W65C02S DATA SHEET

WDC reserves the right to make changes at any time without notice in order to improve design and supply the best possible product. Information contained herein is provided gratuitously and without liability, to any user. Reasonable efforts have been made to verify the accuracy of the information but no guarantee whatsoever is given as to the accuracy or as to its applicability to particular uses. In every instance, it must be the responsibility of the user to determine the suitability of the products for each application. WDC products are not authorized for use as critical components in life support devices or systems. Nothing contained herein shall be construed as a recommendation to use any product in violation of existing patents or other rights of third parties. The sale of any WDC product is subject to all WDC Terms and Conditions of Sales and Sales Policies, copies of which are available upon request.

TABLE OF CONTENTS

INTRODUCTION

The W65C02S Microprocessor Family offers hardware and software compatibility. The small die size makes the W65C02S an excellent choice as an embedded core microprocessor in system-on-a chip designs.

KEY FEATURES OF THE W65C02S

- Totally static operation
- Advanced CMOS family of compatible microprocessors
- Wide operating voltage range (1.2-5.25v)
- Low power consumption
- Enhanced instruction:
  - 70 microprocessor instructions
  - 212 operational codes
  - 16 addressing modes
- 64K-byte addressable memory
- Stop-the-Clock (STP) and WAIT instructions for low power operation
- BE pin controls I/O state of data bus, address bus and RWB
- W65C02S has additional bit-manipulation instructions RMB, SMB, BMB5, BMBR not available on W65C02 and W65C816
- Developer System available directly from WDC
- W65CowDB Developer Board
- W65C02SDS Software Development System
-

SECTION 1

W65C02S 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. 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 (NMIB and IRQB). The stack allows simple implementation of nested subroutines and multiple level interrupts. The stack pointer is initialized by the user's software.

