

APPENDIX A

SUMMARY OF SINGLE CYCLE EXECUTION

This section contains an outline of the data on both the address bus and the data bus for each cycle of the various processor instructions. It tells the system designer exactly what to expect while single cycling through a program.

Note that the processor will not stop in any cycle where R/W is a 0 (write cycle). Instead, it will go right into the next read cycle and stop there. For this reason, some instructions may appear to be shorter than indicated here.

All instructions begin with T0 and the fetch of the OP CODE and continue through the required number of cycles until the next T0 and the fetch of the next OP CODE.

While the basic terminology used in this appendix is discussed in the Programming Manual, it has been defined below for ease of reference while studying Single Cycle Execution.

OP CODE--The first byte of the instruction containing the operator and mode of address.

OPERAND--The data on which the operation specified in the OP CODE is performed.

BASE ADDRESS--The address in Indexed addressing modes which specifies the location in memory to which indexing is referenced. The high order of byte of the base address (AB08 to AB15) is BAH (Base Address High) and the low order byte of the base address (AB00 to AB07) is BAL (Base Address Low).

EFFECTIVE ADDRESS--The destination in memory in which data is to be found. The effective address may be loaded directly as in the case of Page Zero and Absolute Addressing or may be calculated as in Indexing operations. The high order byte of the effective address (AB08 to AB15) is ADH and the low order byte of the effective address (AB00 to AB07) is ADL.

INDIRECT ADDRESS--The address found in the operand of instructions utilizing (Indirect), Y which contains the low order byte of the base address. IAH and IAL represent the high and low order bytes.

JUMP ADDRESS--The value to be loaded into Program Counter as a result of a Jump instruction.

A. 1. SINGLE BYTE INSTRUCTIONS

ASL	DEX	NOP	TAX	TYA
CLC	DEY	ROL	TAY	
CLD	INX	SEC	TSX	
CLI	INY	SED	TXA	
CLV	LSR	SEI	TXS	

These single byte instructions require two cycles to execute. During the second cycle the address of the next instruction in program sequence will be placed on the address bus. However, the OP CODE which appears on the data bus during the second cycle will be ignored. This same instruction will be fetched on the following cycle at which time it will be decoded and executed. The ASL, ROL and LSR instructions apply to the accumulator mode of address.

<u>Tn</u>	<u>Address Bus</u>	<u>Data Bus</u>	<u>R/W</u>	<u>Comments</u>
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	OP CODE (Discarded)	1	
T0	PC + 1	OP CODE	1	Next Instruction

A. 2. INTERNAL EXECUTION ON MEMORY DATA

ADC	CMP	EOR	LDY
AND	CPX	LDA	ORA
BIT	CPY	LDX	SBC

The instructions listed above will execute by performing operations inside the microprocessor using data fetched from the effective address. This total operation requires three steps. The first step (one cycle) is the OP CODE fetch. The second (zero to four cycles) is the calculation of an effective address. The final step is the fetching of the data from the effective address. Execution of the instruction takes place during the fetching and decoding of the next instruction.

A. 2.1. Immediate Addressing (2 cycles)

<u>Tn</u>	<u>Address Bus</u>	<u>Data Bus</u>	<u>R/W</u>	<u>Comments</u>
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	Data	1	Fetch Data
T0	PC + 2	OP CODE	1	Next Instruction

A. 2.2. Zero Page Addressing (3 cycles)

<u>Tn</u>	<u>Address Bus</u>	<u>Data Bus</u>	<u>R/W</u>	<u>Comments</u>
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	ADL	1	Fetch Effective Address
T2	00, ADL	Data	1	Fetch Data
T0	PC + 2	OP CODE	1	Next Instruction

A. 2.3. Absolute Addressing (4 cycles)

<u>Tn</u>	<u>Address Bus</u>	<u>Data Bus</u>	<u>R/W</u>	<u>Comments</u>
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	ADL	1	Fetch low order Effective Address byte
T2	PC + 2	ADH	1	Fetch high order Effective Address byte
T3	ADH, ADL	Data	1	Fetch Data
T0	PC + 3	OP CODE	1	Next Instruction

A. 2.4. Indirect, X Addressing (6 cycles)

<u>Tn</u>	<u>Address Bus</u>	<u>Data Bus</u>	<u>R/W</u>	<u>Comments</u>
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	BAL	1	Fetch Page Zero Base Address
T2	00, BAL	Data (Discarded)	1	
T3	00, BAL + X	ADL	1	Fetch low order byte of Effective Address
T4	00, BAL + X + 1	ADH	1	Fetch high order byte of Effective Address
T5	ADH, ADL	Data	1	Fetch Data
T0	PC + 2	OP CODE	1	Next Instruction

