when the sign (bit 7) changes due to the result exceeding +127 or –128, otherwise V is reset. In the decimal mode, V is reset. The N flag is set if bit 7 of the result is a 1, otherwise N is reset. The Z flag is set if the result is zero, otherwise Z is reset.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Immediate	ADD	#Oper	89	2	2[c]	C2.1
Zero Page	ADD	Oper	64	2	3[c]	C3.1
Zero Page,X	ADD	Oper,X	74	2	4[c]	C4.1

c. Add 1 if in decimal mode.

AND "AND" Memory with Accumulator

Operation: $A \wedge M \rightarrow A$

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	–

AND logically ANDs the contents of the accumulator and addressed memory bit-by-bit and stores the result in the accumulator.

The Z flag is set if the result is zero, otherwise Z is reset. The N flag is set if bit 7 of the result is a 1, otherwise N is reset. C and V flags are unaffected.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Immediate	AND	#Oper	29	2	2	C2.1
Zero Page	AND	Oper	25	2	3	C3.1
Zero Page,X	AND	Oper,X	35	2	4	C4.1
Absolute	AND	Oper	2D	3	4	C6.1
Absolute,X	AND	Oper,X	3D	3	4[a]	C7.1
Absolute,Y	AND	Oper,Y	39	3	4[a]	C8.1
(Indirect)	AND	(Oper)	21	2	5	C9.1
(Indirect),X	AND	(Oper),X	31	2	5[a]	C10.1

a. Add 1 if page boundary is crossed.

ASL Shift Left One Bit (Memory or Accumulator)

Operation: $C \leftarrow 7\ 6\ 5\ 4\ 3\ 2\ 1 \leftarrow 0$

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	Ã

ASL shifts the contents of either the accumulator or the addressed memory location one bit to the left, with bit 0 always being set to 0 and the bit 7 output always being shifted to the C flag. If the accumulator is addressed, memory is unaffected; if memory is addressed, the accumulator is unaffected.

The N flag is set equal to bit 7 of the result (bit 6 of the input); the Z flag is set if the result is equal to 0, otherwise Z is reset. The V flag is unaffected.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Accumulator	ASL	A	0A	1	2	C11.1
Zero Page	ASL	Oper	06	2	5	C12.1
Zero Page,X	ASL	Oper,X	16	2	6	C12.2
Absolute	ASL	Oper	0E	3	6	C12.3
Absolute,X	ASL	Oper,X	1E	3	7	C12.4

ASR Accumulator Shift Right One Bit, Sign Extend

Operation: 7 → 7 6 5 4 3 2 1 0 → C

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	Ã

ASR shifts the contents of the accumulator one bit to the right, with bit 7 being set to the input bit 7 value and the input bit 0 being shifted to the C flag.

The N flag is set equal to bit 7 of the result (the N flag will retain the same value as before since bit 7 copies the input bit 7); the Z flag is set if the result is equal to zero, otherwise Z is reset. The V flag is unaffected.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Accumulator	ASR	A	3A	1	2	C11.1

BAR Branch on Bit(s) Reset

Operation: Branch on ~Memory ∧ Mask _ 0

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

BAR compares an addressed 8-bit location to an 8-bit mask in the instruction. If any bits at the addressed location contain a 0 in a bit position corresponding to a 1 in the mask, a conditional branch is taken to the relative address also contained in the instruction. Bytes 2 and 3 of the instruction contain the ADL and ADH, respectively, of the addressed memory or I/O port. Byte 4 contains the mask. Byte 5 specifies the conditional branch offset.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Bit/Absolute	BAR	Oper,Mask,Dest	E2	5	7[b]	C15.4

b. Add 1 if branch occurs to the same page; add 2 if branch occurs to a different page.

BAS Branch on Bit(s) Set

Operation: Branch on Memory ∧ Mask _ 0

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

BAS compares an addressed 8-bit location to an 8-bit mask in the instruction. If any bits at the addressed location contain a 1 in a bit position corresponding to a 1 in the mask, a conditional branch is taken to the relative address also contained in the instruction. Bytes 2 and 3 of the instruction contain the ADL and ADH, respectively, of the addressed memory or I/O port. Byte 4 contains the mask. Byte 5 specifies the conditional branch offset.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Bit/Absolute	BAS	Oper,Mask,Dest	F2	5	7[b]	C15.4

b. Add 1 if branch occurs to the same page; add 2 if branch occurs to a different page.

BBR Branch on Bit Reset (8)

Operation: Branch on Mb = 0 (b = 0 – 7)

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

BBR tests one of 8 bits in a zero page memory location. If the tested bit is reset to a 0, a conditional branch is taken to the relative address also specified in the instruction. If the bit tested is set, the next sequential instruction is executed. The bit to test is specified by 3 bits of the op code in byte 1 of the instruction. Byte 2 of the instruction designates the zero page address of the byte to be tested. Byte 3 of the instruction specifies the 8-bit relative address for the conditional branch.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Bit/Zero Page				3	5[b]	C15.3
Bit 0	BBR	0,Oper,Dest	0F			
Bit 1	BBR	1,Oper,Dest	1F			
Bit 2	BBR	2,Oper,Dest	2F			
Bit 3	BBR	3,Oper,Dest	3F			
Bit 4	BBR	4,Oper,Dest	4F			
Bit 5	BBR	5,Oper,Dest	5F			
Bit 6	BBR	6,Oper,Dest	6F			
Bit 7	BBR	7,Oper,Dest	7F			

b. Add 1 if branch occurs to the same page; add 2 if branch occurs to a different page.

BBS Branch on Bit Set (8)

Operation: Branch on M_b = 1 (b=0 – 7)

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

BBS tests one of 8 bits in a zero page memory location. If the tested bit is set to a 1, a conditional branch is taken to the relative address also specified in the instruction. If the bit tested is not set, the next sequential instruction is executed. The bit to test is specified by 3 bits of the op code in byte 1 of the instruction. Byte 2 of the instruction designates the zero page address of the byte to be tested. Byte 3 of the instruction specifies the 8-bit relative address for the conditional branch.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Bit/Zero Page				3	5[b]	C15.3
Bit 0	BBS	0,Oper,Dest	8F			
Bit 1	BBS	1,Oper,Dest	9F			
Bit 2	BBS	2,Oper,Dest	AF			
Bit 3	BBS	3,Oper,Dest	BF			
Bit 4	BBS	4,Oper,Dest	CF			
Bit 5	BBS	5,Oper,Dest	DF			
Bit 6	BBS	6,Oper,Dest	EF			
Bit 7	BBS	7,Oper,Dest	FF			

b. Add 1 if branch occurs to the same page; add 2 if branch occurs to a different page.

BCC Branch on Carry Clear

Operation: Branch on C = 0

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

BCC tests the state of the C flag and takes a conditional branch if C is reset, otherwise the next sequential instruction is executed.

Addressing Mode	Assemble Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Relative	BCC	Dest	90	2	2[b]	C13.1

b. Add 1 if branch occurs to the same page; add 2 if branch occurs to a different page.

BCS Branch on Carry Set

Operation: Branch on C = 1

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

BCS takes the conditional branch if the C flag is set (carry is on), otherwise the next sequential instruction is executed.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Relative	BCS	Dest	B0	2	2^b	C13.1

b. Add 1 if branch occurs to the same page; add 2 if branch occurs to a different page.

BEQ Branch on Equal

Operation: Branch on Z = 1

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

BEQ takes the conditional branch if the Z flag is set (result is zero), otherwise the next sequential instruction is executed. BEQ is also referred to as "Branch on Zero".

Addressing Mode	Assemble Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Relative	BEQ	Dest	F0	2	2^b	C13.1

b. Add 1 if branch occurs to the same page; add 2 if branch occurs to a different page.

BIT Test Bits in Memory with Accumulator

Operation: $A \wedge M, M_7 \rightarrow N, M_6 \rightarrow V$

N	V	–	B	D	I	Z	C
M_7	M_6	–	–	–	–	Ã	–

BIT performs a logical "AND" between the addressed byte and the accumulator, but does not alter the contents of the accumulator. The Z flag is set if the result of $A \wedge M$ is zero, otherwise Z is reset. In addition, bits 7 and 6 of addressed memory are copied to the processor status N and V flags, respectively.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Zero Page	BIT	Oper	24	2	3	C3.1
Absolute	BIT	Oper	2C	3	4	C6.1

BMI Branch on Minus

Operation: Branch on N = 1

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

BMI takes the conditional branch if the N flag is set (result is minus), otherwise the next sequential instruction is executed.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Relative	BMI	Dest	30	2	2[b]	C13.1

b. Add 1 if branch occurs to the same page; add 2 if branch occurs to a different page.

BNE Branch on Not Equal

Operation: Branch on Z = 0

N	V	–	B	I	D	Z	C
–	–	–	–	–	–	–	–

BNE takes the conditional branch if the Z flag is not set (result is non-zero), otherwise the next sequential instruction is executed. BNE is also referred to as "Branch on Not Zero".

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Relative	BNE	Dest	D0	2	2[b]	C13.1

b. Add 1 if branch occurs to the same page; add 2 if branch occurs to a different page.

BPL Branch on Plus

Operation: Branch on N = 0

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

BPL is the complementary conditional branch to BMI. BPL tests the N flag and takes the conditional branch if N is reset (result is positive), otherwise the next sequential instruction is executed.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Relative	BPL	Dest	10	2	2[b]	C13.1

b. Add 1 if branch occurs to the same page; add 2 if branch occurs to a different page.

BRA Branch Always

Operation: PC + IMM → PC

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

BRA is an unconditional, or forced, branch. The immediate field is added to the program counter value to form the new program counter value. This allows a short (2-byte) branch to an address +127 or –128 bytes from the current PC value (i.e., the address of next sequential instruction).

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Relative	BRA Dest	80	2	3[a]	C13.2

a. Add 1 if branch occurs to a different page.

BRK Break Command

Operation: Forced Interrupt Request: PC + 1 ↓, P ↓

N	V	–	B	D	I	Z	C
–	–	–	1	–	1	–	–

The BRK command causes the processor to go through an interrupt request sequence under program control. The address in the program counter (which points to the location of the **BRK command +1**) is pushed on the stack, along with the processor status at the beginning of the BRK instruction. The processor then transfers control to the **NMI interrupt vector (FFFC, FFFD)**. Note that the BRK command cannot be masked by setting the I flag.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	BRK	00	1	7	C14.9

BVC Branch on Overflow Clear

Operation: Branch on V = 0

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

BVC tests the status of the V flag and takes the conditional branch if V is not set (result did not overflow). otherwise the next sequential instruction is executed.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Relative	BVC Dest	50	2	2[b]	C13.1

b. Add 1 if branch occurs to the same page; add 2 if branch occurs to a different page.

