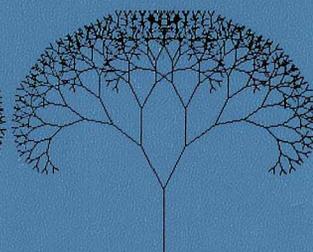
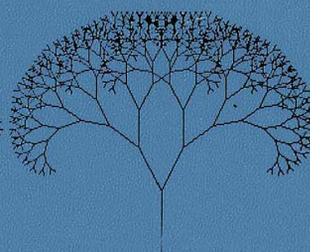
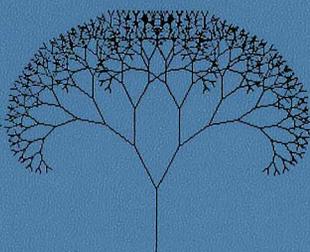
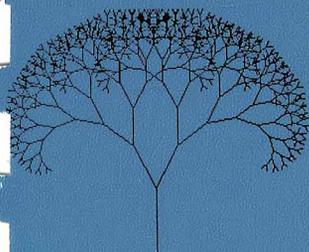


**ASSEMBLY LANGUAGE PROGRAMMING**  
**For The COCO 3**

**An Addendum to ASSEMBLY LANGUAGE PROGRAMMING**  
**For The TRS-80 COLOR COMPUTER**

By  
Laurence A. Tepolt





**ASSEMBLY LANGUAGE PROGRAMMING**  
**For The COCO 3**

**An Addendum to ASSEMBLY LANGUAGE PROGRAMMING**  
**For The TRS-80 COLOR COMPUTER**

By  
Laurence A. Tepolt

Published by TEPCO  
30 Water Street  
Portsmouth, RI 02871

Every effort has been made to ensure that the information provided in this book is accurate and complete. However, neither TEPCO nor the author assume any liability resulting from omissions, inaccuracies, or from the use of information contained herein.

Radio Shack, TRS-80, EDTASM+, Color BASIC, Extended Color BASIC, Disk Extended Color BASIC, and Super Extended Color BASIC are registered trademarks of the Tandy Corporation. IBM is a registered trademark of the International Business Machines Corporation.

Copyright (c) 1987 by Laurence A. Tepolt. All rights reserved. No part of this book may be reproduced in any form or by any means without written permission from the publisher. Printed in the United States of America.

# INTRODUCTION

This book is for the assembly language programmer of the Tandy Color Computer 3. In it are descriptions of the enhancements of the CoCo 3 over the earlier CoCo 1 and 2. Also, ideas and guidelines for using the enhancements are provided.

One purpose of this book is to provide complete, accurate, and understandable descriptions of all the new CoCo 3 capabilities. Another purpose is to help the reader learn assembly language programming. Therefore, the information in this book is organized in a manner to facilitate learning; much like a tutorial. New technical terms are printed bold at their initial occurrence. This indicates the term is important and that the reader should learn its meaning.

This book is written as a continuation of the book "**Assembly Language Programming for the TRS-80 Color Computer**" (henceforth referred to as **The Book**), published by TEPCO. As such, this book starts where **The Book** stopped. Also, this book makes references to **The Book**.

Chapter One is a general introduction to the CoCo 3. It describes the CoCo 3 capabilities, I/O connectors, and provides an internal block diagram.

Chapter Two provides background information on color and describes how to generate the desired colors on the various types of monitors (TV, composite, and RGB).

Chapter Three introduces the concepts of physical and virtual memory and describes how to use them.

Chapter Four describes the new High Resolution display capabilities (text and graphics) and how to use them.

Chapter Five describes the original Low Resolution display capabilities (text, semigraphics, and graphics) and how to use them.

Chapter Six describes the interrupt sources and how to control and use them.

Chapter Seven provides programming guidelines and examples and describes CoCo 3 capabilities not covered elsewhere.

A Cross Reference of Dedicated Addresses is provided at the back of this book instead of an index.



# CONTENTS

<b>1 CoCo 3 Overview</b> . . . . .	<b>1</b>
Program Resources - Display Modes - Input/Output Connectors - Internal Block Diagram	
<b>2 Colors And Monitors</b> . . . . .	<b>9</b>
Color - Palette Registers - Monitors	
<b>3 Physical &amp; Virtual Memory</b> . . . . .	<b>17</b>
New View of Memory - Managing Memory	
<b>4 High Resolution Displays</b> . . . . .	<b>22</b>
ACVC Operation - Text Display - Graphics Display	
<b>5 Low Resolution Displays</b> . . . . .	<b>35</b>
ACVC Operation - Text Display - Graphics Display	
<b>6 Interrupts</b> . . . . .	<b>39</b>
MPU Interrupt Sequences - ACVC Interrupt Generation - Interrupt Programming Guidelines	
<b>7 Concluding Details</b> . . . . .	<b>46</b>
Reset Initialization - Advanced Graphics - Interrupts & Virtual Memory - Odds & Ends	
<b>Appendix A - High Resolution Text Characters and Video codes</b> . . . . .	<b>55</b>
<b>Cross Reference - Dedicated Addresses</b> . . . . .	<b>56</b>

THE HISTORY OF

THE CITY OF BOSTON  
FROM THE FIRST SETTLEMENT  
TO THE PRESENT TIME

BY  
NATHAN OLSZEWSKI  
Author of "The History of the City of Boston"  
and "The History of the City of New York"

NEW YORK  
G. P. PUTNAM'S SONS  
1898

Copyright, 1898, by  
G. P. PUTNAM'S SONS  
New York

Printed and Published by  
G. P. PUTNAM'S SONS  
New York

Illustrations by  
J. H. MANNING  
New York

Published by  
G. P. PUTNAM'S SONS  
New York

# Chapter 1

## COCO 3 OVERVIEW

The Tandy Color Computer 3 (CoCo 3) is an enhanced or improved version of the original CoCo 1 and 2. The improvements consist of additional capabilities. In an effort to maintain compatibility with the original CoCo, the CoCo 3 also provides most of the original capabilities.

This chapter provides general descriptions of the CoCo 3 capabilities. The capabilities are categorized as program, display, and input/output connectors. An internal block diagram is also provided. More detailed descriptions are in later chapters.

### PROGRAM RESOURCES

Program resources are those capabilities which are inherently used by a program. They include the microprocessor, memory, interrupts, and special purpose control registers.

#### Microprocessor

The microprocessor in the CoCo 3 is the MC68B09E. It operates exactly the same as the original MC6809E except it is able to run reliably at twice the speed of the original. In the CoCo 3 and the original CoCos the MPU clock rate (which determines the MPU's speed of operation) is normally 0.89 MHz. In the CoCo 3 the clock rate can be changed, by software, to 1.78 MHz; this doubles the speed at which the MPU performs its operations. All programs (except those which depend on the slower MPU speed such as cassette tape, serial, and disk I/O) should run reliably at the higher MPU speed. Also, other components in the CoCo 3, such as the PIAs, are of the high speed version so they will operate correctly at the high MPU clock rate.

## Memory

The CoCo 3 is available in two versions: those with 128K and those with 512K of memory. The 128K version may be upgraded to 512K. Since the MPU is capable of accessing only memory addresses 0-64K, there is a new function in the CoCo 3 - the **Memory Management Unit (MMU)**. The MMU provides the capability of accessing 512K of memory by intercepting the 16-bit address sent from the MPU and expanding it to 19 bits before sending it to the memory.

## Interrupts

The interrupt sources include those of the original CoCo and the new sources. The original sources are the IRQ type (horizontal and vertical sync), the FIRQ type (serial ready and cartridge), and the NMI type (from the cartridge only). The original IRQ and FIRQ interrupts are controlled as before, through the PIAS. The new interrupt sources may be of either the IRQ or FIRQ types and are listed below.

- Programmable Timer
- Horizontal Sync
- Vertical Sync
- Serial Input Data
- Keyboard/Joystick Fire Buttons
- Cartridge

## Special Purpose Control Registers

The control registers allow a program to control various aspects of the computer's operation such as display modes, colors, memory access, MPU speed, and input/output. The control registers associated with the SAM, VDG, and PIAS are almost unchanged. The new control registers are used to control the new CoCo 3 capabilities.

## DISPLAY MODES

This discussion addresses the various ways information may be displayed on the TV or monitor screen. The display capabilities include most of those originally available plus the newly available as shown in Table 1-1.

The choice of colors has been increased to 63. This includes the nine original colors plus 52 more. However, be aware that specific display modes use only 2, 4, 8, or 16 colors at one time.

Monitors	TV (B/W or color) Composite (monochrome or color) RGB
Low Resolution Display (original)	Text (32 col X 16 row) Semigraphics 4 (SG4) Graphics 1 with Color (G1C) Graphics 1 with Resol (G1R) Graphics 2 with Color (G2C) Graphics 2 with Resol (G2R) Graphics 3 with Color (G3C) Graphics 3 with Resol (G3R) Graphics 6 with Color (G6C) Graphics 6 with Resol (G6R)
High Resolution Display (new)	Text (32/40/64/80 col X 24/25/28) Graphics Vert - 192/200/225 Colors - 2/4/16 Horz - 64/80/128/160/256/320 512/640/1024/1280

Table 1-1. Display Capabilities

## INPUT/OUTPUT CONNECTORS

The input/output connectors are those by which external devices are connected to the Color Computer. External devices either send data to (input) the computer, or receive data from (output) the computer, or both.

The CoCo 3 has all the connectors of the original CoCo and three more. Of the original connectors, only the joystick connectors are changed slightly to incorporate the two fire buttons of the new joysticks. The new connectors are labelled Audio, Video, and RGB. The locations of the connectors as viewed from the rear of the computer are shown in Fig. 1-1. A description of each connector and its signals follow.

### Audio

The audio connector is a phono jack through which an audio signal may only be output. The audio signal is low impedance and may have an amplitude of up to 1.5 volts peak-to-peak. The audio signal may be generated by the computer or may come from the cassette tape recorder or the cartridge. Normally the audio signal is connected to a composite monitor, if in use. Otherwise, the audio signal may be connected to a hi-fi system or some other device meant to receive this type of signal.

