INTRODUCTION

The W65C02 Microprocessor offers complete hardware and software capability with existing 6500 series products as well as significant enhancements. The W65C02 40-pin Dip plug compatible with the NMOS 6502 and acts as a direct replacement in existing systems. The W65C02 44-pin plastic chip carrier (PLCC) offers additional features such as Memory Lock Output (ML-) and Vector Pull (VP-).

In additional to compatibility, the CMOS Family offers many new features unavailable on NMOS products. These include higher reliability, greatly reduced power consumption, increased noise immunity, and other operational enhancements. In addition, ten new instructions (twenty-nine new opcodes) and two new addressing modes have been added. Also the W65C02 is able to run at higher clock speeds than the previous NMOS design.

Because of the popularity of the 6502 instruction set and the extended features listed here, the W65C02 is ideal for new application-specific designs. Also, the small die size makes the W65C02 an excellent choice as a core microprocessor in single-chip microcomputers.

The W65C02 instruction set is downward compatible with the W65C816/802 16-bit Microprocessor Family and with the forth-coming W65C832 32-bit Microprocessor Family.

KEY FEATURES OF THE W65C02

* Advanced CMOS family that is compatible with NMOS 6500 series microprocessors
* Uses single 1.2-6 volt power supply, 5 volts specified
* Low power consumption (2mA @1 MHz) allows battery-powered operation, 10uA standby current
* Enhanced instruction set: 20 additional op codes encompassing ten new instructions enhance software performance compared to existing NMOS 6500 microprocessor instruction set
  --66 microprocessor instructions
  --180 operational codes
  --15 addressing modes
* 65K-byte addressable memory
* 2,4,6,8 or 10MHz operation, timing compatible with W65C816/802
* New Stop-The-Clock (STP) instruction compatible with W65C816/802
* On-board clock generator/ oscillator can be driven by an external single-phase clock input, an RC network, or a crystal circuit
* Early address valid allows use with slower memories
* Early write data for dynamic memories
* 8-bit parallel processing
* Decimal and binary arithmetic
* Pipeline architecture
* Programmable stack pointer
* Variable length stack
* Interrupt capability
* Non-maskable interrupt
* 8-bit bidirectional data bus
* "Ready" input (for single cycle execution)
* Direct memory access capability
* New Wait-For-Interrupt (WAI) instruction compatible with W65C816/802

SECTION 1

W65C02 FUNCTIONAL DESCRIPTION

1.1 Instruction Register and Decode

Instructions fetched from memory are gated onto the internal data bus. These instructions are latched into the instruction register then decoded, along with timing and interrupt signals, to generated control signals for the various registers.

1.2 Timing Control Unit

The Timing Control Unit keeps track of the instruction cycle being monitored. The unit is set to zero each time an instruction fetch is executed and is advanced at the beginning of each PHI1 clock pulse for as many cycles as is required to complete the instruction. Each data transfer between registers depends upon decoding the contents of both the Instruction Register and the Timing Control Unit.

1.3 Arithmetic and Logic Unit

All arithmetic and logic operations take place within the ALU including incrementing and decrementing internal registers (except the program counter). The ALU has no internal memory and is used only to perform logical and transient numerical operations.

1.4 Accumulator

The Accumulator is a general purpose 8-bit register which stores the results of most arithmetic and logic operations. In addition, the accumulator usually contains one of the two data words used in these operations.

1.5 Index Registers

There are two 8-bit Index Registers (X and Y) which may be used to count program steps or to provide an index value to be used in generating an effective address. When executing an instruction which specifies indexed addressing, the CPU fetches the opcode and the base address, and modifies the address by adding the index register to it prior to performing the desired operation. Pre- or post-indexing of indirect addresses is possible.

1.6 Processor Status Register

The 8-bit Processor Status Register contains seven status flags. Some of the flags are controlled by the program, others may be controlled both by the program and the CPU. The 6500 instruction set contains a number of conditional branch instructions which are designed to allow testing of these flags.

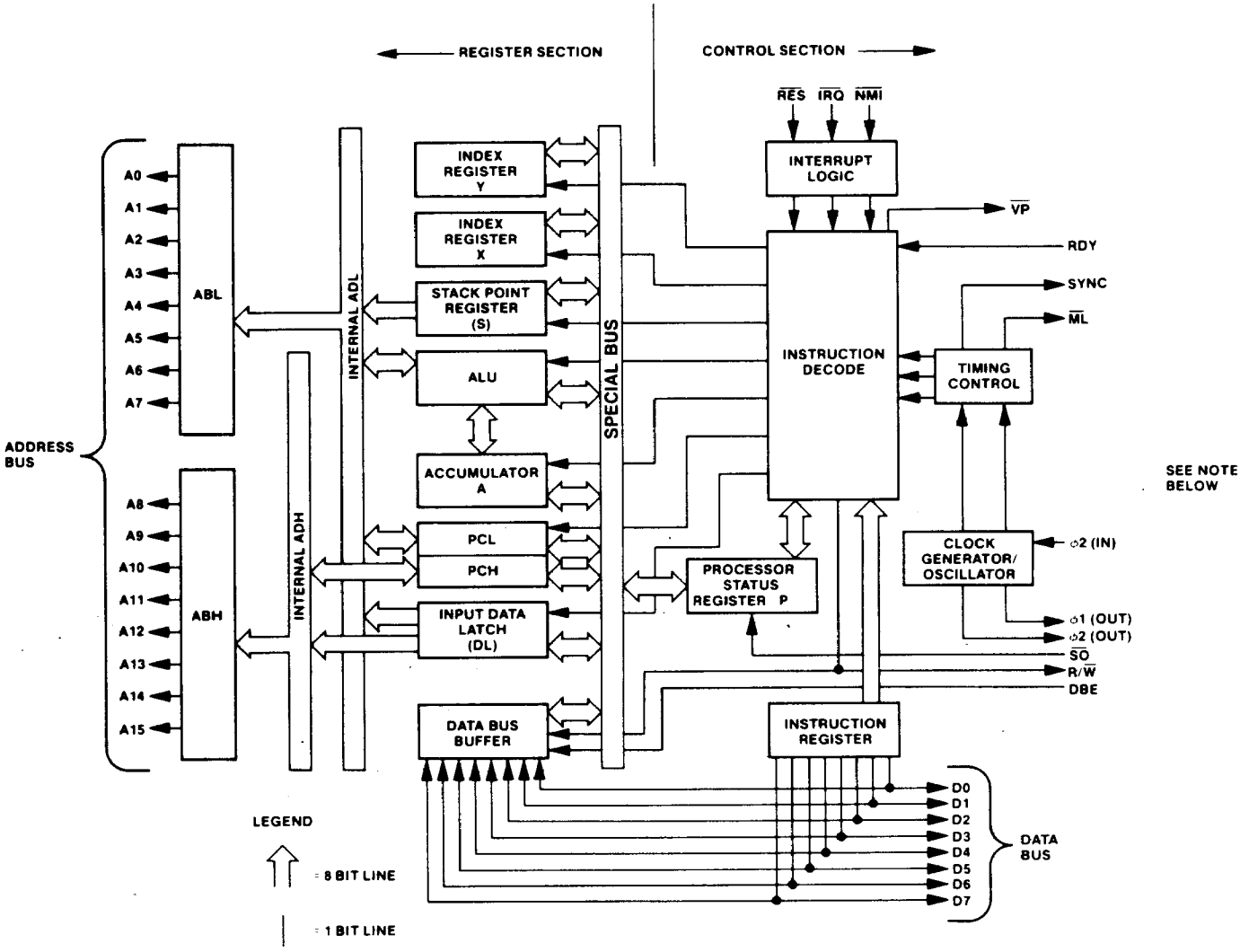
1.7 Program Counter

The 16-bit Program Counter Register provides the addresses which step the microprocessor through sequential program instructions. Each time the microprocessor fetches an instruction from program memory, the lower byte of the program counter (PCL) is placed on the low-order bits of the address bus and the higher byte of the program counter (PCH) is placed on the high-order 8 bits. The counter is incremented each time an instruction or data is fetched from program memory.

1.8 Stack Pointer

The Stack Pointer is an 8-bit register which is used to control the addressing of the variable-length stack. The stack pointer is automatically incremented and decremented under control of the microprocessor to perform stack manipulations under direction of either the program or interrupts (NMI- and IRQ-). The stack allows simple implementation of nested subroutines and multiple level interrupts.

Figure 1-1 W65C02 Internal Architecture Simplified Block Diagram



Note: Refer to Table 6-1 for signal input/output applicability.

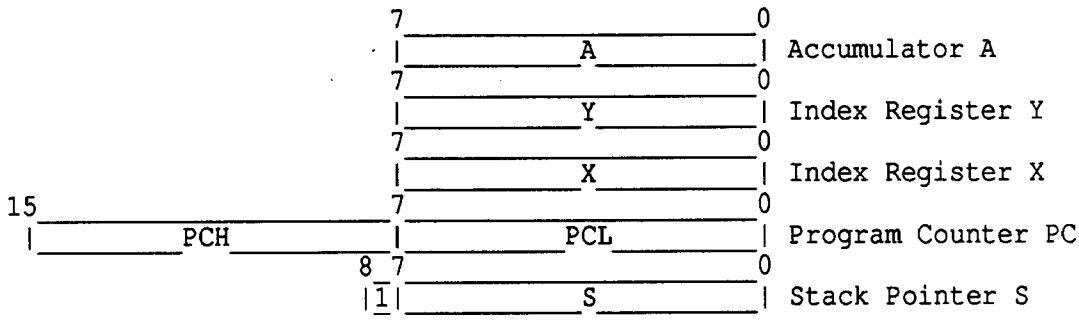


Figure 1-2 W65C02 Microprocessor Programming Model

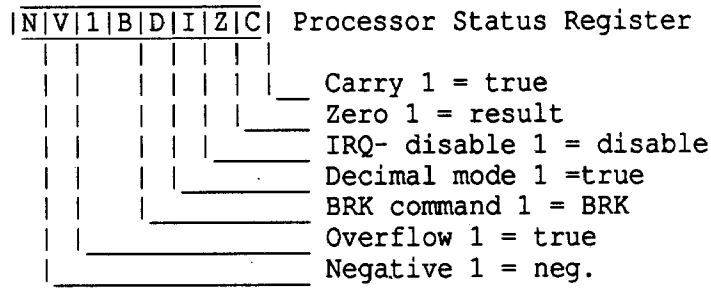


Figure 1-3 W65C02 Status Register Coding

SECTION 2

PIN FUNCTION DESCRIPTION

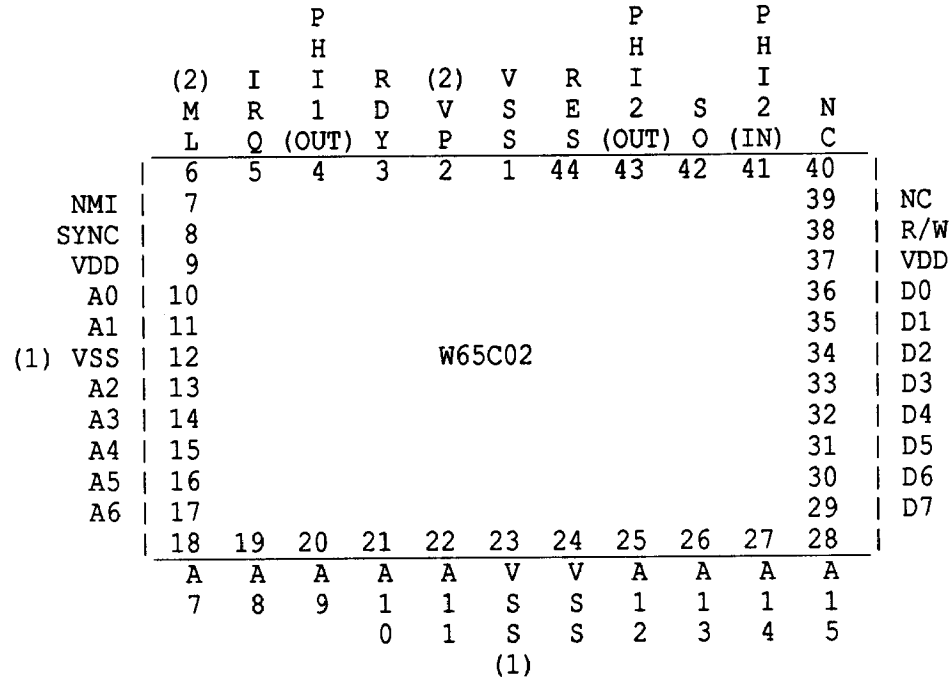


Figure 2-1 W65C02 44 Pin PLCC Pinout

(1) Power supply pins not available on the 40 pin version. These power supply pins have been added for improved performance.
(2) New pins not available on the 40 pin version.