BVS Branch on Overflow Set

Operation: Branch on V = 1

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

BVS tests the V flag and takes the conditional branch if V is set (result did overflow), otherwise the next sequential instruction is executed.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Relative	BVS Dest	70	2	2[b]	C13.1

b. Add 1 if branch occurs to the same page; add 2 if branch occurs to a different page.

CLC Clear Carry Flag

Operation: $0 \rightarrow C$

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	0

CLC initializes the C flag to a 0. This operation should normally precede an ADC loop. It is also useful when used with a ROL instruction to clear a bit in memory.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	CLC	18	1	2	C1.1

CLD Clear Decimal Mode

Operation: $0 \rightarrow D$

N	V	–	B	D	I	Z	C
–	–	–	–	0	–	–	–

CLD resets the D flag in the processor flag register to a 0 (binary mode). This causes all subsequent ADC, ADD and SBC instructions to operate as binary operations.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	CLD	D8	1	2	C1.1

CLI Clear Interrupt Disable Bit

Operation: $0 \rightarrow I$

N	V	–	B	D	I	Z	C
–	–	–	–	–	0	–	–

CLI resets the interrupt disable flag in the processor status register to a 0 (interrupt enabled). This enables the processor to act on an interrupt request (IRQ).

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	CLI	58	1	2	C1.1

CLV Clear Overflow Flag

Operation: $0 \rightarrow V$

N	V	–	B	D	I	Z	C
–	0	–	–	–	–	–	–

This instruction clears the V flag to a 0.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Relative	CLV	B8	1	2	C1.1

CLW Clear W Register and Overflow Flag

Operation: $0 \rightarrow W, V$

N	V	–	B	D	I	Z	C
–	0	–	–	–	–	–	–

CLW clears both halves of the W register and resets the V flag to a 0.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	CLW	52	1	2	C1.1

CMP Compare Memory and Accumulator

Operation: A – M

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	Ã

CMP subtracts the contents of addressed memory from the contents of the accumulator and set/resets the N, Z and C flags accordingly.

The Z flag is set on an equal comparison, otherwise Z is reset. The N flag is set if bit 7 of the result is a 1, otherwise N is reset. The C flag is set when the value in memory is less than or equal to the accumulator, reset when it is greater than the accumulator. The accumulator is not affected.

Result	N Flag	C Flag	Z Flag
Accumulator < Memory	Either	Reset	Reset
Accumulator = Memory	Reset	Set	Set
Accumulator > Memory	Either	Set	Reset

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Immediate	CMP	#Oper	C9	2	2	C2.1
Zero Page	CMP	Oper	C5	2	3	C3.1
Zero Page,X	CMP	Oper,X	D5	2	4	C4.1
Absolute	CMP	Oper	CD	3	4	C6.1
Absolute,X	CMP	Oper,X	DD	3	4[a]	C7.1
Absolute,Y	CMP	Oper,Y	D9	3	4[a]	C8.1
(Indirect)	CMP	(Oper)	C1	2	5	C9.1
(Indirect),X	CMP	(Oper),X	D1	2	5[a]	C10.1

a. Add 1 if a page boundary is crossed.

CPX Compare Memory and Index X

Operation: X – M

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	Ã

CPX subtracts the contents of the addressed memory location from the contents of index register X and sets/resets the N, Z, and C flags accordingly.

The C flag is set if the value in index register X is equal to, or greater than, the value in memory, otherwise C is cleared. The N flag is set if the result of the subtraction is negative (bit 7 is a 1), otherwise N is cleared. The Z flag is set if the value in index register X and the value in memory are equal, otherwise Z is reset.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Immediate	CPX	#Oper	E0	2	2	C2.1
Zero Page	CPX	Oper	E4	2	3	C3.1
Absolute	CPX	Oper	EC	3	4	C6.1

CPY Compare Memory and Index Y

Operation: Y – M

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	Ã

CPY subtracts the contents of the addressed memory location from the contents of index register Y and sets/resets the N, Z, and C flags accordingly.

The C flag is set if the value in index register Y is equal to, or greater than, the value in memory, otherwise C is cleared. The N flag is set if the result of the subtraction is negative (bit 7 is a 1), otherwise N is cleared. The Z flag is set if the value in index register Y and the value in memory are equal, otherwise Z is reset.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Immediate	CPY	#Oper	C0	2	2	C2.1
Zero Page	CPY	Oper	C4	2	3	C3.1
Absolute	CPY	Oper	CC	3	4	C6.1

DEC Decrement Memory by One

Operation: M – 1 → M

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	–

DEC subtracts 1 from the contents of the addressed memory location and stores the result in the addressed memory byte. A value of 0 will be decremented to $FF.

The N flag is set if bit 7 is on as a result of the decrement, otherwise N is reset. The Z flag is set if the result of the decrement is 0, otherwise Z is reset.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Zero Page	DEC	Oper	C6	2	5	C12.1
Zero Page,X	DEC	Oper,X	D6	2	6	C12.2
Absolute	DEC	Oper	CE	3	6	C12.3
Absolute,X	DEC	Oper,X	DE	3	7	C12.4

DEX Decrement Index X by One

Operation: $X - 1 \rightarrow X$

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	–

DEX subtracts one from the value in the X register and stores the result in the X register. The result does not affect or consider carry so that a value of 0 in index register X will be decremented to $FF.

The N flag is set if the X register contains bit 7 on as a result of the decrement, otherwise N is reset. The Z flag is set if the X register is zero as a result of the decrement, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	DEX	CA	1	2	C1.1

DEY Decrement Index Y by One

Operation: $Y - 1 \rightarrow Y$

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	–

DEY subtracts one from the current value in the Y register and stores the result in the Y register. The result does not affect or consider carry so that a value of 0 in index register Y will be decremented to $FF.

The N flag is set if the Y register contains bit 7 on as a result of the decrement, otherwise N is reset. The Z flag is set if the Y register is zero as a result of the decrement, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	DEY	88	1	2	C1.1

EOR "Exclusive-Or" Memory with Accumulator

Operation: $A \vee M \rightarrow A$

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	–

EOR performs a binary "EXCLUSIVE OR" on a bit-by-bit basis between the accumulator and the addressed location and stores the result in the accumulator.

The Z flag is set if the result in the accumulator is 0, otherwise Z is reset. The N flag is set if the result in the accumulator has bit 7 on, otherwise N is reset.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Immediate	EOR	#Oper	49	2	2	C2.1
Zero Page	EOR	Oper	45	2	3	C3.1
Zero Page,X	EOR	Oper,X	55	2	4	C4.1
Absolute	EOR	Oper	4D	3	4	C6.1
Absolute,X	EOR	Oper,X	5D	3	4[a]	C7.1
Absolute,Y	EOR	Oper,Y	59	3	4[a]	C8.1
(Indirect)	EOR	(Oper)	41	2	5	C9.1
(Indirect),X	EOR	(Oper),X	51	2	5[a]	C10.1

a. Add 1 if a page boundary is crossed.

EXC Exchange Accumulator and Memory

Operation: $A \rightarrow M, M \rightarrow A$

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

EXC exchanges the contents in the accumulator with the contents of addressed memory byte.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Zero Page,X	EXC	Oper,X	D4	2	5	C4.3

INC Increment Memory by One

Operation: M + 1 → M

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	–

INC adds 1 to the contents of the addressed memory location. A value of $FF is incremented to 0.

The N flag is set if bit 7 is on as the result of the increment, otherwise N is reset. If the increment causes a zero result, the Z flag is set, otherwise Z is reset.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Zero Page	INC	Oper	E6	2	5	C12.1
Zero Page,X	INC	Oper,X	F6	2	6	C12.2
Absolute	INC	Oper	EE	3	6	C12.3
Absolute,X	INC	Oper,X	FE	3	7	C12.4

INI Increment I by One

Operation: I + 1 → I

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

INI increments the contents of the I register by one.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	INI	BB	1	3	C17.1

INX Increment Index X by One

Operation: X + 1 → X

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	–

INX adds 1 to the value in the X register, storing the result in the X register. A value of $FF is incremented to 0.

The N flag is set if the result of the increment has a one in bit 7, otherwise N is reset. The Z flag is set if the result of the increment is 0, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	INX	E8	1	2	C1.1

INY Increment Index Y by One

Operation: Y + 1 → Y

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	–

INY adds 1 to the value in the Y register, storing the result in the Y register. A value of $FF is incremented to 0.

The N flag is set if the result has a 1 in bit 7, otherwise N is reset. The Z flag is set if the result is zero, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	INY	C8	1	2	C1.1

JMP Jump to New Location

Operation: See below.

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

The JMP instruction establishes a new value for the program counter. It affects only the program counter in the processor and affects no flags in the status register.

In the JMP Absolute instruction, the second and third bytes of the instruction are loaded into the PCL and PCH, respectively, to specify the location of the next instruction. The symbolic operation is (PC + 1) → PCL, (PC + 2) → PCH.

In the JMP (Indirect) instruction, the second and third bytes of the instruction represent the low and high bytes, respectively, of the memory location containing the effective ADL. Once the ADL is fetched, the program counter is incremented with the next memory location containing the ADH.

The JMP (Indirect,X) instruction operates like the JMP (Indirect) instruction except the contents of the X register are added to the indirect low byte of the effective memory location containing the effective ADL (a carry, generated if a page boundary is crossed, is added to the indirect high byte).

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Absolute	JMP	Oper	4C	3	3	C13.3
(Indirect)	JMP	(Oper)	6C	3	5	C13.4
(Indirect,X)	JMP	(Oper,X)	7C	3	6	C13.5

JPI Jump Indirect with Return in I

Operation: PC + 1 → IL, PC + 2 → IH, (I) ⟶ PC, I + 2 → I

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

JPI executes an indirect jump to subroutine, saving the return address, PC+3, in the I register. Bytes 2 and 3 of the instruction are the indirect jump address IAL, IAH. The effective jump address bytes (ADL, ADH) are fetched from memory at locations [IAH, IAL] and [IAH, IAL] + 1, respectively.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
(Indirect)	JPI	0C	3	5	C17.6

JSB Jump to Subroutine (8)

Operation: PC + 1 ↓ , (FFEn) → PCL, (FFEn + 1) → PCH, SP – 2 ⟶ SP

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

JSB is a 1-byte instruction that loads the program counter with a subroutine starting address and leaves a return pointer on the stack to allow the program to return to the next instruction in the main program upon subroutine completion. The JSB instruction stores the program counter address + 1, which points to the instruction following the JSB, onto the stack. The stack byte contains the program count high first, followed by program count low. Three bits of the op code specify one of eight 2-byte vectors in ROM located from $FFE0–$FFEE. The specified vector points to the subroutine starting address. The starting address is transferred to the program counter to direct program execution to continue at that address

JSB affects no processor status flags, causes the stack pointer to be decremented by 2 and substitutes new values into program counter low and program counter high.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Bit/FF Page				1	6	C14.8
(FFE0)	JSB	0	0B			
(FFE2)	JSB	1	1B			
(FFE4)	JSB	2	2B			
(FFE6)	JSB	3	3B			
(FFE8)	JSB	4	4B			
(FFEA)	JSB	5	5B			
(FFEC)	JSB	6	6B			
(FFEE)	JSB	7	7B			

JSR Jump to New Location Saving Return Address

Operation: PC + 3↓, (PC + 1)→ PCL, (PC + 2) → PCH, SP – 2→ SP

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

JSR loads the program counter with the a subroutine starting address and leaves a return pointer on the stack to allow the program to return to the next instruction in the main program upon subroutine completion. To accomplish this, the JSR instruction stores the program counter address + 3, which points to the instruction following the JSR, onto the stack. The stack byte contains the program count high first, followed by program count low. The JSR then transfers the address following the jump instruction to the program counter, thereby directing the program to begin at that new address.