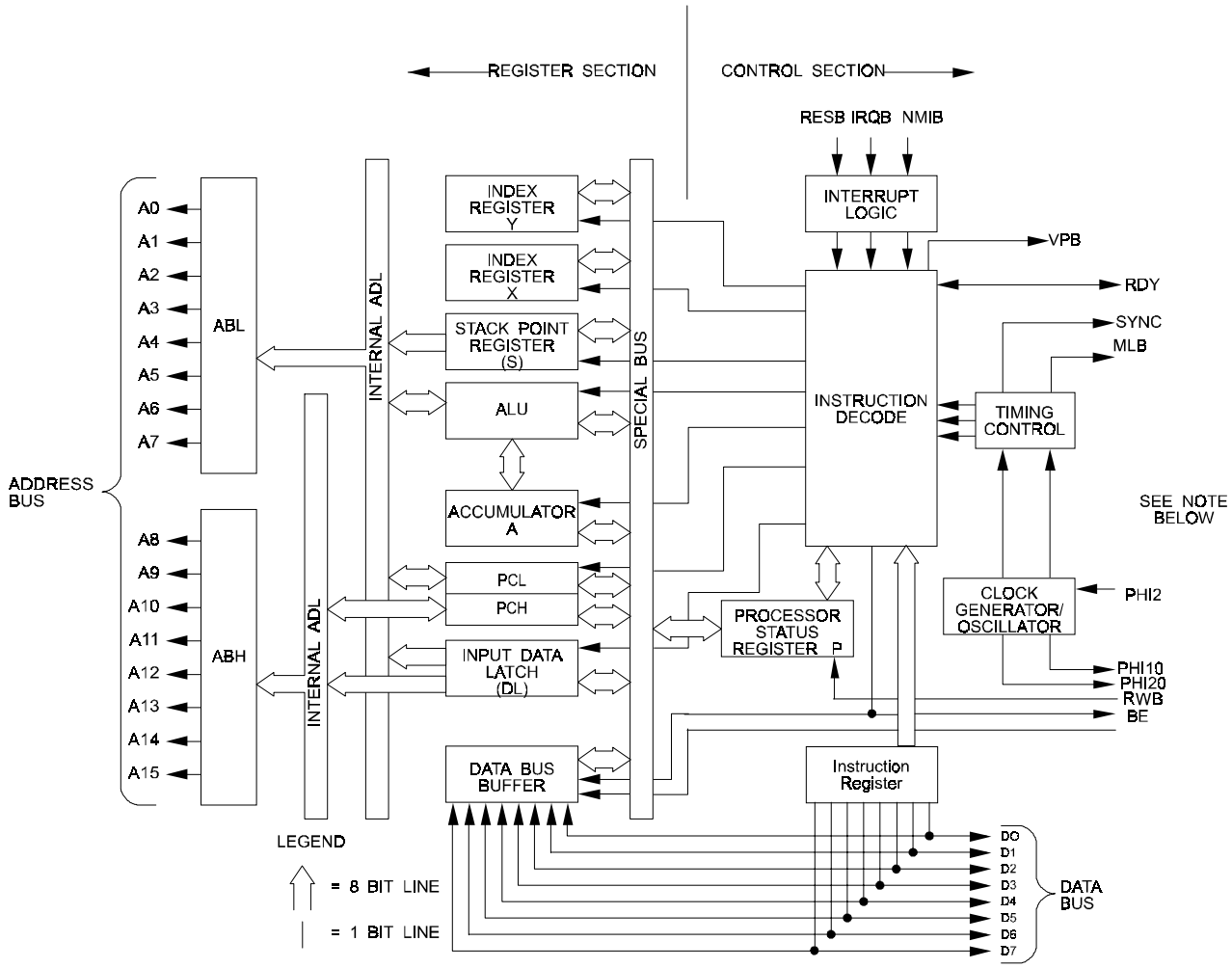


Figure 1-1 W65C02S Internal Architecture Simplified Block Diagram

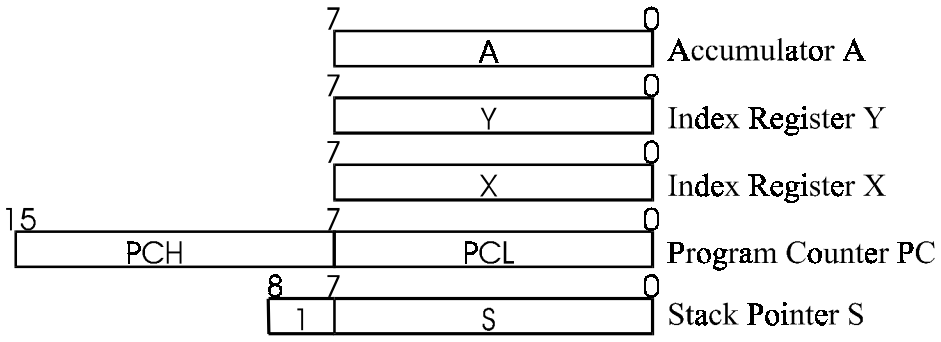


Figure 1-2 W65C02S Microprocessor Programming Model

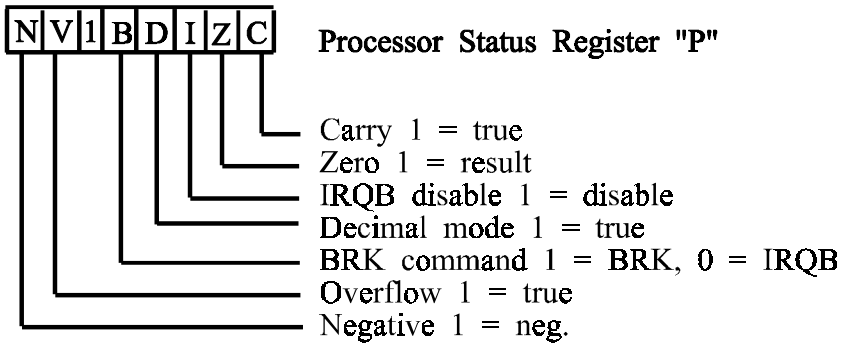


Figure 1-3 W65C02S Status Register Coding

SECTION 2

PIN FUNCTION DESCRIPTION

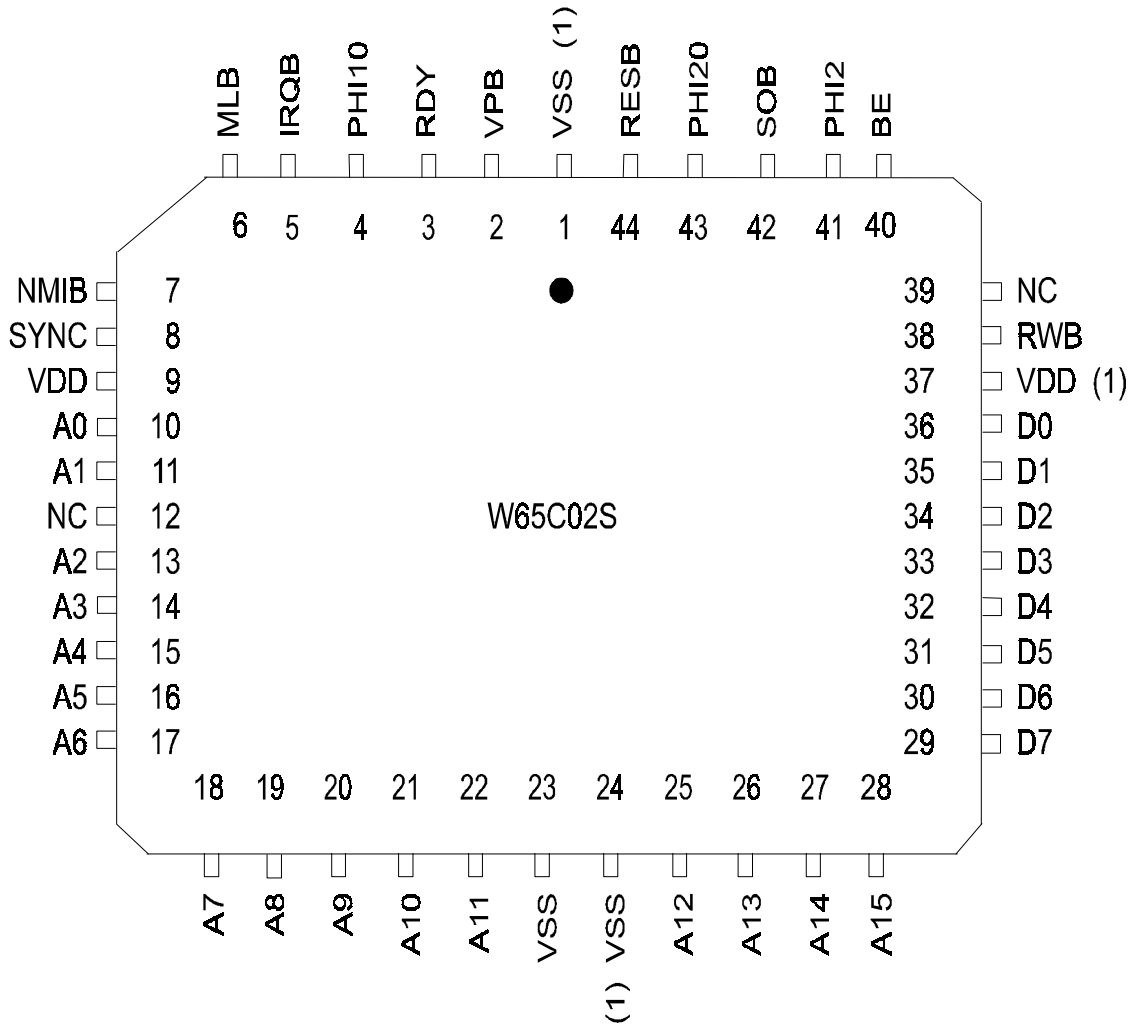


Figure 2-1 W65C02S 44 Pin PLCC Pinout

(1) Power supply pins not available on the 40 pin version. These power supply pins have been added for improved performance. All power supply pins must be connected.

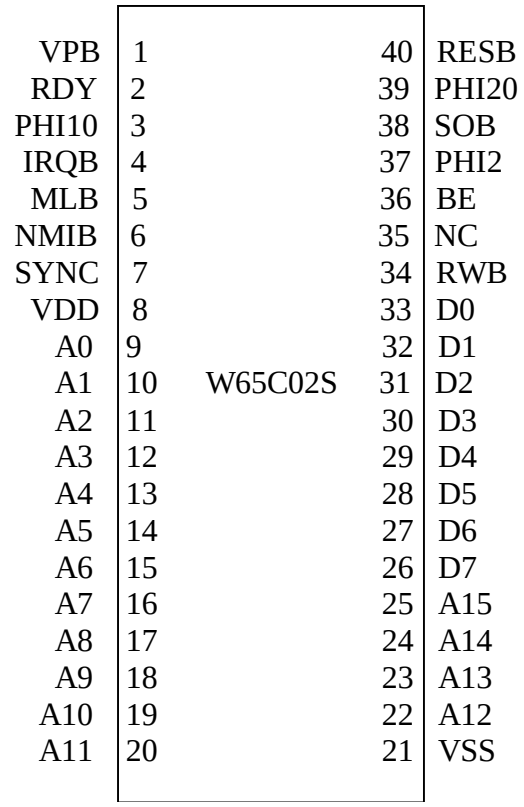


Figure 2-2 W65C02S Pin PDIP Pinout

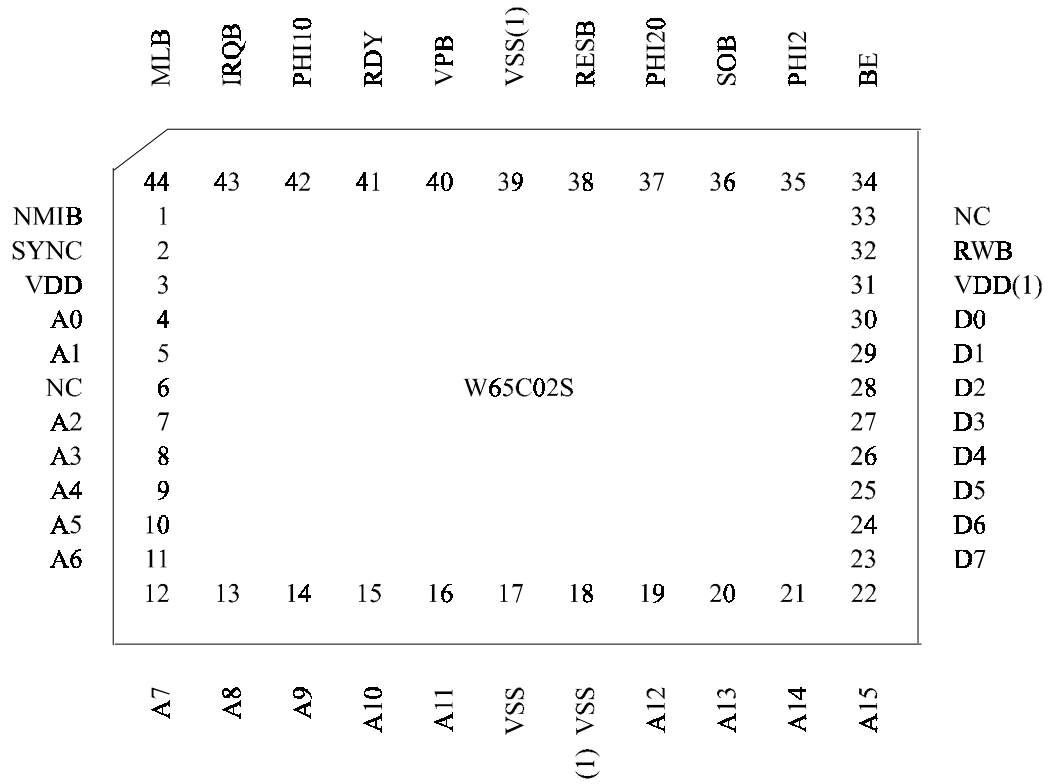


Figure 2-3 W65C02S 44 PIN QFP Pinout

(1) Power supply pins not available on the 40 pin version. These power supply pins have been added for improved performance. All power supply pins must be connected.

Table 2-1 Pin Function Table

Pin	Description
A0-A15	Address Bus
PHI2	Phase 2 In Clock
D0-D7	Data Bus
IRQB	Interrupt Request
MLB	Memory Lock
NC	No Connection
NMIB	Non-Maskable Interrupt
PHI10	Phase 1 Out Clock
PHI20	Phase 2 Out Clock
RDY	Ready
RESB	Reset
RWB	Read/Write
SOB	Set Overflow
SYNC	Synchronize
VPB	Vector Pull
BE	Bus Enable
VDD	Positive Power Supply (+5 volts)
VSS	Internal Logic Ground

2.1 Address Bus

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.

2.2 Data Bus

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

2.3 Interrupt Request

This TTL compatible signal requests that an interrupt sequence begin within the microprocessor. The IRQB is sampled during PHI2 high, if the interrupt disable 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 on the stack. The microprocessor will then set the interrupt disable flag high so that no further interrupts can occur. At the end of this cycle, the PCL will be loaded from address FFFE, and PCH 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.. The IRQB signal going low causes 3 bytes of information to be pushed onto the stack before jumping to the interrupt handler. The first byte is the high byte in the Program Counter. The second byte is the Program Counter low byte. The third byte is the status register value. These values are used to return the processor to it's original state once the interrupt has been handled.

2.4 Memory Lock (MLB)

In a multiprocessor system, MLB indicates the need to defer the rearbitration of the next bus cycle to ensure the integrity of read-modify-write instructions, MLB 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 (NMIB)

A negative-going edge on this input requests that a non-maskable interrupt sequence be generated within the microprocessor. The NMIB is sampled during PHI2 high; 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 NMIB is low and a negative-going edge has not occurred since the last non-maskable interrupt. The NMIB signal going low causes 3 bytes of information to be pushed onto the stack before jumping to the interrupt handler. The first byte is the high byte in the Program Counter. The second byte is the Program Counter low byte. The third byte is the status register value. These values are used to return the processor to it's original state once the interrupt has been handled.

2.6 Phase 1 (PHI10)

Inverted PHI2 signal. With a slight delay of tD01 from PHI2

2.7 Phase 2 In (PHI2)

This is the buffered clock input to the internal clock generator. The clock outputs, PHI10 and PHI20, are derived from this signal.

2.8 Phase 2 Out (PHI20)

This signal is generated from PHI2, PHI20 can be used to provide system timing. There is a slight delay of tD02 from PHI2.

2.9 Read/Write (RWB)

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 prior to the falling edge of PHI2 will halt the microprocessor with the output address lines reflecting the current address being fetched. This assumes the processor setup time is met. This condition will remain through a subsequent PHI2 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. The microprocessor will be released when RDY is high and a falling edge of PHI2 occurs. This again assumes the processor control setup time is met. The RDY pin has an active pullup, when outputting a low level, the pullup is disabled. The RDY pin can still be wire ORed.

