ADVANCED 6502 INTERFACING

BY JOHN M. HOLLAND

BLACKSBURG CONTINUING EDUCATION SERIES™
edited by Titus, Larsen & Titus

The Blacksburg Continuing Education™ Series

The Blacksburg Continuing Education Series™ of books provide a Laboratory—or experiment-oriented approach to electronic topics. Present and forthcoming titles in this series include:

- Advanced 6502 Interfacing
- Analog Instrumentation Fundamentals
- Apple Interfacing
- Basic Business Software
- BASIC Programmer's Notebook
- Circuit Design Programs for the Apple II
- Circuit Design Programs for the TRS-80
- Design of Active Filters, With Experiments
- Design of Op-Amp Circuits, With Experiments
- Design of Phase-Locked Loop Circuits, With Experiments
- Design of Transistor Circuits, With Experiments
- Design of VMOS Circuits, With Experiments
- 8080/8085 Software Design (2 Volumes)
- 8085A Cookbook
- Electronic Music Circuits
- 555 Timer Applications Sourcebook, With Experiments
- Guide to CMOS Basics, Circuits, & Experiments
- How to Program and Interface the 6800
- Introduction to FORTH
- Microcomputer—Analog Converter Software and Hardware Interfacing
- Microcomputer Interfacing With the 8255 PPI Chip
- Microcomputer Design and Maintenance
- NCR Basic Electronics Course, With Experiments
- NCR Data Communications Concepts
- NCR Data Processing Concepts Course
- NCR EDP Concepts Course
- PET Interfacing
- Programming and Interfacing the 6502, With Experiments
- 16-Bit Microprocessors
- 6502 Software Design
- 6801, 68701, and 6803 Microcomputer Programming and Interfacing
- The 68000: Principles and Programming
- 6809 Microcomputer Programming & Interfacing, With Experiments
- TEA: An 8080/8085 Co-Resident Editor/Assembler
- TRS-80 Assembly Language Made Simple
- TRS-80 Interfacing (2 Volumes)
- TRS-80 More Than BASIC

In most cases, these books provide both text material and experiments, which permit one to demonstrate and explore the concepts that are covered in the book. These books remain among the very few that provide step-by-step instructions concerning how to learn basic electronic concepts, wire actual circuits, test microcomputer interfaces, and program computers based on popular microprocessor chips. We have found that the books are very useful to the electronic novice who desires to join the "electronics revolution," with minimum time and effort.

Jonathan A. Titus, Christopher A. Titus, and David G. Larsen
"The Blacksburg Group"

READER SERVICE CARD

To better serve you, the reader, please take a moment to fill out this card, or a copy of it, for us. Not only will you be kept up to date on the Blacksburg Series books, but as an extra bonus, **we will randomly select five cards every month, from all of the cards sent to us during the previous month. The names that are drawn will win, absolutely free, a book from the Blacksburg Continuing Education Series.** Therefore, make sure to indicate your choice in the space provided below. For a complete listing of all the books to choose from, refer to the inside front cover of this book. Please, one card per person. Give everyone a chance.

In order to find out who has won a book in your area, call (703) 953-1861 anytime during the night or weekend. When you do call, an answering machine will let you know the monthly winners. Too good to be true? Just give us a call. Good luck.

If I win, please send me a copy of:

_____

I understand that this book will be sent to me absolutely free, if my card is selected.

For our information, how about telling us a little about yourself. We are interested in your occupation, how and where you normally purchase books and the books that you would like to see in the Blacksburg Series. We are also interested in finding authors for the series, so if you have a book idea, write to The Blacksburg Group, Inc., P.O. Box 242, Blacksburg, VA 24060 and ask for an Author Packet. We are also interested in TRS-80, APPLE, OSI and PET BASIC programs.

My occupation is _____
I buy books through/from _____
Would you buy books through the mail? _____
I'd like to see a book about _____
Name _____
Address _____
City _____
State _____ Zip _____

MAIL TO: BOOKS, BOX 715, BLACKSBURG, VA 24060
!!!!!PLEASE PRINT!!!!!

21836

先进的6502微处理机接口

The Blacksburg Group

According to Business Week magazine (Technology July 6, 1976) large scale integrated circuits or LSI "chips" are creating a second industrial revolution that will quickly involve us all. The speed of the developments in this area is breathtaking and it becomes more and more difficult to keep up with the rapid advances that are being made. It is also becoming difficult for newcomers to "get on board."