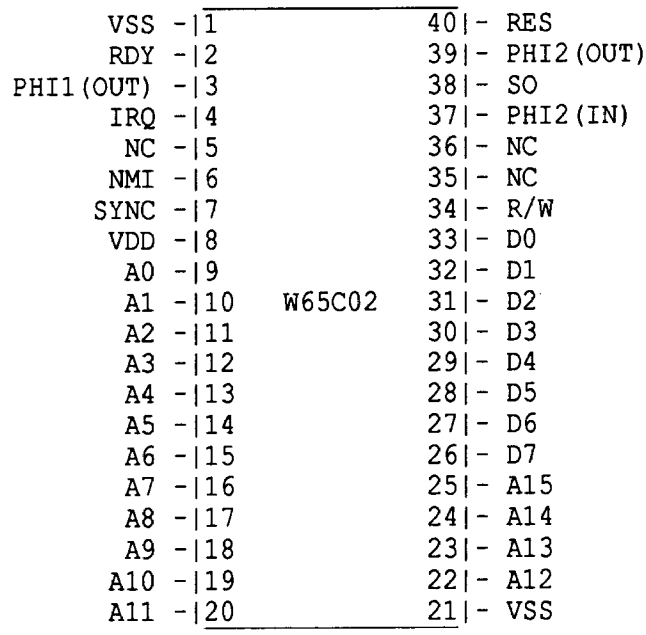


Figure 2-2 W65C02 40 Pin PDIP Pinout

Table 2-1 Pin Function Table

Pin	Description
A0-A15	Address Bus
PHI2(IN)	Phase 2 In Clock
D0-D7	Data Bus
IRQ-	Interrupt Request
ML-	Memory Lock
NC	No Connection
NMI-	Non-Maskable Interrupt
PHI1(OUT)	Phase 1 Out Clock
PHI2(OUT)	Phase 2 Out Clock
RDY	Ready
RES-	Reset
R/W-	Read/Write
SO-	Set Overflow
SYNC	Synchronize
VP-	Vector Pull
VDD	Positive Power Supply (+5 volts)
VSS	Internal Logic Ground

2.1 Address Bus (A0-A15)

The address bus consists of A0-A15 forming a 16-bit address bus for memory and I/O exchanges on the data bus. The output of each address line is TTL compatible, capable of driving one standard TTL load and 130pF.

2.2 Data Bus (D0-D7)

The data lines constitute and 8-bit bidirectional data bus for use during data exchanges between the microprocessor and peripherals. The outputs are three-state buffers capable of driving one TTL load and 130 pF.

2.3 Interrupt Request (IRQ-)

This TTL compatible signal requests that an interrupt sequence begin within the microprocessor. The IRQ- is sampled during PHI2 operation; if the interrupt flag in the processor status register is zero, the current instruction is completed and the interrupt sequence begins during PHI1. The program counter and processor status register are stored in the stack. The microprocessor will then set the interrupt mask flag high so that no further interrupts may occur. At the end of this cycle, the program counter low will be loaded from address FFFE, and program counter high from location FFFF, transferring program control to the memory vector located at these addresses. The RDY signal must be in the high state for an interrupt to be recognized. A 3K ohm external resistor should be used for proper wire-OR operation.

2.4 Memory Lock (ML-) - (44 PLCC only)

In a multiprocessor system, ML- indicates the need to defer the rearbitration of the next bus cycle to ensure the integrity of read-modify-write instructions, ML goes low during ASL, DEC, INC, LSR, ROL, ROR, TRB, TSB memory referencing instructions. This signal is low for the modify and write cycles.

2.5 Non-Maskable Interrupt (NMI-)

A negative-going edge on this input requests that a non-maskable interrupt sequence be generated within the microprocessor. The NMI- is sampled during PHI2; the current instruction is completed and the interrupt sequence begins during PHI1. The program counter is loaded with the interrupt vector from locations FFFA (low byte) and FFFB (high byte), thereby transferring program control to the non-maskable interrupt routine. However, it should be noted this is an edge-sensitive input. As a result, another interrupt will occur if there is another negative-going transition and the program has not returned from a previous interrupt. Also, no interrupt will occur if NMI- is low and a negative-going edge has not occurred since the last non-maskable interrupt.

2.6 Phase 1 Out (PHI1(OUT))

Inverted PHI2(OUT) signal.

2.7 Phase 2 In (PHI2(IN))

This is the buffered clock input to the internal clock generator. The clock outputs, PHI1(OUT) and PHI2(OUT), are derived from this signal.

2.8 Phase 2 Out (PHI2(OUT)

This signal is generated from PHI2(IN). PHI2(IN) provides timing for the system.

2.9 Read/Write (R/W-)

This signal is normally in the high state indicating that the microprocessor is reading data from memory of I/O bus. In the low state the data bus has valid data from the microprocessor to be stored at the addressed memory location.

2.10 Ready (RDY)

This bidirectional signal allows the user to single-cycle the microprocessor on all cycles including write cycles. A negative transition to the low state during or coincident with PHI1 will halt the microprocessor with the output address lines reflecting the current address being fetched. This condition will remain through a subsequent phase two in which the ready signal is low. This feature allows microprocessor interfacing with low-speed memory as well as direct memory access (DMA). The new WAI instruction pulls RDY low signaling the Wait-For-Interrupt condition, thus RDY is a bidirectional pin.

2.11 Reset (RES-)

This input is used to reset the microprocessor. Reset must be held low for at least two clock cycles after VDD reaches operating voltage from a power down. A positive transition of this pin will then cause an initialization sequence to begin. After the system has been operating, a low on this line of a least two cycles will cease microprocessor activity. When a positive edge is detected, there is an initialization sequence lasting six clock cycles. Then the interrupt mask flag is set, the decimal mode is cleared and the program counter is loaded with the restart vector from locations FFFC (low byte) and FFFD (high byte). This is the start location for program control this input should be high in normal operation.

2.12 Set Overflow (SO-)

A negative transition on this line sets the overflow bit in the status code register. The signal is sampled on the trailing edge of PHI1.

2.13 Synchronize (SYNC)

The SYNC output is provided to identify those cycles during which the microprocessor is fetching an opcode. The SYNC line goes high during PHI1 of an opcode fetch and stays high for the remainder of that cycle. If the RDY line is pulled low during the PHI1 clock pulse in which SYNC went high, the processor will stop in its current state and will remain in the state until the RDY line goes high. In this manner, the SYNC signal can be used to control RDY to cause single instruction execution.

2.14 VDD and VSS

VDD is the positive supply voltage and VSS is system logic ground.

2.15 Vector Pull (VP-) - (44 PLCC only)

The Vector Pull output indicates that a vector location is being addressed during an interrupt sequence. VP- is low during the last two interrupt sequence cycles, during which time the processor reads the interrupt vector. The VP- signal may be used to select and prioritize interrupts from several sources by modifying the vector addresses.

SECTION 3

ADDRESSING MODES

Fifteen addressing modes are available to the user of the WDC W65C02 microprocessor. The addressing modes are described in the following paragraphs:

3.1 Immediate Addressing

With immediate addressing, the operand is contained in the second byte of the instruction; no further memory addressing is required.

3.2 Absolute Addressing

For absolute addressing, the second byte of the instruction specifies the eight low order bits of the effective address while the third byte specifies the eight high order bits. Therefore, this addressing mode allows access to the total 65K bytes of addressable memory.

3.3 Zero Page Addressing

Zero page addressing allows shorter code and execution times by only fetching the second byte of the instruction and assuming a zero high address byte. The careful use of zero page addressing can result in significant increase in code efficiency.

3.4 Implied Addressing

In the implied addressing mode, the address containing the operand is implicitly stated in the operation code of the instruction.

3.5 Accumulator Addressing

This form of addressing is represented with a one byte instruction and implies an operation on the accumulator. The opcodes are included under implied addressing.

3.6 Zero Page Indexed Indirect Addressing: (IND,X)

With zero page indexed indirect addressing (usually referred to as Indirect X) the second byte of the instruction is added to the contents of the X index register; the carry is discarded. The result of this addition points to a memory location on page zero whose contents is the low order eight bits of the effective address. The next memory location in page zero contains the high order eight bits of the effective address. Both memory locations specifying the high and low order bytes of the effective address must in page zero.

3.7 Absolute Indexed Indirect Addressing (Jump Instruction only)

With absolute indexed indirect addressing, the contents of the second and third instruction bytes are added to the X register. The result of this addition points to a memory location containing the lower order eight bits of the effective address. The next memory location contains the higher-order eight bits of the effective address (opcode 7C).

3.8 Indirect Indexed Addressing: (IND), Y

This form of addressing is usually referred to as Indirect, Y. The second byte of the instruction points to a memory location in page zero. The contents of this memory location is added to the contents of the Y index register, the result being the low order eight bits of the effective address. The carry from this addition is added to the contents of the next page zero memory location, the result being the high order eight bits of the effective address.

3.9 Zero Page Indexed Addressing

Zero page absolute addressing is used in conjunction with the index register and is referred to as "Zero Page, X" or "Zero Page, Y". The effective address is calculated by adding the second byte to the contents of the index register. Since this is a form of "Zero Page" addressing, the content of the second byte references a location in page zero. Additionally, due to the "Zero Page" addressing nature of this mode, no carry is added to the high order eight bits of memory and crossing of page boundaries does not occur.

3.10 Absolute Indexed Addressing

Absolute indexed addressing is used in conjunction with X and Y index register and is referred to as "Absolute, X", and "Absolute, Y". The effective address is formed by adding the contents of X and Y to the address contained in the second and third bytes of the instruction. This mode allows the index register to contain the index or count value and the instruction to contain the base address. This type of indexing allows any location referencing and the index to modify multiple fields resulting in reduced coding and execution time.

3.11 Relative Addressing

Relative addressing is used only with branch instruction, it established a destination for the conditional branch.