2.11 Reset (RESB)

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 a reset 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 will be a reset sequence lasting seven clock cycles. The interrupt disable flag is set, the decimal mode is cleared and the program counter is loaded with the reset vector from locations FFFC (low byte) and FFFD (high byte). This is the start location for program control. This input should be pulled high in normal operation.

2.12 Set Overflow (SOB)

A negative transition on this line sets the overflow bit in the status code register. The signal is sampled on the rising edge of PHI2.

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 the clock cycle of an opcode fetch and stays high for the entire cycle. If the RDY line is pulled low during the clock cycle 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 (VPB)

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

2.16 Bus Enable (BE)

The BE asynchronous input signal allows external control of the Address and Data Buffers, as well as the RWB signal. When BE is high, the RWB and Address Buffers are active. When BE is low, these buffers are disabled.

SECTION 3

ADDRESSING MODES

The W65C02S is capable of directly addressing 64 KBytes of memory. This address space has special significance within certain addressing modes, as follows:

3.1 Reset and Interrupt Vectors

The Reset and Interrupt Vectors use the majority of the fixed addresses between FFFA and FFFF.

3.2 Stack

The Stack may be use memory from 0100 to 01FF. The effective address of Stack and Stack Relative addressing modes will be always be within this range.

3.3 Data Address Space

The Program Address and Data Address space is contiguous throughout the 64 KByte address space. Words, arrays, records, or any data structures may span the 64 KByte address space.

3.4 Addressing Mode Descriptions

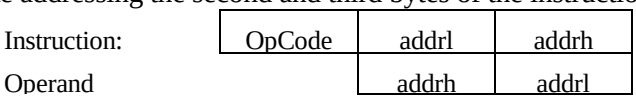
The following addressing mode descriptions provide additional detail as to how effective addresses are calculated. Sixteen addressing modes are available for the W65C02S.

3.4.1 Immediate Addressing-#

The operand is the second byte of the instruction.

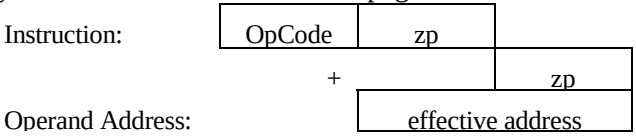
3.4.2 Absolute-a

With Absolute addressing the second and third bytes of the instruction form the 16-bit address.



3.4.3 Zero Page-zp

The second byte of the instruction is the zero page address.



3.4.4 Accumulator-A

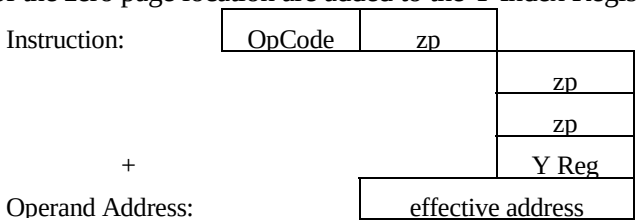
This form of addressing always uses a single byte instruction. The operand is the Accumulator.

3.4.5 Implied-i

Implied addressing uses a single byte instruction. The operand is implicitly defined by the instruction.

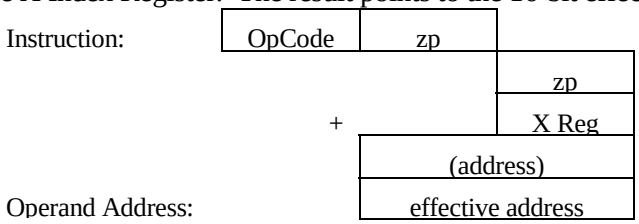
3.4.6 Zero Page Indirect Indexed-(zp),Y

This address mode is often referred to as Indirect,Y. The second byte of the instruction is the zero page address and the contents of the zero page location are added to the Y Index Register to form the effective address.



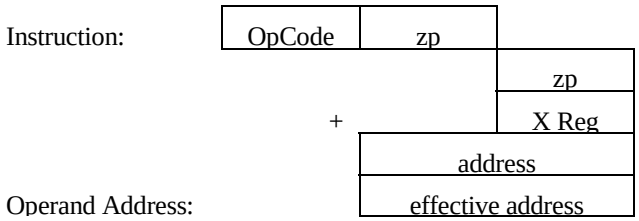
3.4.7 Zero Page Indexed Indirect-(zp,X)

This address mode is often referred to as Indirect,X. The second byte of the instruction is the zero page address and is added to the X Index Register. The result points to the 16-bit effective address.



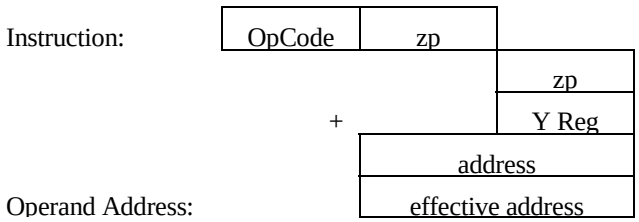
3.4.8 Zero Page Indexed With X-zp,X

The second byte of the instruction is the zero page address and is added to the X Index Register to form the 16-bit effective address.



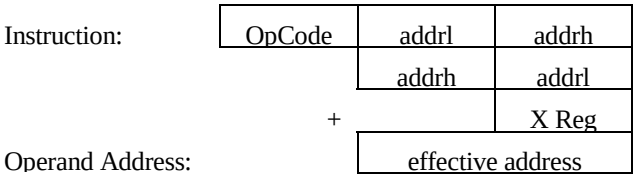
3.4.9 Zero Page Indexed With Y-zp,Y

The second byte of the instruction is the zero page address and is the Y Index Register to form the 16-bit effective address.



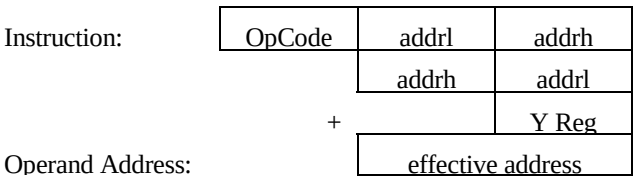
3.4.10 Absolute Indexed With X-a,X

The second and third bytes of the instruction are added to the X Index Register to form the 16-bits of the effective address.



3.4.11 Absolute Indexed With Y-a,Y

The second and third bytes of the instruction are added to the Y Index Register to form the l6 bits of the effective address.

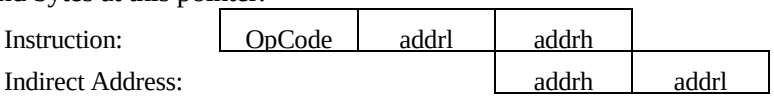


3.4.12 Program Counter Relative-r

This address mode, referred to as Relative Addressing, is used only with the Branch instructions. If the condition being tested is met, the second byte of the instruction is added to the Program Counter, which has been updated to point to the OpCode of the next instruction. The offset is a signed 8-bit quantity in the range from -128 to 127.

3.4.13 Absolute Indirect-(a)

The second and third bytes of the instruction form an address to a pointer. The Program Counter is loaded with the first and second bytes at this pointer.



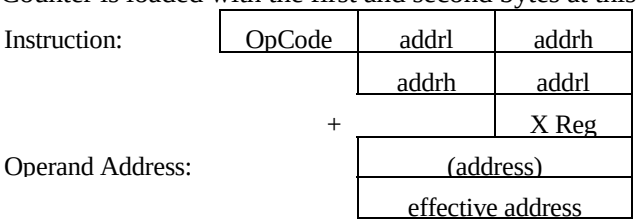
New PC = (indirect address)
with JML:
New PC = (indirect address)

3.4.14 Stack-s

Stack addressing refers to all instructions that push or pull data from the stack, such as Push, Pull, Jump to Subroutine, Return from Subroutine, Interrupts, and Return from Interrupt.

3.4.15 Absolute Indexed Indirect-(a,x)

The second and third bytes of the instruction are added to the X Index Register to form an address to a pointer.
The program Counter is loaded with the first and second bytes at this pointer.



3.4.16 Zero Page Indirect-(zp)
The second byte of the instruction is zero page address. It points to the 16 bit effective address.

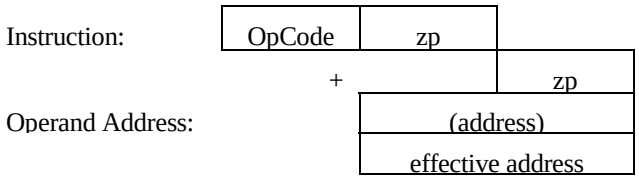


Table 3-2 Addressing Mode Summary

Address Mode	Instruction Times in Memory Cycle		Memory Utilization in Number of Program Sequence Bytes	
	Original 8-bit NMOS 6502	New W65C02S	Original 8-bit NMOS 6502	New W65C02S
1. Immediate	2	2	2	2
2. Absolute	4 (3)	4 (3)	3	3
3. Zero Page	3 (3)	3 (3)	2	2
4. Accumulator	2	2	1	1
5. Implied	2	2	1	1
6. Zero Page Indirect Indexed (d),y	5 (1)	5 (1)	2	2
7. Zero Page Indexed Indirect (d,x)	6	6	2	2
8. Zero Page, X	4 (3)	4 (3)	2	2
9. Zero Page, Y	4	4	2	2
10. Absolute, X	4 (1,3)	4 (1,3)	3	3
11. Absolute, Y	4 (1)	4 (1)	3	3
12. Relative	2 (2)	2 (2)	2	2
13. Absolute Indirect (Jump)	5	5	3	3
14. Stack	3-7	3-7	1-3	1-4

Notes (these are indicated in parentheses):

1. Page boundary, add 1 cycle if page boundary is crossed when forming address.
2. Branch taken, add 1 cycle if branch is taken.
3. Read-Modify-Write, add 2 cycles.