### Video

The video connector is a phono jack through which a composite video signal is only output. The video signal is of 75 ohm impedance and has an amplitude of 1.0 volts peak-to-peak. The composite video signal is normally sent to a composite monitor where it causes the expected picture to be

displayed. Otherwise, it could be connected to the video-in jack of a VCR and the pictures could be recorded. The composite video signal is a mixture of the luminance, color, and horizontal and vertical synchronization analog signals generated in the computer.

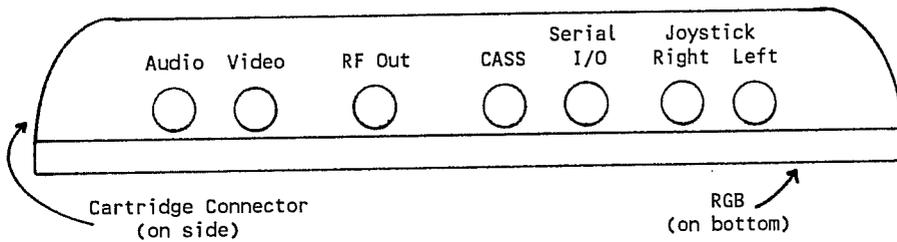


Fig. 1-1. I/O Connectors (rear view)

### RF Out

The RF Out connector is a phono jack through which an RF TV signal is only output. The RF signal is a mixture of the composite video and audio signals and is output on TV channel 3 or 4 (switch selectable). Its impedance is 75 ohms.

### CASS

The CASS connector is a 5-pin DIN jack through which the cassette tape recorder motor is controlled and data is either output to or input from the cassette recorder. Fig. 1-2 shows the pin assignments of this connector as viewed from the rear of the computer.

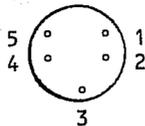


Fig. 1-2. CASS Connector (rear view)

- Pins 1 and 5 - The cassette motor relay (in the computer) contacts are connected between pins 1 and 5. When the relay is energized under program control, the contacts close and there is electrical continuity between pins 1 and 5 which causes the cassette motor to run. A program may de-energize the relay which causes the contacts to open and the cassette motor to stop.
- Pin 2 - This pin receives data (as an analog audio signal) from the cassette tape recorder.
- Pin 3 - This pin is connected to the computer's ground (chassis).

- Pin 4 - Out this pin is sent data (as an analog audio signal) to the cassette recorder.

### Serial I/O

The serial I/O connector is a 4-pin DIN jack through which serial (RS-232) data may be sent or received. Fig. 1-3 shows the pin assignments of the serial I/O connector as viewed from the rear of the computer.

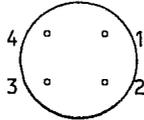


Fig. 1-3. Serial I/O Connector (rear view)

- Pin 1 - This pin receives the Carrier Detected (CD), or more simply the Ready, signal from the external device.
- Pin 2 - This pin receives data (RS-232IN) from the external device.
- Pin 3 - This is connected to the computer's ground (chassis).
- Pin 4 - Out this pin is sent data (RS-232OUT) to the external device.

### Joysticks - Right/Left

The right and left joystick connectors are each a 6-pin DIN jack through which signals are received that indicate the joystick position and whether fire buttons 1 or 2 are depressed. Fig. 1-4 shows the pin assignments as viewed from the rear of the computer.

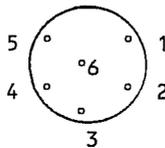


Fig. 1-4. Joystick Connector (rear view)

- Pin 1 - This pin receives an analog voltage (0-5 volts) that represents the left/right position of the joystick.
- Pin 2 - This pin receives an analog voltage (0-5 volts) that represents the forward/back position of the joystick.
- Pin 3 - This is connected to the computer's ground (chassis).

- Pin 4 - This receives a signal from fire button 1 that indicates whether it is depressed (0 volts) or not (+5 volts).
- Pin 5 - Out this pin is sent +5 volts from the computer to the joystick.
- Pin 6 - This receives a signal from fire button 2 that indicates whether it is depressed (0 volts) or not (+5 volts).

## RGB

The RGB monitor connector is a 10-post (on 0.1 inch centers) jack through which analog signals are sent to control an RGB monitor. Fig. 1-5 shows the post assignments as viewed from the bottom of the computer.

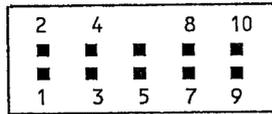


Fig. 1-5. RGB Connector (bottom view)

- Pins 1 and 2 - These are connected to the computer's ground (chassis).
- Pin 3 - Out this pin is sent the analog signal that controls the red color intensity.
- Pin 4 - Out this pin is sent the analog signal that controls the green color intensity.
- Pin 5 - Out this pin is sent the analog signal that controls the blue color intensity.
- Pin 6 - Unused.
- Pin 7 - Out this pin is sent the audio signal.
- Pin 8 - Out this pin is sent the horizontal synchronization pulse (positive).
- Pin 9 - Out this pin is sent the vertical synchronization pulse (positive).
- Pin 10 - From this pin the computer receives a voltage indicating whether an RGB monitor is connected (0 volts) or not (+5 volts).

## Cartridge Connector

The cartridge connector is a 40-pin jack that provides access to the main internal busses (data and address) and several control signals. Its signals are described in **The Book**.

## INTERNAL BLOCK DIAGRAM

This section provides general descriptions of the major components in the CoCo 3 and a block diagram that shows how they are interconnected. The major components are the MPU, RAM, ROM, ACVC, and PIAs. The internal block diagram is shown in Fig. 1-6.

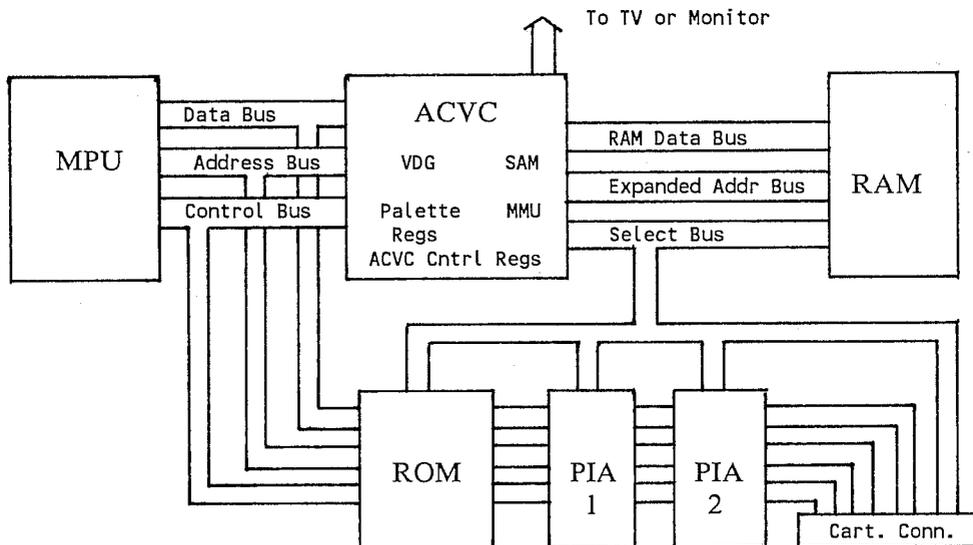


Fig. 1-6. CoCo 3 Block Diagram

## MPU

The MPU is an MC68B09E. It executes instructions in RAM or ROM and responds to interrupts. It is described in **The Book**.

## RAM

The RAM, or random access memory, is the component in which are stored instructions and/or data that may be needed by some other component. This type of memory is volatile; its content is lost if electrical power to it is turned off. The RAM size may be of 128K or 512K bytes.

## ROM

The ROM, or read-only memory, is a type of memory such that its

content (instructions and/or data) is fixed or unchangeable. The ROM size is 32K bytes. In ROM are the programs: Color BASIC, Extended Color BASIC, Super Extended BASIC, and Reset Initialization.

## ACVC

The ACVC, or **advanced color video chip**, is the main controller of the computer. Some people call it the GIME chip. It controls the operating modes and synchronizes the operation of the other major components. The functional areas in the ACVC are described below.

- VDG - The video display generator is that which controls and generates the original CoCo display modes. It generates a video signal which is sent to the TV or monitor.
- SAM - The synchronous address multiplexer is used in conjunction with the VDG to generate the original CoCo display modes and to control the MPU clock rate and ROM/RAM map mode.
- Palette Registers - There are sixteen palette registers which can be loaded by a program with the codes that correspond to the colors that program will use. Then, to generate the desired colors, the program specifies the appropriate palette registers.
- MMU - The memory management unit serves as a programmable interface between the MPU and RAM. It intercepts the 16-bit address from the MPU and expands it to nineteen bits before sending it to RAM. This allows the MPU to access up to 512K of memory. The MMU is composed of the logic necessary to perform the address expansion and sixteen **page address registers (PARS)**. The PARS exist as two sets of eight PARS each.
- ACVC Control Registers - These include the other registers used to control the new high resolution display modes, interrupts, and other miscellaneous operating modes.

## PIAs

The PIAs are the peripheral interface adapters which are primarily used to perform I/O to the external devices and the keyboard. Their operation and use are described in **The Book**.

## Chapter 2

# COLORS And MONITORS

This chapter describes color, the monitors usable with the CoCo 3, and how the CoCo 3 generates the available colors. First, color is introduced in both the technical and psychological sense.

### COLOR

In this discussion color is a human perception. This perception is the end product of a chain of events that starts with raw light. In this case raw means the light is only its physical entity (electromagnetic radiation). Next, light enters the eye where it is focused, detected, processed somewhat, and the resulting information is transmitted to the brain. There this information is processed further (at a subconscious level) until qualities such as shape, motion, and color are detected and one finally becomes conscious of them (that is, perceives them). Throughout this complex processing path are opportunities for false perceptions, or illusions, to be generated. In particular, the perceptions, true and false, generated by our cerebral color processing are extensively used by color TVs and monitors to cause us to perceive the desired results.

### Light and Color Vision

Light is electromagnetic radiation or waves much like radio waves but of a much higher frequency. For example, let's say audible sound is analogous to electromagnetic radiation. On our 'analog piano' the bass notes (on the far left) correspond to the lower frequency radio and TV waves. The treble notes (on the far right) correspond to the higher frequency waves such as X rays and gamma rays. Then, the seven notes in the middle would correspond to the frequencies of the colors red, orange, yellow, green, blue, indigo, and violet. This rainbow series of colors in Fig. 2-1 is the same as generated by a prism on which is directed a beam of

truly white light. There are actually many more colors between the major ones identified in Fig. 2-1. Also, a prism can be used in reverse such that it generates a beam of white light when the rainbow series of colors is properly directed toward it.