3.12 Zero Page Indirect Addressing: Indirect

In this form of addressing, the second byte of the instruction contains the low order eight bits of a memory location. The high order eight bits is always zero. The contents of the fully specified memory location is the low order byte of the effective address. The next memory location contains the high order byte of the effective address.

3.13 Absolute Indirect Addressing (Jump Instruction Only)

The second byte of the instruction contains the low order eight bits of memory location. The high order eight bits of that memory location is contained in the third byte of the instruction. The contents of the fully specified memory location is the low order byte of the effective address. The next memory location contains the high order byte of the effective address which is loaded into the 16 bits of the program counter (opcode 6C).

SECTION 4

TIMING, AC AND DC CHARACTERISTICS

4.1 Absolute Maximum Ratings

Table 4-1 Absolute Maximum Ratings

Rating	Symbol	Value
Supply Voltage	VDD	-0.3 to +7.0V
Input Voltage	VIN	-0.3 to VDD +0.3V
Operating Temperature	TA	0 °C to +70°C
Storage Temperature	TS	-55 °C to +150°C

This device contains input protection against damage due to high static voltages or electric fields; however, precautions should be taken to avoid application of voltages higher than the maximum rating.

Note:

Exceeding these ratings may result in permanent damage. Functional operation under these conditions is not implied.

4.2 DC Characteristics VDD = 5.0V +/- 5%, VSS = 0V, TA = 0oC to +70oC

Table 4-2 DC Characteristics

Parameter	Symbol	Min	Max	Unit
Input High Voltage	Vih			
PHI2(IN), NMI-, RES-		0.7VDD	VDD+0.3	V
RDY, IRQ-, Data, SO-		VSS+2.0	VDD+0.3	V
Input Low Voltage	Vil			
PHI2(IN), NMI-, RES-		VSS-0.3	VSS+0.2	V
RDY, IRQ-, Data, SO-		VSS-0.3	VSS+0.8	V
Input Leakage Current (Vin = 0.4 to 2.4, VDD=5.25V)	Iin			
RES-, NMI-, RDY, IRQ-, SO- (Internal Pullup)		-100	1	uA
PHI2(IN)		-1	1	uA
Data, (Off State)		-10	10	uA
Output High Voltage(Ioh=-100uA,VDD=4.75V)				
SYNC,Data,A0-A15,R/W-,ML-,VP-	Voh	2.4	-	V
Output Low Voltage (Iol=1.6mA, VDD-4.75V)				
SYNC,Data,A0-A15,R/W-,ML-,VP-	Vol	-	VSS+0.4	V
Supply Current (No Load)	Icc		4	mA/MHz
Standby Current Outputs Unloaded	Isby	-		
PHI2(IN),RES-,NMI-,RDY,IRQ-,SO- = VDD; DATA = VSS or VDD			10	uA
Capacitance (Vin=0V, TA=25oC, f=1MHz)				
Logic,PHI2(IN)	Cin	-	10	pF
A0-A15, R/W-, Data (Off State)	Cts	-	15	
* Not inspected during production test; verified on a sample basis.				

4.3 General AC Characteristic Equations VDD= 5.0V +/- 5%, VSS= 0V,
TA= 0oC to +70oC

Table 4-3A W65C02 General AC Characteristic Equations, 4-7MHz

Parameter	Symbol	4 MHz Min	4 MHz Max	5 MHz Min	5 MHz Max	6 MHz Min	6 MHz Max	7 MHz Min	7 MHz Max	Unit
Cycle Time	tCYC	250	DC	200	DC	165	DC	140	DC	nS
Clock Pulse Width Low	tPWL	.125	10	.10	10	.082	10	.07	10	uS
Clock Pulse Width High	tPWH	125	-	100	-	82	-	70	-	nS
Fall Time, Rise Time	tF,tR	-	10	-	10	-	5	-	5	nS
DelayTime,PHI2(IN)-PHI1(OUT)	tDO1	-	20	-	20	-	20	-	20	nS
DelayTime,PHI2(IN)-PHI2(OUT)	tDO2	-	40	-	40	-	40	-	40	nS
Address Hold Time	tAH	10	-	10	-	10	-	10	-	nS
Address Setup Time	tADS	-	75	-	67	-	60	-	50	nS
Access Time	tACC	130	-	115	-	87	-	80	-	nS
Read Data Hold Time	tDHR	10	-	10	-	10	-	10	-	nS
Read Data Setup Time	tDSR	30	-	25	-	20	-	17	-	nS
Write Data Delay Time	tMDS	-	70	-	65	-	60	-	45	nS
Write Data Hold Time	tDHW	10	-	10	-	10	-	10	-	nS
Processor Control Setup Time	tPCS	30	-	25	-	20	-	17	-	nS
Processor Control Hold Time	tPCH	10	-	10	-	10	-	10	-	nS
Capacitive Load *1	CEXT	-	100	-	100	-	35	-	35	pF

Table 4-3B W65C02 General AC Characteristic Equations, 8-10MHz

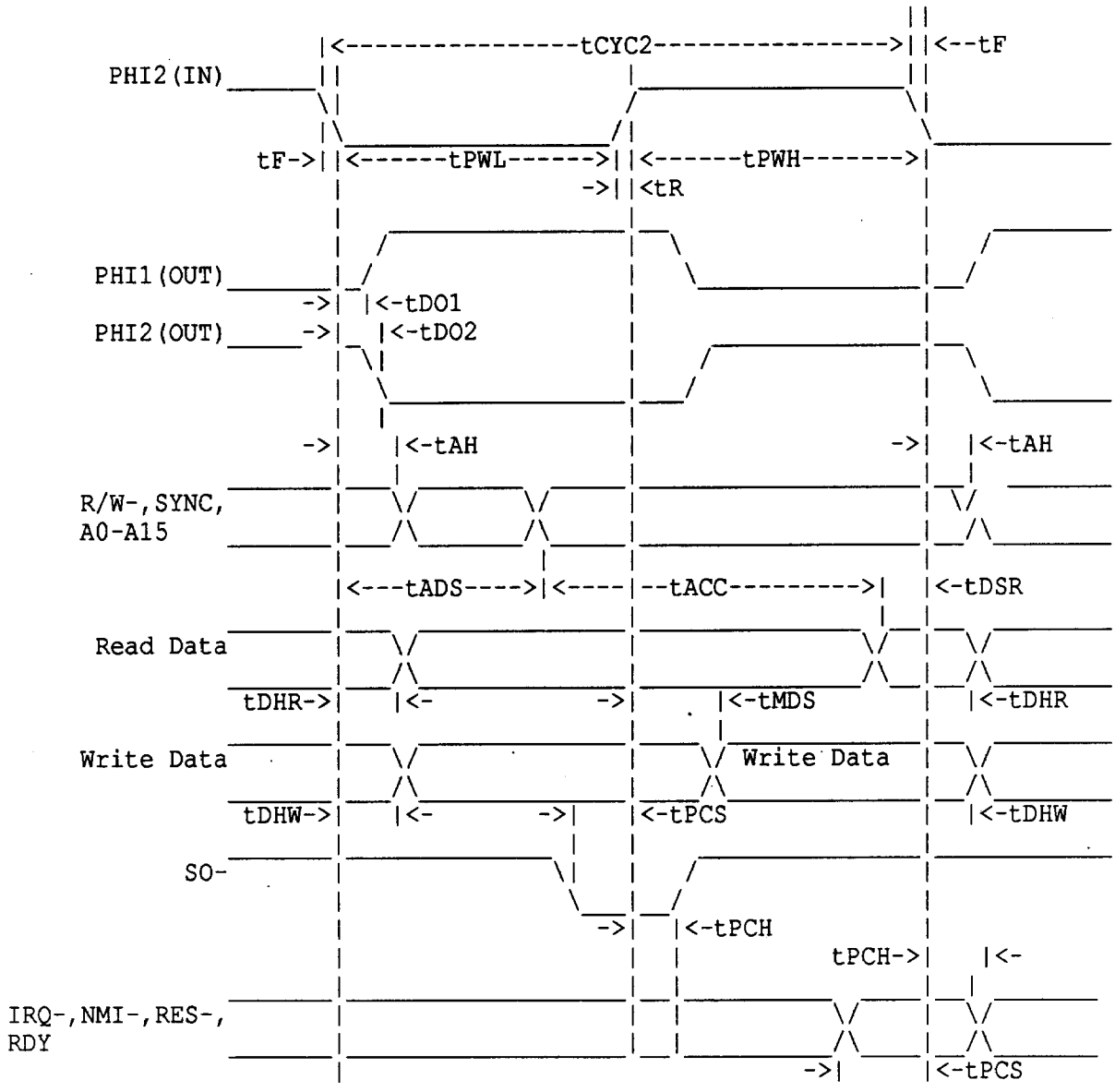
Parameter	Symbol	8 MHz Min	8 MHz Max	9 MHz Min	9 MHz Max	10 MHz Min	10 MHz Max	Unit
Cycle Time	tCYC	125	DC	110	DC	100	DC	nS
Clock Pulse Width Low	tPWL	.062	10	.055	10	.05	10	uS
Clock Pulse Width High	tPWH	62	-	55	-	50	-	nS
Fall Time, Rise Time	tF,tR	-	5	-	5	-	5	nS
Delay Time, PHI2(IN)-PHI1(OUT)	tD01	-	20	-	20	-	20	nS
Delay Time, PHI2(IN)-PHI2(OUT)	tD02	-	40	-	40	-	40	nS
Address Hold Time	tAH	10	-	10	-	10	-	nS
Address Setup Time	tADS	-	40	-	40	-	40	nS
Access Time	tACC	70	-	70	-	70	-	nS
Read Data Hold Time	tDHR	10	-	10	-	10	-	nS
Read Data Setup Time	tDSR	15	-	15	-	15	-	nS
Write Data Delay Time	tMDS	-	40	-	40	-	40	nS
Write Data Hold Time	tDHW	10	-	10	-	10	-	nS
Processor Control Setup Time	tPCS	15	-	15	-	15	-	nS
Processor Control Hold Time	tPCH	10	-	10	-	10	-	nS
Capacitive Load *1	CEXT	-	35	-	35	-	35	pF

*1 Applied to Address, Data, R/W

Table 4-3C W65C02 General AC Characteristics, 40KHz

Parameter	Symbol	40 KHz Min	Max	Unit
Cycle Time	tCYC	-	25	uS
Clock Pulse Width Low	tPWL	12.5	13	uS
Clock Pulse Width High	tPWH	12.5	-	uS
Fall Time, Rise Time	tF,tR	-	10	nS
DelayTime,PHI2(IN)-PHI1(OUT)	tDO1	-	20	nS
DelayTime,PHI2(IN)-PHI2(OUT)	tDO2	-	40	nS
Address Hold Time	tAH	10	-	nS
Address Setup Time	tADS	-	2	uS
Access Time	tACC	35	-	uS
Read Data Hold Time	tDHR	100	-	nS
Read Data Setup Time	tDSR	1.5	-	uS
Write Data Delay Time	tMDS	-	2	uS
Write Data Hold Time	tDHW	10	-	nS
Processor Control Setup Time	tPCS	1.5	-	uS
Processor Control Hold Time	tPCH	100	-	nS
Capacitive Load *1	CEXT	-	100	pF

*1 Applied to Address, Data, R/W



Timing Notes:

1. Timing measurement points are 1.5V and 1.5V.

Figure 4-1 General Timing Diagram

SECTION 5

ORDERING INFORMATION

W 65C02 P -4

Description
W-Standard

Product Identification Number

Package
P- 40 lead plastic dual in line
PL-44 leaded plastic chip carrier

Temperature/Processing
Blank- 0oC to +70oC

Performance Designator
Designators selected for speed and power.
-4 4MHz -6 6MHz -8 8MHz -10 10MHz

General sales or technical assistance, and information about devices supplied to a custom specification may be requested from:

The Western Design Center, Inc.
2166 East Brown Road
Mesa, Arizona 85213
Phone: 602-962-4545 Fax: 602-835-6442

WARNING:

MOS CIRCUITS ARE SUBJECT TO DAMAGE FROM STATIC DISCHARGE

Internal static discharge circuits are provided to minimize part damage due to environmental static electrical charge build-ups. Industry established recommendations for handling MOS circuits include:

1. Ship and store product in conductive shipping tubes or conductive foam plastic. Never ship or store product in non-conductive plastic containers or non-conductive plastic foam material.
2. Handle MOS parts only at conductive work stations.
3. Ground all assembly and repair tools.