SECTION 4

TIMING, AC AND DC CHARACTERISTICS

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
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 = 0°C to +70°C

Table 4-2 DC Characteristics

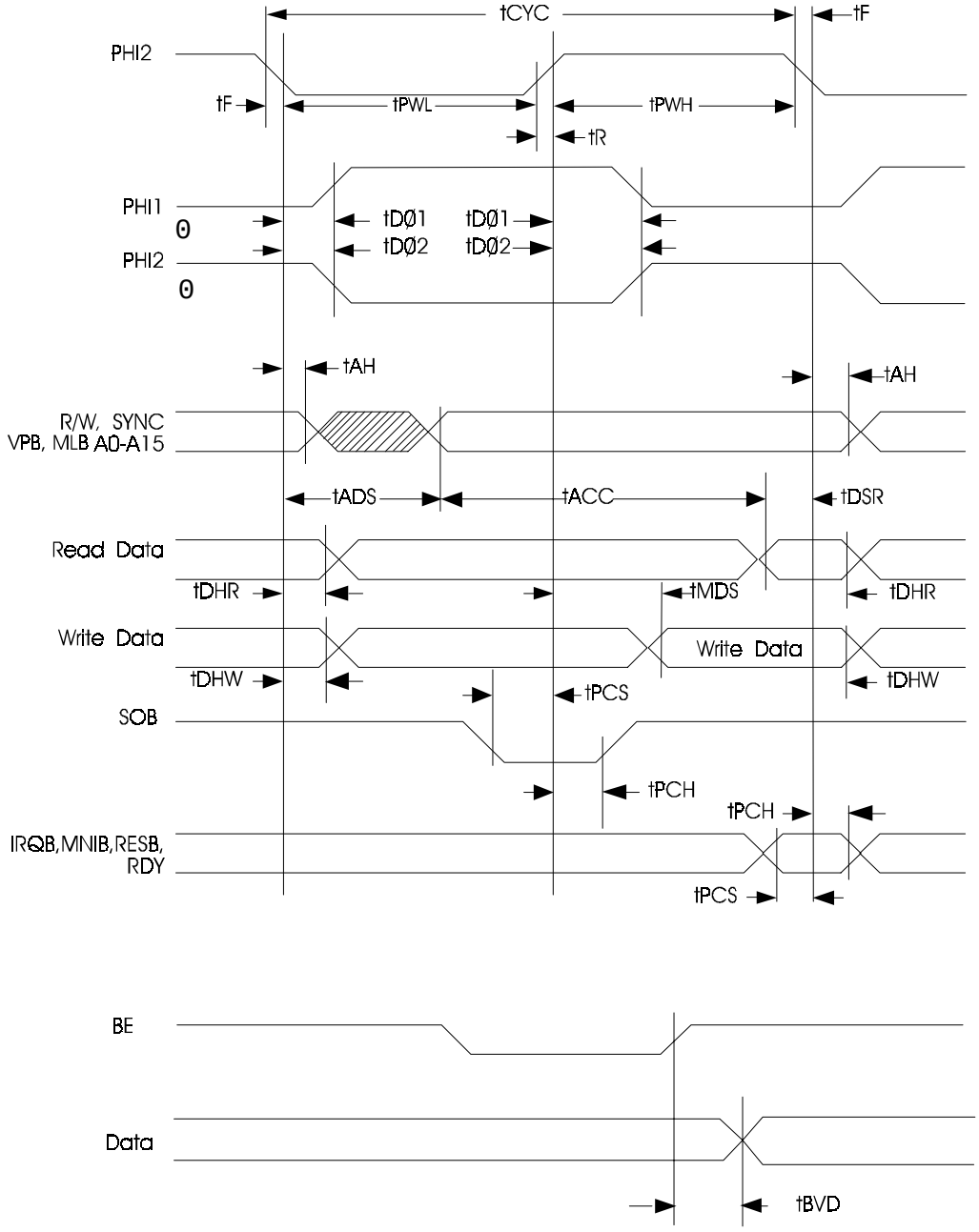
Parameter	Symbol	Min	Max	Unit
Input High Voltage PHI2, NMIB, RESB RDY, IRQB, Data, SOB	Vih	 VDD-0.2 VDD-0.2	 VDD+0.3 VDD+0.3	 V V
Input Low Voltage PHI2, NMIB, RESB RDY, IRQB, Data, SOB	Vil	 VSS-0.3 VSS-0.3	 VSS+0.2 VSS+0.2	 V V
Input Leakage Current (Vin=0.4 to 2.4, VDD=5.25V) All input pins Data, (off state)	Iin	 -1 -10	 1 10	 uA uA
Output High Voltage (Ioh=-100uA, VDD=4.75V) SYNC, Data, A0-A15, RWB, MLB, VPB, PHI10, PHI20	Voh	 2.4	 -	 V
Output Low Voltage (Iol=1.6mA, VDD-4.75V) SYNC, Data, A0-A15, RWB, MLB, VPB, PHI10 PHI20	Vol	 -	 0.4	 V
Supply Current (no load)	Icc		1.5	mA/MHz
Standby Current Outputs Unloaded PHI2, RESB, NMIB, RDY, IRQB, SOB=VDD; DATA=VSS or VDD	Isby	-	1	uA
*Capacitance (Vin=0V, TA=25°C, f-1MHz) PHI2, BE, SOB,PHI20, RESB, VPB, RDY, PHI10,IRQB,MLB,NMIB,SYNC A0-A15, RWB, Data (Off state) *Not inspected during production test; verified on a sample basis.	 Cin Cts	 - -	 10 15	 pF

4.3 General AC Characteristic EquationsVDD=5.0V +/- 5%, VSS= 0V, TA= 0°C to +70°C(1)

Table 4-3 W65C02S General AC Characteristic Equations, 14 MHz

Parameter	Symbol	14MHz		Unit
		Min	Max	
Cycle Time	tCYC	70		nS
Clock Pulse Width Low	tPWL	35	-	nS
Clock Pulse Width High	tPWH	35	-	nS
Fall Time, Rise Time	tF,tR	-	5	nS
Delay Time, PHI2, PHI10	tD01	-	22	nS
Delay Time, PHI2, PHI20	tD02	-	22	nS
Address Hold Time	tAH	10	-	nS
Address Setup Time	tADS	-	30	nS
Access Time	tACC	30	-	nS
Read Data Hold Time	tDHR	10	-	nS
Read Data Setup Time	tDSR	10	-	nS
Write Data Delay Time	tMDS	-	25	nS
Write Data Hold Time	tDHW	10	-	nS
Processor Control Setup Time	tPCS	10	-	nS
Processor Control Hold Time	tPCH	10	-	nS
Capacitive Load (2)	CEXT	-	35	pF
BE to Valid Data (3)	tBVD	-	25	nS

1. Custom testing available for voltage range, temperature and timing.
2. Applied to Address, Data, RWB
3. BE to High Impedance State is not testable but should be the same amount of time as BE to Valid Data.



Timing Notes:
1. Timing measurement points are 1.5V.
2. Custom testing available for voltage range, temperature and timing.