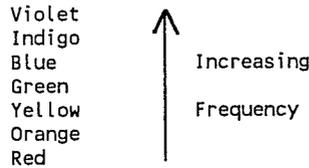


Fig. 2-1. Rainbow Colors

The basic truths are that each pure or rainbow color is electromagnetic radiation of a specific frequency and white light is a mixture of all the pure colors. And for completeness, black is the absence of light. Also, human perception of **monochrome** (single-frequency) colors agrees with these basic truths.

Now let's combine colors. In this discussion colors are combined in the **additive** sense, such as when shining two differently colored lights on one spot. This is different from combining colors in the **subtractive** sense, such as when coloring paper with red and green crayons. What color is perceived when red and green lights are shown on a white surface? Yellow! Even though the light reflected is a mixture of red and green (electromagnetic radiation of two different frequencies), the human visual processing generates the illusion of yellow. In fact, with a minimum of three well chosen colors one can generate the illusion of any of the other pure colors by combining the three chosen colors in appropriate proportions.

The three most commonly chosen colors, also called **primary** colors, are red, green, and blue. These primary colors are emitted by dots on all color TV and monitor screens and their relative brightness is controlled to produce the desired color illusions. Fig. 2-2 shows the primary colors and the illusory colors generated where they overlap.

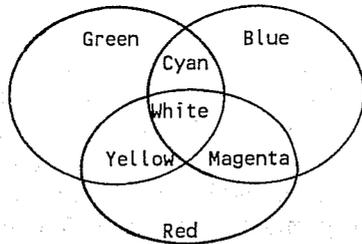


Fig. 2-2. Additive Primary Colors

## **Describing Colors**

The basic attributes of color are its **hue, saturation, and brightness**. Of course there are many others such as texture, glare, and glitter, but the three basic attributes serve well when describing the colors generated by the CoCo 3.

- **Hue** - This is a classification of the color. The hues include all the rainbow colors and those between adjacent rainbow colors such as greenish yellow.
- **Saturation** - This is a measure of the vividness or amount of color in the sample. For example, medium blue is more saturated than pale blue.
- **Brightness** - A measure of how light or dark the sample color is. It is easiest to imagine the color sample has a background that may vary from black, through the grays, to white. A bright color has a white background and a dark color has a black background.

## **Using Colors**

The goal is to create a screen image that will have the desired effect on the viewer. The raw materials are the available colors. The main tool used to create the image is the knowledge of how to draw and work with colors. Some items of that knowledge are presented here. More detailed information can be found in drawing and art books at a library.

- **Color Contrast** - Color differences between objects are accentuated when they are juxtaposed. For example, of two objects of different brightness, one will appear brighter and the other darker than they actually are.
- **Advancing/Retreating Colors** - These colors seem to provide a 3-dimensional image. The reddish colors appear closer to the viewer and the bluish colors appear farther away.

## **PALETTE REGISTERS**

This section describes how to generate the available colors on the two types of monitors (composite and RGB) that may be used. The colors are specified with palette registers whose operation is different for the two monitor types. In this section a TV is considered to be a type of composite monitor.

There are sixteen palette registers (numbered 0-15) that are accessed by their dedicated addresses (FFB0-FFBF). This is shown in Table 2-1. Each palette register is composed of only six bits (5-0).

A color is selected for future use by writing its color code in one or more of the palette registers. A palette register's 6-bit size provides for 64 (0-63) different color codes. A program would normally proceed by first loading the palette registers with the color codes which correspond to the colors to be used. When the program is to display a color on the screen it does so by specifying the palette register number which contains the appropriate color code. That is, the colors are selected indirectly.

Palette Register	Dedicated Address	Palette Register	Dedicated Address
0	FFB0	8	FFB8
1	FFB1	9	FFB9
2	FFB2	10	FFBA
3	FFB3	11	FFBB
4	FFB4	12	FFBC
5	FFB5	13	FFBD
6	FFB6	14	FFBE
7	FFB7	15	FFBF

Table 2-1. Palette Register Addresses

### Composite Color Set

The composite color set is the set of colors generated on a composite monitor (or TV) by the 64 color codes. When using a composite monitor the bits of the palette registers are interpreted as shown in Fig. 2-3. The six bits (5-0) generate a color by specifying its hue, saturation, and brightness.

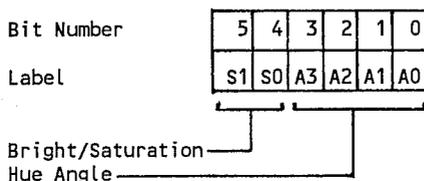


Fig. 2-3. Composite Palette Register

Bits 3-0 select a hue from a hue wheel by specifying an angle. The angles are number 0-15 and select a hue as shown in Fig. 2-4. The primary colors are underlined and all the others are the illusory colors. Each illusory color, except gray, is a combination of the two primary colors it is between. Gray is a combination of all three primary colors. For example, indigo is mostly blue and some red.

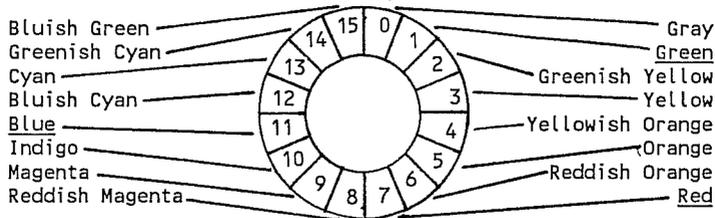


Fig. 2-4. Hue Wheel

Bits 4 and 5 select both the brightness and saturation. They can't be selected independently. Table 2-2 shows the four bit combinations and the corresponding brightness and saturation.

S1	S0	Brightness	Saturation
0	0	low	high
0	1	medium	medium
1	0	high	low
1	1	very high	very low

Table 2-2. Brightness and Saturation

The exception to this scheme is the color generated by color code 63. It is not a bluish green of very high brightness and very low saturation. It is white. Some example of colors and their codes follow.

Binary Code	Color
000000	Black
000111	Dark Red
011011	Medium Blue

### RGB Color Set

The RGB color set is the set of colors generated on an RGB monitor by the 64 color codes. In this case the bits of the palette registers are interpreted as shown in Fig. 2-5. The six bits (5-0) generate a color by specifying its primary color components and their intensities.

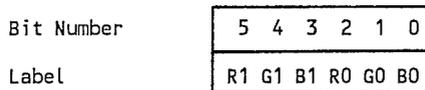


FIG. 2-5. RGB Palette Register

The red intensity is specified by the R1 and R0 bits, green by G1 and G0, and blue by B1 and B0. The primary color intensities that may be selected by each pair of bits are shown in Table 2-3.

X1	X0	Intensity
0	0	none
0	1	low
1	0	medium
1	1	high

Table 2-3. Primary Color Intensities

The relative intensities of the primary colors required to generate a particular display color can be approximately determined from the hue wheel in Fig. 2-4. For example, orange is mostly red and some green. Pastel colors are generated by mixing light gray and the desired hue. Examples of colors and their binary codes are shown below.

Color Code	Color	Components
000000	Black	None
000111	Dark Gray	A little of all
100110	Orange	Mostly red + little green
111100	Pink	Light gray + little red
110110	Bright Yellow	Red + green
111111	White	All

### Alternate Color Set

Setting bit 5 of dedicated address FF98 invokes the alternate color set. In this set all the same colors are available; they are just specified by a different color code. Enabling the alternate color set has the effect of shifting all the colors, except gray, half way around the hue wheel in Fig. 2-4. Thus the hue angle specifies a different color than with the normal color set. The purpose of the alternate color set is to simulate the original artifact color effect on a TV. With the original CoCo, sometimes the artifact colors would be of one set and other times of another set.

### MONITORS

This section describes the three general types of monitors (TV, composite, and RGB) and how they perform. Since the monitors do perform differently it would be best for a program to ask the user which monitor type is in use. The program would then adjust its displays and color sets accordingly. The most common and reasonably priced monitors are those capable of medium to high resolution display. A monitor of very high resolution is required to accurately display the highest resolution graphics displays of the CoCo 3.

### Television Set

For some, a TV may be the lowest cost monitor but then it provides the lowest quality picture. Presented here are highlights of TV monitor use.

- **Black & White TV** - Only black, white, and shades of gray are available. For the most ledgeable text the appropriate palette registers should be loaded with the codes of black and white.
- **Color TV** - The colors available are those of the composite color set. Artifacts are also available for generating a few more colors in display modes that normally provide few colors.
- **Display** - The display resolution is about 260H x 250V. This is sufficient for a highly readable 40 columns of text. In the new high resolution displays about 8% of the screen image extends off the left edge of the picture tube.
- **European** - In Europe, TVs require a slightly different video signal. To provide this, European CoCo 3s are outfitted with a PAL circuit. Also, bit 3 of dedicated address FF98 must be set. This selects a 50 Hz vertical synchronization rate (rate at which each frame is redrawn on the screen). Presumably, a European ROM version would set this bit.

### **Composite Monitor**

A wide variety of monochrome and color composite monitors is available. Just be sure the one you purchase is meant to receive an analog composite video signal (not the IBM PC type). They all display the full screen image. If you want sound capability the monitor must have an amplifier and speaker.

- **Monochrome** - These monitors provide just green or amber on a black background. For the highest quality display the appropriate palette registers should be loaded with codes for white and black. Also, the color signal (which will confuse a monochrome monitor) should be turned off by setting bit 4 of dedicated address FF98. Normally these monitors have a resolution of 800-1000H x 300V which provides a very ledgeable 80-column text display.
- **Color** - The colors available are those of the composite color set. The medium resolution models have a resolution of 260H x 300V and are those normally used with Commodore and Apple II computers. They provide a very ledgeable 40-column text display. The high resolution (640H x 300V) models provide a readable 80-column text display.
- **European** - European composite monitors require a slightly different video signal. To provide this, European CoCo 3s are outfitted with a PAL circuit. Also, bit 3 of dedicated address FF98 must be set. This

selects a 50 Hz vertical synchronization rate (rate at which each frame is redrawn on the screen). Presumably, a European ROM version would set this bit.