SECTION 6

APPLICATION INFORMATION

Table 6-1 W65C02 Instruction Set-Alphabetical Sequence

ADC-Add Memory to Accumulator with Carry
AND-"AND" Memory with Accumulator
ASL-Shift One Bit Left
BCC-Branch on Carry Clear
BCS-Branch on Carry Set
BEQ-Branch on Result Zero
* BIT-Test Memory Bits w/Accumulator
BMI-Branch on Result Minus
BNE-Branch on Result Not Zero
BPL-Branch on Result Plus
BRA-Branch Always
BRK-Force Break
BVC-Branch on Overflow Clear
BVS-Branch on Overflow Set
CLC-Clear Carry Flag
CLD-Clear Decimal Mode
CLI-Clear Interrupt Disable Bit
CLV-Clear Overflow Flag
* CMP-Compare Memory and Accumulator
CPX-Compare Memory and Index X
CPY-Compare Memory and Index Y
* DEC-Decrement by One
DEX-Decrement Index X by One
DEY-Decrement Index Y by One
* EOR-"Exclusive-or" Memory with Accumulator
* INC-Increment by One
INX-Increment Index X by One
INY-Increment Index Y by One
* JMP-Jump to New Location
JSR-Jump to New Location Saving Return Address
* LDA-Load Accumulator with Memory
LDX-Load Index X with Memory
LDY-Load Index Y with Memory
LSR-Shift One Bit Right
NOP-No Operation
* ORA-"OR" Memory with Accumulator
PHA-Push Accumulator on Stack
PHP-Push Processor Status on Stack
PHX-Push Index X on Stack
PHY-Push Index Y on Stack
PLA-Pull Accumulator from Stack
PLP-Pull Processor Status from Stack
PLX-Pull Index X from Stack
PLY-Pull Index Y from Stack
ROL-Rotate One Bit Left
ROR-Rotate One Bit Right
RTI-Return from Interrupt
RTS-Return from Subroutine
* SBC-Subtract Memory from Accumulator with Borrow
SEC-Set Carry Flag
SED-Set Decimal Mode
SEI-Set Interrupt Disable Bit
* STA-Store Accumulator in Memory
# STP-Stop the Clock
STX-Store Index X in Memory
STY-Store Index Y in Memory
STZ-Store Zero in Memory
TAX-Transfer Accumulator to Index X
TAY-Transfer Accumulator in Index Y
TRB-Test and Reset Memory Bits with Accumulator
TSB-Test and Set Memory Bits with Accumulator
TSC-Transfer Stack Pointer Register to Accumulator
TSX-Transfer Stack Pointer to Index X
TXA-Transfer Index X to Accumulator
TXS-Transfer Index X to Stack Pointer
TYA-Transfer Index Y to Accumulator
WAI-Wait for Interrupt

- New instruction
* - Old instruction with New Addressing Modes

Table 6-2 Vector Locations

FFFE,F-IRQ-/BRK Hardware/Software
FFFC,D-RESET- Hardware
FFFA,B-NMI- Hardware

The VP output is low during the two cycles used for vector location access. When an interrupt is executed, D=0 and I=1 in Status Register P.

Table 6-3 Opcode Matrix

LSD / MSD	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	BRK	ORA ind, X			TSB • zpg	ORA zpg	ASL zpg		PHP	ORA imm	ASL A		TSB • abs	ORA abs	ASL abs		0
1	BPL rel	ORA ind, Y	ORA* ind		TRB • zpg	ORA zpg, X	ASL zpg, X		CLC	ORA abs, Y	INC* A		TRB • abs	ORA abs, X	ASL abs, X		1
2	JSR abs	AND ind, X			BIT zpg	AND zpg	ROL zpg		PLP	AND imm	ROL A		BIT abs	AND abs	ROL abs		2
3	BMI rel	AND ind, Y	AND* ind		BIT * zpg, X	AND zpg, X	ROL zpg, X		SEC	AND abs, Y	DEC* A		BIT * abs, X	AND abs, X	ROL abs, X		3
4	RTI	EOR ind, X				EOR zpg	LSR zpg		PHA	EOR imm	LSR A		JMP abs	EOR abs	LSR abs		4
5	BVC rel	EOR ind, Y	EOR* ind			EOR zpg, X	LSR zpg, X		CLI	EOR abs, Y	PHY•			EOR abs, X	LSR abs, X		5
6	RTS	ADC ind, X			STZ • zpg	ADC zpg	ROR zpg		PLA	ADC imm	ROR A		JMP ind	ADC abs	ROR abs		6
7	BVS rel	ADC ind, Y	ADC* ind		STZ • zpg, X	ADC zpg, X	ROR zpg, X		SEI	ADC abs, Y	PLY•		JMP * ind, X	ADC abs, X	ROR abs, X		7
8	BRA • rel	STA ind, X			STY zpg	STA zpg	STX zpg		DEY	BIT * imm	TXA		STY abs	STA abs	STX abs		8
9	BCC rel	STA ind, Y	STA* ind		STY zpg, X	STA zpg, X	STX zpg, Y		TYA	STA abs, Y	TXS		STZ • abs	STA abs, X	STZ • abs, X		9
A	LDY imm	LDA ind, X	LDX imm		LDY zpg	LDA zpg	LDX zpg		TAY	LDA imm	TAX		LDY abs	LDA abs	LDX abs		A
B	BCS rel	LDA ind, Y	LDA* ind		LDY zpg, X	LDA zpg, X	LDX zpg, Y		CLV	LDA abs, Y	TSX		LDY abs, X	LDA abs, X	LDX abs, Y		B
C	CPY imm	CMP ind, X			CPY zpg	CMP zpg	DEC zpg		INY	CMP imm	DEX	WAI•	CPY abs	CMP abs	DEC abs		C
D	BNE rel	CMP ind, Y	CMP* ind			CMP zpg, X	DEC zpg, X		CLD	CMP abs, Y	PHX•	STP•		CMP abs, X	DEC abs, X		D
E	CPX imm	SBC ind, X			CPX zpg	SBC zpg	INC zpg		INX	SBC imm	NOP		CPX abs	SBC abs	INC abs		E
F	BEQ rel	SBC ind, Y	SBC* ind			SBC zpg, X	INC zpg, X		SED	SBC abs, Y	PLX•			SBC abs, X	INC abs, X		F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

NOTE: • = New Instruction
* = Old Instruction with New Addressing Mode

Table 6-4 Operation, Operation Codes and Status Register

MNE-MONIC	OPERATION	IMMEDIATE OP	n	#	ABSOLUTE OP	n	#	ZERO PAGE OP	n	#	(4) IMPLIED OP	n	#	(IND, X) OP	n	#	(1) (IND), Y OP	n	#	ZPG,X OP	n	#	(1) ABS,X OP	n	#	(1) ABS,Y OP	n	#	RELATIVE (2) OP	n	#	INDIRECT OP	n	#	ZPG,Y OP	n	#	PROCESSOR STATUS CODE 7 6 4 3 2 1 0 N V B D I Z C	MNE-MONIC
ADC	A + M + C → A (3)	69	2	2	6D	4	3	65	3	2				61	6	2	71	5	2	75	4	2	7D	4	3	79	4	3				72	5	2				N V • • • Z C	ADC
AND	A∧M → A	29	2	2	2D	4	3	25	3	2				21	6	2	31	5	2	35	4	2	3D	4	3	39	4	3				32	5	2				N • • • • Z •	AND
ASL	C ← [7 0] ←0				0E	6	3	06	5	2	0A	2	1							16	6	2	1E	6	3													N • • • • Z C	ASL
BCC	BRANCH IF C=0 (2)																												90	2	2							• • • • • • •	BCC
BCS	BRANCH IF C=1 (2)																												B0	2	2							• • • • • • •	BCS
BEQ	BRANCH IF Z=1 (2)																												F0	2	2							• • • • • • •	BEQ
BIT	A ∧ M (5)	89	2	2	2C	4	3	24	3	2										34	4	2	3C	4	3													M7 M6 • • Z •	BIT
BMI	BRANCH IF N=1 (2)																												30	2	2							• • • • • • •	BMI
BNE	BRANCH IF Z=0 (2)																												D0	2	2							• • • • • • •	BNE
BPL	BRANCH IF N=0 (2)																												10	2	2							• • • • • • •	BPL
BRA	BRANCH ALWAYS(2)																												80	2	2							• • • • • • •	BRA
BRK	BREAK										00	7	1																									• • 1 0 1 • •	BRK
BVC	BRANCH IF V=0 (2)																												50	2	2							• • • • • • •	BVC
BVS	BRANCH IF V=1 (2)																												70	2	2							• • • • • • •	BVS
CLC	0 → C										18	2	1																									• • • • • • 0	CLC
CLD	0 → D										D8	2	1																									• • • 0 • • •	CLD
CLI	0 → I										58	2	1																									• • • • 0 • •	CLI
CLV	0 → V										B8	2	1																									• 0 • • • • •	CLV
CMP	A-M	C9	2	2	CD	4	3	C5	3	2				C1	6	2	D1	5	2	D5	4	2	DD	4	3	D9	4	3				D2	5	2				N • • • • Z C	CMP
CPX	X-M	E0	2	2	EC	4	3	E4	3	2																												N • • • • Z C	CPX
CPY	Y-M	C0	2	2	CC	4	3	C4	3	2																												N • • • • Z C	CPY
DEC	DECREMENT				CE	6	3	C6	5	2	3A	2	1							D6	6	2	DE	6	3													N • • • • Z •	DEC
DEX	X-1 → X										CA	2	1																									N • • • • Z •	DEX
DEY	Y-1 → Y										88	2	1																									N • • • • Z •	DEY
EOR	A ∀ M →A	49	2	2	4D	4	3	45	3	2				41	6	2	51	5	2	55	4	2	5D	4	3	59	4	3				52	5	2				N • • • • Z •	EOR
INC	INCREMENT				EE	6	3	E6	5	2	1A	2	1							F6	6	2	FE	6	3													N • • • • Z •	INC
INX	X + 1 → X										E8	2	1																									N • • • • Z •	INX
INY	Y + 1 → Y										C8	2	1																									N • • • • Z •	INY
JMP	JUMP TO NEW LOC				4C	3	3							7C	6	3																6C	6	3				• • • • • • •	JMP
JSR	JUMP SUB				20	6	3																															• • • • • • •	JSR
LDA	M → A	A9	2	2	AD	4	3	A5	3	2				A1	6	2	B1	5	2	B5	4	2	BD	4	3	B9	4	3				B2	5	2				N • • • • Z •	LDA
LDX	M → X	A2	2	2	AE	4	3	A6	3	2																BE	4	3							B6	4	2	N • • • • Z •	LDX
LDY	M → Y	A0	2	2	AC	4	3	A4	3	2										B4	4	2	BC	4	3													N • • • • Z •	LDY
LSR	0→ [7 0] → C				4E	6	3	46	5	2	4A	2	1							56	6	2	5E	6	3													0 • • • • Z C	LSR
NOP	NO OPERATION										EA	2	1																									• • • • • • •	NOP
ORA	AVM → A	09	2	2	0D	4	3	05	3	2				01	6	2	11	5	2	15	4	2	1D	4	3	19	4	3				12	5	2				N • • • • Z •	ORA
PHA	A →Ms S-1 →S										48	3	1																									• • • • • • •	PHA
PHP	P →Ms S-1 →S										08	3	1																									• • • • • • •	PHP
PHX	X →Ms S-1 →S										DA	3	1																									• • • • • • •	PHX
PHY	Y →Ms S-1 →S										5A	3	1																									• • • • • • •	PHY
PLA	S+1 →S Ms →A										68	4	1																									N • • • • Z •	PLA
PLP	S+1 →S Ms →P										28	4	1																									N V 1 D I Z C	PLP
PLX	S+1 →S Ms →X										FA	4	1																									N • • • • Z •	PLX
PLY	S+1 →S Ms →Y										7A	4	1																									N • • • • Z •	PLY
ROL	[C ← 7 0 ←C]				2E	6	3	26	5	2	2A	2	1							36	6	2	3E	6	3													N • • • • Z C	ROL
ROR	[C → 7 0]				6E	6	3	66	5	2	6A	2	1							76	6	2	7E	6	3													N • • • • Z C	ROR
RTI	RTRN INT										40	6	1																									N V 1 D I Z C	RTI
RTS	RTRN SUB										60	6	1																									• • • • • • •	RTS
SBC	A-M-C̄ →A (3)	E9	2	2	ED	4	3	E5	3	2				E1	6	2	F1	5	2	F5	4	2	FD	4	3	F9	4	3				F2	5	2				N V • • • Z C	SBC
SEC	1 →C										38	2	1																									• • • • • • 1	SEC
SED	1 →D										F8	2	1																									• • • 1 • • •	SED
SEI	1 →I										78	2	1																									• • • • 1 • •	SEI
STA	A →M				8D	4	3	85	3	2				81	6	2	91	6	2	95	4	2	9D	5	3	99	5	3				92	5	2				• • • • • • •	STA
STP	STOP (1 → φ2)										DB	3	1																									• • • • • • •	STP
STX	X →M				8E	4	3	86	3	2																											96	4	2	• • • • • • •	STX
STY	Y →M				8C	4	3	84	3	2										94	4	2																	• • • • • • •	STY
STZ	00 →M				9C	4	3	64	3	2										74	4	2	9E	5	3													• • • • • • •	STZ
TAX	A →X										AA	2	1																									N • • • • Z •	TAX
TAY	A →Y										A8	2	1																									N • • • • Z •	TAY
TRB	Ā∧M →M (6)				1C	6	3	14	5	2																													• • • • • Z •	TRB
TSB	A∨M →M (6)				0C	6	3	04	5	2																													• • • • • Z •	TSB
TSX	S →X										BA	2	1																									N • • • • Z •	TSX
TXA	X →A										8A	2	1																									N • • • • Z •	TXA
TXS	X →S										9A	2	1																									• • • • • • •	TXS
TYA	Y →A										98	2	1																									N • • • • Z •	TYA
WAI	0 →RDY										CB	3	1																									• • • • • • •	WAI