Figure 4-1 General Timing Diagram

SECTION 5
OPERATION TABLES

Table 5-1 W65C02S Instruction Set-Alphabetical Sequence

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

Table 5-2 Vector Locations

FFFE,F	BRK/IRQB	Hardware/Software
FFFC,D	RESETB	Hardware
FFFA,B	NMIB	Hardware

The VPB 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 5-3 OpCode Matrix

MSD																	MSD
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	
0	BRK s 7,2	ORA(zp,X) 6,2			TSB•zp 5,2	ORA zp 3,2	ASL zp 5,2	RMB0 zp 5,2	PHP s 3,1	ORA# 2,2	ASL A 2,1		TSB•a 6,3	ORAa 4,3	ASLa 6,3	BBR0 r 5,3	0
1	BPL r 2,2	ORA(zp),Y 5,2	ORA*(zp) 5,2		TRB•izp, 5,2	ORA zp,X 4,2	ASL zp,X 6,2	RMB1 zp 5,2	CLC i 2,1	ORA a,Y 4,3	INC*A 2,1		TRB•a 6,3	ORA a,X 4,3	ASL a,X 6,3	BBR1 r 5,3	1
2	JSR a 6,3	AND(zp,X) 6,2			BIT zp 3,2	AND zp 3,2	ROL zp 5,2	RMB2 zp 5,2	PLP s 4,1	AND# 2,2	ROL A 2,1		BIT a 4,3	AND a 4,3	ROL a 6,3	BBR2 R 5,3	2
3	BMI r 2,2	AND(zp),Y 5,2	AND(zp), 5,2		BIT zp,X 4,2	AND zp,X 4,2	ROL zp,X 6,2	RMB3zp 5,2	SEC i 2,1	AND a, Y 4,3	DEC*A 2,1		BIT*a,X 4,3	AND a,X 4,3	ROL a,X 6,3	BBR3 r 5,3	3
4	RTI s 6,1	EOR(zp,X) 6,2				EOR zp 3,2	LSR zp 5,2	RMB4zp 5,2	PHA s 3,1	EOR# 2,2	LSR A 2,1		JMP a 3,3	EOR a, 4,3	LSR a 6,3	BBR4 r 5,3	4
5	BVC r 2,2	EOR(zp),Y 5,2	EOR (zp) 5,2			EOR zp,X 4,2	LSR zp,X 6,2	RMB5zp 5,2	CLI i 2,1	EOR a,Y 4,3	PHY•s 3,1			EOR a,X 4,3	LSR a,X 6,3	BBR5 r 5,3	5
6	RTS s 6,1	ADC(zp,X) 6,2			STZ•zp 3,2	ADC zp 4,2	ROR zp 5,2	RMB6zp 5,2	PLA s 4,1	ADC# 2,2	ROR A 2,1		JMP(a) 6,3	ADC a, 4,3	ROR a 6,3	BBR6 r 5,3	6
7	BVS r 2,2	ADC(zp),Y 5,2	ADC(zp) 5,2		STZ• zp,X 4,2	ADC zp,X 4,2	ROR zp,X 6,2	RMB7zp 5,2	SEI i 2,1	ADC a,Y 4,3	PLY s 4,1		JMP*(a,X) 6,3	ADC a,X 4,3	ROR a,X 6,3	BBR7 r 5,3	7
8	BRA•r 3,2	STA(zp,X) 6,2			STY zp 3,2	STA zp 3,2	STX zp 3,2	SMB0p 5,2	DEY i 2,1	BIT*# 2,2	TXA i 2,1		STY a, 4,3	STA a, 4,3	STX a 4,3	BBS0 r 5,3	8
9	BCC r 2,2	STA(zp),Y 6,2	STA(zp) 5,2		STY zp,X 4,2	STA zp,X 4,2	STX zp,Y 4,2	SMB1zp 5,2	TYA i 2,1	STA a,Y 5,3	TXS i 2,1		STZ a, 4,3	STA a,X 5,3	STZ a,X 5,3	BBS1 r 5,3	9
A	LDY# 2,2	LDA(zp,X) 6,2	LDX# 2,2		LDY zp 3,2	LDA zp 3,2	LDX zp 3,2	SMB2 zp 5,2	TAY i 2,1	LDA# 2,2	TAX i 2,1		LDY a, 4,3	LDA a, 4,3	LDX a 4,3	BBS2 r 5,3	A
B	BCS r 2,2	LDA(zp),Y 5,2	LDA*(zp) 5,2		LDY zp,X 4,2	LDA zp,X 4,2	LDX zp,Y 4,2	SMB3zp 5,2	CLV i 2,1	LDA a,Y 4,3	TSX i 2,1		LDY a,X 4,3	LDA a,X 4,3	LDX a,Y 4,3	BBS3 r 5,3	B
C	CPY# 2,2	CMP(zp,X) 6,2			CPY zp 3,2	CMP zp 3,2	DEC zp 5,2	SMB4zp 5,2	INY i 2,1	CMP# 2,2	DEX i 2,1	WAI•i 3,1	CPY a 4,3	CMP a, 4,3	DEC a 6,3	BBS4 r 5,3	C
D	BNE r 2,2	CMP(zp),Y 5,2	CMP*(zp) 5,2			CMP zp,X 3,2	DEC zp,X 6,2	SMB5zp 5,2	CLD i 2,1	CMP a,Y 4,3	PHX• s 3,1	STP•i 3,1		CMP a,X 4,3	DEC a,X 6,3	BBS5 r 5,3	D
E	CPX# 2,2	SBC(zp,X) 6,2			CPX zp 3,2	SBC zp 3,2	INC zp 5,2	SMB6zp 5,2	INX i 2,1	SBC# 2,2	NOP i 2,1		CPX a 4,3	SBC a 4,3	INC a 6,3	BBS6 r 5,3	E
F	BEQ r 2,2	SBC(zp),Y 5,2	SBC*(zp) 5,2			SBC zp,X 4,2	INC zp,X 6,2	SMB7zp 5,2	SED i 2,1	SBC a,Y 4,3	PLX s 4,1			SBC a,X 4,3	INC a,X 6,3	BBS7 r 5,3	F
	0	1	2	3	4	5	6	7	8	9	A	B	C	D	E	F	

•= New instruction * = Old instruction with new addressing modes

	OPERATION v AND w OR x EXCLUSIVE OR																PROCESSOR STATUS CODE							
																	7	6	5	4	3	2	1	0
MNEMONIC	~ NOT	#	A	ZP	A	I	(ZP),Y	(ZP,X)	ZP,X	ZP,Y	A,X	A,Y	R	(A)	S	(A,X)	(ZP)	N	V	1	B	D	I	Z	C
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16								
ADC	A+M+C→A	69	6D	65			71	61	75		7D	79					73	N	V	.	.	.	.	Z	C
AND	A∨M→A	29	2D	25			31	21	35		3D	39					32	N	.	.	.	.	.	Z	.
ASL	C←[7 0]←0		0E	6	0A				16		1E							N	.	.	.	.	.	Z	C
BBR0	BRANCH ON BIT 0 RESET												0F					.	.	.	.	.	.	.	.
BBR1	BRANCH ON BIT 1 RESET												1F					.	.	.	.	.	.	.	.
BBR2	BRANCH ON BIT 2 RESET												2F					.	.	.	.	.	.	.	.
BBR3	BRANCH ON BIT 3 RESET												3F					.	.	.	.	.	.	.	.
BBR4	BRANCH ON BIT 4 RESET												4F					.	.	.	.	.	.	.	.
BBR5	BRANCH ON BIT 5 RESET												5F					.	.	.	.	.	.	.	.
BBR6	BRANCH ON BIT 6 RESET												6F					.	.	.	.	.	.	.	.
BBR7	BRANCH ON BIT 7 RESET												7F					.	.	.	.	.	.	.	.
BBS0	BRANCH ON BIT 0 SET												8F					.	.	.	.	.	.	.	.
BBS1	BRANCH ON BIT 1 SET												9F					.	.	.	.	.	.	.	.
BBS2	BRANCH ON BIT 2 SET												AF					.	.	.	.	.	.	.	.
BBS3	BRANCH ON BIT 3 SET												BF					.	.	.	.	.	.	.	.
BBS4	BRANCH ON BIT 4 SET												CF					.	.	.	.	.	.	.	.
BBS5	BRANCH ON BIT 5 SET												DF					.	.	.	.	.	.	.	.
BBS6	BRANCH ON BIT 6 SET												EF					.	.	.	.	.	.	.	.
BBS7	BRANCH ON BIT 7 SET												FF					.	.	.	.	.	.	.	.
BCC	BRANCH C = 0												90					.	.	.	.	.	.	.	.
BCS	BRANCH IF C = 1												B0					.	.	.	.	.	.	.	.
BEQ	BRANCH IF Z = 1												F0					.	.	.	.	.	.	.	.
BIT	A∨M	89	2C	24					34		3C							M_7	M_6	.	.	.	Z	.	.
BMI	BRANCH IF N = 1												30					.	.	.	.	.	.	.	.
BNE	BRANCH IF Z = 0												D0					.	.	.	.	.	.	.	.
BPL	BRANCH IF N = 0												10					.	.	.	.	.	.	.	.
BRA	BRANCH ALWAYS												80					.	.	.	.	.	.	.	.
BRK	BREAK														0			.	.	.	1	0	1	.	.
BVC	BRANCH IF V = 0												50					.	.	.	.	.	.	.	.
BVS	BRANCH IF V = 1												70					.	.	.	.	.	.	.	.
CLC	0→C					18												.	.	.	.	.	.	.	0
CLD	0→D					D8												.	.	.	.	0	.	.	.
CLI	0→I					58												.	.	.	.	.	0	.	.
CLV	0→V					B8												.	0	.	.	.	.	.	.
CMP	A-M	C9	CD	C5			D1	C1	D5		DD	D9					D2	N	.	.	.	.	.	Z	C

Note: M_7 = memory bit #7
M_6 = memory bit #6

Table 5-4 Operation, Operation Codes and Status Register (continued)

Mnemonic	Operation ∨ AND w OR x Exclusive OR ~ NOT	#	A	zp	A	i	(zp),Y	(zp,X)	zp,X	zp,Y	a,X	a,Y	r	(a)	s	(a,X)	(zp)	Processor Status Code 7 6 5 4 3 2 1 0 N V 1 B D I Z C
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
CPX	X-M	E0	EC	E4														N . . . . . Z C
CPY	Y-M	C0	CC	C4														N . . . . . Z C
DEC	Decrement		CE	C6	3A				D6		DE							N . . . . . Z .
DEX	X-1 → X					CA												N . . . . . Z .
DEY	Y-1 → Y					88												N . . . . . Z .
EOR	A ∨ M → A	49	4D	45			51	41	55		5D	59					52	N . . . . . Z .
INC	Increment		EE	E6	1A				F6		FE							N . . . . . Z .
INX	X+1 → X					E8												N . . . . . Z .
INY	Y+1 → Y					C8												N . . . . . Z .
JMP	Jump to New Location		4C					7C						6C		7C		. . . .
JSR	Jump to Subroutine		20														B2	N . . . . . Z .
LDA	M → A	A9	AD	A5			B1	A1	B5		BD	B9						N . . . . . Z .
LDX	M → X	A2	AE	A6						B6		BE						N . . . . . Z .
LDY	M → Y	A0	AC	A4					B4		BC							N . . . . . Z .
LSR	0 → [7 6 5 4 3 2 1 0] → C		4E	46	4A				56		5E							0 . . . . . Z C
NOP	No Operation					EA												. . . .
ORA	A ∨ M → A	09	0D	05			11	01	15		1D	19					12	N . . . . . Z .
PHA	A → Ms, S-1 → S														48			. . . .
PHP	P → Ms, S-1 → S														08			. . . .
PHX	X → Ms, S-1 → S														DA			. . . .
PHY	Y → Ms, S-1 → S														5A			. . . .
PLA	S+1 → S, Ms → A														68			N . . . . . Z .
PLP	S+1 → S, Ms → P														28			N V . B D I Z C
PLX	S+1 → S, Ms → X														FA			N . . . . . Z .
PLY	S+1 → S, Ms → Y														7A			N . . . . . Z .
RMB0	Reset Memory Bit 0			07														. . . .
RMB1	Reset Memory Bit 1			17														. . . .
RMB2	Reset Memory Bit 2			27														. . . .
RMB3	Reset Memory Bit 3			37														. . . .
RMB4	Reset Memory Bit 4			47														. . . .
RMB5	Reset Memory Bit 5			57														. . . .
RMB6	Reset Memory Bit 6			67														. . . .
RMB7	Reset Memory Bit 7			77														. . . .
ROL	[7 6 5 4 3 2 1 0] ← C ←		2E	26	2A				36		3E							N . . . . . Z C
ROR	→ [7 6 5 4 3 2 1 0] → C		6E	66	6A				76		7E							N . . . . . Z C