A. 2.5. Absolute, X or Absolute, Y Addressing (4 or 5 cycles)

<u>Tn</u>	<u>Address Bus</u>	<u>Data Bus</u>	<u>R/W</u>	<u>Comments</u>
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	BAL	1	Fetch low order byte of Base Address
T2	PC + 2	BAH	1	Fetch high order byte of Base Address
T3	ADL: BAL + index register ADH: BAH + C	Data*	1	Fetch data (no page crossing) Carry is \emptyset or 1 as required from previous add operation
T4*	ADL: BAL + index register ADH: BAH + 1	Data	1	Fetch data from next page
T0	PC + 3	OP CODE	1	Next Instruction

*If the page boundary is crossed in the indexing operation, the data fetched in T3 is ignored. If page boundary is not crossed, the T4 cycle is bypassed.

A. 2.6. Zero Page, X or Zero Page, Y Addressing Modes (4 cycles)

<u>Tn</u>	<u>Address Bus</u>	<u>Data Bus</u>	<u>R/W</u>	<u>Comments</u>
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	BAL	1	Fetch Page Zero Base Address
T2	00, BAL	Data (Discarded)	1	
T3	00, BAL + index register	Data	1	Fetch Data (no page crossing)
T0	PC + 2	OP CODE	1	Next Instruction

A. 2.7. Indirect, Y Addressing Mode (5 or 6 cycles)

<u>Tn</u>	<u>Address Bus</u>	<u>Data Bus</u>	<u>R/W</u>	<u>Comments</u>
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	IAL	1	Fetch Page Zero Indirect Address
T2	00, IAL	BAL	1	Fetch low order byte of Base Address
T3	00, IAL + 1	BAH	1	Fetch high order byte of Base Address
T4	ADL: BAL + Y ADH: BAH + C	Data*	1	Fetch Data from same page Carry is 0 or 1 as required from previous add operation
T5*	ADL: BAL + Y ADH: BAH + 1	Data	1	Fetch Data from next page
T0	PC + 2	OP CODE	1	Next Instruction

*If page boundary is crossed in indexing operation, the data fetch in T4 is ignored. If page boundary is not crossed, the T5 cycle is bypassed.

A. 3. STORE OPERATIONS

STA
STX
STY

The specific steps taken in the Store Operations are very similar to those taken in the previous group (Internal execution on memory data). However, in the Store Operation, the fetch of data is replaced by a WRITE (R/W = 0) cycle. No overlapping occurs and no shortening of the instruction time occurs on indexing operations.

A. 3.1. Zero Page Addressing (3 cycles)

<u>Tn</u>	<u>Address Bus</u>	<u>Data Bus</u>	<u>R/W</u>	<u>Comments</u>
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	ADL	1	Fetch Zero Page Effective Address
T2	00, ADL	Data	0	Write internal register to memory
T0	PC + 2	OP CODE	1	Next Instruction

A. 3.2. Absolute Addressing (4 cycles)

<u>Tn</u>	<u>Address Bus</u>	<u>Data Bus</u>	<u>R/W</u>	<u>Comments</u>
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	ADL	1	Fetch low order byte of Effective Address
T2	PC + 2	ADH	1	Fetch high order byte of Effective Address
T3	ADH, ADL	Data	0	Write internal register to memory
T0	PC + 3	OP CODE	1	Next Instruction

A. 3.3. Indirect, X Addressing (6 cycles)

<u>Tn</u>	<u>Address Bus</u>	<u>Data Bus</u>	<u>R/W</u>	<u>Comments</u>
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	BAL	1	Fetch Page Zero Base Address
T2	00, BAL	Data (Discarded)	1	
T3	00, BAL + X	ADL	1	Fetch low order byte of Effective Address
T4	00, BAL + X + 1	ADH	1	Fetch high order byte of Effective Address
T5	ADH, ADL	Data	0	Write internal register to memory
T0	PC + 2	OP CODE	1	Next Instruction

A. 3.4. Absolute, X or Absolute, Y Addressing (5 cycles)

<u>Tn</u>	<u>Address Bus</u>	<u>Data Bus</u>	<u>R/W</u>	<u>Comments</u>
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	BAL	1	Fetch low order byte of Base Address
T2	PC + 2	BAH	1	Fetch high order byte of Base Address
T3	ADL: BAL + index register ADH: BAH + C	Data (Discarded)	1	
T4	ADH, ADL	Data	0	Write internal register to memory
T0	PC + 3	OP CODE	1	Next Instruction