Notes:
1. Add 1 to "n" if page boundary is crossed, except STA and STZ.
2. Add 1 to "n" if branch occurs to same page. Add 2 to "n" if branch occurs to different page.
3. Add 1 to "n" if decimal mode.
4. Accumulator address is included in Implied address.
5. "N" and "V" flags are unchanged in immediate mode.
6. "Z" flag indicates A∧M result (same as BIT instruction).

X Index X
Y Index Y
A Accumulator
M Memory per effective address
Ms Memory per stack pointer

\+ Add
\- Subtract
∧ And
∨ Or
∀ Exclusive or

n No. Cycles
No. Bytes
M6 Memory Bit #6
M7 Memory Bit #7

Table 6-5 Instruction Operation (13)

ADDRESS MODE		CYCLE	(14) $\overline{VP}$	(14) $\overline{ML}$	(14) VDA	(14) VPA	(15) ADDRESS BUS	DATA BUS	R/$\overline{W}$
1. Immediate-#		1.	1	1	1	1	PBR,PC	OpCode	1
(LDY,CPY,CPX,LDX,ORA,		2.	1	1	0	1	PBR,PC+1	IDL	1
AND,EOR,ADC,BIT,LDA,	(1)	2a.	1	1	0	1	PBR,PC+2	IDH	1
CMP,SBC,REP,SEP)									
(14 OpCodes)									
(2 and 3 bytes)									
(2 and 3 cycles)									
2a. Absolute-a		1.	1	1	1	1	PBR,PC	OpCode	1
(BIT,STY,STZ,LDY,		2.	1	1	0	1	PBR,PC+1	AAL	1
CPY,CPX,STX,LDX,		3.	1	1	0	1	PBR,PC+2	AAH	1
ORA,AND,EOR,ADC,		4.	1	1	1	0	DBR,AA	Data Low	1/0
STA,LDA,CMP,SBC)	(1)	4a.	1	1	1	0	DBR,AA+1	Data High	1/0
(18 OpCodes)									
(3 bytes)									
(4 and 5 cycles)									
2b. Absolute-(R-M-W)-a		1.	1	1	1	1	PBR,PC	OpCode	1
(ASL,ROL,LSR,ROR		2.	1	1	0	1	PBR,PC+1	AAL	1
DEC,INC,TSB,TRB)		3.	1	1	0	1	PBR,PC+2	AAH	1
(6 OpCodes)		4.	1	0	1	0	DBR,AA	Data Low	1
(3 bytes)	(1)	4a.	1	0	1	0	DBR,AA+1	Data High	1
(6 and 8 cycles)	(3)	5.	1	0	0	0	DBR,AA+1	IO	1
	(1)	6a.	1	0	1	0	DBR,AA+1	Data High	0
		6.	1	0	1	0	DBR,AA	Data Low	0
2c. Absolute (JUMP)-a		1.	1	1	1	1	PBR,PC	OpCode	1
(JMP)(4C)		2.	1	1	0	1	PBR,PC+1	New PCL	1
(1 OpCode)		3.	1	1	0	1	PBR,PC+2	New PCH	1
(3 bytes)		1.	1	1	1	1	PBR,NEW PC	New OpCode	1
(3 cycles)									
2d. Absolute (Jump to		1.	1	1	1	1	PBR,PC	OpCode	1
subroutine)-a		2.	1	1	0	1	PBR,PC+1	New PCL	1
(JSR)		3.	1	1	0	1	PBR,PC+2	New PCH	1
(1 OpCode)		4.	1	1	0	0	PBR,PC+2	IO	1
(3 bytes)		5.	1	1	1	0	0,S	PCH	0
(6 cycles)		6.	1	1	1	0	0,S-1	PCL	0
(different order from		1.	1	1	1	1	PBR,NEW PC	New OpCode	1
N6502)									
*3a. Absolute Long-al		1.	1	1	1	1	PBR,PC	OpCode	1
(ORA,AND,EOR,ADC		2.	1	1	0	1	PBR,PC+1	AAL	1
STA,LDA,CMP,SBC)		3.	1	1	0	1	PBR,PC+2	AAH	1
(8 OpCodes)		4.	1	1	0	1	PBR,PC+3	AAB	1
(4 bytes)		5.	1	1	1	0	AAB,AA	Data Low	1/0
(5 and 6 cycles)	(1)	5a.	1	1	1	0	AAB,AA+1	Data High	1/0
*3b. Absolute Long (JUMP)-al		1.	1	1	1	1	PBR,PC	OpCode	1
(JMP)		2.	1	1	0	1	PBR,PC+1	New PCL	1
(1 OpCode)		3.	1	1	0	1	PBR,PC+2	New PCH	1
(4 bytes)		4.	1	1	0	1	PBR,PC+3	New BR	1
(4 cycles)		1.	1	1	1	1	NEW PBR,PC	OpCode	1

ADDRESS MODE		CYCLE	$\overline{VP}$	$\overline{ML}$	VDA	VPA	ADDRESS BUS	DATA BUS	R/$\overline{W}$
*3c. Absolute Long (JUMP to		1.	1	1	1	1	PBR,PC	OpCode	1
Subroutine Long)-al		2.	1	1	0	1	PBR,PC+1	New PCL	1
(JSL)		3.	1	1	0	1	PBR,PC+2	New PCH	1
(1 OpCode)		4.	1	1	1	0	0,S	PBR	0
(4 bytes)		5.	1	1	0	0	0,S	IO	1
(7 cycles)		6.	1	1	0	1	PBR,PC+3	New PBR	1
		7.	1	1	1	0	0,S-1	PCH	0
		8.	1	1	1	0	0,S-2	PCL	0
		1.	1	1	1	1	NEW PBR,PC	New OpCode	1
4a. Direct-d		1.	1	1	1	1	PBR,PC	OpCode	1
(BIT,STZ,STY,LDY,		2.	1	1	0	1	PBR,PC+1	DO	1
CPY,CPX,STX,LDX,	(2)	2a.	1	1	0	0	PBR,PC+1	IO	1
ORA,AND,EOR,ADC,		3.	1	1	1	0	0,D+DO	Data Low	1/0
STA,LDA,CMP,SBC)	(1)	3a.	1	1	1	0	0,D+DO+1	Data High	1/0
(18 OpCodes)									
(2 bytes)									
(3,4 and 5 cycles)									
4b. Direct (R-M-W)-d		1.	1	1	1	1	PBR,PC	OpCode	1
(ASL,ROL,LSR,ROR		2.	1	1	0	1	PBR,PC+1	DO	1
DEC,INC,TSB,TRB)	(2)	2a.	1	1	0	0	PBR,PC+1	IO	1
(6 OpCodes)		3.	1	0	1	0	0,D+DO	Data Low	1
(2 bytes)	(1)	3a.	1	0	1	0	0,D+DO+1	Data High	1
(5,6,7 and 8 cycles)	(3)	4.	1	0	0	0	0,D+DO+1	IO	1
	(1)	5a.	1	0	1	0	0,D+DO+1	Data High	0
		5.	1	0	1	0	0,D+DO	Data Low	0
5. Accumulator-A		1.	1	1	1	1	PBR,PC	OpCode	1
(ASL,INC,ROL,DEC,LSR,ROR)		2.	1	1	0	0	PBR,PC+1	IO	1
(6 OpCodes)									
(1 byte)									
(2 cycles)									
6a. Implied i		1.	1	1	1	1	PBR,PC	OpCode	1
(DEY,INY,INX,DEX,NOP,		2.	1	1	0	0	PBR,PC+1	IO	1
XCE,TYA,TAY,TXA,TXS,									
TAX,TSX,TCS,TSC,TCD,									
TDC,TXY,TYX,CLC,SEC,									
CLI,SEI,CLV,CLD,SED)									
(25 OpCodes)									
(1 byte)									
(2 cycles)									
*6b. Implied i		1.	1	1	1	1	PBR,PC	OpCode	1
(XBA)		2.	1	1	0	0	PBR,PC+1	IO	1
(1 OpCode)		3.	1	1	0	0	PBR,PC+1	IO	1
(1 byte)									
(3 cycles)									
#6c. Wait-for-Interrupt						RDY			
(WAI)		1.	1	1	1	1	1 PBR,PC	OpCode	1
(1 OpCode)	(9)	2.	1	1	0	0	1 PBR,PC+1	IO	1
(1 byte)		3.	1	1	0	0	0 PBR,PC+1	IO	1
(3 cycles)	$\overline{IRQ}$,$\overline{NMI}$	1.	1	1	1	1	1 PBR,PC+1	IRQ(BRK)	1