### **RGB Monitor**

A variety of these monitors is available. Just be sure the one you purchase is meant to receive analog R, G, B, horizontal, and vertical signals (not the IBM PC type). If you want sound capability the monitor must have an amplifier and speaker.

- Colors - The colors available are those of the RGB color set. Artifact colors are not available.
- Display - The full screen is displayed and, with a typical resolution of 640H x 300V, 80-column text is quite readable.

## Chapter 3

# PHYSICAL & VIRTUAL MEMORY

This chapter describes a new way of viewing memory and how the much larger memory amount is accessed and used. Such concepts as virtual and physical memory and memory mapping and pages are described.

### NEW VIEW OF MEMORY

Now that the amount of memory available in the CoCo 3 does not match the MPU addressing capability, there are two classifications of memory. In the original CoCo the two classifications were the same, but they are now disconnected from each other.

#### Virtual/Physical Memory

**Virtual memory** is that memory which a program segment assumes to be available. This includes that in which reside the program segment instructions and its local data area. The virtual memory extent is limited by the MPU 16-bit address to 0-64K. The virtual memory may be RAM, ROM, or some of both.

**Physical memory** is the total amount of usable memory in the computer. Not all of it may be in used at the same time, but it is available. Physical memory includes RAM and ROM. The reduced size, complexity, and cost of memory have resulted in the construction of computers with far more physical memory than the MPU addressing extent can cover. A CoCo 3 with 128K of RAM also has an internal 32K of ROM; this provides 160K of physical memory. A 512K CoCo 3 has 544K of physical memory.

#### Memory Pages

The memory extent, either virtual or physical, is no longer considered as a seamless continuum but rather as segmented into pages. A memory page is an 8K block (where each block is a continuous range of addresses) of

memory that starts on an 8K boundary. For example, addresses 0-1FFF and A000-BFFF each constitute a valid memory page. Also, each page is identified by its page number.

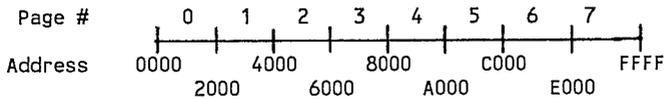


Fig. 3-1. Virtual Memory Pagination

The virtual memory extent is divided into eight 8K pages as shown in Fig. 3-1. They are numbered 0-7. The page number that any particular address occupies is determined by the three MSBs of that address. For example, DC23 is in page six as shown below.

$$\begin{aligned} \text{DC23} &= \underline{1101} \ 1100 \ 0010 \ 0011 \\ \text{Page \#} &= 6 \end{aligned}$$

The 512K physical memory extent is divided into sixty-four 8K pages as shown in Fig. 3-2. The pages are numbered 0-3F. The page number of any particular address is determined by the six MSBs of that physical address. In the CoCo 3, physical memory is considered to start at the 512K address point and extend downward toward zero. Therefore, in a 128K system, the physical memory extent proceeds from the 512K point down to 384K. Thus, the physical memory pages available in a 128K computer are 30-3F.

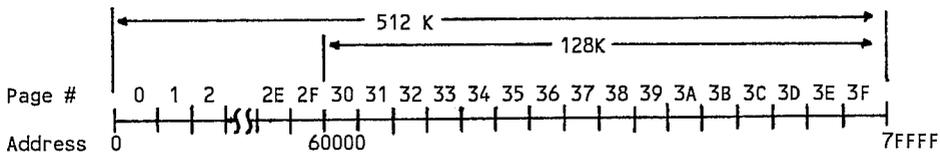


Fig 3-2. Physical Memory Pagination

## MANAGING MEMORY

Memory management is a new program responsibility; controlling the virtual and physical memory pages. Its purpose is to associate the required virtual memory pages with the appropriate physical memory pages. It is done by a program segment (the memory manager) that controls the memory management unit.

## RAM Memory Mapping

Memory mapping is the process of associating virtual memory pages with physical pages. That is, each virtual page is assigned, or pointed, to a particular physical page. It is in physical memory that programs and data

are stored. The virtual memory pages are like windows which allow viewing or accessing items in physical memory. In a technical sense, the virtual address extent is said to be mapped onto the physical address extent. A mapping example of a 128K computer is shown in Fig. 3-3. The memory mapping for a particular program may be static or changed as the program proceeds.

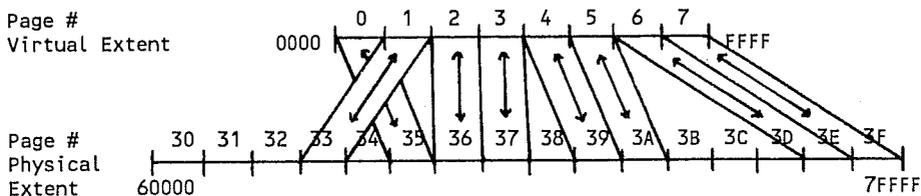


Fig. 3-3. Memory Mapping

### Memory Management Unit

The memory management unit (MMU) is a programmable hardware device that assigns virtual pages to physical pages. Up to eight virtual pages (a 64K extent) may be assigned to a maximum of eight physical pages (a 64K extent) at any one time.

The MMU is composed of two sets of eight **page address registers** (PARs) each and its control logic. Each PAR is six bits (5-0) long. Each of the eight PARs (numbered 0-7) of either set always corresponds with the eight virtual pages (0-7). The two sets of PARs, the **executive** set and the **task** set, and their dedicated addresses are shown in Table 3-1. A virtual page is mapped to a physical page by writing the physical page number into the dedicated address of that virtual page's corresponding PAR. For example, virtual page 5 is mapped to physical page 2C by storing a 2C in FFAD (using the task set).

Executive Set		Task Set	
PAR #	Ded. Addr.	PAR #	Ded. Addr.
0	FFA0	0	FFA8
1	FFA1	1	FFA9
2	FFA2	2	FFAA
3	FFA3	3	FFAB
4	FFA4	4	FFAC
5	FFA5	5	FFAD
6	FFA6	6	FFAE
7	FFA7	7	FFAF

Table 3-1. PAR Sets

The MMU converts a virtual address to a physical address by generating a 19-bit address from the 16-bit virtual address and the content of a PAR. The three MSBs of the virtual address select one of the eight

PARs whose content forms the six MSBs of the physical address. The lower thirteen bits of the physical address are the same as those of the virtual address. Fig. 3-4 shows an MMU block diagram and the conversion process.

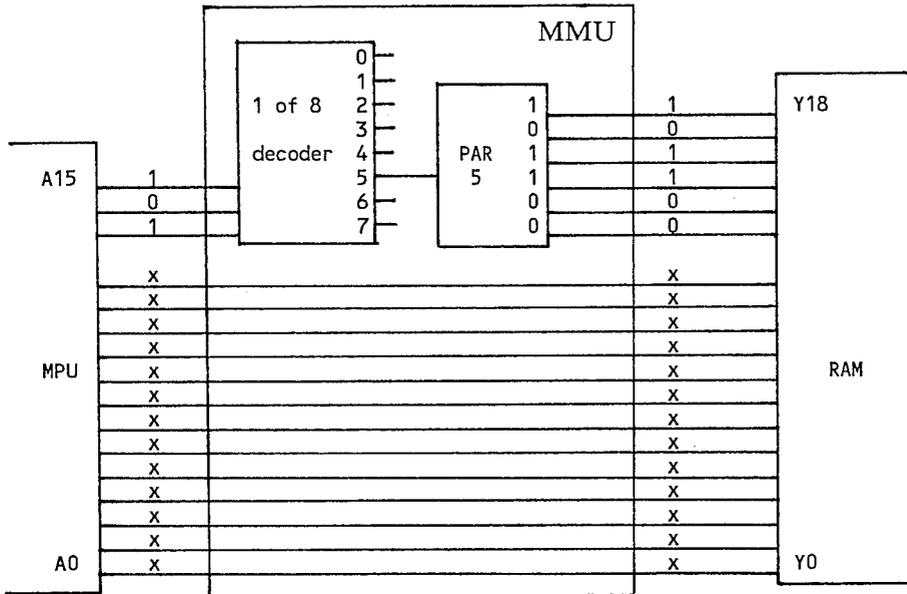


Fig. 3-4. MMU Operation

MMU operation is controlled by two bits in the ACVC control registers. Bit 0 of FF91 selects either the executive (bit clear) or task (bit set) set of PARs for use in accessing physical RAM. Normally a large program or a system of programs has a controller segment which calls the other program segments as required. The CoCo 3 is designed for the controller segment to always be mapped by the executive PAR set. The controller, before it is to call another program segment, will map that segment with the task PAR set. Then it will switch to the task PAR set and call the segment. This arrangement also calls for a small part of the task virtual memory extent to always map part (the interface) of the controller. It is through the interface that MPU control is either passed to the called segment or returned to the controller. Typically the kernel of an operating system is the controller just described. The application or task programs (utilities and user programs) are the called segments that run under the task PAR set.

Bit 6 of FF90 enables (bit set) or disables (bit clear) the operation of the MMU. When enabled, the MMU operates as already described. When disabled, The MMU always sets the expanded address bits Y18-Y16. This causes the 64K virtual extent to always be mapped onto the uppermost 64K

of physical memory.

The exception to the previously described MMU operation is when the virtual address is in the range FF00-FFFF. These (the dedicated addresses) are not expanded and sent to RAM but are always routed to their appropriate device and/or control register. Therefore, there are 8K minus 256 RAM addresses in virtual page 7.

The 128K CoCo 3 has a physical memory addressing idiosyncrasy. The content of physical memory is present in all the groups of physical pages. It is present in its normal physical page group of 30-3F, and the other page groups of 00-0F, 10-1F, and 20-2F. This is so because the two MSBs of the expanded address (Y18 and Y17) are not used.

### ROM Memory Mapping

Physical memory may be selected as all RAM or part RAM and ROM. The all-RAM mode is selected by setting the SAMTY bit. It is set by writing anything into dedicated address FFDF.

The ROM/RAM mode is selected by clearing the SAMTY bit. It is cleared by writing anything into dedicated address FFDE. In the ROM/RAM mode the physical memory of pages 3C-3F are not RAM but ROM. The specific ROM assigned to these pages is controlled by bits 0 and 1 of dedicated address FF90. The three possible ROM configurations are shown in Fig. 3-5.