The JSR instruction affects no flags, causes the stack pointer to be decremented by 2 and substitutes new values into the program counter low and the program counter high.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Absolute	JSR Oper	20	3	5	C14.8

LAB Load Absolute to Accumulator

Operation: |A| → A

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	–

LAB loads the absolute value of the contents in the accumulator to the accumulator. If the original contents of the accumulator are negative, LAB performs a twos complement. Exception: An accumulator value of $80 will produce the result $80.

The Z flag is set if the result is zero, otherwise Z is reset. The N flag is set if bit 7 of the result is a 1, otherwise N is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Accumulator	LAB A	13	1	3	C11.2

LAI Load Accumulator Indirect through I

Operation: (I) → A

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

LAI loads the byte pointed to by the I register into the accumulator.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	LAI	EB	1	3	C17.2

LAN Load A Indirect and Increment I

Operation: (I) → A, I + 1 → I

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

LAN loads the byte pointed to by the I register into the accumulator. The I register is incremented by 1.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	LAN	AB	1	3	C17.3

LDA Load Accumulator with Memory

Operation: M → A

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	–

LDA copies the contents of addressed memory to the accumulator.

The Z flag is set if the accumulator is zero as a result of the LDA, otherwise Z is reset. The N flag is set if bit 7 of the accumulator is a 1, otherwise N is reset.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Immediate	LDA	#Oper	A9	2	2	C2.1
Zero Page	LDA	Oper	A5	2	3	C3.1
Zero Page,X	LDA	Oper,X	B5	2	4	C4.1
Absolute	LDA	Oper	AD	3	4	C6.1
Absolute,X	LDA	Oper,X	BD	3	4[a]	C7.1
Absolute,Y	LDA	Oper,Y	B9	3	4[a]	C8.1
(Indirect)	LDA	(Oper)	A1	2	5	C9.1
(Indirect),X	LDA	(Oper),X	B1	2	5[a]	C10.1

a. Add 1 if a page boundary is crossed.

LDX Load Index X with Memory

Operation: M → X

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	–

LDX copies the contents of addressed memory to index register X.

The N flag is set if bit 7 of the loaded value is a 1, otherwise N is reset. The Z flag is set if the value loaded is zero, otherwise Z is reset.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Immediate	LDX	#Oper	A2	2	2	C2.1
Zero Page	LDX	Oper	A6	2	3	C3.1
Zero Page,Y	LDX	Oper,Y	B6	2	4	C5.1
Absolute	LDX	Oper	AE	3	4	C6.1
Absolute,Y	LDX	Oper,Y	BE	3	4[a]	C8.1

a. Add 1 if a page boundary is crossed.

LDY Load Index Y with Memory

Operation: M → Y

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	–

LDY copies the contents of memory to index register Y.

The N flag is set if bit 7 of the loaded value is a 1, otherwise N is reset. The Z flag is set if the loaded value is zero, otherwise Z is reset. The other processor status flags and processor registers are unaffected.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Immediate	LDY	#Oper	A0	2	2	C2.1
Zero Page	LDY	Oper	A4	2	3	C3.1
Zero Page,X	LDY	Oper,X	B4	2	4	C4.1
Absolute	LDY	Oper	AC	3	4	C6.1
Absolute,X	LDY	Oper,X	BC	3	4[a]	C7.1

a. Add 1 if a page boundary is crossed.

LII Load I Indirect through I

Operation: (I) → I

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

LII loads the two bytes at the memory address pointed to by the I register into the I register.

No processor status flags or other processor registers are affected.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	LII	9B	1	5	C17.5

LSR Logical Shift Right One Bit (Memory or Accumulator)

Operation: $0 \rightarrow 7\ 6\ 5\ 4\ 3\ 2\ 1\ 0 \rightarrow C$

N	V	–	B	D	I	Z	C
0	–	–	–	–	–	Ã	Ã

LSR shifts the contents of the accumulator, or of a specified memory location, one bit to the right, with the high bit of the result always being set to 0, and the shifted low bit being stored in the C flag.

The N flag is always reset to a 0. The Z flag is set if the result of the shift is 0, otherwise Z is reset. The C flag is set equal to bit 0 of the input.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Accumulator	LSR	A	4A	1	2	C11.1
Zero Page	LSR	Oper	46	2	5	C12.1
Zero Page,X	LSR	Oper,X	56	2	6	C12.2
Absolute	LSR	Oper	4E	3	6	C12.3
Absolute,X	LSR	Oper,X	5E	3	7	C12.4

MPA Multiply and Accumulate

Operation: $A \times Y + W \rightarrow W$

N	V	–	B	D	I	Z	C
Ã	Ã	–	–	–	–	–	–

MPA performs a signed multiply of the value in the accumulator by the value in the Y register, adds the result of the multiplication to the value in the W register, then stores the accumulated result in the W register. The accumulator contents are not altered, however, the contents of the Y register are destroyed.

The N flag is set if the result of the accumulation in the W register is negative (e.g., bit 15 is a 1); otherwise N is reset. The V flag is set if bit 15 of the accumulated result changes due to the value exceeding +32767 ($7FFF) or –32768 ($8000), otherwise V is reset. In case of overflow, the most positive value ($7FFF) or most negative value ($8000) is left in the W register depending on overflow polarity.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	MPA	12	1	6	C1.2

MPY Multiply

Operation: A x Y → A, Y

N	V	–	B	D	I	Z	C
Ã	0	–	–	–	–	–	–

MPY performs a signed multiply of the value in the accumulator by the value in the Y register. The upper half of the result is stored in the accumulator and the lower half of the result is stored in Y register. The original contents of the A and Y registers are destroyed.

The V flag is reset. The N flag copies the sign of the result, accumulator bit 7.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	MPY	02	1	6	C1.2

NEG Negate Accumulator

Operation: ~A → A

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	–

NEG performs a twos complement of the contents in the accumulator and stores the result in the accumulator. The value $80 will produce the result $80.

The N flag is set if bit 7 of the result in the accumulator is a 1, otherwise N is reset. The Z flag is set if the result in the accumulator is zero, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Accumulator	NEG A	1A	1	2	C11.1

NOP No Operation

Operation: PC + 1 → PC

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

NOP performs no operation except incrementing the program counter by 1.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	NOP	EA	1	2	C1.1

NXT Next Instruction

Operation: (I) → PC, I + 2 → I

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

NXT loads the two bytes at the address pointed to by the I register into the program counter and increments the I register by 2. The low order address byte is loaded first.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	NXT	8B	1	4	C17.4

ORA "OR" Memory with Accumulator

Operation: A∨ M → A

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	–

The ORA instruction performs a binary "OR" on a bit-by-bit basis between the accumulator and addressed memory and stores the result in the accumulator.

The Z flag is set if the result in the accumulator is 0, otherwise Z is reset. The N flag is set if the result in the accumulator has bit 7 on, otherwise N is reset.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Immediate	ORA	#Oper	09	2	2	C2.1
Zero Page	ORA	Oper	05	2	3	C3.1
Zero Page,X	ORA	Oper,X	15	2	4	C4.1
Absolute	ORA	Oper	0D	3	4	C6.1
Absolute,X	ORA	Oper,X	1D	3	4[a]	C7.1
Absolute,Y	ORA	Oper,Y	19	3	5[a]	C8.1
(Indirect)	ORA	(Oper)	01	2	6	C9.1
Indirect),X	ORA	(Oper),X	11	2	5[a]	C10.1

a. Add 1 if a page boundary is crossed.

PHA Push Accumulator on Stack

Operation: A ↓, SP–1 → SP

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

PHA copies the current value of the accumulator to the stack and decrements the stack pointer by 1 to point to the next available stack location.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PHA	48	1	3	C14.1

PHI Push I on Stack

Operation: IH ↓, SP–1→ SP, IL ↓, SP–1 → SP

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

PHI copies the contents of the I register to the stack, high byte first, and decrements the stack pointer by 2 to point to the next available stack location.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PHI	CB	1	4	C14.2

PHP Push Processor Status on Stack

Operation: P ↓, SP–1 → SP

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

PHP copies the contents of the processor status register, unchanged, to the stack and decrements the stack pointer by 1 to point to the next available stack location.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PHP	08	1	3	C14.1

PHW Push W on Stack

Operation: WH ↓, SP–1 → SP, WL ↓, SP–1 → SP

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

PHW copies the contents of the W register to the stack, high byte first, and decrements the stack pointer by 2 to point to the next available stack location.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PHW	23	1	4	C14.2

PHX Push Index X on Stack

Operation: X ↓, SP–1 → SP

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

PHX copies the contents of index register X to the stack and decrements the stack pointer by 1 to point to the next available stack location.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PHX	DA	1	3	C14.1

PHY Push Index Y on Stack

Operation: Y ↓, SP–1 → SP

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

PHY copies the contents of index register Y to the stack and decrements the stack pointer by 1 to point to the next available stack location.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PHY	5A	1	3	C14.1

PIA Pull I from Stack, Load Accumulator, Increment I

Operation: SP + 1 → SP, IL ↑, SP + 1 → SP, IH ↑,
(I) → A, (I) → X, I +1 → I

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	–

PIA first loads the I register with two bytes from the stack, low byte first. The stack pointer is incremented by 1 before each byte is copied. Next, the byte pointed to by the I register is then loaded into the accumulator and to the X register, and then the I register is incremented by 1.

The N flag is set if bit 7 of the accumulator is 1 as a result of the instruction, otherwise N is reset. The Z flag is set if the accumulator value is zero, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PIA	FB	1	6	C14.6

PLA Pull Accumulator from Stack

Operation: SP + 1 → SP, A ↑

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	–

PLA increments the stack pointer by 1 then loads the accumulator with the byte from the stack.