Table 5-4 Operation, Operation Codes and Status Register (continued)

Mnemonic	Operation v AND w OR x Exclusive OR ~ NOT	#	A	zp	A	i	(zp),Y	(zp,X)	zp,X	zp,Y	A,X	A,Y	r	(A)	s	(A,X)	(zp)	Processor Status Code 7 6 5 4 3 2 1 0 N V 1 B D I Z C
		1	2	3	4	5	6	7	8	9	10	11	12	13	14	15	16	
RTI	Return from Interupt														40			N V . . D I Z C
RTS	Return from Subroutine														60			. . . . .
SBC	A-M-(~C)6A	E9	ED	E5			F1	E1	F5		FD	F9					F2	N V . . . . Z C
SEC	16C					38												 1
SED	16D					F8												. . . . 1 . . .
SEI	16I					78												. . . . . 1 . .
SMB0	Set Memory Bit 0			87														. . . . .
SMB1	Set Memory Bit 1			97														. . . . .
SMB2	Set Memory Bit 2			A7														. . . . .
SMB3	Set Memory Bit 3			B7														. . . . .
SMB4	Set Memory Bit 4			C7														. . . . .
SMB5	Set Memory Bit 5			D7														. . . . .
SMB6	Set Memory Bit 6			E7														. . . . .
SMB7	Set Memory Bit 7			F7														. . . . .
STA	A6M		8D	85			91	81	95		9D	99					92	. . . . .
STP	STOP(16 FH2)					DB												. . . . .
STX	X6M		8E	86						96								. . . . .
STY	Y6M		8C	84					94									. . . . .
STZ	006M		9C	64					74		9E							. . . . .
TAX	A6X					AA												N . . . . Z .
TAY	A6Y					A8												N . . . . Z .
TRB	~AvM6M		1C	14														. . . . . Z .
TSB	AvM6M		0C	04														. . . . . Z .
TSX	S6X					BA												N . . . . Z .
TXA	X6A					8A												N . . . . Z .
TXS	X6S					9A												. . . . .
TYA	Y6A					98												N . . . . Z .
WAI	C6RDY					CB												. . . . .

Address Mode	Note	Cycle	VPB	MLB	SYNC	Address Bus	Data Bus	RWB
1. Immediate # LDY,CPY,CPX,LDX,ORA,AND,EOR, ADC,BIT,LDA,CMP,SBC,12 OpCodes, 2 bytes, 2 & 3 cycles	(6)	1 2	1 1	1 1	1 0	PC PC+1	OpCode ID	1 1
2a. Absolute a BIT,STY,STZ,LDY,CPY,CPX,STX, LDX,ORA,AND,EOR,ADC,STA,LDA, CMP,SBC 16 OpCodes, 3 bytes, 4 & 5 cycles	(6)	1 2 3 4	1 1 1 1	1 1 1 1	1 0 0 0	PC PC+1 PC+2 AA	OpCode AAL AAH Data	1 1 1 1/0
2b. Absolute (R-M-W) a ASL,ROL,LSR,ROR,DEC,INC,TSB,TRB 8 OpCodes, 3 bytes, 6 cycles		1 2 3 4 5 6	1 1 1 1 1 1	1 1 1 0 0 0	1 0 0 0 0 0	PC PC+1 PC+2 AA AA AA	OpCode AAL AAH Data IO Data	1 1 1 1 1 0
2c. Absolute (JUMP) a JMP (4C) 1 OpCode, 3 bytes, 3 cycles		1 2 3 1	1 1 1 1	1 1 1 1	1 0 0 1	PC PC+1 PC+2 New PC	OpCode New PCL New PCH New OpCode	1 1 1 1
2d. Absolute (JUMP to subroutine) a JSR (20) 1 OpCode, 3 bytes, 6 cycles (different order from N6502)		1 2 3 4 5 6 1	1 1 1 1 1 1 1	1 1 1 1 1 1 1	1 0 0 0 0 0 1	PC PC+1 S S S-1 PC+2 New PC	OpCode New PCL IO PCH PCL New PCH New OpCode	1 1 1 0 0 1 1
3a. Zero Page zp BIT,STZ,STY,LDY,CPY,CPX,STX,LDX,ORA, AND,EOR,ADC,STA,LDA,CMP,SBC 16 OpCodes, 2 bytes, 3 and 4 cycles	(6)	1 2 3	1 1 1	1 1 1	1 0 0	PC PC+1 0,zp	OpCode zp Data	1 1 1/0
3b. Zero Page (R-M-W) zp ASL,ROL,LSR,ROR,DEC,INC,TSB,TRB 8 OpCodes, 2 bytes, 5 cycles		1 2 3 4 5	1 1 1 1 1	1 1 0 0 0	1 0 0 0 0	PC PC+1 0,zp 0,zp+1 0,zp	OpCode zp Data IO Data	1 1 1 1 0
3c. Memory Bit Manipulation (R-M-W) RMB,SMB, zp 16 Op Codes 2 bytes 5 cycles		1 2 3 4 5	1 1 1 1 1	1 1 0 0 0	1 0 0 0 0	PC PC+1 0,zp 0,zp 0,zp	OpCode zp Data IO Data	1 1 1 1 0
4. Accumulator A ASL,INC,ROL,DEC,LSR,ROR 6 OpCodes, 1 byte, 2 cycles		1 2	1 1	1 1	1 0	PC PC+1	OpCode IO	1 1
5a. Implied I DEY,INY,INX,DEX,NOP,TYA,TAY,TXA TXS,TAX,TSX,CLC,SEC,CLI,SEI,CLV,CLD, SED 18 OpCodes, 1 byte, 2 cycles		1 2	1 1	1 1	1 0	PC PC+1	OpCode IO	1 1

Address Mode	Note	Cycle	VPB	MLB	SYNC	Address Bus	Data Bus	RWB
5b. Wait for Interrupt						RDY		
WAI (CB)		1	1	1	1	PC	OpCode	1
1 OpCode	(4)	2	1	1	0	PC+1	IO	1
1 byte		3	1	1	0	PC+1	IO	1
3 cycles, IRQB NMIB		1	1	1	1	PC+1	IRQ(BRK)	1
5c. Stop the Clock						RDY		
STP (DB)		1	1	1	1	PC	OpCode	1
1 OpCode, 1 byte, 3 cycles		2	1	1	0	PC+1	IO	1
RESB=1		3	1	1	0	PC+1	IO	1
RESB=0		1c	1	1	0	PC+1	RES	1
RESB=0		1b	1	1	0	PC+1	(BRK)	1
RESB=1		1a	1	1	0	PC+1	RES	1
(see 16a. Stack Hardware Interrupt)		1	1	1	1	PC+1	(BRK)	1
							RES	
							(BRK)	
							BEGIN	
6. Zero Page Indexed-(zp),Y		1	1	1	1	PC	OpCode	1
ORA,AND,EOR,ADC,STA,LDA,CMP,SBC	(1)	2	1	1	0	PC+1	zp	1
8 OpCodes, 2 bytes		3	1	1	0	0,zp	AAL	1
5,6, and 7 cycles	(6)	4	1	1	0	0,zp+1	AAH	1
		5	1	1	0	AA+Y	Data	1/0
7. Zero Page Indirect (zp,X)		1	1	1	1	PC	OpCode	1
ORA,AND,EOR,ADC,STA,LDA,CMP,SBC		2	1	1	0	PC+1	zp	1
8 OpCodes, 2 bytes,		3	1	1	0	PC+1	IO	1
6 and 7 cycles	(6)	4	1	1	0	0,zp+X	AAL	1
		5	1	1	0	0,zp+X+1	AAH	1
		6	1	1	0	AA	Data	1/0
8a. Zero Page,X zp,X		1	1	1	1	PC	OpCode	1
BIT,STZ,STY,LDY,ORA,AND,EOR,ADC	(6)	2	1	1	0	PC+1	zp	1
STA,LDA,CMP,SBC		3	1	1	0	PC+1	IO	1
12 OpCodes,2 bytes,4 and 5 cycles		4	1	1	0	0,zp+X	Data	1/0
8b. Zero Page, X (R-M-W) zp,X		1	1	1	1	PC	OpCode	1
ASL,ROL,LSR,ROR,DEC,INC		2	1	1	0	PC+1	zp	1
6 OpCodes, 2 bytes	(1)	3	1	1	0	PC+1	IO	1
6 and 7 cycles		4	1	0	0	0,zp+X	Data	1
		5	1	0	0	0,zp+X+1	IO	1
		6	1	0	0	0,zp+X	Data	0
9. Zero Page, Y zp,Y		1	1	1	1	PC	OpCode	1
STX,LDX	(1)	2	1	1	0	PC+1	zp	1
2 OpCodes		3	1	1	0	PC+1	IO	1
2 bytes, 4 and 5 cycles		4	1	1	0	0,zp+Y	Data	1/0
10a. Absolute, X a,X		1	1	1	1	PC	OpCode	1
BIT,LDY,STZ,ORA,AND,EOR,ADC		2	1	1	0	PC+1	AAL	1
STA,LDA,CMP,SBC	(1)	3	1	1	0	PC+2	AAH	1
11 OpCodes	(6)	4	1	1	0	AA+X	Data	1/0
3 bytes, 4,5 and 6 cycles								
10b. Absolute, X(R-M-W) a,X		1	1	1	1	PC	OpCode	1
ASL,ROL,LSR,ROR,DEC,INC		2	1	1	0	PC+1	AAL	1
6 OpCodes, 3 bytes		3	1	1	0	PC+2	AAH	1
7 cycles	(1)	4	1	1	0	AAH,AAL+X	IO	1
		5	1	0	0	AA+X	DATA	1
		6	1	0	0	AA+X+1	IO	1
		7	1	0	0	AA+X	DATA	0