Physical Page #	3C	3D	3E	3F	Bit 1	Bit 0
	Ext BASIC	BASIC	Cart. ROM		0	x
	Ext BASIC	BASIC	Reset Init	Sup Ext BASIC	1	0
	Cartridge ROM				1	1

Fig. 3-5. Physical ROM Configurations

## Chapter 4

# HIGH RESOLUTION DISPLAYS

This chapter describes the new high resolution displays and how to control and use them. The new displays are categorized as those that display text or graphics. The text display shows text in 32, 40, 64, or 80 columns where each character's attributes (such as color, underline, and blink) are individually specified. The graphics displays include several modes of different resolution where the highest resolution is 1280H x 225V. A disadvantage is most of the high resolution displays extend slightly past the left edge of a TV screen.

The new displays are controlled by the ACVC control registers, of which the original CoCo had none. In the CoCo 3 the ACVC includes all the functions necessary to create a video display from the content of a buffer area. A buffer area is a continuous section of physical memory in which are put codes that specify characters and their attributes when in a text display mode or that control the colors of pixels (picture elements) when in a graphics display mode. The ACVC control registers that control the high resolution displays and their dedicated addresses are shown in Table 4-1.

ACVC Register	Ded. Address
Init0	FF90
Video Mode	FF98
Video Resolution	FF99
Border Color	FF9A
Vertical Scroll	FF9C
Vertical Offset1	FF9D
Vertical Offset0	FF9E
Horizontal Offset	FF9F

Table 4-1. ACVC Hi-Res Control Registers

## ACVC OPERATION

The ACVC continuously reads a given buffer area in physical memory and generates a video signal. The buffer is completely read sixty times a second, the rate at which the monitor screen image is redrawn or refreshed. Since the ACVC directly accesses physical memory, no virtual memory pages need point to the buffer are if the MPU is not about the change the buffer content.

The memory area of a buffer is best thought of as a rectangular grid whose horizontal and vertical sizes correspond with the display mode in use. For example, when displaying 40-column by 24-row text, each character is specified by two bytes. This buffer is composed of 1920 (2 x 40 x 24) memory locations organized as an 80 x 24 grid. The upper left corner of this buffer, which starts at address A000, is shown in Figure 4-1. In this buffer the locations of pairs of cells correspond to character positions on the monitor screen. Thus, the character specified by the contents of A000 and A001 is positioned in the upper left corner of the screen, and the character of A050 and A051 is positioned just below that. All buffers start at the upper left of the screen and end at the lower right.

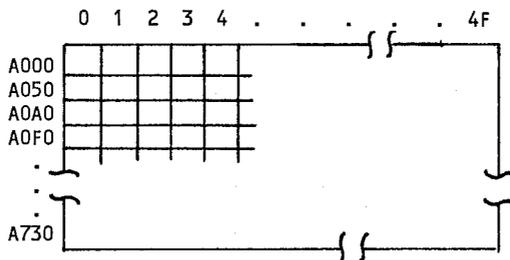


Fig. 4-1. Text Buffer Grid

## Init0 Register

Bit 7 of this register controls the ACVC general display mode. When set the display modes are those (low resolution) of the original CoCo. When clear, the new high resolution display modes are available.

## Video Mode Register

The bits of this register are shown in Table 4-2. Bits 7 and 2-0 control only the high resolution displays and just they are described here.

Bit 7 - BP	Bit 3 - H50
Bit 6 - Not Used	Bit 2 - LPR2
Bit 5 - BPI	Bit 1 - LPR1
Bit 4 - MOCH	Bit 0 - LPRO

Table 4-2. Video Mode Register

- BP - When clear this bit selects a high resolution text display mode. When set a high resolution graphics display mode is selected.
- LPR2-0 - In these bits is put a three bit code which selects the vertical size of a pixel or character. Only certain codes are valid for particular display modes.

### Video Resolution Register

The video resolution register controls the horizontal and vertical resolution and how many colors are available. That is, its content selects a particular display mode from the text or graphics categories. Its bit assignments are shown in Table 4-3.

Bit 7 - Not Used	Bit 3 - HRES1
Bit 6 - VRES1	Bit 2 - HRES0
BIT 5 - VRES0	Bit 1 - CRES1
BIT 4 - HRES2	Bit 0 - CRES0

Table 4-3. Video Resolution Register

- VRES1-0 - These specify the vertical resolution and the height of the screen image and buffer. The vertical resolution is the number of rows that make up the display image within the border. In a text mode, VRES1-0 control the number of text lines. In a graphics mode they control the number of pixel rows. The vertical resolution codes and their results are shown in Table 4-4.

VRES1	VRES0	Graphic Rows	Text Rows	Image Height
0	0	192	24	Normal
0	1	200	25	Larger
1	0	N o t U s e d		
1	1	225	28	Full Screen

Table 4-4. Vertical Resolutions

- HRES2-0 and CRES1-0 - Together these specify the particular text or graphics mode by selecting the horizontal resolution and number of available colors. The text and graphics sections of this chapter describe codes to put in these bits to select the desired display mode.

### Border Register

This is a 6-bit register in which is put a color code that determines the border color. The color code may be any of the available sixtyfour codes. The border color is specified directly with a color code, not indirectly via a palette register.

## Vertical Offset Registers 0 & 1

The starting address of a buffer area is indicated to the ACVC via these registers. This is done by writing the upper sixteen bits (Y18-Y3) of the starting physical address into these registers. Y18-Y11 are written to vertical offset 1 and Y10-Y3 are written to vertical offset 0. Thus, a buffer is limited to starting on an 8-byte boundary, or the starting address is limited to the binary form: xxx xxxx xxxx xxxx x000.

## Vertical Scroll Register

This register is only effective in the text modes and provides the smooth scroll capability. A smooth scroll is when the text appears to move up smoothly instead of jumping up one line at a time. The bit assignments of this register are shown in Table 4-5.

Bit 7 - Not used	Bit 3 - SCEN
Bit 6 - Not Used	Bit 2 - SC2
Bit 5 - Not Used	Bit 1 - SC1
Bit 4 - Not Used	Bit 0 - SC0

Table 4-5. Vertical Scroll Register

- SCEN - Vertical scrolling is enabled when this bit is clear and disabled when set.
- SC2-0 - In these bits is put a vertical scroll value of 0-7. Each incrementally higher value will scroll the text up one eighth of a line. In doing so, new buffer area is exposed at the bottom of the screen. This should have been previously loaded with the new line of text to appear at the screen bottom. When the scroll value reaches seven, the smooth scroll process is completed by setting the scroll value to zero and incrementing the buffer starting address (vertical offset 1 and 0) such that the buffer starts at the new top line of text.

## Horizontal Offset Register

This register provides the capability of horizontally shifting the screen image. This capability is useful when the text image is wider than the screen image; the screen image can be shifted horizontally to view the desired text. In graphics modes a wide landscape can exist in the buffer over which the screen can be shifted left or right. The bit assignments of this register are shown in Table 4-6.

Bit 7 - HE	Bit 3 - X3
Bit 6 - X6	Bit 2 - X2
Bit 5 - X5	Bit 1 - X1
Bit 4 - X4	Bit 0 - X0

Table 4-6. Horizontal Offset Register

- HE - When this bit is clear the screen buffer horizontal size corresponds with the selected display mode. When this bit is set the screen buffer width is 256 bytes and the buffer must start at a 256 byte boundary. In a text display mode this provides a width of 128 columns since each character is specified by two bytes. In a graphics display mode this provides a buffer whose horizontal size is larger than expected for the selected display mode.
- x6-0 - In these bits is put a shift value which moves the screen image to the left. Larger values cause a larger shift than small values. Zero causes no shift. Each increment of the shift value causes the displayed area of the buffer to move two bytes to the right. In a text mode this causes the image to shift left one column. In a graphics mode this causes the image to shift left by 16, 8, or 4 pixels depending on the number of colors available, either 2, 4, or 16, respectively.

## TEXT DISPLAY

This section describes how to use and select the text displays. This information includes the displayable character set, their video codes, and how to set up the ACVC control registers.

### Character Selection

The content of a text screen buffer is composed of pairs (bytes 0 and 1) of bytes. In each pair is put the codes to select the desired character and its attributes. The screen buffer always starts at an even address and the first buffer byte is the first byte (byte 0) of a pair. Thus, byte 0 is always at an even address and byte 1 is always at an odd address.

A character is displayed by putting its video display code in byte 0 and its attribute code in byte 1. The video display code ranges from 0 to 7F where each code corresponds with one of the 128 displayable characters. The displayable characters include the alphabet, numbers, punctuation, and a Roman extension (unique European characters). The displayable characters and their video display codes are shown in Appendix A. Notice that the video codes 32-127 are the same as the ASCII codes of those characters. The attribute code specifies other characteristics such as color, blink, and underline. The bit assignments of the attribute byte are shown in Table 4-7.

Bit 7 - Blink	Bit 3 - FRC0
Bit 6 - Undln	Bit 2 - BKC2
Bit 5 - FRC2	Bit 1 - BKC1
Bit 4 - FRC1	Bit 0 - BKC0

Table 4-7. Attribute Byte

- **Blink** - When set the displayed character blinks; otherwise it doesn't blink. The blink rate depends on the content of the Timer 0 and 1 registers and bit 5 of the Init 1 register. They are described in Chapter Six.
- **Undln** - When set the character is underlined; otherwise it is not.
- **FRC2-0** - The foreground color bits select one of the eight palette registers 8-15. An FRC code of 0 selects palette register 8, 1 selects 9, etc. The character color is that specified by the color code in the selected palette register.
- **BKC2-0** - The background color bits select one of the eight palette registers 0-7. A BKC code of 0 selects palette register 0, 1 selects 1, etc. The character background color is that specified by the color code in the selected palette register.

### Display Formats

The basic display formats are of 32, 40, 64, and 80 columns. All four formats also have other selectable attributes such as vertical character size, number of rows, border color, and the expanded horizontal size. Changing some of the attributes will change the screen buffer size and dimensions.