A. 3.5. Zero Page, X or Zero Page, Y Addressing Modes (4 cycles)

<u>Tn</u>	<u>Address Bus</u>	<u>Data Bus</u>	<u>R/W</u>	<u>Comments</u>
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	BAL	1	Fetch Page Zero Base Address
T2	00, BAL	Data (Discarded)	1	
T3	ADL: BAL + index register	Data	0	Write internal register to memory
T0	PC + 2	OP CODE	1	Next Instruction

A. 3.6. Indirect, Y Addressing Mode (6 cycles)

<u>Tn</u>	<u>Address Bus</u>	<u>Data Bus</u>	<u>R/W</u>	<u>Comments</u>
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	IAL	1	Fetch Page Zero Indirect Address
T2	00, IAL	BAL	1	Fetch low order byte of Base Address
T3	00, IAL + 1	BAH	1	Fetch high order byte of Base Address
T4	ADL: BAL + Y ADH: BAH	Data (Discarded)	1	
T5	ADH, ADL	Data	0	Write Internal Register to memory
T0	PC + 2	OP CODE	1	Next Instruction

A. 4. READ--MODIFY--WRITE OPERATIONS

ASL	LSR
DEC	ROL
INC	ROR

The Read--Modify--Write operations involve the loading of operands from the operand address, modification of the operand and the resulting modified data being stored in the original location.

Note: The ROR instruction will be available on MCS650X microprocessors after June, 1976.

A. 4.1. Zero Page Addressing (5 cycles)

<u>Tn</u>	<u>Address Bus</u>	<u>Data Bus</u>	<u>R/W</u>	<u>Comments</u>
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	ADL	1	Fetch Page Zero Effective Address
T2	00, ADL	Data	1	Fetch Data
T3	00, ADL	Data	0	
T4	00, ADL	Modified Data	0	Write modified Data back to memory
T0	PC + 2	OP CODE	1	Next Instruction

A. 4.2. Absolute Addressing (6 cycles)

<u>Tn</u>	<u>Address Bus</u>	<u>Data Bus</u>	<u>R/W</u>	<u>Comments</u>
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	ADL	1	Fetch low order byte of Effective Address
T2	PC + 2	ADH	1	Fetch high order byte of Effective Address
T3	ADH, ADL	Data	1	
T4	ADH, ADL	Data	0	
T5	ADH, ADL	Modified Data	0	Write modified Data back into memory
T0	PC + 3	OP CODE	1	Next Instruction

A. 4.3. Zero Page, X Addressing (6 cycles)

<u>Tn</u>	<u>Address Bus</u>	<u>Data Bus</u>	<u>R/W</u>	<u>Comments</u>
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	BAL	1	Fetch Page Zero Base Address
T2	00, BAL	Data (Discarded)	1	
T3	ADL: BAL + X (without carry)	Data	1	Fetch Data
T4	ADL: BAL + X (without carry)	Data	0	
T5	ADL: BAL + X (without carry)	Modified Data	0	Write modified Data back into memory
T0	PC + 2	OP CODE	1	Next Instruction

A. 4.4. Absolute, X Addressing (7 cycles)

<u>Tn</u>	<u>Address Bus</u>	<u>Data Bus</u>	<u>R/W</u>	<u>Comments</u>
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	BAL	1	Fetch low order byte of Base Address
T2	PC + 2	BAH	1	Fetch high order byte of Base Address
T3	ADL: BAL + X ADH: BAH + C	Data (Discarded)	1	
T4	ADL: BAL + X ADH: BAH + C	Data	1	Fetch Data
T5	ADH, ADL	Data	0	
T6	ADH, ADL	Modified Data	0	Write modified Data back into memory
T0	PC + 3	OP CODE	1	New Instruction

A. 5. MISCELLANEOUS OPERATIONS

BCC	BRK	PHP
BCS	BVC	PLA
BEQ	BVS	PLP
BMI	JMP	RTI
BNE	JSR	RTS
BPL	PHA	

A. 5.1. Push Operation--PHP, PHA (3 cycles)

<u>Tn</u>	<u>Address Bus</u>	<u>Data Bus</u>	<u>R/W</u>	<u>Comments</u>
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	OP CODE (Discarded)	1	
T2	Stack Pointer*	Data	0	Write Internal Register into Stack
T0	PC + 1	OP CODE	1	Next Instruction

*Subsequently referred to as "Stack Ptr."