The N flag is set if bit 7 of the loaded value is a 1, otherwise N is reset. If the accumulator is zero as a result of the PLA, the Z flag is set, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PLA	68	1	4	C14.4

PLI Pull I from Stack

Operation: SP + 1 → SP, IL ↑, SP + 1 → SP, IH ↑

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

PLI loads the I register with two bytes from the stack, low byte first. The stack pointer is incremented by 1 before each byte is copied.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PLI	DB	1	6	C14.6

PLP Pull Processor Status from Stack

Operation: SP + 1 → SP, P ↑

N	V	–	B	D	I	Z	C
Ã	Ã	–	–	Ã	Ã	Ã	Ã

PLP increments the stack pointer by 1 then copies the byte in the stack to the processor status register.

Six bits in the processor status register are replaced.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PLP	28	1	4	C14.4

PLW Pull W from Stack

Operation: SP + 1 → SP, WL ↑, SP + 1 → SP, WH ↑

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

PLW loads the W register with two bytes from the stack, low byte first. The stack pointer is incremented by 1 before each byte is transferred.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PLW	33	1	5	C14.5

PLX Pull Index X from Stack

Operation: SP + 1 → SP, X ↑

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	–

PLX adds 1 to the stack pointer then copies the byte in the stack to the X register.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PLX	FA	1	4	C14.4

PLY Pull Index Y from Stack

Operation: SP + 1 → SP, Y ↑

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	–

PLY adds 1 to the stack pointer then copies the contents of the byte in the stack pointed to by the stack pointer into the Y register.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PLY	7A	1	4	C14.4

PSH Push A, X and Y on Stack

Operation: A ↓, SP–1 → SP, X ↓, SP–1 → SP, Y ↓, SP–1 → SP

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

PSH copies the contents of the accumulator, X register and Y register to the stack in this order. The stack pointer is decremented by 1 after each byte is transferred.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PSH	22	1	5	C14.3

PUL Pull Y, X and A from Stack

Operation: SP + 1 → SP, Y ↑, SP + 1 → SP, X ↑, SP + 1 → SP, A ↑

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

PUL pulls the contents of three bytes from the stack to the Y register, X register and accumulator in this order. The stack pointer is incremented by one before each byte is transferred.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	PUL	32	1	6	C14.7

RBA Reset Bit(s) in Memory

Operation: ~Mask ∧ M → M

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

RBA resets to a "0", in the addressed memory location, all bits set to a 1 in a mask. Byte 2 of the instruction contains the mask and bytes 3 and 4 of the instruction specify the absolute address of the operand.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Bit/Absolute	RBA	Mask,Addr	C2	4	7	C15.2

RMB Reset Memory Bit (8)

Operation: $0 \rightarrow M_b$ (b = 0 –7)

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

RMB resets to "0" one of eight bits in an addressed memory or I/O port location. The first byte of the instruction specifies the RMB operation code and 1 of 8 bits (bit 0 to bit 7) to be reset. The second byte of the instruction designates the zero page address of the byte to be operated upon.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Bit/Zero Page				2	5	C15.1
Bit 0	RMB	0,Oper	07			
Bit 1	RMB	1,Oper	17			
Bit 2	RMB	2,Oper	27			
Bit 3	RMB	3,Oper	37			
Bit 4	RMB	4,Oper	47			
Bit 5	RMB	5,Oper	57			
Bit 6	RMB	6,Oper	67			
Bit 7	RMB	7,Oper	77			

RND Round

Operation: WH + WL7 → A

N	V	–	B	D	I	Z	C
Ã	Ã	–	–	–	–	–	–

The most significant bit of WL is added to WH and the result is stored in the accumulator. The maximum positive value stored in the accumulator is 127 ($7F) while the maximum negative value stored is –128 ($80). W remains unchanged.

The N flag is set if bit 7 in the accumulator is a 1, otherwise N is reset. If a RND is attempted when WH = $7F and WL7 = 1, then $7F is loaded in the accumulator, the V flag is set, and the contents of the Y register are destroyed; otherwise V is reset and Y remains unchanged.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	RND	42	1	2	C1.1

ROL Rotate Left One Bit (Memory or Accumulator)

Operation: C ← 7 6 5 4 3 2 1 0 ← C

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	Ã

ROL shifts either the accumulator or addressed memory left one bit, with the input C flag being stored in bit 0 and with the input bit 7 being stored in the C flag.

The C flag is set equal to the input bit 7. The N flag is set equal to the result bit 7. The Z flag is set if the result in memory or the accumulator is 0, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Accumulator	ROL A 2A	1	2	C11.1	
Zero Page	ROL Oper	26	2	5	C12.1
Zero Page,X	ROL Oper,X	36	2	6	C12.2
Absolute	ROL Oper	2E	3	6	C12.3
Absolute,X	ROL Oper,X	3E	3	7	C12.4

ROR Rotate Right One Bit (Memory or Accumulator)

Operation: C → 7 6 5 4 3 2 1 0 → C

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	Ã

ROR shifts either the accumulator or addressed memory right one bit, with the input C flag shifted into bit 7 and the input bit 0 shifted into the C flag.

The C flag is set to input bit 0 and the N flag is set equal to the input C flag. The Z flag is set if the result in memory or the accumulator is 0; otherwise Z is reset.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Accumulator	ROR	A	6A	1	2	C11.1
Zero Page	ROR	Oper	66	2	5	C12.1
Zero Page,X	ROR	Oper,X	76	2	6	C12.2
Absolute	ROR	Oper	6E	3	6	C12.3
Absolute,X	ROR	Oper,X	7E	3	7	C12.4

RTI Return from Interrupt

Operation: SP + 1 → SP, P ↑, SP + 1 → SP, PCL ↑,
SP + 1 → SP, PCH ↑

N	V	–	B	D	I	Z	C
Ã	Ã	–	–	Ã	Ã	Ã	Ã

RTI transfers from the stack the processor status and the program counter location for the instruction which was interrupted. By virtue of the interrupt having stored this data before executing the instruction, and due to the fact that the RTI reinitializes the processor to the same state as when it was interrupted, the combination of interrupt plus RTI allows truly reentrant coding.

The RTI instruction reinitializes 6 flags to the state which they were when the interrupt was taken, and sets the program counter back to its pre-interrupt state. RTI affects no other registers in the processor.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	RTI	40	1	6	C14.11

RTS Return from Subroutine

Operation: SP + 1 → SP, PCL ↑, SP + 1 → SP, PCH ↑

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

RTS loads the program counter with two bytes from the stack, low byte first. The stack pointer is incremented by 2.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	RTS	60	1	5	C14.10

SBA Set Bit(s) in Memory

Operation: Mask ∨ M → M

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

SBA sets to a "1" in the addressed memory location all bits set to a 1 in a mask. Byte 2 of the instruction contains the mask and bytes 3 and 4 of the instruction specify the absolute address of the operand. SBA logically ORs the mask with the operand value to form the new operand value.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Bit/Absolute	SBA Mask,Addr	D2	4	7	C15.2

SBC Subtract Memory from Accumulator with Borrow

Operation: A – M – ~C → A (~C = Borrow)

N	V	–	B	D	I	Z	C
Ã	Ã	–	–	–	–	Ã	Ã

SBC subtracts the value in the addressed memory location and borrow (i.e., the complement of the C flag) from the value in the accumulator and stores the result in the accumulator.

The N flag is set if bit 7 of the result is a 1, otherwise N is reset. The Z flag is set if the result is zero, otherwise Z is reset. The C flag is set if the result is greater than, or equal to, zero; otherwise C is reset. In the binary mode, the V flag is set if the result exceeds +127 or –128, otherwise V is reset. In the decimal mode, V is reset.

No other processor status flags or processor registers are affected.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Immediate	SBC	#Oper	E9	2	2[c]	C2.1
Zero Page	SBC	Oper	E5	2	3[c]	C3.1
Zero Page,X	SBC	Oper,X	F5	2	4[c]	C4.1
Absolute	SBC	Oper	ED	3	4[c]	C6.1
Absolute,X	SBC	Oper,X	FD	3	4[a,c]	C7.1
Absolute,Y	SBC	Oper,Y	F9	3	4[a,c]	C8.1
(Indirect)	SBC	(Oper)	E1	2	5[c]	C9.1
(Indirect),X	SBC	(Oper),X	F1	3	5[a,c]	C10.1

a. Add 1 when page boundary is crossed.

c. Add 1 if in decimal mode.

SEC Set Carry Flag

Operation: 1 → C

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	1

SEC initializes the C flag to a 1. This operation should normally precede a SBC loop.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	SEC	38	1	2	C1.1

SED Set Decimal Mode

Operation: $1 \rightarrow D$

N	V	–	B	D	I	Z	C
–	–	–	–	1	–	–	–

SED sets the D flag to a 1 (decimal mode). This makes all subsequent ADC, ADD, and SBC instructions operate as a decimal arithmetic operation (until a CLD resets the D flag to a 0).

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	SED	F8	1	2	C1.1

SEI Set Interrupt Disable Status

Operation: $1 \rightarrow I$

N	V	–	B	D	I	Z	C
–	–	–	–	–	1	–	–

SEI sets the I flag to a 1 (disable interrupt).

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	SEI	78	1	2	C1.1

SMB Set Memory Bit (8)

Operation: $1 \rightarrow M_b$ (b = 0 - 7)

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

SMB sets to "1" one of the bits in an 8-bit data field specified by the zero page address (memory or I/O port). The first byte of the instruction specifies the SMB operation and 1 of 8 bits to be set. The second byte of the instruction designates the zero page address of the byte or I/O port to be operated upon.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Bit/Zero Page			2	5	C15.1
Bit 0	SMB 0,Oper	87			
Bit 1	SMB 1,Oper	97			
Bit 2	SMB 2,Oper	A7			
Bit 3	SMB 3,Oper	B7			
Bit 4	SMB 4,Oper	C7			
Bit 5	SMB 5,Oper	D7			
Bit 6	SMB 6,Oper	E7			
Bit 7	SMB 7,Oper	F7			

STA Store Accumulator in Memory

Operation: A → M

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

STA copies the contents of the accumulator to addressed memory.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Zero Page	STA	Oper	85	2	3	C3.2
Zero Page,X	STA	Oper,X	95	2	4	C4.2
Absolute	STA	Oper	8D	3	4	C6.2
Absolute,X	STA	Oper,X	9D	3	5	C7.2
Absolute,Y	STA	Oper,Y	99	3	5	C8.2
(Indirect)	STA	(Oper)	81	2	5	C9.2
(Indirect),X	STA	(Oper),X	91	2	6	C10.2

STI Store Immediate to Memory

Operation: Oper → M

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

STI stores the contents of the immediate field (byte 2 of the instruction) to the zero page address specified by byte 3 of the instruction.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Zero Page	STI	#Oper1, Oper2	B2	3	4	C2.2

STX Store Index X in Memory

Operation: X → M

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

STX copies the contents of the X register to the addressed memory location.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Zero Page	STX	Oper	86	2	3	C3.2
Zero Page,Y	STX	Oper,Y	96	2	4	C5.2
Absolute	STX	Oper	8E	3	4	C6.2

STY Store Index Y in Memory

Operation: Y → M

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

STY copies the contents of the Y register to the addressed memory location.