It has been our objective, as The Blacksburg Group, to develop timely and effective educational materials that will permit students, engineers, scientists, technicians and others to quickly learn how to use new technologies and electronic techniques. We continue to do this through several means, textbooks, short courses, seminars and through the development of special electronic devices and training aids.

Our group members make their home in Blacksburg, found in the Appalachian Mountains of southwestern Virginia. While we didn't actively start our group collaboration until the Spring of 1974, members of our group have been involved in digital electronics, minicomputers and microcomputers for some time.

Some of our past experiences and on-going efforts include the following:

–The design and development of what is considered to be the first popular hobbyist computer. The Mark-8 was featured in Radio-Electronics magazine in 1974. We have also designed several 8080-based computers, including the MMD-1 system. Our most recent computer is an 8085-based computer for educational use, and for use in small controllers.

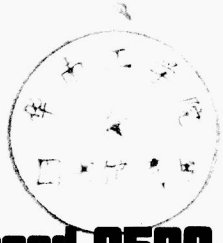
—The Blacksburg Continuing Education Series™ covers subjects ranging from basic electronics through microcomputers, operational amplifiers, and active filters. Test experiments and examples have been provided in each book. We are strong believers in the use of detailed experiments and examples to reinforce basic concepts. This series originally started as our Bugbook series and many titles are now being translated into Chinese, Japanese, German and Italian.

–We have pioneered the use of small, self-contained computers in hands-on courses for microcomputer users. Many of our designs have evolved into commercial products that are marketed by E&L Instruments and PACCOM, and are available from Group Technology, Ltd., Check, VA 24072.

–Our short courses and seminar programs have been presented throughout the world. Programs are offered by The Blacksburg Group, and by the Virginia Polytechnic Institute Extension Division. Each series of courses provides hands-on experience with real computers and electronic devices. Courses and seminars are provided on a regular basis, and are also provided for groups, companies and schools at a site of their choosing. We are strong believers in practical laboratory exercises, so much time is spent working with electronic equipment, computers and circuits.

Additional information may be obtained from Dr. Chris Titus, the Blacksburg Group, Inc. (703) 951-9030 or from Dr. Linda Leffel, Virginia Tech Continuing Education Center (703) 961-5241.

Our group members are Mr. David G. Larsen, who is on the faculty of the Department of Chemistry at Virginia Tech, and Drs. Jon Titus and Chris Titus who work full-time with The Blacksburg Group, all of Blacksburg, VA.



Advanced 6502 Interfacing

by John M. Holland

Howard W. Sams & Co., Inc.
4300 WEST 62ND ST. INDIANAPOLIS, INDIANA 46268 USA

FIRST EDITION
FIRST PRINTING—1982

International Standar 1836-4
Library of Congress C ber: 81-86551

Edited by: *Charlie Buffington*
Illustrated by: *D.B. Clemons*

Printed in the United States of America.

Preface

When the first practical large-scale computers became available in the 1950s and 1960s, there was a great deal of concern about the social and economic impact that they would have on our way of life. An "automation revolution" was envisioned (and feared) by many to be just around the corner. The prospect of giant computers running whole factories without the aid of humans seemed inevitable to most people. When years passed, and the revolution did not come as quickly as first thought, the vision began to fade. When the revolution did begin, few noticed, as it was led not by "giants" but by "midgets."

Although these "midgets," or microprocessors, have been commonly available since the early 1970s, the full scope of their power and usefulness has only begun to be realized in the industrial and consumer marketplaces. The 6502 family must be counted among the microprocessor families that have been proven to have excellent technical and economic merit. The availability of 6502-based products, such as the AIM 65, Apple, PET 2001, and several single-board computers, makes the 6502 an excellent choice for a wide variety of control applications.

Before the 6502 can be of any use in control applications, it must be properly programmed and interfaced to the system it is to control. Although there are many interface boards available for the 6502 and other microcomputers, there is (and will remain) a strong need for people who know how to effectively interface these computers to "the outside world." It is the purpose of this book to provide the designer with some guidelines that he can use in this task.