Address Mode	Note	Cycle	VPB	MLB	SYNC	Address Bus	Data Bus	RWB
11. Absolute, Y a,Y LDX,ORA,AND,EOR,ADC,STA,LDA, CMP,SBC 9 OpCodes, 3 bytes 4,5 and 6 cycles	 (1) (6)	1 2 3 4	1 1 1 1	1 1 1 1	1 0 0 0	PC PC+1 PC+2 AA+Y	OpCode AAL AAH Data	1 1 1 1/0
12a. Relative r BPL,BMI,BVC,BVS,BCC BCS,BNE,BEQ,BRA 9 OpCodes, 2 bytes 2,3 and 4 cycles	 (2) (3)	1 2 1	1 1 1	1 1 1	1 0 1	PC PC+1 New PC	OpCode OFF OpCode	1 1 1
12b. Relative Bit Branch r BBRX, BBSX 16 OpCodes 3 bytes 5,6,7 cycles	(2)\ (3)	1 2 3 4 5	1 1 1 1 1	1 1 1 1 1	1 0 0 0 1	PC PC+1 0,zp 0,zp 0,zp	OpCode DO Data IO Data	1 1 1 1 1
13. Absolute Indirect (a) JMP (6C) 1 OpCode, 3 bytes 6 cycles		1 2 3 4 5 6 1	1 1 1 1 1 1 1	1 1 1 1 1 1 1	1 0 0 0 0 0 1	PC PC+1 PC+2 PC+2 0,AA 0,AA+1 NEW PC	OpCode AAL AAH AAH New PCL New PCH OpCode	1 1 1 1 1 1 1
14a. Stack (Hardware Interrupts) s IRQ,NMI,RES 4 hardware interrupts 0 bytes 7 cycles	 (4) (5)	1 2 3 4 5 6 7 1	1 1 1 1 1 0 0 1	1 1 1 1 1 1 1 1	1 0 0 0 0 0 0 1	PC PC 01,S 01,S-1 01,S-2 01,VA 01,VA+1 01,AAV	IO IO PCH PCL P AAVL AAVH New OpCode	1 1 1/0 1/0 1/0 1 1 1
14b. Stack (Software Interrupts) s BRK(00) 1 OpCode 2 bytes 7 cycles		1 2 3 4 5 6 7 1	1 1 1 1 1 0 0 1	1 1 1 1 1 1 1 1	1 0 0 0 0 0 0 1	PC PC+1 01,S 01,S-1 01,S-2 01,VA 01,VA+1 01,AAV	OpCode Signature PC+2H PC+2L P AAVL AAVH New OpCode	1 1 0 0 0 1 1
14c. Stack (Return from Interrupt) s RTI (40) 1 Op Code 1 byte 6 cycles (different order from N6502)		1 2 3 4 5 6 1	1 1 1 1 1 1 1	1 1 1 1 1 1 1	1 0 0 0 0 0 1	PC PC+1 01,S 01,S+1 01,S+2 01,S+3 New PC	OpCode IO IO P New PCL New PCH New OpCode	1 1 1 1 1 1 1

Address Mode	Note	Cycle	VPB	MLB	SYNC	Address Bus	Data Bus	RWB
14d. Stack (Return from Subroutine) s RTS 1 OpCode 1 byte 6 cycles		1 2 3 4 5 6 1	1 1 1 1 1 1 1	1 1 1 1 1 1 1	1 0 0 0 0 0 1	PC PC+1 0,S 0,S+1 0,S+2 NEW PC PC	OpCode IO IO PCL PCH IO OpCode	1 1 1 1 1 1 1
14e. Stack (Push) s PHP,PHA,PHY,PHX 4 Op Codes 1 byte 3 cycles		1 2 3	1 1 1	1 1 1	1 0 0	PC PC+1 0,S-1	OpCode IO REG	1 1 0
14f. Stack (Pull) s PLP,PLA,PLY,PLX Different than N6502 4 Op Codes 1 byte 4 cycles		1 2 3 4 1	1 1 1 1 1	1 1 1 1 1	1 0 0 0 1	PC PC+1 0,S 0,S+1 PC+1	OpCode New OpCode IO REG OpCode	1 1 1 1 1
15. Absolute Indexed Indirect (a) JMP (7C) 1 OpCode 3 bytes 6 cycles	(1)	1 2 3 4 5 6 1	1 1 1 1 1 1 1	1 1 1 1 1 1 1	1 0 0 0 0 0 1	PC PC+1 PC+2 PC+2 AA+X AA+X+1 NEW PC	OpCode AAL AAH IO New PCL New PCH OpCode	1 1 1 1 1 1 1
16. Zero Page (zp) ORA,AND,EOR,ADC,STA,LDA,CMP,SBC 8 OpCodes 2 bytes 5 and 6 cycles	(6)	1 2 3 4 5	1 1 1 1 1	1 1 1 1 1	1 0 0 0 0	PC PC+1 0,D 0,D+1 AA	OpCode zp AAL AAH Data	1 1 1 1 1/0

Notes:

1. Add 1 cycle for indexing across page boundaries, or write. This cycle contains invalid addresses.
2. Add 1 cycle if branch is taken.
3. Add 1 cycle if branch is taken across page boundaries.
4. Wait at cycle 2 for 2 cycles after NMIB or IRQB active input.
5. RWB remains high during Reset.
6. Add 1 cycle for decimal mode

AAH	Absolute Address High	PC	Program Counter
AAL	Absolute Address Low	PCH	Program Counter High
AAVH	Absolute Address Vector High	PCL	Program Counter Low
AAVL	Absolute Address Vector Low	R-M-W	Read-Modify-Write
C	Accumulator	REG	Register
DEST	Destination	S	Stack Address
ID	Immediate Data	SRC	Source
IO	Internal Operation	SO	Stack Offset
OFF	Offset	VA	Vector Address
P	Status Register	x,y	Index Register
		zp	Zero Page Address

SECTION 6
CAVEATS

Table 6-1 Microprocessor Operational Enhancements

Function	NMOS 6502	W65C02S
Indexed addressing across page boundary	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). OpCode Bytes Cycles 02,22,42,62,82 C2, E2 2 2 X3,OB-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 instruction 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.
Oscillator.	Requires external active components.	Crystal or RC network will oscillate when connected between PHI2 and PHI10.
Assertion of Ready (RDY) during write operations.	Ignored.	Stops processor during PHI2, and WAI instruction pulls RDY low.
Clock inputs.	Two non-overlapping clock inputs (PHI1 and PHI2) are required.	PHI2 is the only required clock.
Unused input-only pins.	Must be connected to low impedance signal to avoid noise problems.	Must be tied to VDD through a 10KΩ resistor.