The 32 and 40 column formats display characters whose size is approximately that of the original CoCo text characters. This size is very ledgeable on a TV monitor. The 64 and 80 column formats display characters of a smaller size. They are marginally ledgeable on a TV monitor and quite readable on a high resolution RGB or composite monitor.

The border color is specified independently of the text foreground and background colors by writing a color code into the border register. With the 32 and 64 column formats the image is centered with the border extending completely around it. With the 40 and 80 column formats the image extends the full width of the screen and the border exists at only the top and bottom of the screen.

The number of lines of displayed text may be 24, 25, or 28. Normally a program uses the 24 lines for the work area. With 25 lines the twentyfifth line usually serves to display the program's status or for the user to enter commands. With 28 lines the four extra lines may serve to display status, for user command entry, or to provide a small help screen. Changing the number of displayed lines changes the vertical size of the buffer. The number of lines is controlled with the VRES 1 and 0 bits.

The height of characters can be changed slightly. This results in reducing the number of displayable lines. The height is controlled by the LPR2-0 bits. The usable values and their effect on a 24 line display are shown in Table 4-7.

LPR2	LPR1	LPR0	Lines of Text
0	1	1	24
1	0	0	21
1	0	1	19
1	1	0	17.5

Table 4-7. Character Heights

In the horizontally expanded mode (HE=1) the buffer width is 256 bytes and the buffer must start at a 256 byte boundary. This allows text lines of up to 128 characters. The desired portion of text is displayed by horizontal scrolling via the horizontal offset register.

BP=0 and Init0 bit 7=0

32 COLUMN

---

Rows	24	25	28
VRES1-0	00	01	11
HRES2-0	0x0	0x0	0x0
CRES1-0	x1	x1	x1
Buffer Size (HE=0)	64H x 24V	64H x 25V	64H x 28V
Buffer Size (HE=1)	256H x 24V	256H x 25V	256H x 28V

40 COLUMN

---

Rows	24	25	28
VRES1-0	00	01	11
HRES2-0	0x1	0x1	0x1
CRES1-0	x1	x1	x1
Buffer Size (HE=0)	80H x 24V	80H x 25V	80H x 28V
Buffer Size (HE=1)	256H x 24V	256H x 25V	256H x 28V

64 COLUMN

---

Rows	24	25	28
VRES1-0	00	01	11
HRES2-0	1x0	1x0	1x0
CRES1-0	x1	x1	x1
Buffer Size (HE=0)	128H x 24V	128H x 25V	128H x 28V
Buffer Size (HE=1)	256H x 24V	256H x 25V	256H x 28V

80 COLUMN

---

Rows	24	25	28
VRES1-0	00	01	11
HRES2-0	1x1	1x1	1x1
CRES1-0	x1	x1	x1
Buffer Size (HE=0)	160H x 24V	160H x 25V	160H x 28V
Buffer Size (HE=1)	256H x 24V	256H x 25V	256H x 28V

Table 4-8. Hi-Res Text Mode Selection

## Display Selection

Table 4-8 shows how to select the various display formats by loading the ACVC control registers with the appropriate values.

## Demonstration Program

The program in Listing 4-1 demonstrates vertical smooth scrolling and horizontal scrolling. A large buffer area is established and cleared, over which the screen will be moved by the left joystick. The buffer area dimensions are 256H x 32V. This provides for 32 lines of text where each may be up to 128 characters long. Then the ACVC is pointed to this area and the 40-column text mode with horizontal expansion is selected. The last part of the program reads the left joystick position. Its forward/back position is used to scroll the text up or down and its left/right position is used to scroll left or right.

Addr	Instr		Source code	
0AF8	1A 50	100	TEXTA	ORCC #50 LOCKOUT INTERRUPT
0AFA	8E 4000	110		LDX #SCRBUF ADDR OF DISPL AREA
0AFD	CC 200C	120		LDD #200C SPACE + ATTR BYTES
0B00	ED 81	130	TEXTA1	STD ,X++ CLEAR DISPLAY AREA
0B02	8C 8000	140		CMPX #SCRBUF+32*256 DONE ?
0B05	23 F9	150		BLS TEXTA1 NO - JUMP BACK
0B07	8E 5000	160		LDX #SCRBUF+8*256 TEXT SCR POS
0B0A	CE 0B58	170		LDU #TEXTB ADDR OF TEXT STRING
0B0D	A6 C0	180	TEXTA2	LDA ,U+ MOVE STRING AND ATTR
0B0F	ED 81	190		STD ,X++ TO BUFFER AREA
0B11	1183 0BB1	200		CMPU #TEXTC DONE ?
0B15	25 F6	210		BLO TEXTA2 NO - JUMP
0B17	CC E800	220		LDD #E800 PHYSICAL BUFF ADDRESS
0B1A	FD FF9D	230		STD \$FF9D PUT IN VERTOFFSET1+0
0B1D	FD 0B56	240		STD VERPOS AND SAVE IT
0B20	7F FF9C	250		CLR \$FF9C CLEAR VERT SCROLL
0B23	86 80	260		LDA #80 SET HE=1 AND
0B25	B7 FF9F	270		STA \$FF9F HORZ OFFSET=0
0B28	86 4E	280		LDA #4E SELECT
0B2A	B7 FF90	290		STA \$FF90 HIGH RES DISPLAY
0B2D	CC 0305	300		LDD #0305 40 COLUMN
0B30	FD FF98	310		STD \$FF98 TEXT MODE
0B33	AD9F A00A	320	TEXTA3	JSR [\$A00A] READ JOYSTICK POS
0B37	B6 015D	330		LDA \$15D GET VERT POSITION
0B3A	84 07	340		ANDA #07 JUST LOWER 3 BITS
0B3C	B7 FF9C	350		STA \$FF9C PUT IN VERT SCROLL
0B3F	B6 015D	360		LDA \$15D GET VERT POSITION
0B42	84 38	370		ANDA #38 JUST UPPER 3 BITS
0B44	48	380		LSLA ADJUST TO A 256
0B45	48	390		LSLA BYTE INCREMENT
0B46	BB 0B57	400		ADDA VERPOS+1 ADD TO STRT ADDR
0B49	B7 FF9E	410		STA \$FF9E TO VERT OFFSET 0
0B4C	B6 015C	420		LDA \$15C GET HORIZON POSIT
0B4F	8A 80	430		ORA #80 SET HE BIT
0B51	B7 FF9F	440		STA \$FF9F TO HORIZ OFFSET
0B54	20 DD	450		BRA TEXTA3 DO AGAIN
		460	*****	
0B56	470	VERPOS	RMB 2	BUF ADDR SAVE AREA
0B58	480	TEXTB	FCC	/THIS TEST MESSAGE IS LONGER

```

THAN A 40 COLUMN DISPLAY TO DEMONSTRATE HORIZONTAL SCROLLING./
OBB1      490 TEXTC  RMB  1      END OF TEXT
          4000      500 SCRBUF EQU  $4000    START ADDR OF BUFFER
          510      END

```

### Listing 4-1. Hi-Res Text Demonstration

## GRAPHICS DISPLAY

This section describes the operation and use of the high resolution graphics displays. All twenty-two graphics display modes, of which BASIC uses just four, are equally and completely described.

### Pixel Control

In all graphics modes the monitor screen is divided into pixels by an imaginary grid. The horizontal and vertical resolutions specify the number of pixels in a row and column, respectively. The higher resolutions cause the grid lines to be closer together and there to be more smaller pixels over the screen extent. A particular display mode causes the screen to be divided into a fixed number of rows and columns of pixels. Associated with a display is its buffer which is organized such that there is a correspondence between the position of a group of bits and the pixel it controls. A graphics display buffer organization is similar to that of a text display. The first byte of the buffer controls pixels at the upper left corner of the screen. The following bytes control pixels to the right in that row. The last buffer byte controls the pixels at the lower right corner of the screen.

A pixel is controlled by specifying its color. Its color is specified by its controlling bits selecting a palette register. Each of the twenty-two display modes use one of three pixel control schemes. The scheme in use is determined by the CRES1-0 bits. Three of their combinations, shown in Fig. 4-2, determine how each byte in a buffer is interpreted for pixel control.

CRES1	CRES0	Format	Byte Content
0	0	A	P <sub>0</sub>
0	1	B	P <sub>1</sub> P <sub>0</sub> P <sub>1</sub> P <sub>0</sub> P <sub>1</sub> P <sub>0</sub> P <sub>1</sub> P <sub>0</sub>
1	0	C	P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub> P <sub>3</sub> P <sub>2</sub> P <sub>1</sub> P <sub>0</sub>

Fig. 4-2. Buffer Byte Formats

The buffer bytes of display modes where CRES1-0 equal 00 are of format A. Each bit of a byte controls one pixel of a horizontal group of eight pixels. For example, bit 7 controls the leftmost pixel of a group, bit 6 controls the next pixel to the right, etc. A complete set of pixel groups forms one horizontal row. Therefore, the buffer width is one eighth

the horizontal resolution. The state (0 or 1) of a bit controls the pixel color by selecting a palette register (either 0 or 1, respectively). Thus, any two colors of the sixty-three available may be used at one time.

The buffer bytes of display modes where  $CRES_{1-0}$  equal 01 are of format B. Two bits of each byte control one pixel of a horizontal group of four pixels. For example, bits 7-6 control the leftmost pixel of a group, bits 5-4 control the next pixel to the right, etc. A complete set of pixel groups forms one row. Therefore, the buffer width is one-fourth the horizontal resolution. A pixel color is selected by its controlling pair of bits having a value (0-3) which selects a palette register (0-3, respectively). Thus, up to four colors may be displayed at one time.

The buffer bytes of display modes where  $CRES_{1-0}$  equal 10 are of format C. Four bits of each byte control one pixel of a horizontal group of two pixels. For example, bits 7-4 control the left pixel of a group and bits 3-0 control the right pixel. A complete set of pixel groups forms one row. Therefore, the buffer width is one-half the horizontal resolution. A pixel color is selected by its controlling bits having a value (0-15) which selects a palette register (0-15). Thus, up to sixteen colors may be displayed at one time.

### Display Control

The graphics display modes are enabled by setting the BP bit of the video mode register and clearing bit 7 of the Init0 register. The ACVC is directed to the desired buffer area by putting its physical address in vertical offset registers 1 and 0. The HRES and CRES bits control the horizontal resolution and number of colors that may be displayed. The HRES bits specifically select the buffer width and the CRES bits select the format of the buffer bytes. The HRES0 bit also controls the horizontal size of the working screen area. When clear the working screen area is centered on the screen and it is bordered on the left and right. When set the displayed area extends the full width of the screen. The border color is controlled as in the text modes, via the border register.