Addressing Mode	Assembly Language Form		Op Code	No. Bytes	No. Cycles	Ref.
Zero Page	STY	Oper	84	2	3	C3.2
Zero Page,X	STY	Oper,X	94	2	4	C4.2
Absolute	STY	Oper	8C	3	4	C6.2

TAW Transfer Accumulator to W

Operation: A → WH, 0 → WL

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	–

TAW copies the contents of the accumulator to the upper half of the W register and stores zero in the lower half of the W register.

The N flag copies bit 7 of the WH register. The Z flag is set if the WH register is 0, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	TAW	62	1	2	C1.1

TAX Transfer Accumulator to Index X

Operation: A → X

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	–

TAX copies the contents of accumulator to the index register X without disturbing the contents of the accumulator.

The N flag copies bit 7 of the X register. The Z flag is set if the X register is 0, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	TAX	AA	1	2	C1.1

TAY Transfer Accumulator to Index Y

Operation: A → Y

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	–

TAY copies the contents of the accumulator to index register Y without altering the contents of the accumulator.

The N flag copies bit 7 of the Y register. The Z flag is set if the Y register is 0, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	TAY	A8	1	2	C1.1

TIP Transfer I to Program Counter

Operation: I → PC

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

TIP copies the contents of the I register to the program counter. This instruction functions as a return from subroutine (RTS) to the JPI instruction.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	TIP	03	1	2	C1.1

TSX Transfer Stack Pointer to Index X

Operation: S → X

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	–

TSX transfers the value in the stack pointer to the index register X.

The N flag copies bit 7 of the X register. The Z flag is set if the X register is zero, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	TSX	BA	1	2	C1.1

TWA Transfer W to Accumulator

Operation: WH → A

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	–

TWA copies the contents of the upper half of the W register to the accumulator.

The N flag copies bit 7 of the accumulator. The Z flag is set if the accumulator is 0, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	TWA	72	1	2	C1.1

TXA Transfer Index X to Accumulator

Operation: X → A

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	–

TXA copies the value in the index register X to the accumulator without disturbing the contents of the index register X.

The N flag copies bit 7 of the accumulator. The Z flag is set if the accumulator is 0, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	TXA	8A	1	2	C1.1

TXS Transfer Index X to Stack Pointer

Operation: X → S

N	V	–	B	D	I	Z	C
–	–	–	–	–	–	–	–

TXS copies the value in the index register X to the stack pointer. TXS changes only the stack pointer, making it equal to the contents of index register X.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	TXS	9A	1	2	C1.1

TYA Transfer Index Y to Accumulator

Operation: Y → A

N	V	–	B	D	I	Z	C
Ã	–	–	–	–	–	Ã	–

TYA copies the value in index register Y to the accumulator without disturbing the contents of the Y register.

The N flag copies bit 7 of the accumulator. The Z flag is set if the accumulator is 0, otherwise Z is reset.

Addressing Mode	Assembly Language Form	Op Code	No. Bytes	No. Cycles	Ref.
Implied	TYA	98	1	2	C1.1

C. CPU INSTRUCTION ADDRESSING MODES

This appendix describes the addressing modes for the CPU instructions. The following attributes are provided for each addressing mode:

One number of bytes and execution time in clock cycles.

The instructions using the addressing mode.

The format of the instruction.

The instruction operation by clock cycle.

The instruction formats are:

Addressing Mode	No. Bytes	Op Code	Operand	Operand Description
Absolute	3	XXX	nn	nn = 2-byte absolute address
Absolute,X	3	XXX	nn	nn = 2-byte absolute address
Absolute,Y	3	XXX	nn	nn = 2-byte absolute address
Accumulator	1	XXX		
Bit/FF Page (JSB)	1	XXX	b	b = bit 0 –7
Bit/Zero Page (BBR/BBS)	3	XXX	b,n,r	b = bit 0–7, n = 1-byte zero page address, r = 1-byte relative offset
Bit/Zero Page (RMB/SMB)	2	XXX	b,n	b = bit 0–7, n = 1-byte zero page address
Bit/Absolute (RBA/SBA)	4	XXX	bm,nn	bm = byte mask, nn = 2-byte absolute address
Bit/Absolute (BAR/BAS)	5	XXX	nn,bm,rb	bm = byte mask, nn = 2-byte absolute address, r = 1-byte relative offset
Immediate	2	XXX	#n	n = 1-byte constant
Implied	1	XXX		
Indirect Absolute	3	XXX	(nn)	nn = 2-byte absolute address
(Indirect)	2	XXX	(n)	n = 1-byte zero page address of addrLH
(Indirect),X	2	XXX	(n),X	n = 1-byte zero page address of addrLH
Relative	2	XXX	r	r = 1-byte relative offset
Zero Page	2	XXX	n	n = 1-byte zero page address
Zero Page ,X	2	XXX	n,X	n = 1-byte zero page address
Zero Page ,Y	2	XXX	n,Y	n = 1-byte zero page address

C1 IMPLIED ADDRESSING

C1.1 General – 1 Byte, 2 Cycles

Instructions: CLC, CLD, CLI, CLW, CLV, DEX, DEY, INX, INY, NOP, RND, SEC, SED, SEI, TAW, TAX, TAY, TIP, TSX, TWA, TXA, TXS, and TYA

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Data (Ignored)	Interpret OP CODE
3	2201	Next OP CODE	Fetch Next OP CODE	Execute instruction, Increment PC to 2202

C1.2 Multiply – 1 Byte, 6 Cycles

Instruction: MPA and MPY

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Data (Ignored)	Interpret OP CODE
3	2201	Next OP CODE	Fetch Data (Ignored)	Execute multiply steps
4	2201	Next OP CODE	Fetch Data (Ignored)	
5	2201	Next OP CODE	Fetch Data (Ignored)	
6	2201	Next OP CODE	Fetch Data (Ignored)	
7	2201	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2202

C2 IMMEDIATE ADDRESSING

C2.1 Read – 2 Bytes, 2 Cycles

Instructions: ADC, ADD, AND, CMP, CPX, CPY, EOR, LDA, LDX, LDY, ORA, and SBC

Format: XXX #n n= 1-byte constant

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Fetch Data	Data	Increment PC to 2202
3	2202	Next OP CODE	Fetch Next OP CODE	Execute instruction, Increment PC to 2203

C2.2 Write – 2 Bytes, 4 Cycles

Instruction: STI

Format: XXX #n n= 1-byte constant

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Fetch Data	Data	Increment PC to 2202
3	2202	Fetch ADL	ADL	Increment PC to 2203
4	00, ADL	Data	Write Data	Hold PC
5	2203	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2204

C3 ZERO PAGE ADDRESSING

C3.1 Read – 2 Bytes, 3 Cycles

Instructions: ADC, ADD, AND, BIT, CMP, CPX, CPY, EOR, LDA, LDX, LDY, ORA, and SBC

Format: XXX n n = zero page address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	ADL	Fetch ADL	Interpret OP CODE, Increment PC to 2202
3	00, ADL	Data	Fetch Data	Hold PC
4	2202	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2203

C3.2 Write – 2 Bytes, 3 Cycles

Instructions: STA, STX, and STY.

Format: XXX n n = zero page address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	ADL	Fetch ADL	Interpret OP CODE, Increment PC to 2202
3	00, ADL	Data	Write Data	Hold PC
4	2202	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2203

C4 ZERO PAGE,X ADDRESSING

C4.1 Read – 2 Bytes, 4 Cycles

Instructions: ADC, ADD, AND, CMP, EOR, LDA, LDY, ORA, and SBC

Format: XXX n,X n = 1-byte zero page address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	BAL	Fetch BAL	Interpret OP CODE, Increment PC to 2202
3	00, BAL	Data (Ignored)	Fetch Data (Ignored)	Add BAL+X, Hold PC
4	00, BAL+X	Data	Fetch Data	Hold PC
5	2202	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2203

C4.2 Write – 2 Bytes, 4 Cycles

Instructions: STA and STY

Format: XXX n,X n = 1-byte zero page address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	BAL	Fetch BAL	Interpret OP CODE, Increment PC to 2202
3	00, BAL	Data (Ignored)	Fetch Data (Ignored)	Add BAL+X, Hold PC
4	00, BAL+X	Data	Write Data	Hold PC
5	2202	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2203

C4.3 Write – 2 Bytes, 5 Cycles

Instructions: EXC

Format: XXX n,X n = 1-byte zero page address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	BAL	Fetch BAL	Interpret OP CODE, Increment PC to 2202
3	00, BAL	Data (Ignored)	Fetch Data (Ignored)	Add BAL+X, Hold PC
4	00, BAL+X	Data	Fetch Data	Hold PC
5	00, BAL+X	Data	Write Data	Hold PC
6	2202	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2203

C5 ZERO PAGE,Y ADDRESSING

C5.1 Read – 2 Bytes, 4 Cycles

Instruction: LDX

Format: XXX n,Y n = 1-byte zero page address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	BAL	Fetch BAL	Interpret OP CODE, Increment PC to 2202
3	00, BAL	Data (Ignored)	Fetch Data (Ignored)	Add BAL+Y, Hold PC
4	00, BAL+Y	Data	Fetch Data	Hold PC
5	2202	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2203

C5.2 Write – 2 Bytes, 4 Cycles

Instruction: STX

Format: XXX n,Y n = 1-byte zero page address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	BAL	Fetch BAL	Interpret OP CODE, Increment PC to 2202
3	00, BAL	Data (Ignored)	Fetch Data (Ignored)	Add BAL+Y, Hold PC
4	00, BAL+Y	Data	Write Data	Hold PC
5	2202	Next OP CODE	Fetch Next OP CODE	Finish Instruction, Increment PC to 2203

C6 ABSOLUTE ADDRESSING

C6.1 Read – 3 Bytes, 4 Cycles

Instructions: ADC, AND, BIT, CMP, CPX, CPY, EOR, LDA, LDX, LDY, ORA, and SBC

Format: XXX nn nn = 2-byte absolute address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	ADL	Fetch ADL	Interpret OP CODE, Increment PC to 2202
3	2202	ADH	Fetch ADH	Hold ADL, Increment PC to 2203
4	ADH, ADL	Data	Fetch Data	Hold PC
5	2203	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2203