ADDRESS MODE		CYCLE		$\overline{VP}$	$\overline{ML}$	VDA	VPA	RDY	ADDRESS BUS	DATA BUS	R/$\overline{W}$
#6d. Stop-the-Clock											
(STP)			1.	1	1	1	1	1	PBR,PC	OpCode	1
(1 OpCode)			2.	1	1	0	0	1	PBR,PC+1	IO	1
(1 byte)	RES-=1		3.	1	1	0	0	1	PBR,PC+1	IO	1
(3 cycles)	RES-=0		1c.	1	1	0	0	1	PBR,PC+1	RES(BRK)	1
	RES-=0		1b.	1	1	0	0	1	PBR,PC+1	RES(BRK)	1
	RES-=1		1a.	1	1	0	0	1	PBR,PC+1	RES(BRK)	1
(See 21a. Stack			1.	1	1	1	1	1	PBR,PC+1	BEGIN	1
Hardware Interrupt)											
7. Direct Indirect			1.	1	1	1	1		PBR,PC	OpCode	1
Indexed-(d),y											
(ORA,AND,EOR,ADC,			2.	1	1	0	1		PBR,PC+1	DO	1
STA,LDA,CMP,SBC)		(2)	2a.	1	1	0	0		PBR,PC+1	IO	1
(8 OpCodes)			3.	1	1	1	0		0,D+DO	AAL	1
(2 bytes)			4.	1	1	1	0		0,D+DO+1	AAH	1
(5,6,7 and 8 cycles)		(4)	4a.	1	1	0	0		DBR,AAJ,AAL+YL	IO	1
			5.	1	1	1	0		DBR,AA+Y	Data Low	1/0
		(1)	5a.	1	1	1	0		DBR,AA+Y+1	Data High	1/0
8. Direct Indirect			1.	1	1	1	1		PBR,PC	OpCode	1
Indexed Long-[d],y			2.	1	1	0	1		PBR,PC+1	DO	1
(ORA,AND,EOR,ADC,		(2)	2a.	1	1	0	0		PBR,PC+1	IO	1
STA,LDA,CMP,SBC)			3.	1	1	1	0		0,D+DO	AAL	1
(8 OpCodes)			4.	1	1	1	0		0,D+DO+1	AAH	1
(2 bytes)			5.	1	1	1	0		0,D+DO+2	AAB	1
(6,7 and 8 cycles)			6.	1	1	1	0		AAB,AA+Y	Data Low	1/0
		(1)	6a.	1	1	1	0		AAB,AA+Y+1	Data High	1/0
9. Direct Indexed			1.	1	1	1	1		PBR,PC	OpCode	1
Indirect-(d,x)			2.	1	1	0	1		PBR,PC+1	DO	1
(ORA,AND,EOR,ADC,		(2)	2a.	1	1	0	0		PBR,PC+1	IO	1
STA,LDA,CMP,SBC)			3.	1	1	0	0		PBR,PC+1	IO	1
(8 OpCodes)			4.	1	1	1	0		0,D+DO+X	AAL	1
(2 bytes)			5.	1	1	1	0		O,D+DO+X+1	AAH	1
(6,7 and 8 cycles)			6.	1	1	1	0		DBR,AA	Data Low	1/0
		(1)	6a.	1	1	1	0		DBR,AA+1	Data High	1/0
10a.Direct,X-d,x			1.	1	1	1	1		PBR,PC	OpCode	1
(BIT,STZ,STY,LDY,			2.	1	1	0	1		PBR,PC+1	DO	1
ORA,AND,EOR,ADC,		(2)	2a.	1	1	0	0		PBR,PC+1	IO	1
STA,LDA,CMP,SBC)			3.	1	1	0	0		PBR,PC+1	IO	1
(11 OpCodes)			4.	1	1	1	0		0,D+DO+X	Data Low	1/0
(2 bytes)		(1)	4a.	1	1	1	0		0,D+DO+X+1	Data High	1/0
(4,5 and 6 cycles)											
10b.Direct,X(R-M-W)-d,x			1.	1	1	1	1		PBR,PC	OpCode	1
(ASL,ROL,LSR,ROR,			2.	1	1	0	1		PBR,PC+1	DO	1
DEC,INC)		(2)	2a.	1	1	0	0		PBR,PC+1	IO	1
(6 OpCodes)			3.	1	1	0	0		PBR,PC+1	IO	1
(2 bytes)			4.	1	0	1	0		0,D+DO+X	Data Low	1
(6,7,8 and 9 cycles)		(1)	4a.	1	0	1	0		0,D+DO+X+1	Data High	1
		(3)	5.	1	0	0	0		0,D+DO+X+1	IO	1
		(1)	6a.	1	0	1	0		0,D+DO+X+1	Data High	0
			6.	1	0	1	0		0,D+DO+X	Data Low	0

ADDRESS MODE		CYCLE	$\overline{VP}$	$\overline{ML}$	VDA	VPA	ADDRESS BUS	DATA BUS	R/$\overline{W}$
11. Direct,Y-d,y		1.	1	1	1	1	PBR,PC	OpCode	1
(STX,LDX)		2.	1	1	0	1	PBR,PC+1	DO	1
(2 OpCodes)	(2)	2a.	1	1	0	0	PBR,PC+1	IO	1
(2 bytes)		3.	1	1	0	0	PBR,PC+1	IO	1
(4,5 and 6 cycles)		4.	1	1	1	0	0,D+DO+Y	Data Low	1/0
	(1)	4a.	1	1	1	0	0,D+DO+Y+1	Data High	1/0
12a.Absolute,X-a,x		1.	1	1	1	1	PBR,PC	OpCode	1
(BIT,LDY,STZ,		2.	1	1	0	1	PBR,PC+1	AAL	1
ORA,AND,EOR,ADC,		3.	1	1	0	1	PBR,PC+2	AAH	1
STA,LDA,CMP,SBC)	(4)	3a.	1	1	0	0	DBR,AAH,AAL+XL	IO	1
(11 OpCodes)		4.	1	1	1	0	DBR,AA+X	Data Low	1/0
(3 bytes)	(1)	4a.	1	1	1	0	DBR,AA+X+1	Data High	1/0
(4,5 and 6 cycles)									
12b.Absolute,X(R-M-W)-a,x		1.	1	1	1	1	PBR,PC	OpCode	1
(ASL,ROL,LSR,ROR,		2.	1	1	0	1	PBR,PC+1	AAL	1
DEC,INC)		3.	1	1	0	1	PBR,PC+2	AAH	1
(6 OpCodes)		4.	1	1	0	0	DBR,AAH,AAL+XL	IO	1
(3 bytes)		5.	1	0	1	0	DBR,AA+X	Data Low	1
(7 and 9 cycles)	(1)	5a.	1	0	1	0	DBR,AA+X+1	Data High	1
	(3)	6.	1	0	0	0	DBR,AA+X+1	IO	1
	(1)	7a.	1	0	1	0	DBR,AA+X+1	Data High	0
		7.	1	0	1	0	DBR,AA+X	Data Low	0
*13. Absolute Long,X-al,x		1.	1	1	1	1	PBR,PC	OpCode	1
(ORA,AND,EOR,ADC,		2.	1	1	0	1	PBR,PC+1	AAL	1
STA,LDA,CMP,SBC)		3.	1	1	0	1	PBR,PC+2	AAH	1
(8 OpCodes)		4.	1	1	0	1	PBR,PC+3	AAB	1
(4 bytes)		5.	1	1	1	0	AAB,AA+X	Data Low	1/0
(5 and 6 cycles)	(1)	5a.	1	1	1	0	AAB,AA+X+1	Data High	1/0
14. Absolute,Y-a,y		1.	1	1	1	1	PBR,PC	OpCode	1
(LDX,ORA,AND,EOR,ADC,		2.	1	1	0	1	PBR,PC+1	AAL	1
STA,LDA,CMP,SBC)		3.	1	1	0	1	PBR,PC+2	AAH	1
(9 OpCodes)	(4)	3a.	1	1	0	0	DBR,AAH,AAL+Y	IL	1
(3 bytes)		4.	1	1	1	0	DBR,AA+Y	Data Low	1/0
(4,5 and 6 cycles)	(1)	4a.	1	1	1	0	DBR,AA+Y+1	Data High	1/0
15. Relative-r		1.	1	1	1	1	PBR,PC	OpCode	1
(BPL,BMI,BVC,BVS,BCC,		2.	1	1	0	1	PBR,PC+1	OFF	1
BCS,BNE,BEQ,BRA)	(5)	2a.	1	1	0	0	PBR,PC+1	IO	1
(9 OpCodes)	(6)	2b.	1	1	0	0	PBR,PC+1	IO	1
(2 bytes)		1.	1	1	1	1	PBR,PC+Offset	OpCode	1
(2,3 and 4 cycles)									
*16. Relative Long-rl		1.	1	1	1	1	PBR,PC	OpCode	1
(BRL)		2.	1	1	0	1	PBR,PC+1	OFF Low	1
(1 OpCode)		3.	1	1	0	1	PBR,PC+2	OFF High	1
(3 bytes)		4.	1	1	0	0	PBR,PC+2	IO	1
(4 cycles)		1.	1	1	1	1	PBR,PC+Offset	OpCode	1
17a.Absolute Indirect-(a)		1.	1	1	1	1	PBR,PC	OpCode	1
(JMP)		2.	1	1	0	1	PBR,PC+1	AAL	1
(1 OpCode)		3.	1	1	0	1	PBR,PC+2	AAH	1
(3 bytes)		4.	1	1	1	0	0,AA	New PCL	1
(5 cycles)		5.	1	1	1	0	0,AA+1	New PCH	1
		1.	1	1	1	1	PBR,NEW PC	OpCode	1