The VRES bits select the vertical resolution and the working screen area height as shown in Table 4-4. The LPR bits select the vertical size of the pixels by specifying the number of rows a pixel occupies. Normally the LPR bits are clear. Their values and effects are shown in Table 4-10. An LPR value of seven causes the top row of pixels to be displayed over the full screen height.

Some programmers prefer the screen to be blank while it is being updated or redrawn. The screen is blanked by setting both  $CRES_1$  and  $CRES_0$ . After the program has manipulated the buffer content,  $CRES_{1-0}$  are set to their original value to allow viewing of the new image.

LPR2	LPR1	LPR0	Rows
0	0	0	1
0	0	1	1
0	1	0	2
0	1	1	7
1	0	0	8
1	0	1	9
1	1	0	10
1	1	1	all

Table 4-9. Pixel Heights

Vertical scrolling is not performed with the vertical scroll register but with the vertical offset 1 and 0 registers. The screen image is moved up one pixel row by incrementing the physical buffer address (in vertical offset registers 1 and 0) by the buffer's width. Downward motion by one row is done by decrementing the buffer address by its width. While moving the image, new buffer area will be exposed to view; it should have been previously cleared or loaded with the desired image.

The available horizontal scroll capability is more coarse than the vertical scroll. The image is moved to the left by increasing the value in bits x6-0 of the horizontal offset register. The image is moved 4, 8, or 16 pixels (depending on CRES) for each increment or decrement of the horizontal offset register. That is, the displayed buffer area is moved left or right two bytes for each horizontal offset increment or decrement. When using a normal buffer (HE=0) the image will appear on one side of the screen as it is scrolled off the other (wrap-around). When using a horizontally expanded buffer (HE=1) the buffer is 256 byte wide. This provides a panorama over which the screen may be scrolled.

### Display Selection

Table 4-10 shows how to select the various horizontal display formats and other pertinent information. The buffer widths are for HE=0. When HE=1 their widths are 256 bytes.

BP=1 and Init0 bit 7=0

HRES = 0      Buffer Width = 16      Screen Width = Short

---

CRES1-0	Horizontal Resolution	Palette Regs.	Byte Format
0	128	0-1	A
1	64	0-3	B
2	Not Used		
3	Blank		

HRES = 1      Buffer Width = 20      Screen Width = Full

---

CRES1-0	Horizontal Resolution	Palette Regs.	Byte Format
0	160	0-1	A
1	80	0-3	B
2	Not Used		
3	Blank		

HRES = 2      Buffer Width = 32      Screen Width = Short

---

CRES1-0	Horizontal Resolution	Palette Regs.	Byte Format
0	256	0-1	A
1	128	0-3	B
2	64	0-15	C
3	Blank		

HRES = 3      Buffer Width = 40      Screen Width = Full

---

CRES1-0	Horizontal Resolution	Palette Regs.	Byte Format
0	320	0-1	A
1	160	0-3	B
2	80	0-15	C
3	Blank		

HRES = 4      Buffer Width = 64      Screen Width = Short

---

CRES1-0	Horizontal Resolution	Palette Regs.	Byte Format
0	512	0-1	A
1	256	0-3	B
2	128	0-15	C
3	Blank		

Please attach the 'Change Page' to page 34 of the Addendum. It corrects an error about the available Hi-Res graphics horizontal resolutions. Specifically, the 1024 and 1280 horizontal resolutions are not available on the CoCo 3. This error occurred because the monitor in use when determining all the Hi-Res graphics modes was not able to show these highest resolutions and so it was assumed they existed. Subsequent checking (after printing of this book) of the video and RGB signals with an oscilloscope showed they were not available. We apologize and are embarrassed because we try to be the best source of assembly language information for the CoCo.

'Change Page'

HRES = 5      Buffer Width = 80      Screen Width = Full

---

CRES1-0	Horizontal Resolution	Palette Regs.	Byte Format	
0	640	0-1	A	HSCREEN 3
1	320	0-3	B	HSCREEN 1
2	160	0-15	C	
3	Blank	-	-	

HRES = 6      Buffer Width = 128      Screen Width = Short

---

CRES1-0	Horizontal Resolution	Palette Regs.	Byte Format	
0	64	0-1	X	
1	512	0-3	B	
2	256	0-15	C	
3	Blank	-	-	

HRES = 7      Buffer Width = 160      Screen Width = Full

---

CRES1-0	Horizontal Resolution	Palette Regs.	Byte Format	
0	80	0-1	X	
1	640	0-3	B	HSCREEN 4
2	320	0-15	C	HSCREEN 2
3	Blank	-	-	

Table 4-10. Horizontal Format Selections

Byte Format X - Each pixel is controlled by two consecutive bytes. When bit 7 of the first byte (at even address) is clear the pixel is the color of palette register 0. When set the pixel is the color of palette register 1. The other 15 bits (6-0 at even addresses and 7-0 at odd addresses) are unused.

## Chapter 5

# LOW RESOLUTION DISPLAYS

This chapter provides an overview of the low resolution displays and describes any differences from their operation in the original CoCo. The low resolution displays are most of those available in the original CoCo. The operation and use of these displays are completely described in **The Book**. The low resolution display modes available in the CoCo 3 are text, SG4, and all the graphics modes (G1C - G6R). In the original CoCo the displays were generated and controlled by the synchronous address multiplexer (SAM) and the video display generator (VDG). In the CoCo 3 these functions are embedded in the ACVC.

### ACVC OPERATION

This section describes parts of the ACVC which control the generation of the low resolution displays. These include the SAM, VDG, and other control registers. Table 5-1 lists these and their dedicated addresses.

REGISTER	DED. ADDR.
Init0	FF90
VDG	FF22
Vert Offset 1,0	FF9D, FF9E
SAM Vert Offset	FFC6-FFD3
SAM Mode	FFC0-FFC5
Video Resolution	FF99
Horizontal Offset	FF9F

Table 5-1. Lo-Res Control Registers

#### Init0

Bit 7 controls the ACVC general display mode. When set the low resolution displays are available. When clear the high resolution displays

are available.

## V DG

The V DG operating mode is controlled by the upper six bits of dedicated address FF22. The value put in these bits selects one of the low resolution display modes.

## SAM Mode

The SAM mode register is of three bits (v2-v0) which select the size of the buffer area. Each bit is set or cleared by writing anything to its appropriate dedicated address as shown in Table 5-2.

Bit	Set	Clear
V2	FFC5	FFC4
V1	FFC3	FFC2
V0	FFC1	FFC0

Table 5-2. SAM Mode register

## Vertical Offset

The physical address of the display buffer is put in vertical offset registers 1 and 0 and the SAM vertical offset register. The SAM vertical offset register is of seven bits (F6-F0) that are set or cleared by writing anything to their appropriate dedicated address as shown in Table 5-3. The buffer starting address is apportioned to the vertical offset registers as shown in Fig. 5-1.

Bit	Set	Clear
F6	FFD3	FFD2
F5	FFD1	FFD0
F4	FFCF	FFCE
F3	FFCD	FFCC
F2	FFCB	FFCA
F1	FFC9	FFC8
F0	FFC7	FFC6

Table 5-3. SAM Vertical Offset Register

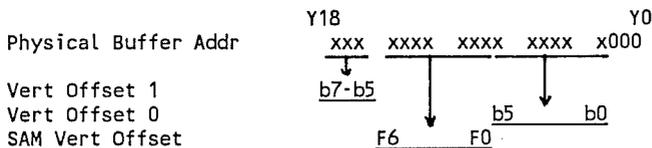


Fig. 5-1. Buffer Address Apportionment

## Video Resolution Register

The bits available for use are VRES1-0 and HRES2. They are normally clear. In the text mode there are thirty-two columns when HRES2 is clear and sixty-four when set. VRES1-0 control the vertical resolution. The valid values that may be put in VRES1-0 and the number of text or graphics rows they select are shown in Table 5-4.

VRES1	VRES0	Text	G1x/G2C	G2R/G3x	G6x
0	0	16	64	96	192
0	1	16.75	67	100	200
1	1	19	75	112	225

Table 5-4. Vertical Resolution

## Horizontal Offset Register

It operates similarly as described in Chapter Four. Its disadvantage in the low resolution display modes is the image scrolls up one row when shifted and HE=0.

## TEXT DISPLAY

The text display provides thirty-two and sixty-four (HRES2=1) column text and semigraphics 4. This section describes CoCo 3 -unique attributes of the text displays. More detailed information is provided in **The Book**.

## Character Selection

The displayable characters are the alphabet, numerals, and punctuation marks. They are displayed by putting their video display codes (Appendix C, **The Book**) in the buffer. The semigraphics shapes, which occupy one character position, are displayed by putting their video display codes in the buffer.

The semigraphic shapes are of a selectable color on a black background. Bit seven of their video display code is always set. Bits 3-0 select one of sixteen shapes (see page 292 of **Color Computer 3 Extended BASIC Manual**). In bits 6-4 is put a value (0-7) which selects a palette register (0-7) containing the desired color code.

## Display Selection

The text display is selected by putting the correct values into the VDG and SAM mode registers as shown in Table 5-5. The text VDG codes (bits 7-3 of FF22) and their effects are shown in Fig. 5-2.

VDG Bit	State	Result
7	0	Text Display Mode
6	0	Black Border
	1	Border is green (bit 3=0) or orange (bit 3=1)
5	0	Palette regs used per bit 3
	1	Fore/back pal regs reversed & border = black
4	0	Lower case displayed with reverse video
	1	True lower case displayed
3	0	Char. color= pal reg 12, backgnd = pal reg 13
	1	Char. color= pal reg 14, backgnd = pal reg 15

Fig 5-2. VDG Text Control

## GRAPHICS DISPLAY

The graphics displays available are G1C, G1R, G2C, G2R, G3C, G3R, G6C, and G6R. They are selected by putting the values from Table 5-5 into the VDG and SAM mode registers. These display modes are described in **The Book**. The difference is that the bits of a buffer byte that control a pixel now do so by selecting a palette register. The color selection bit (c) determines the border color and which palette registers may be selected. When c is clear the border is green and when set the border is buff.