C6.2 Write – 3 Bytes, 4 Cycles

Instructions: STA, STX, and STY

Format: XXX nn nn = 2-byte absolute address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	ADL	Fetch ADL	Interpret OP CODE, Increment PC to 2202
3	2202	ADH	Fetch ADH	Hold ADL, Increment PC to 2203
4	ADH, ADL	Data	Write Data	Hold PC
5	2203	Next OP CODE	Fetch Next OP CODE	Finish Instruction, Increment PC to 2203

C7 ABSOLUTE,X ADDRESSING

C7.1 Read

Instructions: ADC, AND, CMP, EOR, LDA, LDY, ORA, and SBC

Format: XXX nn,X nn = 2-byte absolute address

C7.1a No Page Crossing – 3 Bytes, 4 Cycles

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	BAL	Fetch BAL	Interpret OP CODE, Increment PC to 2202
3	2202	BAH	Fetch BAH	Add BAH+X, Increment PC to 2203
4	BAH, BAL+X	Data	Fetch Data	Hold PC
5	2203	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2204

C7.1b Page Crossing – 3 Bytes, 5 Cycles

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	BAL	Fetch BAL	Interpret OP CODE, Increment PC to 2202
3	2202	BAH	Fetch BAH	Add BAL+X, Increment PC to 2203
4	BAH, BAL+X	Data (Ignored)	Fetch Data (Ignored)	Add BAH+Carry (1), Hold PC
5	BAH+C, BAL+X	Data	Fetch Data	Hold PC
6	2203	Next OP CODE	Fetch Next OP CODE	Finish instruction Increment PC to 2204

C7.2 Write – 3 Bytes, 5 Cycles

Instruction: STA

Format: XXX nn,X nn = 2-byte absolute address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	BAL	Fetch BAL	Interpret OP CODE, Increment PC to 2202
3	2202	BAH	Fetch BAH	Add BAL+X, Increment PC to 2203
4	BAH, BAL+X	Data (Ignored)	Fetch Data (Ignored)	Add BAH+Carry (0 or 1), Hold PC
5	BAH+C, BAL+X	Data	Write Data	Hold PC
6	2203	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2204

C8 ABSOLUTE,Y ADDRESSING

C8.1 Read

Instructions: ADC, AND, CMP, EOR, LDA, LDX, ORA, and SBC

Format: XXX nn,Y nn = 2-byte absolute address

C8.1a No Page Crossing – 3 Bytes, 4 Cycles

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	BAL	Fetch BAL	Interpret OP CODE, Increment PC to 2202,
3	2202	BAH	Fetch BAH	Add BAL+Y, Increment PC to 2203
4	BAH, BAL+Y	Data	Data	Hold PC
5	2203	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2204

C8.1b Page Crossing – 3 Bytes, 5 Cycles

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	BAL	Fetch BAL	Interpret OP CODE, Increment PC to 2202
3	2202	BAH	Fetch BAH	Add BAL+Y, Increment PC to 2203
4	BAH, BAL+Y	Data (Ignored)	Fetch Data (Ignored)	Add BAH+Carry (1), Hold PC
5	BAH+C, BAL+Y	Data	Fetch Data	Hold PC
6	2203	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2204

C8.2 Write – 3 Bytes, 5 Cycles

Instruction: STA

Format: XXX nn,Y nn = 2-byte absolute address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	BAL	Fetch BAL	Interpret OP CODE, Increment PC to 2202
3	2202	BAH	Fetch BAH	Add BAL+Y, Increment PC to 2203
4	BAH, BAL+Y	Data (Ignored)	Fetch Data (Ignored)	Add BAH+Carry (0 or 1), Hold PC
5	BAH+C, BAL+Y	Data	Write Data	Hold PC
6	2203	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2204

C9 INDIRECT ADDRESSING

C9.1 Read - 2 Bytes, 5 Cycles

Instructions: ADC, AND, CMP, EOR, LDA, ORA, and SBC

Format: XXX (n) n = Zero page address of absolute address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	IAL	Fetch IAL	Interpret OP CODE, Increment PC to 2202
3	00, IAL	BAL	Fetch BAL	Add 1 to IAL, Hold PC
4	00, IAL+1	BAH	Fetch BAH	Hold BAL, Hold PC
5	BAH, BAL	Data	Fetch Data	Hold PC
6	2202	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2203

C9.2 Write – 2 Bytes, 5 Cycles

Instruction: STA

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Load OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	IAL	Fetch IAL	Interpret OP CODE, Increment PC to 2202
3	00, IAL	BAL	Fetch BAL	Add 1 to IAL, Hold PC
4	00, IAL+1	BAH	Fetch BAH	Hold BAL, Hold PC
5	BAH, BAL	Data	Write Data	Hold PC
6	2202	Next OP CODE	Fetch Next OP CODE	Increment PC to 2203

C10 INDIRECT,X ADDRESSING

C10.1 Read

Instructions: ADC, AND, CMP, EOR, LDA, ORA, and SBC

Format: XXX (n),X n = Zero page address of absolute address

C10.1a No Page Crossing – 2 Bytes, 5 Cycles

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	IAL	Fetch IAL	Interpret OP CODE, Increment PC to 2202
3	00, IAL	BAL	Fetch BAL	Add 1 to IAL, Hold PC
4	00, IAL+1	BAH	Fetch BAH	Add BAL+X, Hold PC
5	BAH, BAL+X	Data	Fetch Data	Hold PC
6	2202	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2203

C10.1b Page Crossing – 2 Bytes, 6 Cycles

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Load OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	IAL	Fetch IAL	Interpret OP CODE, Increment PC to 2202
3	00, IAL	BAL	Fetch BAL	Add 1 to IAL, Hold PC
4	00, IAL+1	BAH	Fetch BAH	Add BAL to X, Hold PC
5	BAH, BAL+X	Data (Ignored)	Fetch Data (Ignored)	Add BAH+Carry (1), Hold PC
6	BAH+C, BAL+X	Data	Fetch Data	Hold PC
7	2202	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2203

C10.2 Write – 2 Bytes, 6 Cycles

Instruction: STA

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Load OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	IAL	Fetch IAL	Interpret OP CODE, Increment PC to 2202
3	00, IAL	BAL	Fetch BAL	Add 1 to IAL, Hold PC
4	00, IAL+1	BAH	Fetch BAH	Add BAL to X, Hold PC
5	BAH, BAL+X	Data (Ignored)	Fetch Data (Ignored)	Add BAH+Carry (0 or 1) Hold PC
6	BAH+C, BAL+X	Data	Write Data	Hold PC
7	2202	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2203

C11 ACCUMULATOR ADDRESSING

C11.1 1 Byte, 2 Cycles

Instructions: ASL, ASR, LSR, ROL, ROR, and NEG

Format: XXX

Operation: ROL example

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE (ROL)	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch OP CODE (Ignored)	Interpret OP CODE (ROL), Hold PC
3	2201	Next OP CODE	Fetch Next OP CODE	Shift through the Adder, Increment PC to 2202
4	2202	?	Fetch Second Byte	Store result into A, Interpret next OP CODE

C11.2 1 Byte, 3 Cycles

Instruction: LAB

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Data (Ignored)	Decode OP CODE, Hold PC
3	2201	Next OP CODE	Fetch Data (Ignored)	
4	2201	Next OP CODE	Fetch Next OP CODE	Increment PC to 2202
5	2202	?	Fetch Second Byte	Store result into A, Interpret next OP CODE

C12 READ/MODIFY/WRITE

C12.1 Zero Page Addressing – 2 Bytes, 5 Cycles

Instructions: ASL, DEC, Increment, LSR, ROL, and ROR

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE (ROL)	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	ADL	Fetch ADL	Decode OP CODE (ROL), Increment PC to 2202
3	00, ADL	Data	Fetch Data	Hold PC
4	00, ADL	Data	Fetch Data (Ignored)	Perform rotate, Hold PC
5	00, ADL	Shifted Data	Write Data	Set flags, Hold PC
6	2202	OP CODE	Fetch Next OP CODE	Increment PC to 2203

C12.2 Zero Page,X Addressing – 2 Bytes, 6 Cycles

Instructions: ASL, DEC, Increment, LSR, ROL, and ROR

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE (ROL)	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	BAL	Fetch BAL	Decode OP CODE (ROL), Increment PC to 2202
3	00, BAL	Data (Ignored)	Fetch Data (Ignored)	Add BAH+X, Hold PC
4	00, BAL+X	Data	Fetch Data	Hold PC
5	00, BAL+X	Data	Fetch Data (Ignored)	Perform rotate, Hold PC
6	00, BAL+X	Shifted Data	Write Data	Set flags, Hold PC
7	2202	OP CODE	Fetch Next OP CODE	Increment PC to 2203

C12.3 Absolute Addressing – 3 Bytes, 6 Cycles

Instructions: ASL, DEC, Increment, LSR, ROL, and ROR

Format: XXX nn nn = 2-byte absolute address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	ADL	Fetch ADL	Decode OP CODE (ROL), Increment PC to 2202
3	2202	ADH	Fetch ADH	Hold ADL, Increment PC to 2203
4	ADH, ADL	Data	Fetch Data	Hold PC
5	ADH, ADL	Data	Fetch Data (Ignored)	Perform rotate, Hold PC
6	ADH, ADL	Shifted Data	Write Data	Set flags, Hold PC
7	2203	OP CODE	Fetch Next OP CODE	Increment PC to 2204

C12.4 Absolute,X Addressing – 3 Bytes, 7 Cycles

Instructions: ASL, DEC, Increment, LSR, ROL, and ROR

Format: XXX nn,X nn = 2-byte absolute address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE (ROL)	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	BAL	Fetch BAL	Decode OP CODE (ROL), Increment PC to 2202
3	2202	BAH	Fetch BAH	Add BAL+X, Increment PC to 2203
4	BAH, BAL+X	Data (Ignored)	Fetch Data (Ignored)	Add BAH+Carry (0 or 1), Hold PC
5	BAH+C, BAL+X	Data	Fetch Data	Hold PC
6	BAH+C, BAL+X	Data	Fetch Data (Ignored)	Perform rotate, Hold PC
7	BAH+C, BAL+X	Shifted Data	Write Data	Set Flags, Hold PC
8	2203	OP CODE	Fetch Next OP CODE	Increment PC to 2204

C13 BRANCH/JUMP

C13.1 Conditional Branch

Instructions: BCC, BCS, BEQ, BMI, BNE, BPL, BVC, and BVS

Format: XXX r r = 1-byte relative offset

C13.1a Branch Not Taken – 2 Bytes, 2 Cycles

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Offset	Fetch Offset	Interpret OP CODE, Set previous instruction flags, Increment PC to 2202
3	2202	Next OP CODE	Fetch Next OP CODE	Check flags, Increment PC to 2203