A. 5.2. Pull Operations--PLP, PLA (4 cycles)

<u>Tn</u>	<u>Address Bus</u>	<u>Data Bus</u>	<u>R/W</u>	<u>Comments</u>
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	OP CODE (Discarded)	1	
T2	Stack Ptr.	Data (Discarded)	1	
T3	Stack Ptr. + 1	Data	1	Fetch Data from Stack
T0	PC + 1	OP CODE	1	Next Instruction

A. 5.3. Jump to Subroutine--JSR (6 cycles)

<u>Tn</u>	<u>Address Bus</u>	<u>Data Bus</u>	<u>R/W</u>	<u>Comments</u>
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	ADL	1	Fetch low order byte of Subroutine Address
T2	Stack Ptr.	Data (Discarded)	1	
T3	Stack Ptr.	PCH	0	Push high order byte of program counter to Stack
T4	Stack Ptr. - 1	PCL	0	Push low order byte of program counter to Stack
T5	PC + 2	ADH	1	Fetch high order byte of Subroutine Address
T0	Subroutine Address (ADH, ADL)	OP CODE	1	Next Instruction

A. 5.4. Break Operation--(Hardware Interrupt)-BRK (7 cycles)

<u>Tn</u>	<u>Address Bus</u>	<u>Data Bus</u>	<u>R/W</u>	<u>Comments</u>
T0	PC	OP CODE	1	Fetch BRK OP CODE (or force BRK)
T1	PC + 1 (PC on hardware interrupt)	Data (Discarded)	1	
T2	Stack Ptr.	PCH	0	Push high order byte of program counter to Stack
T3	Stack Ptr. - 1	PCL	0	Push low order byte of program counter to Stack
T4	Stack Ptr. - 2	P	0	Push Status Register to Stack
T5	FFFE (NMI-FFFA) (RES-FFFC)	ADL	1	Fetch low order byte of interrupt vector
T6	FFFF (NMI-FFFB) (RES-FFFD)	ADH	1	Fetch high order byte of interrupt vector
T0	Interrupt Vector (ADH, ADL)	OP CODE	1	Next Instruction

A. 5.5. Return from Interrupt-RTI (6 cycles)

<u>Tn</u>	<u>Address Bus</u>	<u>Data Bus</u>	<u>R/W</u>	<u>Comments</u>
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	Data (Discarded)	1	
T2	Stack Ptr.	Data (Discarded)	1	
T3	Stack Ptr. + 1	Data	1	Pull P from Stack
T4	Stack Ptr. + 2	Data	1	Pull PCL from Stack
T5	Stack Ptr. + 3	Data	1	Pull PCH from Stack
T0	PCH, PCL	OP CODE	1	Next Instruction

A. 5.6. Jump Operation--JMP

A.5.6.1. Absolute Addressing Mode (3 cycles)

<u>Tn</u>	<u>Address Bus</u>	<u>Data Bus</u>	<u>R/W</u>	<u>Comments</u>
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	ADL	1	Fetch low order byte of Jump Address
T2	PC + 2	ADH	1	Fetch high order byte of Jump Address
T0	ADH, ADL	OP CODE	1	Next Instruction

A.5.6.2. Indirect Addressing Mode (5 cycles)

<u>Tn</u>	<u>Address Bus</u>	<u>Data Bus</u>	<u>R/W</u>	<u>Comments</u>
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	IAL	1	Fetch low order byte of Indirect Address
T2	PC + 2	IAH	1	Fetch high order byte of Indirect Address
T3	IAH, IAL	ADL	1	Fetch low order byte of Jump Address
T4	IAH, IAL + 1	ADH	1	Fetch high order byte of Jump Address
T0	ADH, ADL	OP CODE	1	Next Instruction

A. 5.7. Return from Subroutine--RTS (6 cycles)

<u>Tn</u>	<u>Address Bus</u>	<u>Data Bus</u>	<u>R/W</u>	<u>Comments</u>
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	Data (Discarded)	1	
T2	Stack Ptr.	Data (Discarded)	1	
T3	Stack Ptr. + 1	PCL	1	Pull PCL from Stack
T4	Stack Ptr. + 2	PCH	1	Pull PCH from Stack
T5	PCH, PCL (from Stack)	Data (Discarded)	1	
T0	PCH, PCL + 1	OP CODE	1	Next Instruction

A. 5.8. Branch Operation--BCC, BCS, BEQ, BMI, BNE, BPL, BVC, BVS (2, 3, or 4 cycles)

<u>Tn</u>	<u>Address Bus</u>	<u>Data Bus</u>	<u>R/W</u>	<u>Comments</u>
T0	PC	OP CODE	1	Fetch OP CODE
T1	PC + 1	Offset	1	Fetch Branch Offset
T2*	PC + 2 + offset (w/o carry)	OP CODE	1	Offset Added to Program Counter
T3**	PC + 2 + offset (with carry)	OP CODE	1	Carry Added

*Skip if branch not taken

**Skip if branch not taken; skip if branch operation doesn't cross page boundary.