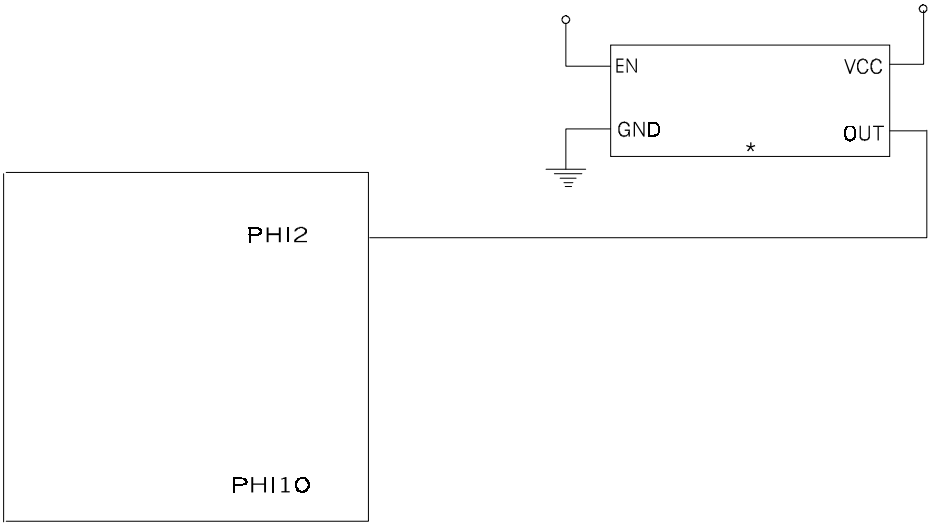


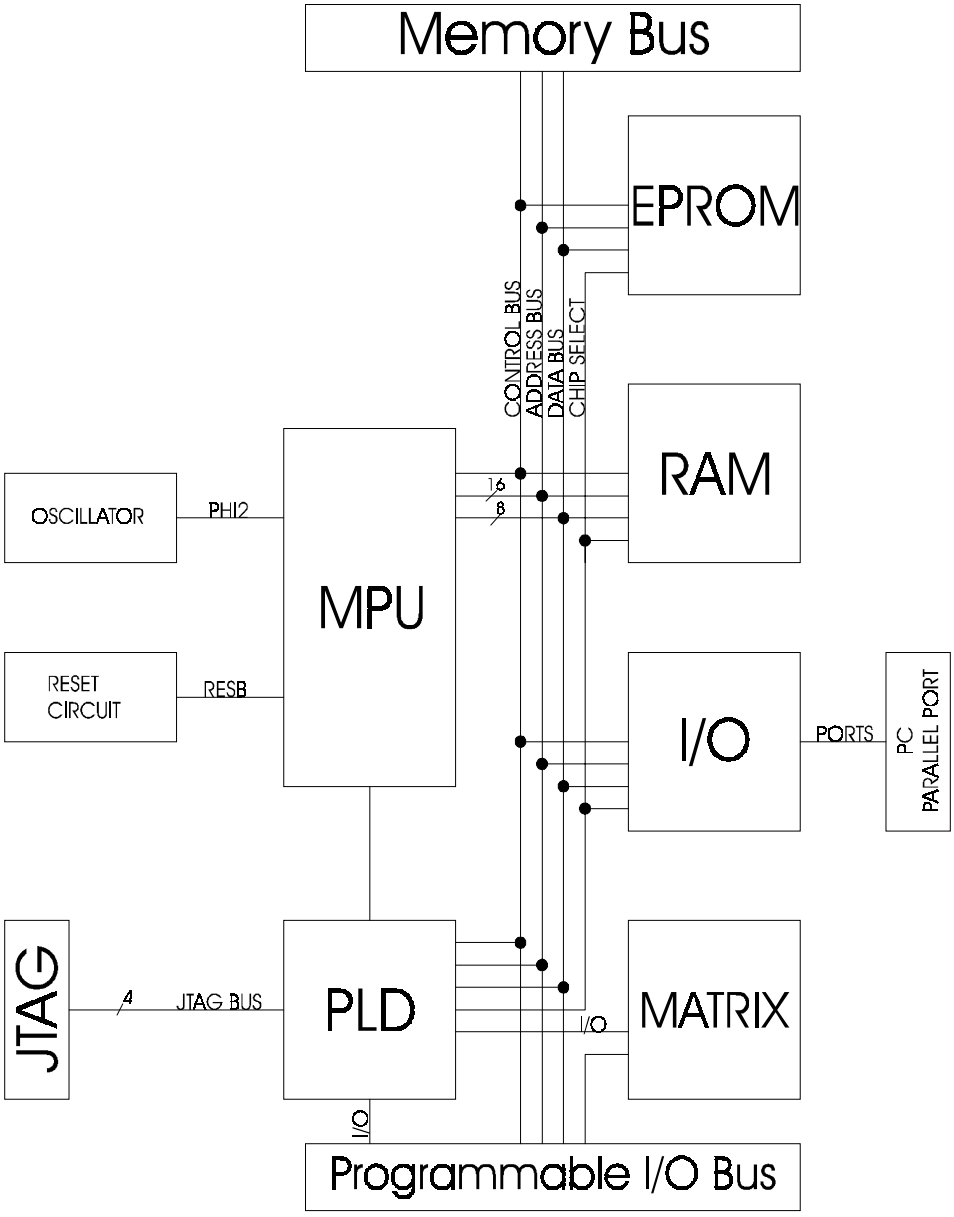
Figure 6-1 8MHz Oscillator

6.2 The BRK instruction for both the NMOS 6502 and 65C02 is a 2 byte instruction. The NMOS device simply skips the second byte (i.e. doesn't care about the second byte) by incrementing the program counter twice. The 65C02 does the same thing except the assembler is looking for that second byte as a "signature byte". With either device (NMOS or CMOS), the second byte is not used. It is important to realize that if a return from interrupt is used it will return to the location after the second or signature byte.

*CTS Package oscillator MX0-45T-XXX
XXX=frequency of operation

SECTION 7

W65C02DB Developer Board



The W65C02DB is used for W65C02 core microprocessor System-Chip Development, W65C02S (chip) System Development, or Embedded W65C02DB (board) Development.

Features:

W65C02S 8-bit MPU, total access to all control lines, Memory Bus, Programmable I/O Bus, PC Interface, 20 I/O lines, easy oscillator change, 32K SRAM, 32K EPROM, W65C22S Versatile Interface Adapter VIA peripheral chip, on-board matrix, PLD for Memory map decoding and ASIC design.

The PLD chip is a XILINX XC9572 for changing the chip select and I/O functions if required. To change the PLD chip to suit your own setup, you need XILINX Data Manager for the XC9572 CPLD chip. The W65C02DB includes an on-board programming header for JTAG configuration. For more details refer to the circuit diagram. The on-board W65C02S and the W65C22S devices have measurement points for core power consumption. Power input is provided by an optional power board which plugs into the 10 pin power header.

An EPROM programmer or an EPROM emulator is required to use the board. WDC's Software Development System includes a W65C02S Assembler and Linker, W65C02S C-Compiler and Optimizer, and W65C02S Simulator/Debugger. WDC's PC IO daughter board can be used to connect the Developer Board to the parallel port of a PC.

Memory map:

CS1B:	8000-FFFF	⇒	EPROM (27C256)
CS3B:	0000-00EF & 0100-7FFF	⇒	SRAM (62C256)
CS2B:	00F0-00FF	⇒	VIA(W65C22S)

7.1 Cross-Debugging Monitor Program

The Cross-Debugging Monitor Programs of the Developer Boards are located in the directory <drive>:\WDC_SDS\DEBUG\WDCMON\

This directory contains the source and the batch files for all of the monitor programs. These programs can be burned into an EPROM and used with the WDC evaluation boards (Developer Boards) and the WDC IO (or ZIO-1) daughter board to interface to the parallel port of a PC. Then, the WDCDB.EXE debugger can be used to download programs, single step, set breakpoints, examine memory, etc.

The monitors have been designed to run correctly with either a W65C02 MPU (WDCMON_1), W65C816 MPU (WDCMON_2), W65C134 MCU (WDC134), or W65C265 MCU (WDC265). It detects the appropriate CPU type on RESET and operates accordingly.

BUILDING

The batch files assemble the program and link it producing Motorola S-Record output. This can be changed by using a different option with the WDCLN linker.

HARD CORE MODEL

W65C02C Core Information

8.1 The W65C02C uses the same instruction set as the W65C02S.

8.2 The only functional difference between the W65C02S and W65C02C is the RDY pin. The W65C02S RDY pin is bi-directional utilizing an active pullup. The W65C02C RDY function is split into 2 pins, RDYINput and WAITN. The WAITN output goes low when a WAI instruction is executed.

8.3 The ESD and latch-up buffers have been removed.

8.4 The output from the core is the buffer N-channel and the P-channel drive transistors.

8.5 The following inputs, if not used, must be pulled to the high state: RDY input, IRQB, NMIB, BE and SOB.

8.6 The timing of the W65C02C is the same as the W65C02S.

SECTION 9
SOFT CORE RTL MODEL

W65C02 RTL-Code in Verilog

9.1 The RTL-Code (**R**egister **T**ransfer **L**evel) in Verilog is a synthesizable model. It can be synthesized into different technologies such as FLEX10K FPGA technology from ALTERA. The behavorial of this model is equivalent to the original W65C02C hardcore. The W65C02 RTL-Code is available in two versions. The W65C02C softcore model and the W65C02S standard chip model. The standard chip model includes the softcore and the buffer ring in RTL-Code. Synthesizable cores are useful in ASIC design, but if a minimum amount of gates are needed, the hardcore model of the W65C02C should be used.

SECTION 10

FIRM CORE MODEL

UNDER CONSTRUCTION

SECTION 11

ORDERING INFORMATION

W65C02S8PL-14	
Description W65C = standard product	W65C
Product Identification Number	02S
Foundry Process Blank = 1.2u 8 = .8u	8
Package P = Plastic Dual-In-Line, 40 pins PL = Plastic Leaded Chip Carrier, 44 pins Q = Quad Flat Pack, 44 pins	PL
Temperature/Processing Blank = 0°C to + 70°C	
Speed Designator -14 = 14MHz	-14

To receive general sales or technical support on standard product or information about our module library licenses, contact us at:

The Western Design Center, Inc.
2166 East Brown Road
Mesa, Arizona 85213 USA
Phone: 602-962-4545 Fax: 602-835-6442
e-mail: information@wdesignc.com
WEB: http://www.wdesignc.com

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.