ADDRESS MODE		CYCLE	$\overline{VP}$	$\overline{ML}$	VDA	VPA	ADDRESS BUS	DATA BUS	R/$\overline{W}$
*17b.Absolute Indirect-(a)		1.	1	1	1	1	PBR,PC	OpCode	1
(JML)		2.	1	1	0	1	PBR,PC+1	AAL	1
(1 OpCode)		3.	1	1	0	1	PBR,PC+2	AAH	1
(3 bytes)		4.	1	1	1	0	0,AA	New PCL	1
(6 cycles		5.	1	1	1	0	0,AA+1	New PCH	1
		6.	1	1	1	0	0,AA+2	New PBR	1
		1.	1	1	1	1	NEW PBR,PC	OpCode	1
#18. Direct Indirect-(d)		1.	1	1	1	1	PBR,PC	OpCode	1
(ORA,AND,EOR,ADC,		2.	1	1	0	1	PBR,PC+1	DO	1
STA,LDA,CMP,SBC)	(2)	2a.	1	1	0	0	PBR,PC+1	IO	1
(8 OpCodes)		3.	1	1	1	0	0,D+DO	AAL	1
(2 bytes)		4.	1	1	1	0	0,D+DO+1	AAH	1
(5,6 and 7 cycles)		5.	1	1	1	0	DBR,AA	Data Low	1/0
	(1)	5a.	1	1	1	0	PBR,AA+1	Data High	1/0
*19. Direct Indirect Long-[d]		1.	1	1	1	1	PBR,PC	OpCode	1
(ORA,AND,EOR,ADC		2.	1	1	0	1	PBR,PC+1	DO	1
STA,LDA,CMP,SBC	(2)	2a.	1	1	0	0	PBR,PC+1	IO	1
(8 OpCodes)		3.	1	1	1	0	O,D+DO	AAL	1
(2 bytes)		4.	1	1	1	0	O,D+DD+1	AAH	1
(6,7 and 8 cycles)		5.	1	1	1	0	O,D+DO+2	AAB	1
		6.	1	1	1	0	AAB,AA	Data Low	1/0
	(1)	6a.	1	1	1	0	AAB,AA+1	Data High	1/0
20a.Absolute Indexed Indirect-									
(a,x)		1.	1	1	1	1	PBR,PC	OpCode	1
(JMP)		2.	1	1	0	1	PBR-PC+1	AAL	1
(1 OpCode)		3.	1	1	0	1	PBR-PC+2	AAH	1
(3 bytes)		4.	1	1	0	0	PBR,PC+2	IO	1
(6 cycles)		5.	1	1	0	1	PBR,AA+X	New PCL	1
		6.	1	1	0	1	PBR,AA+X+1	New PCH	1
		1.	1	1	1	1	PBR,NEW PC	OpCode	1
*20b.Absolute Indexed Indirect-		1.	1	1	1	1	PBR,PC	OpCode	1
(a,x)		2.	1	1	0	1	PBR,PC+1	AAL	1
(JSR)		3.	1	1	1	0	O,S	PCH	0
(1 OpCode)		4.	1	1	1	0	O,S-1	PCL	0
(3 bytes)		5.	1	1	0	1	PBR,PC+2	AAH	1
(8 cycles)		6.	1	1	0	0	PBR,PC+2	IO	1
		7.	1	1	0	1	PBR,AA+X	New PCL	1
		8.	1	1	0	1	PBR,AA+X+1	New PCH	1
		1.	1	1	1	1	PBR,NEW PC	New OpCode	1
21a.Stack (Hardware		1.	1	1	1	1	PBR,PC	IO	1
Interrupts)-s	(3)	2.	1	1	0	0	PBR,PC	IO	1
(IRQ,NMI,ABORT,RES)	(7)	3.	1	1	1	0	O,S	PBR	0
(4 hardware interrupts)	(10)	4.	1	1	1	0	O,S-1	PCH	0
(0 bytes)	(10)	5.	1	1	1	0	O,S-2	PCL	0
(7 and 8 cycles)	(10)(11)	6.	1	1	1	0	O,S-3	P	0
		7.	0	1	1	0	O,VA	AAVL	1
		8.	0	1	1	0	O,VA+1	AAVH	1
		1.	1	1	1	1	O,AAV	New OpCode	1

ADDRESS MODE		CYCLE	$\overline{VP}$	$\overline{ML}$	VDA	VPA	ADDRESS BUS		DATA BUS	R/$\overline{W}$
21b.Stack (Software		1.	1	1	1	1	PBR,PC		OpCode	1
Interrupts)-s	(3)	2.	1	1	0	1	PBR,PC+1		Signature	1
(BRK,COP)	(7)	3.	1	1	1	0	0,S		PBR	0
(2 OpCodes)		4.	1	1	1	0	0,S-1		PCH	0
(2 bytes)		5.	1	1	1	0	0,S-2		PCL	0
(7 and 8 cycles)		6.	1	1	1	0	0,S-3	(16)	P	0
		7.	0	1	1	0	0,VA		AAVL	1
		8.	0	1	1	0	0,VA+1		AAVH	1
		1.	1	1	1	1	0,AAV		New OpCode	1
21c.Stack (Return from		1.	1	1	1	1	PBR,PC		OpCode	1
Interrupt)-s		2.	1	1	0	0	PBR,PC+1		IO	1
(RTI)	(3)	3.	1	1	0	0	PBR,PC+1		IO	1
(1 Op Code)		4.	1	1	1	0	0,S+1		P	1
(1 byte)		5.	1	1	1	0	0,S+2		PCL	1
(6 and 7 cycles)		6.	1	1	1	0	0,S+3		PCH	1
(different order from	(7)	7.	1	1	1	0	0,S+4		PBR	1
N6502)		1.	1	1	1	1	PBR,PC		New OpCode	1
21d.Stack (Return from		1.	1	1	1	1	PBR,PC		OpCode	1
Subroutine)-s		2.	1	1	0	0	PBR,PC+1		IO	1
(RTS)		3.	1	1	0	0	PBR,PC+1		IO	1
(1 Op Code)		4.	1	1	1	0	0,S+1		PCL	1
(1 byte)		5.	1	1	1	0	0,S+2		PCH	1
(6 cycles)		6.	1	1	0	0	NEW PC-1		IO	1
		1.	1	1	1	1	PBR,PC		OpCode	1
*21e.Stack (Return from		1.	1	1	1	1	PBR,PC		OpCode	1
Subroutine Long)-s		2.	1	1	0	0	PBR,PC+1		IO	1
(RTL)		3.	1	1	0	0	PBR,PC+1		IO	1
(1 Op Code)		4.	1	1	1	0	0,S+1		New PCL	1
(1 byte)		5.	1	1	1	0	0,S+2		New PCH	1
(6 cycles)		6.	1	1	1	0	0,S+3		New PBR	1
		1.	1	1	1	1	NEW PBR,PC		New OpCode	1
21f.Stack (Push)-s		1.	1	1	1	1	PBR/PC		OpCode	1
(PHP,PHA,PHY,PHX,		2.	1	1	0	0	PBR,PC+1		IO	1
PHD,PHK,PHB)	(1)(11)	3a.	1	1	1	0	0,S		REG High	1
(7 Op Codes)		3.	1	1	1	0	0,S-1		REG Low	1
(1 byte)										1
(3 and 4 cycles)										
21g.Stack (Pull)-s		1.	1	1	1	1	PBR,PC		OpCode	1
(PLP,PLA,PLY,PLX,PLD,PLB)		2.	1	1	0	0	PBR,PC+1		IO	1
(Different than N6502)		3.	1	1	0	0	PBR,PC+1		IO	1
(6 Op Codes)		4.	1	1	1	0	0,S+1		REG Low	1
(1 byte)	(1)	4a.	1	1	1	0	0,S+2		REG High	1
(4 and 5 cycles)										
*21h.Stack (Push Effective		1.	1	1	1	1	PBR,PC		OpCode	1
Indirect Address)-s		2.	1	1	0	1	PBR,PC+1		DO	1
(PEI)	(2)	2a.	1	1	0	0	PBR,PC+1		IO	1
(1 Op Code)		3.	1	1	1	0	0,D+DO		AAL	1
(2 bytes)		4.	1	1	1	0	0,D+DO+1		AAH	1
(6 and 7 cycles)		5.	1	1	1	0	0,S-1		AAH	0
		6.	1	1	1	0	0,S-1		AAL	0

ADDRESS MODE		CYCLE	$\overline{VP}$	$\overline{ML}$	VDA	VPA	ADDRESS BUS	DATA BUS	R/$\overline{W}$
*21i.Stack (Push Effective		1.	1	1	1	1	PBR,PC	OpCode	1
Absolute Address)-s		2.	1	1	0	1	PBR,PC+1	AAL	1
(PEA)		3.	1	1	0	1	PBR,PC+2	AAH	1
(1 Op Code)		4.	1	1	1	0	O,S	AAH	0
(3 bytes)		5.	1	1	1	0	O,S-1	AAL	0
(5 cycles)									
*21j.Stack (Push Effective		1.	1	1	1	1	PBR,PC	OpCode	1
Program Counter Relative		2.	1	1	0	1	PBR,PC+1	OFF Low	1
Address)-s		3.	1	1	0	1	PBR,PC+2	OFF High	1
(PER)		4.	1	1	0	0	PBR,PC+2	IO	1
(1 Op Code)		5.	1	1	1	0	O,S	PCH+OFF+ Carry	0
(3 bytes)									
(6 cycles)		6.	1	1	1	0	O,S-1	PCL+OFF	0
*22. Stack Relative-d,s		1.	1	1	1	1	PBR,PC	OpCode	1
(ORA,AND,EOR,ADL,		2.	1	1	0	1	PBR,PC+1	SO	1
STA,LDA,CMP,SBC)		3.	1	1	0	0	PBR,PC+1	IO	1
(8 Op Codes)		4.	1	1	1	0	O,S+SO	Data Low	1/0
(2 bytes)	(1)	4a.	1	1	1	0	O,S+SO+1	Data High	1/0
(4 and 5 cycles)									
*23. Stack Relative Indirect		1.	1	1	1	1	PBR,PC	OpCode	1
Indexed-(d,s),y		2.	1	1	0	1	PBR,PC+1	SO	1
(ORA,AND,EOR,ADC,STA,LDA,		3.	1	1	0	0	PBR+PC+1	IO	1
CMP,SBC)		4.	1	1	1	0	O,S+SO	AAL	1
(8 Op Codes)		5.	1	1	1	0	O,S+SO+1	AAH	1
(2 bytes)		6.	1	1	0	0	O,S+SO+1	IO	1
(7 and 8 Cycles)		7.	1	1	1	0	DBR,AA+Y	Data Low	1/0
	(1)	7a.	1	1	1	0	DBR,AA+Y+1	Data High	1/0
*24a.Block Move Positive		1.	1	1	1	1	PBR,PC	OpCode	1
(forward)-xyc		2.	1	1	0	1	PBR,PC+1	DBA	1
(MVP)		3.	1	1	0	1	PBR,PC+2	SBA	1
(1 Op Code)	N-2	4.	1	1	1	0	SBA,X	SRC Data	1
(3 bytes)	Byte	5.	1	1	1	0	DBA,Y	DEST Data	0
(7 cycles)	C=2	6.	1	1	0	0	DBA,Y	IO	1
		7.	1	1	0	0	DBA,Y	IO	1
x=Source Address									
y=Destination		1.	1	1	1	1	PBR,PC	OpCode	1
c=# of bytes to move-1		2.	1	1	0	1	PBR,PC+1	DBA	1
x,y Decrement		3.	1	1	0	1	PBR,PC+2	SBA	1
MVP is used when the	N-1	4.	1	1	1	0	SBA,X-1	SRC Data	1
dest. start address	Byte	5.	1	1	1	0	DBA,Y-1	DEST Data	0
is higher (more	C=1	6.	1	1	0	0	DBA,Y-1	IO	1
positive) than the source		7.	1	1	0	0	DBA,Y-1	IO	1
start address.									
		1.	1	1	1	1	PBR,PC	Op Code	1
FFFFFF		2.	1	1	0	1	PBR,PC+1	DBA	1
^ Dest Start	N Byte	3.	1	1	0	1	PBR,PC+2	SBA	1
| |	Last	4.	1	1	1	0	SBA,X-2	SRC Data	1
| Source Start	C=0	5.	1	1	1	0	DBA,Y-2	DEST Data	0
| | Dest End		6.	1	1	0	0	DBA,Y-2	IO	1
| Source End		7.	1	1	0	0	DBA,Y-2	IO	1
000000		1.	1	1	1	1	PBR,PC+3	New OpCode	1