C13.1b Branch Taken with Positive Offset, No Page Boundary Crossing – 2 Bytes, 3 Cycles

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Offset (+50)	Fetch Offset	Interpret OP CODE, Set previous instruction flags, Increment PC to 2202
3	2202	Next OP CODE	Fetch Next OP CODE	Check flags, Add Offset to PCL, Increment PC to 2203
4	2252	Next OP CODE	Fetch Next OP CODE	Transfer results to PCL, Increment PC to 2253

C13.1c Branch Taken with Negative Offset, Page Boundary Crossed – 2 Bytes, 4 Cycles

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Offset (–50)	Fetch Offset	Interpret OP CODE, Set previous instruction flags, Increment PC to 2202
3	2202	Next OP CODE	Fetch Next OP CODE	Check flags, Add offset to PCL
4	22B2	Data	Fetch Data (Ignored)	Store Adder result in PCL and subtract 1 from PCH
5	21B2	Next OP CODE	Fetch Next OP CODE	Put out new PCH, Increment PC to 21B3

C13.2 Unconditional Branch Crossing – 2 Bytes, 3 Cycles with Positive Offset, No Page Boundary

Instruction: BRA

Format: XXX r r = 1-byte relative offset

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Offset (+40)	Fetch Offset	Interpret OP CODE, Increment PC to 2202
3	2202	Next OP CODE	Fetch Next OP CODE	Add offset to PC, Increment PC to 2203
4	2242	Next OP CODE	Fetch Next OP CODE	Increment PC to 2243

C13.3 Jump Absolute – 3 Bytes, 3 Cycles

Instruction: JMP

Format: XXX nn nn = 2-byte absolute address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	ADL	Fetch ADL	Interpret OP CODE, Increment PC to 2202
3	2202	ADH	Fetch ADH	
4	ADH, ADL	Next OP CODE	Fetch Next OP CODE	ADH, ADL to PC, Increment PC

C13.4 Jump (Indirect,X) – 3 bytes, 6 cycles

Instruction: JMP

Format: XXX (nn,X) nn = 2-byte absolute address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	IAL	Fetch IAL	Interpret OP CODE, Increment PC to 2202
3	2202	IAH	Fetch IAH	Hold IAL
4	IAH, IAL+X	Data	Fetch Data (Ignored)	Add IAH + carry
5	IAH+C, IAL+X	ADL	Fetch ADL	Add 1 to [IAH, IAL]
6	[IAH+C, IAL+X]+1	ADH	Fetch ADH	
7	ADH, ADL	Next OP CODE	Fetch Next OP CODE	ADH, ADL to PC, Increment PC

C13.5 JMP (Indirect) – 3 Bytes, 5 Cycles

Instruction: JMP

Format: XXX (nn) nn = 2-byte absolute address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	IAL	Fetch IAL	Interpret OP CODE, Increment PC to 2202
3	2202	IAH	Fetch IAH	Hold IAL
4	IAH, IAL	ADL	Fetch ADL	Add 1 to [IAH, IAL]
5	[IAH, IAL]+1	ADH	Fetch ADH	
6	ADH, ADL	Next OP CODE	Fetch Next OP CODE	ADH, ADL to PC,

C14 STACK

C14.1 Push 1 Byte – 1 byte, 3 Cycles

Instructions: PHA, PHP, PHX, and PHY

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE (PHA)	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Next OP CODE (Ignored)	Interpret OP CODE (PHA), Hold PC
3	01FF	(A)	Write A to Stack	Decrement SP to 01FE
4	2201	Next OP CODE	Fetch Next OP CODE	Increment PC to 2203

C14.2 Push 2 Bytes – 1 Byte, 4 Cycles

Instructions: PHI and PHW

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE (PHI)	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Next OP CODE (Ignored)	Interpret OP CODE (PHI), Hold PC
3	01FF	(IH)	Write IH to Stack	Decrement SP to 01FE
4	01FE	(IL)	Write IL to Stack	Decrement SP to 01FD
5	2201	Next OP CODE	Fetch Next OP CODE	Increment PC to 2203

C14.3 Push 3 Bytes – 1 Byte, 5 Cycles

Instruction: PSH

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE (PSH)	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Next OP CODE (Ignored)	Interpret OP CODE (PSH), Hold PC
3	01FF	(A)	Write A to Stack	Decrement SP to 01FE
4	01FE	(X)	Write X to Stack	Decrement SP to 01FD
5	01FD	(Y)	Write Y to Stack	Decrement SP to 01FC
6	2201	Next OP CODE	Fetch Next OP CODE	Increment PC to 2202

C14.4 Pull 1 Byte – 1 Byte, 4 Cycles

Instructions: PLA, PLP, PLX, and PLY

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Next OP CODE and Discard	Interpret OP CODE, Hold PC
3	01FE	Discarded Data	Read Stack	Increment SP to 01FF
4	01FF	Data	Fetch Data	Save Stack
5	2201	Next OP CODE	Fetch Next OP CODE	Data to A

C14.5 Pull 2 Bytes – 1 Byte, 5 Cycles

Instruction: PLW

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Next OP CODE and Discard	Interpret OP CODE, Hold PC
3	01FD	Data	Fetch Data (Ignored)	Increment SP to 01FE
4	01FE	WL	Fetch Data from Stack	Increment SP to 01FF
5	01FF	WH	Fetch Data from Stack	Data to WL, Save Stack
6	2201	Next OP CODE	Fetch Next OP CODE	Data to WH

C14.6 Pull 2 Bytes – 1 Byte, 6 Cycles

Instructions: PLI and PIA

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish Previous Instruction
2	2201	Next OP CODE	Fetch Data (Ignored)	Interpret OP CODE, Hold PC
3	01FD	Data	Fetch Data (Ignored)	Increment SP to 01FE
4	01FE	IL	Fetch Data	Increment SP to 01FF
5	01FF	IH	Fetch Data	Data to IL
6	IH, IL	Data	Fetch Data	Data to IH, Save Stack, Increment I if PIA
7	2201	Next OP CODE	Fetch New OP CODE	Data to A if PIA

C14.7 Pull 3 Bytes – 1 Byte, 6 Cycles

Instructions: PUL

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE (PUL)	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Next OP CODE and Discard	Interpret OP CODE (PUL), Hold PC
3	01FC	Data	Fetch Data (Ignored)	Increment SP to 01FD
4	01FD	Y	Fetch Data from Stack	Increment SP to 01FE
5	01FE	X	Fetch Data from Stack	Data to Y, Increment SP to 01FF
6	01FF	A	Fetch Data from Stack	Data to X
7	2201	Next OP CODE	Fetch Next OP CODE	Data to A

C14.8 Jump to Subroutine – 3 Bytes, 5 Cycles

Instruction: JSR

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE(JSR)	Finish previous instruction, Increment PC to 2201
2	2201	New ADL	Fetch ADL	Decode OP CODE (JSR), Increment PC to 2202
3	2202	New ADH	Fetch ADH	Hold ADL
4	01FF	PCH	Write PCH to Stack	Hold ADL and ADH, Decrement SP to 01FE
5	01FE	PCL	Write PCL to Stack	Hold ADL and ADH, Decrement SP to 01FD
6	ADH, ADL	New OP CODE	Fetch New OP CODE	ADL to PCL, ADH to PCH

C14.9 Break – 1 Byte, 7 Cycles

Instruction: BRK

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Data (Ignored)	Force BRK instruction, Increment PC to 2202
3	01FF	PCH	Store PCH on Stack	Decrement SP to 01FE
4	01FE	PCL	Store PCL on Stack	Decrement SP to 01FD
5	01FD	P	Store P on Stack	Decrement SP to 01FC
6	FFF0	New PCL	Fetch Vector Low	Put Away Stack
7	FFF1	New PCH	Fetch Vector High	Vector Low
8	PCH, PCL	OP CODE	Fetch Interrupt Program	Increment PC to PC + 1

C14.10 Return from Subroutine – 1 Byte, 5 Cycles

Instruction: RTS

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	3300	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 3301
2	3301	Data	Fetch Data (Ignored)	Decode OP CODE (RTS)
3	3301	Data	Fetch Data (Ignored)	Increment SP to 01FE
4	01FE	PCL (02)	Fetch PCL	Increment SP to 01FF, Hold PCL
5	01FF	PCH (22)	Fetch PCH	
6	2202	Next OP CODE	Fetch Next OP CODE	Increment PC to 2203

C14.11 Return From Interrupt – 1 Byte, 6 Cycles

Instruction: RTI

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	3300	OP CODE (RTI)	Fetch OP CODE	Finish previous instruction, Increment PC to 3301
2	3301	Data	Fetch Data (Ignored)	Decode OP CODE (RTI)
3	3301	Data	Fetch Data (Ignored)	Increment SP to 01FD
4	01FD	P	Fetch P from Stock	Increment SP to 01FE
5	01FE	PCL (02)	Fetch PCL from Stack	Increment SP to 01FF, Hold PCL
6	01FF	PCH (22)	Fetch PCH from Stack	M to PCL,
Store SP				
7	2202	OP CODE	Fetch OP CODE	Increment New PC

C15 BIT

C15.1 Reset/Set Memory Bit – 2 Bytes, 5 Cycles

Instructions: RMB0 – RMB7, and SMB0 – SMB7

Format: XXX n n = 1-byte zero page address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	ADL	Fetch ADL	Interpret OP CODE, Increment PC to 2202
3	00, ADL	Data	Fetch Data	Hold PC
4	00, ADL	Data	Fetch Data (Ignored)	Reset (RMB)/set (SMB) bit, Hold PC
5	00, ADL	Modified Data	Write Modified Data	Hold PC
6	2202	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2203

C15.2 Reset/Set Bits in Memory – 4 Bytes, 7 Cycles

Instructions: RBA and SBA

Format: XXX bm,nn bm = 1-byte bit mask, nn = 2-byte absolute address

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	BM	Fetch BM	Interpret OP CODE, Increment PC to 2202
3	2202	ADL	Fetch ADL	Hold BM in Accumulator* Increment PC to 2203
4	2203	ADH	Fetch ADH	Hold ADL, Increment PC to 2204
5	ADH, ADL	Data	Fetch Data	Hold PC
6	ADH, ADL	Data	Fetch Data (Ignored)	Reset(RBA)/Set (SBA) bit, Hold PC
7	ADH, ADL	Modified Data	Write Modified Data	Hold PC
8	2204	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2205

* Note that Accumulator contents are altered by RBA and SBA.