Since this is an advanced book, it would be best for the newcomer to refer to *Programming and Interfacing the 6502*, by

Marvin L. De Jong. Additional information on programming can be found in *6502 Software Design*, by Leo J. Scanlon.

JOHN M. HOLLAND

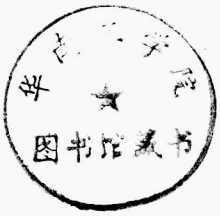
Acknowledgments

I wish to extend my deepest personal thanks to the many people whose comments have contributed to this book. Most especially, I wish to thank my technical editor, Dr. Christopher A. Titus of The Blacksburg Group, Inc. His long-suffering efforts over my manuscript have provided both the work and myself with many fresh ideas.

I would also like to thank my technician, Mr. William Grady Spiegel, for his excellent help in the checking and editing of this work. Additional thanks must also go to Mr. Robert "Bob" Morris, formerly of Beacon Inc., for his help in obtaining Rockwell literature and sample parts.

J.M.H.

This book is dedicated to the scientists, engineers, technicians, and hobbyists who have the "spark." On the tireless work of these men and women, mankind is finding its destiny in the stars.



Contents

CHAPTER 1

CHAPTER 2

CHAPTER 3

CHAPTER 4

CHAPTER 5

CHAPTER 6

CHAPER 7

CHAPTER 8

APPENDIX A

APPENDIX B

APPENDIX C

1

System Architecture

The 6500 family of microprocessors is made up of third-generation microprocessor devices that incorporate an 8-bit bidirectional data bus and a unidirectional address bus consisting of from 12 to 16 address lines. All of the 6500 family microprocessors, except the 6500/1, contain the same silicon chip. The 6502 and 6512 are housed in 40-pin DIP packages, while the 6503, 6504, 6505, 6513, 6514, and 6515 are housed in 28-pin DIP packages.

The 6512 offers a data bus enable (DBE) signal not found on the more popular 6502. The reason that the 6512 chip is not often used is that systems complex enough to require direct memory access (DMA), which is made possible by the DBE signal, almost invariably have external data bus buffering. Bus access in such systems is accomplished at the buffers. The other processors previously mentioned are offered in the less expensive 28-pin package by simply eliminating certain control and bus signals. In production designs, the savings in component cost derived from using these smaller devices is approximately 10%, and the savings in board area is approximately 0.6 square inch. The instruction sets are identical for all devices.

The 6500/1 is a stand-alone single-chip microcomputer that contains four 8-bit I/O ports, two internal timers, ROM, and R/W memory. The instruction set includes the full 6502 set, plus several special instructions associated with its internal timers and ports. Since this device contains its own I/O ports, and is not normally associated with complex interface problems, it is not discussed in depth in this book.

The architecture of the 6500 family is very similar to that of the Motorola 6800 family, the most noticeable difference being

the absence of the 6800 valid memory address (VMA) signal. This similarity means that both families can be considered as a resource pool in designing interfaces for either microprocessor family. This combination represents a very substantial level of support. Most of the interface circuits presented in this book can be used with the MC6800 family microprocessors, provided that the designer simply ANDs VMA with $\phi 2$ and uses this signal where the $\phi 2$ signal is used (mostly in address decoder circuits).

MEMORY VERSUS REGISTERS

The 6500 (and 6800) microprocessors are designed around a memory-oriented concept as opposed to the register-oriented concept found in processors such as the 8080 and Z80. Both concepts have their strong and weak points, and it is important that the designer recognize and utilize the positive features. Under the memory-oriented concept, interfaces to the outside world are treated exactly as if they were memory locations. There are no special input or output instructions in the instruction set. Any instruction that can be used with memory can also be used to communicate with peripheral devices.

Since one of the advantages of the memory-oriented concept is the ability of the microprocessor to do numerous logic and mathematical operations directly on memory (without first transferring data or data pointers to registers), this same power can be applied to the interface devices.

PAGE ZERO AND INDIRECT ADDRESSING

Page zero (Z-page) has special significance in the 6502. Besides the absolute mode (that can access any memory location), there are two additional modes of addressing that are associated with page zero. These two modes are called *Z-page* and *indirect addressing*.

When the microprocessor uses Z-page addressing to access memory in page zero, it must generate a 16-bit address just as in the absolute addressing mode, but only two bytes are required in the program. The microprocessor recognizes the op code as implying not only the operation (LDA, STA, EOR, etc.), but also the high-address byte. The microprocessor will always generate a high byte of the address that is zero. The lower eight bits of address are specified in the byte that follows the op code.