ADDRESS MODE	CYCLE	$\overline{VP}$	$\overline{ML}$	VDA	VPA	ADDRESS BUS	DATA BUS	R/$\overline{W}$
*24b.Block Move Negative (backward)-xyc (MVN) (1 Op Code) (3 bytes) (7 cycles)	1.	1	1	1	1	PBR,PC	OpCode	1
	2.	1	1	0	1	PBR,PC+1	DBA	1
	3.	1	1	0	1	PBR,PC+2	SBA	1
N-2 Byte C=2	4.	1	1	1	0	SBA,X	SRC Data	1
	5.	1	1	1	0	DBA,Y	DEST Data	0
	6.	1	1	0	0	DBA,Y	IO	1
	7.	1	1	0	0	DBA,Y	IO	1
x=Source Address y=Destination c=# of bytes to move-1 x,y Increment MVN is used when the dest. start address is lower (more negative) than the source start address.	1.	1	1	1	1	PBR,PC	OpCode	1
	2.	1	1	0	1	PBR,PC+1	DBA	1
	3.	1	1	0	1	PBR,PC+2	SBA	1
N-1 Byte C=1	4.	1	1	1	0	SBA,X+1	SRC Data	1
	5.	1	1	1	0	DBA,Y+1	DEST Data	0
	6.	1	1	0	0	DBA,Y+1	IO	1
	7.	1	1	0	0	DBA,Y+1	IO	1
FFFFF Source End N Byte Dest.End C=0 Source Start Dest.Start 000000	1.	1	1	1	1	PBR,PC	OpCode	1
	2.	1	1	0	1	PBR,PC+1	DBA	1
	3.	1	1	0	1	PBR,PC+2	SBA	1
	4.	1	1	1	0	SBA,X+2	SRC Data	1
	5.	1	1	1	0	DBA,Y+2	DEST Data	0
	6.	1	1	0	0	DBA,Y+2	IO	1
	7.	1	1	0	0	DBA,Y+2	IO	1
	1.	1	1	1	1	PBR,PC+3	New OpCode	1

Notes: Be aware that notes #4-7, 9 and 10 apply to the W65C02 and W65C816. All other notes apply to the W65C816 only.

1. Add 1 byte (for immediate only) for M=0 or X=0 (i.e. 16-bit data), add 1 cycle for M=0 or X=0.
2. Add 1 cycle for direct register low (DL) not equal 0.
3. Special case for aborting instruction. This is the last cycle which may be aborted or the Status, PBR or DBR registers will be udpated.
4. Add 1 cycle for indexing across page boundaries, or write, or X=0. When X=1 or in the emulation mode, this cycle contains invalid addresses.
5. Add 1 cycle if branch is taken.
6. Add 1 cycle if branch is taken across page boundaries in 6502 emulation mode (E=1).
7. Subtract 1 cycle for 6502 emulation mode (E=1).
8. Add 1 cycle for REP, SEP.
9. Wait at cycle 2 for 2 cycles after NMI- or IRQ- active input.
10. R/W- remains high during Reset.
11. BRK bit 4 equals "0" in Emulation mode.
12. PHP and PLP.
13. Some OpCodes shown are compatible only with the W65C816.
14. VDA and VPA are not valid outputs on the W65C02 but are valid on the W65C816. The two signals, VDA and VPA, are included to point out the upward compatibility to the W65C816. When VDA and VPA are both a one level, this is equivalent to SYNC being a one level.
15. The PBR is only applicable to the W65C816.
16. COP Latches.

AAB	Absolute Address Bank
AAH	Absolute Address High
AAL	Absolute Address Low
AAVH	Absolute Address Vector High
AAVL	Absolute Address Vector Low
C	Accumulator
D	Direct Register
DBA	Destination Bank Address
DBR	Data Bank Register
DEST	Destination
DO	Direct Offset
IDH	Immediate Data High
IDL	Immediate Data Low
IO	Internal Operation
OFF	Offset
P	Status Register
PBR	Program Bank Register
PC	Program Counter
PCH	Program Counter High
PCL	Program Counter Low
R-M-W	Read-Modity-Write
REG	Register
S	Stack Address
SBA	Source Bank Address
SRC	Source
SO	Stack Offset
VA	Vector Address
x,y	Index Register

* = New W65C816/802 Addressing Modes
\# = New W65C02 Addressing Modes
Blank = NMOS 6502 Addressing Modes

SECTION 7

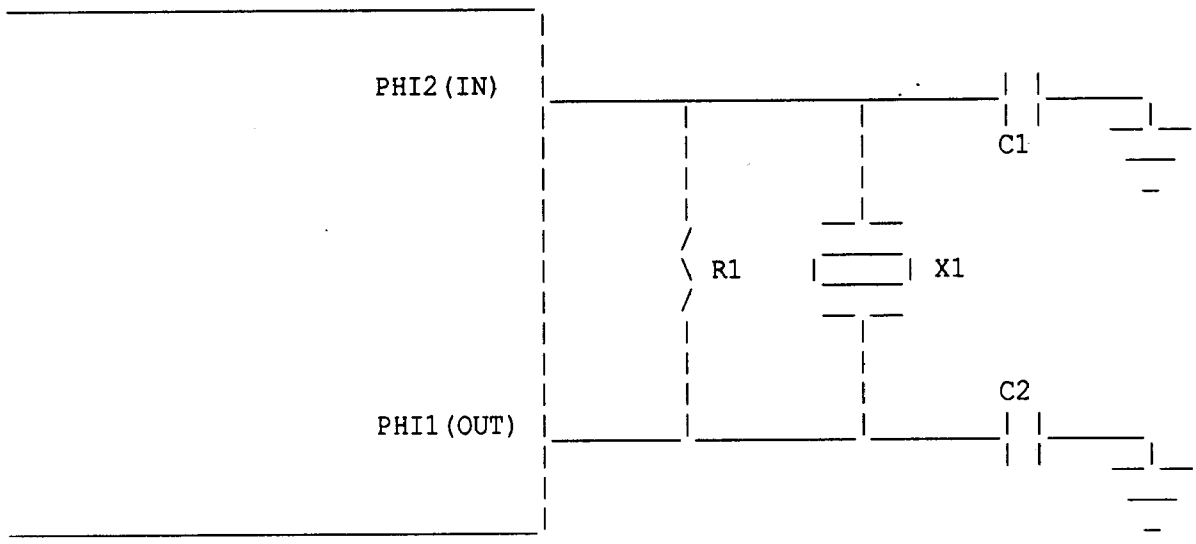
CAVEATS

Table 7-1 Microprocessor Operational Enhancements

Function	NMOS 6502	W65C02
Indexed addressing across page boundry.	Extra read of invalid address.	Extra read of last instruction byte.
Execution of invalid opcodes.	Some terminate only by reset. Results are undefined	All are NOP's (reserved for future use). Op Code — Bytes — Cycles X2 — 2 — 2 X3,X7,0B-BB, EB,FB — 1 — 1 44 — 2 — 3 54, D4, F4 — 2 — 4 5C — 3 — 8 DC, FC — 3 — 4
Jump indirect, operand = XXFF.	Page address does not increment.	Page address increments, one additional cycle.
Read/modify/write instruct at effective address.	One read and two write cycles.	Two read and one write cycle.
Decimal flag.	Indeterminate after reset.	Initialized to binary mode (D=0) after reset and interrupts.
Flags after decimal operation.	Invalid N, V and Z flags.	Valid flags. One additional cycle.
Interrupt after fetch of BRK instruction.	Interrupt vector is loaded; BRK vector is ignored.	BRK is executed, then interrupt is executed.
Ready.	Input.	Bidirectional, WAI instruction pulls low.
Read/modify/write instructions absolute indexed in same page.	Seven cycles.	Six cycles.

Table 7-2 Microprocessor Hardware Enhancements

Function	NMOS 6502	W65C02
Oscillator.	Requires external active components.	Crystal or RC network will oscillate when connected between PHI2(IN) and PHI1(OUT).
Assertion of Ready (RDY) during write operations.	Ignored.	Stops processor during PHI2, and bidirectional WAI instruction pulls RDY low.
Clock inputs.	Two non-overlapping clock inputs (PHI1 and PHI2) are required.	PHI2(IN) is the only required clock.
Unused input-only pins	Must be connected to low impedance signal to avoid noise problems.	Connected internally by a high-resistance to VDD (approximately 1 Megohm).



C1, C2 = 51 pF
R1 = 200K
X1 = 1MHz

Figure 7-1 Crystal Circuit for Internal Oscillator

SECTION 8

TOOLBOX DESIGN SYSTEM

In-Circuit Emulator for the W65C02, W65C816, W65C134 and W65C265

8.1 Features

* Hardware Breakpoint on any combination of address, data, read/write, instruction fetch, and user-defined I/O
* Instruction single-stepping and single cycle capability
* Target memory display and editor
* Microprocessor register display and editor
* 2048-cycle real-time trace buffer providing logic analyzer-type features
* W65C816/W65C02 cycle disassembler
* Can ICE targets with clock rates from 32 KHz to 4 Mhz
* Ability to save and load target memory to/from disk files
* 64K RAM/ROM overlay in target system address space
* C-language based ICE software

8.2 Product Overview

The Toolbox Design System In-Circuit Emulator (ICE) is a hardware/software debugger for WDC's W65C series of microprocessors and microcontrollers. These include the W65C02, W65C816, W65C134 (C02 core) and the W65C265 (816 core).

An Apple IIgs acts as a host system. An interface card connected to the Toolbox plugs into one of the Apple IIgs expansion slots. ICE Software running on the host is used to control the Toolbox and to control ICEIng.

To "ICE" a system, a cable from the Toolbox plugs into the target system microprocessor socket. From the host system, the user can single step the microprocessor or run at the normal clock rate. When runnning at the normal clock rate, the Toolbox is "transparent" to the target system.

Manufactured by The Western Design Center, Inc.

Sales and Technical Support by:

Hall Effects
PO Box 41624
Mesa, AZ 85274-1624 USA
Phone (602 731-9516

Apple IIgs is a trademark of Apple Computer, Inc.

8.3 Publications supporting the W65C02

Programming the 65816
William Labiak
SYBEX, Inc.
2344 Sixth Street
Berkeley, CA 94710

The 6502, 65C02 and 65816 Handbook
Steve Hendrix
Weber Systems, Inc.
8437 Mayfield Road
Chesterland, OH 44026

65816/65802 Assembly Language Programming
Michael Fisher
Osborne McGraw-Hill
2600 Tenth Street
Berkeley, CA 94710

Programming the 65816 Including the 6502, 65C02, and 65802
David Eyes and Ron Lichty
Prentice Hall Press
A Division of Simon & Schuster, Inc.
Gulf & Western Bldg.
One Gulf & Western Plaza
New York, NY 10023