C15.3 Branch on Bit Reset/Set, Branch Not Taken – 3 Bytes, 5 Cycles

Instructions: BBR0 – BBR7, and BBS0 – BBS7

Format: XXX n,r n = 1-byte zero page address, r = offset

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	ADL	Fetch ADL	Interpret OP CODE, Increment PC to 2202
3	00, ADL	Data	Fetch Data	Hold PC
4	00, ADL	Data	Fetch Data (Ignored)	Test the bit, Hold PC
5	2202	Offset	Fetch Branch Offset	Increment PC to 2203
6	2203	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2204

C15.4 Branch on Bit(s) Reset/Set, Branch Not Taken – 5 Bytes, 7 Cycles

Instructions: BAR and BAS

Format: XXX nn, bm, r nn = 2-byte absolute address, bm = 1-byte bit mask,
r = offset

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	ADL	Fetch ADL	Interpret OP CODE, Increment PC to 2202
3	2202	ADH	Fetch ADH	Hold ADL, Increment PC to 2203
4	ADH, ADL	Data	Fetch Data	Hold PC
5	2203	BM	Fetch BM	Hold ADL, ADH Hold PC
6	2203	BM	Fetch BM (Ignored)	Test bits, Increment PC to 2204
7	2204	Offset	Fetch Branch Offset	Increment PC to 2205
8	2205	Next OP CODE	Fetch Next OP CODE	Finish Instruction, Increment PC to 2206

C16 JSB – 1 Byte, 6 Cycles

Instructions: JSB0 – JSB7

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE (JSB0)	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Data (Ignored)	Interpret OP CODE (JSB0), Hold PC
3	01FF	PCH	Write PCH to Stack	Decrement SP to 01FE
4	01FE	PCL	Write PCL to Stack	Decrement SP to 01FD
5	FFE0	New PCL	Fetch PCL	Save SP
6	FFE1	New PCH	Fetch PCH	Hold PCL
7	PCH, PCL	First OP CODE	Fetch First OP CODE	Increment PC to PC+1

C17 THREADED CODE

C17.1 INI – 1 Byte, 3 Cycles

Instruction: INI

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Data (Ignored)	Interpret OP CODE, Hold PC
3	IH, IL	Data	Fetch Data (Ignored)	Increment I
4	2201	Next OP CODE	Fetch Next OP CODE	Finish instruction, Increment PC to 2202

C17.2 LAI – 1 Byte, 3 Cycles

Instruction: LAI

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Data (Ignored)	Interpret OP CODE, Hold PC
3	IH, IL	Data	Fetch Data	
4	2201	Next OP CODE	Fetch Next OP CODE	Data to A, Increment PC to 2202

C17.3 LAN – 1 Byte, 3 Cycles

Instruction: LAN

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Data (Ignored)	Interpret OP CODE, Hold PC
3	IH, IL	Data	Fetch Data	Increment I
4	2201	Next OP CODE	Fetch Next OP CODE	Finish instruction, Data to A Increment PC to 2202

C17.4 NXT – 1 Byte, 4 Cycles

Instruction: NXT

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Fetch Data (Ignored)	Interpret OP CODE
3	IH, IL	New PCL	Fetch New PCL	
4	[IH, IL] + 1	New PCH	Fetch New PCH	Hold PCL
5	PCH, PCL	Next OP CODE	Fetch New OP CODE	Finish instruction, Increment PC to PC + 1

C17.5 LII – 1 Byte, 5 Cycles

Instruction: LII

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	Next OP CODE	Ignore	Interpret OP CODE, Hold PC
3	IH, IL	New IL	Fetch IL	
4	[IH, IL] + 1	New IH	Fetch IH	Hold IL
5	New IH, IL	Data	Fetch Data (Ignored)	New IH, IL → I
6	2201	Next OP CODE	Fetch New OP CODE	Increment PC to 2202

C17.6 JPI – 3 Bytes, 5 Cycles

Instruction: JPI

Format: XXX

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Increment PC to 2201
2	2201	IAL	Fetch IAL	Interpret OP CODE, Increment PC to 2202
3	2202	IAH	Fetch IAH	Save IAL,
4	IAH, IAL	ADL	Fetch ADL	[IAH, IAL] + 1
5	[IAH, IAL] + 1	ADH	Fetch ADH	Save ADL
6	ADH, ADL	Next OP CODE	Fetch New OP CODE	Finish instruction, Increment PC to [ADH, ADL] + 1

C18 START AND INTERRUPT SEQUENCES

C18.1 Start Sequence (Power-On)

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	?	?	Fetch Data (Ignored)	Hold During Reset
2	? + 1	?	Fetch Data (Ignored)	First Start State
3	01, SP	?	Fetch Data (Ignored)	Second Start State
4	01, SP–1	?	Fetch Data (Ignored)	Third Start State
5	01, SP–2	?	Fetch Data (Ignored)	Fourth Start State
6	FFFE	PCL	Fetch PCL	
7	FFFF	PCH	Fetch PCH	Hold PCL
8	PCH, PCL	First OP CODE	Fetch First OP CODE	Increment PC to [PCH, PCL] + 1

C18.2 Interrupt Sequence (NMI) *

Operation:

Cycle	Address Bus	Data Bus	Bus Operation	CPU Operation
1	2200	OP CODE	Fetch OP CODE	Finish previous instruction, Hold PC
2	2200	OP CODE	Fetch OP CODE	Force a BRK Instruction, Hold PC
3	01FF	PCH	Store PCH on Stack	Decrement SP to 01FE
4	01FE	PCL	Store PCL on Stack	Decrement SP to 01FD
5	01FD	P	Store P on Stack	Decrement SP to 01FC
6	FFFC	PCL	Fetch PCL	
7	FFFD	PCH	Fetch PCH	Hold PCL
8	PCH, PCL	Next OP CODE	Fetch Interrupt Program First OP CODE	Increment PC to PC + 1

* All other interrupts are identical except for different vector locations at cycles 6 and 7.

This page is intentionally blank.

INSIDE BACK COVER NOTES

Headquarters
Rockwell Semiconductor Systems
4311 Jamboree Road,
P.O. Box C
Newport Beach, CA 92658-8902
Phone: (714) 221-4600
Fax: (714) 221-6375

European Headquarters
Rockwell Semiconductor Systems
S.A.R.L.
Les Taissounieres B1
Route des Dolines
Sophia Antipolis Cedex
06905 Valbonne
France
Phone: (33) 93 00 33 35
Fax: (33) 93 00 33 03

For more information:
Call 1-800-854-8099
International information:
Call 1-714-833-6996

URL Address:
http://www.nb.rockwell.com
E-Mail Address:
literature@nb.rockwell.com

REGIONAL SALES OFFICES

US Southwest Office
Rockwell Semiconductor Systems
5000 Birch Street
Suite 400
Newport Beach, CA 92660
Phone: (714) 222-9119
Fax: (714) 222-0620

US Southwest Satellite Office
Rockwell Semiconductor Systems
1000 Business Center Circle
Suite 215
Thousand Oaks, CA 91320
Phone: (805) 376-0559
Fax: (805) 376-8180

US South Central Office
Rockwell Semiconductor Systems
2001 North Collins Blvd
Suite 103
Richardson, TX 75080
Phone: (214) 379-9310
Fax: (214) 479-9317

US Southeast Office
Rockwell Semiconductor Systems
900 Ashwood Parkway
Suite 400
Atlanta, GA 30338
Phone: (770) 393-1830
Fax: (770) 395-1419

US Southeast Satellite Office
Rockwell Semiconductor Systems
One Prestige Place
2600 McCormick Drive
Suite 350
Clearwater, FL 34619
Phone: (813) 799-8406
Fax: (813) 799-8306

US Northwest Office
Rockwell Semiconductor Systems
US Northwest Office
3600 Pruneridge Avenue
Suite 100
Santa Clara, CA 95051
Phone: (408) 249-9696
Fax: (408) 249-7113

US North Central Office
Rockwell Semiconductor Systems
Two Pierce Place
Chancellory Park
Suite 810
Itasca, IL 60143
Phone: (708) 773-3454
Fax: (708) 773-3907

US Northeast Office
Rockwell Semiconductor Systems
239 Littleton Road
Suite 4A
Westford, MA 01886
Phone: (508) 692-7660
Fax: (508) 692-8185

Australia
Rockwell Semiconductor Systems
Rockwell Australia Pty Limited
Suite 603, 51 Rawson Street
Epping, NSW 2121
Australia
Phone: (61-2) 9869 4088
Fax: (61-2) 9869 4077

Europe Mediterranean
Rockwell Semiconductor Systems
c/o Rockwell Automation S.r.l.
Via Di Vittorio, 1
20017 Mazzo Di Rho (MI)
Italy
Phone: (39 2) 93179911
Fax (39 2) 93179913

Europe North
Rockwell Semiconductor Systems, Ltd.
Berkshire Court
Western Road
Bracknell
Berkshire RG12 1RE
England
Phone: +44 1344 486 444
Fax: +44 1344 486 555

Europe South
Rockwell Semiconductor Systems
S.A.R.L.
Tour GAN
Cedex 13
92082 Paris La Defense 2
France
Phone: (33-1) 49-06-3980
Fax: (33-1) 49-06-3990

Germany
Rockwell Semiconductor Systems
Rockwell Int'l GmbH Germany
Paul-Gerhardt-Allee 50 a
81245 Munchen
Germany
Phone: (49-89) 829-1320
Fax: (49-89) 834-2734

Hong Kong
Rockwell Int'l (Asia Pacific) Ltd.
13th Floor, Suites 8-10,
Harbour Centre
25 Harbour Road
Wanchai,
Hong Kong
Phone: (852) 2 827-0181
Fax: (852) 2 827-6488

Japan
Rockwell Int'l Japan Co., Ltd.
Shimomoto Bldg
1-46-3 Hatsudai, Shibuya-ku
Tokyo, 151
Japan
Phone: (81-3) 5371 1520
Fax: (81-3) 5371 1501

Korea
Rockwell-Collins Int'l, Inc.
Room No. 1508
Korea Textile Centre Building
944-31, Daechi-3dong
Kangnam P.O. Box 2037
Kangnam-ku
Seoul
Korea
Phone: (82-2) 565-2880
Fax: (82-2) 565-1440

Singapore
Rockwell Semiconductor Systems
Singapore Branch
1 Kim Seng Promenade
#09-01 Great World City East Tower
Singapore 237994
Phone: (65) 737-7355
Fax: (65) 737-9077

Taiwan
Rockwell Int'l Taiwan Company, Ltd.
Room 2808 International Trade Bldg.
333, Keelung Road, Section I
Taipei,
Taiwan
10548 ROC
Phone: (886-2) 720-0282
Fax: (886-2) 757-6760

Rockwell
Semiconductor Systems