In indirect addressing, page zero can be used to store up to 128 16-bit addresses or pointers. When the microprocessor encounters an indirect op code in the program, it will do two

read operations in page zero. The first read will be at the Z-page location addressed by the byte that follows the op code, and the second will be at the next higher Z-page location. The two bytes read during these fetches will form the low and high bytes, respectively, of the memory location at which the op code will be executed. Thus, at times, these memory locations will act as if they were actually registers in the CPU chip. While it is not within the scope of this book to go deeply into programming the 6500 microprocessor family, it is essential that the interface designer understand the power of the indirect addressing technique for him to efficiently interface to these processors.

Some designers incorporate I/O interfaces into the Z-page memory space to take advantage of the abbreviated instructions mentioned earlier. This use of Z-page *is not recommended* unless the scope of the system is very limited. There are two reasons for this: first, the interfaces use up memory locations which could be used more effectively to store pointers, and, second, portions of a program that are not concerned with the I/O operations may attempt to use this address space as memory, and thus cause strange and unexpected operation of peripherals. In any system that is running a diversity of software, page zero can become a very crowded and busy place.

PAGE ONE AND THE STACK

In the 6500 family microprocessors, page one of memory ($0100 to $01FF) contains the stack. While the location of the stack may be manipulated within this page, the high-order address byte of all stack operations remains #01. Because of this, addressing interfaces within this page is not advisable.

PAGE $FF AND THE VECTORS

Page $FF, the highest address page, contains the NMI (nonmaskable interrupt), RES (reset), and IRQ (interrupt request) vectors. In a dedicated control system these vectors are usually in ROM (read-only memory), while in most general purpose systems they are contained in a small block of R/W (read/write) memory. Systems that contain these vectors in R/W memory usually utilize a technique known as *ROM first RAM* or *bootstrapping*, where a ROM device occupies this portion of the microprocessor's address space when the power-on reset is activated. This ROM contains a reset vector or address that causes the microprocessor to jump to the beginning of the ini-

tialization program. This program, by communicating with an I/O port, causes the ROM to be disabled, and a block of R/W memory to be enabled, whenever an address that contains $FF in the high byte is on the address bus. The software then causes the appropriate vectors to be written into the R/W memory.

The locations of these vectors, or address pointers, are as follows:

NMI	low address byte	FFFA
NMI	high address byte	FFFB
RESET	low address byte	FFFC
RESET	high address byte	FFFD
IRQ	low address byte	FFFE
IRQ	high address byte	FFFF

Before using page $FF for I/O operations, the designer must ascertain that no conflict exists between the memory address used in the interface, and the R/W memory or ROM.

THE SYSTEM CLOCK

The 6500 processors are dynamic devices, meaning that they cannot be halted without the loss of information stored in the registers. While the clock speed may be varied over a reasonable range, most applications utilize either the 1-MHz or 2-MHz version of the CPU chips, and operate the clock at that speed. In actuality, the 2-MHz parts are simply derived from sorting the CPU chips, and are otherwise identical to the slower parts. Running either device below 500 kHz is generally inadvisable. While the 6500 processors are advertised as containing a built-in clock oscillator circuit, they contain only an MOS inverter which can serve as a simple oscillator.

Many of the early clock oscillator circuits recommended by the manufacturers were not reliable when used with a variety of crystals. One factor that affected this was the rise time of the system power. Occasionally, the oscillator would start at either a subharmonic or harmonic of the desired frequency, and occasionally the oscillation would be in the parallel mode for series crystals and vice versa. In most cases, if a crystal is operating in the wrong mode it tends to have an error in frequency of from 1% to 20%.

The oscillator circuit shown in Fig. 1-1 has been widely published in applications literature. Resistor R1 is used to balance the duty cycle of the oscillator (make it closer to 50%). While the circuit is very reliable with high values of R1, it becomes

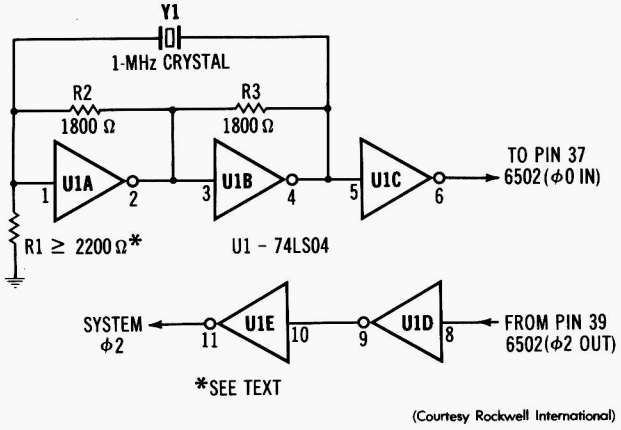


Fig. 1-1. Factory recommended clock circuit for the 6500 series microprocessors.

less reliable as R1 approaches 2200 ohms. Sometimes it is not possible to obtain a 50% duty cycle without losing reliability.

A slightly modified version of the oscillator is shown in Fig. 1-2 and is recommended here for oscillators that are running at the system clock rate (and, therefore, require a 50% duty cycle). The interstage coupling capacitor (C1) in the second version reduces the dc loop gain, and thus causes a 50% duty cycle to be obtained.

In some cases, it is economically beneficial to use a crystal at twice or four times the system clock frequency. These crystals are smaller and often less expensive than their lower frequency counterparts. The use of a divider after the oscillator eliminates the problem associated with obtaining a 50% duty cycle.

In operation, the clock determines the system timing. In conjunction with the read/write line, the clock controls the flow of data on the data bus. The two half cycles of the clock are referred to as $\phi 1$, and $\phi 2$. During $\phi 1$, the processor puts the address of the next memory operation on the address bus. During some cycles this is meaningless, as the processor has not yet fully constructed the address desired. On the next half cycle ($\phi 2$), data will be transferred to or from the addressed device.

THE READ/WRITE LINE AND PIPELINING

The 6500 microprocessors utilize a technique known as *pipelining* to increase the throughput on the chips. Pipelining means that the CPU may still be processing previously received data while reading new data. The read/write line determines

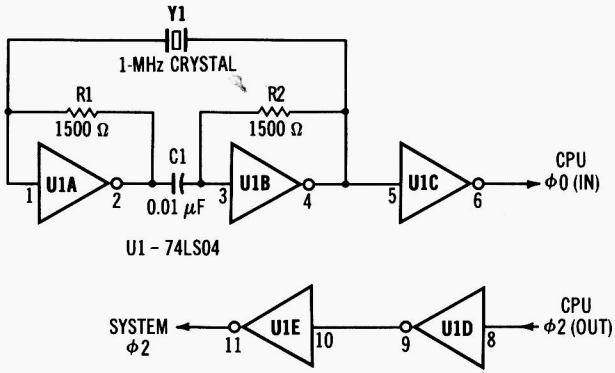


Fig. 1-2. Modified version of the circuit shown in Fig. 1-1. This clock circuit uses capacitive coupling to obtain duty cycle balancing.

the direction of data flow on the data bus when $\phi 2$ is active. This line will be high (logic true or near +5 volts) if data is to be read from the addressed memory location or interface device, and it will be low (near ground) if the transfer is to be from the processor to the device (write).

A special word of caution is in order at this point. Since the 6500 processors do not generate a VMA or valid memory address signal as do the 6800 processors, an interesting bug can arise. It was mentioned earlier that during some cycles when the full desired address had not yet been calculated, an incorrect intermediate address is output on the address bus. Normally, this address is the base address of an impending indexed address operation. For example, if the processor were calculating the memory address to be used for LDA ($1000),Y, it would go through one cycle in which the incorrect address of ($1000) was read from memory. This value would then be discarded by the processor, and the correct value of the address ($1000+Y) would be read. Since these incorrect addresses always force the read/write line to a read state (logic one), the effect of the fetch of the incorrect memory address is that some data value is placed on the data bus, but it is ignored. This is not true in fetches from interface devices. Since some LSI interface chips contain up to 16 register addresses, indexed addressing is often used to initialize or reinitialize groups of registers. The problem arises when a port of the device is operating in the *handshaking* (see Chapter 2) mode and inadvertently receives a false read operation as described above. This may cause an undesired reset of the data-ready flag in the status register of the LSI chip,