In the graphics display modes with resolution (GxR) each pixel is controlled by one bit. Its value (0-1) selects one of two palette registers. When c is clear the selectable palette registers are 8 and 9. When set they are 10 and 11.

In the graphics display modes with color (GxC) each pixel is controlled by two bits. Its value (0-3) selects one of four palette registers. When c is clear the selectable palette registers are 0-3. When set they are 4-7.

Display Mode	VDG - FF22	SAM Mode		
	7 6 5 4 3	V2	V1	V0
Text + SG4	0 x x x x	0	0	0
G1C	1 0 0 0 c	0	0	1
G1R	1 0 0 1 c	0	0	1
G2C	1 0 1 0 c	0	1	0
G2R	1 0 1 1 c	0	1	1
G3C	1 1 0 0 c	1	0	0
G3R	1 1 0 1 c	1	0	1
G6C	1 1 1 0 c	1	1	0
G6R	1 1 1 1 c	1	1	0

Table 5-5. Low resolution Display Selection

## Chapter 6

# INTERRUPTS

This chapter describes the new interrupt sources and how to use and control them. The new interrupt sources generate IRQ or FIRQ interrupts via the ACVC. The interrupts of the original CoCo and their sources are described in **The Book**.

### MPU INTERRUPT SEQUENCES

An interrupt may be considered as an external (to the MPU) event that causes the MPU to suspend its current processing and directs the MPU to process some other program (an interrupt handler). The interrupt handler performs its task in response to the interrupt and finally returns. Returning sends the MPU to the point of interruption in the suspended program so it may resume processing. The four types of interrupts (IRQ, FIRQ, NMI, and SWI) and how the MPU responds to them are described in detail in **The Book**.

### Interrupt Sequences

This section provides an overview of the sequences through which an MPU goes in response to an interrupt. Also, any characteristics unique to the CoCo 3 are described.

The IRQ, NMI, and SWI types of interrupts cause the MPU to set the E bit of the CC register and to push all its registers, except s, onto the s stack. Thus, the registers' contents at the time of interruption are saved. Then the PC register is loaded from the appropriate pair of dedicated addresses (see Table 6-1). This directs, or vectors, the MPU to the appropriate interrupt handler. The FIRQ interrupt sequence is similar to the others, but E is cleared and only the CC and PC registers are pushed onto the s stack.

The table of interrupt vectors exists in ROM at addresses BFF2-BFFF. This table also appears at addresses FFF2-FFFF, where the MPU expects it to be. Table 6-1 shows the interrupt vector table and its content. Notice that this table is also used by the Reset sequence.

Interrupt	Address	Content
RESET	FFFE-FFFF	8C1B
NMI	FFFC-FFFD	FEFD
SWI	FFFA-FFFB	FEFA
IRQ	FFF8-FFF9	FEF7
FIRQ	FFF6-FFF7	FEF4
SWI2	FFF4-FFF5	FEF1
SWI3	FFF2-FFF3	FEEE

Table 6-1. Interrupt Vector Table

### Jump Tables

The interrupt vectors point the MPU to the **primary jump table** which is in RAM. This table, in Table 6-2, is composed of long branch (LBRA) instructions. Because this table is in RAM it can be modified so that a particular type of interrupt will cause the MPU to branch to your interrupt handler. The primary jump table in Table 6-2 is shown as set up by the Reset Initialization program (described in Chapter Seven). Because an LBRA instruction contains the value (limited to +/- 32K) added to the PC register content to form the destination address, it may be easier to modify this table by replacing the LBRAs with Jumps. Then the operands would simply be the destination addresses.

Address	Content	Purpose
FEED	LBRA \$100	Branch to SWI3 intrpt handler
FEF1	LBRA \$103	Branch to SWI2 intrpt handler
FEF4	LBRA \$10F	Branch to FIRQ intrpt handler
FEF7	LBRA \$10C	Branch to IRQ intrpt handler
FEFA	LBRA \$106	Branch to SWI intrpt handler
FEFD	LBRA \$109	Branch to NMI intrpt handler

Table 6-2. Primary Jump Table

Notice that the LBRAs of Table 6-2 branch to another series of addresses (\$100-\$10F). This is the **secondary jump table** which exists to provide compatibility with the original CoCo. The secondary jump table is shown in Table 6-3 and is the same as the original CoCo's only interrupt jump table. This table is composed of jump instructions in RAM which may be modified to point to your interrupt handlers. Table 6-3 is shown as set up by Extended Color BASIC 2.0.

Address	Content	Purpose
0100	-	Jump to SWI3 intrpt handler
0103	-	Jump to SWI2 intrpt handler
0106	-	Jump to SWI intrpt handler
0109	-	Jump to NMI intrpt handler
010C	JMP \$894C	Jump to IRQ intrpt handler
010F	JMP \$A0F6	Jump to FIRQ intrpt handler

Table 6-3. Secondary Jump Table

## ACVC INTERRUPT GENERATION

The new interrupt sources are connected to the ACVC which may be programmed to generate IRQ and/or FIRQ interrupts in response to their activation. The ACVC is programmed by its control registers shown in Table 6-4. Be aware that the ACVC generated interrupts are in addition to those generated by the PIAs and cartridge. The addition of the ACVC interrupts is shown in Fig. 6-1. The IRQ interrupts may now come from PIA 1 and/or the ACVC and the FIRQ interrupts may come from PIA 2 and/or the ACVC.

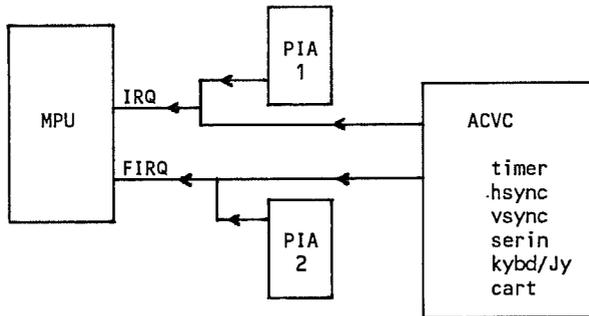


Fig. 6-1. Addition of ACVC Interrupts

Register	Dedicated Address
Init0	FF90
Init1	FF91
IRQEN	FF92
FIRQEN	FF93
Timer 1	FF94
Timer 0	FF95

Table 6-4. ACVC Interrupt Control Registers

### Init0 Register

Bits 4 and 5 of this register control the ACVC's generation of IRQ and FIRQ interrupts. When bit 5 is clear ACVC IRQ interrupt generation

is disabled. When set its generation is enabled. When bit 4 is clear ACVC FIRQ interrupt generation is disabled. When set its generation is enabled.

### Init1 Register

Bit 5 of this register controls the rate at which the timer value (in Timer 1 and 0) is decremented. When clear the timer value is decremented once every 63.5 microseconds or at a rate of about 15,748 times a second. When set the timer decrementation rate is once every 70 nanoseconds or about 14,300,000 times a second.

### IRQEN Register

This register serves the dual purposes of selecting those event(s) which will cause an IRQ interrupt and identifying/releasing a pending IRQ interrupt request. Only the lower six bits of this register are used as shown in Table 6-5. Setting certain bits of this register enables IRQ interrupt generation, if enabled by Init0, by their corresponding sources. Reading this register indicates to the ACVC that the MPU has responded to its IRQ interrupt request. That is, reading this register causes the ACVC to deactivate its pending IRQ interrupt request. Also, reading this register indicates which of the six sources (from Table 6-5) caused the current interrupt by one of its bits being set. This register is typically read by an interrupt handler.

Bit 7 - Not used	Bit 3 - Vbord
Bit 6 - Not used	Bit 2 - SerIn
Bit 5 - Timer	Bit 1 - Kybd/Jy
Bit 4 - Hbord	Bit 0 - Cart

Table 6-5. IRQEN Register

- Timer - When clear no IRQ interrupt is caused by Timer 1 and 0. When set an IRQ interrupt is generated, if enabled by Init0, when the timer value (in Timer 1 and 0) decrements to zero.
- Hbord - When clear no IRQ interrupt is caused by the video horizontal scanning circuits. When set an IRQ interrupt is generated, if enabled by Init0, when the video horizontal scan reaches the right hand border. This provides an interrupt once every 63.5 microseconds.
- Vbord - When clear no IRQ interrupt is caused by the video vertical scanning circuits. When set an IRQ interrupt is generated, if enabled by Init0, when the video vertical scan reaches the bottom border. This provides an interrupt once every 16.6 milliseconds or sixty times a second.

- SerIn - When clear no IRQ interrupt is generated because of serial input data. When set an IRQ interrupt is generated, if enabled by Init0, when the serial input data changes from a 0 to a 1.
- Kybd/Jy - When clear no IRQ interrupt is caused by the keyboard keys or joystick fire buttons. When set an IRQ interrupt is generated, if enabled by Init0, when any joystick fire button is depressed or when a keyboard key is depressed. However, first the keyboard must have been prepared by sending zeroes to bits PB7-0 of PIA 1.
- Cart - When clear no IRQ interrupt is caused by the CART signal from pin 8 of the cartridge connector. When set an IRQ interrupt is generated, if enabled by Init0, when the CART signal is active (low).

### **FIRQEN Register**

This register is of the same format and operation as the IRQEN register except that it controls FIRQ interrupt generation. Table 6-5 shows this register's internal format. Besides controlling FIRQ interrupt generation, this register serves to release and identify a pending FIRQ interrupt request just as the IRQEN register does for IRQ interrupts.

### **Timer 1 and 0 Registers**

Timer 1 and 0 control the rate of high resolution text character blinking and of the Timer interrupt. This rate is controlled by putting a 12-bit value into the Timer 1 and 0 registers. In Timer 0 is put the lower eight bits and in Timer 1 is put the upper four bits of the 12-bit timer value. The ACVC then decrements the timer value at a rate determined by bit 5 of the Init1 register. When the value reaches zero a text character blinks if so selected and an IRQ and/or FIRQ interrupt is generated if so selected. Then the ACVC automatically reloads Timer 1 and 0 with the value last put in them and the decrementing begins again.

## **INTERRUPT PROGRAMMING GUIDELINES**

This section provides programming guidelines for using ACVC generated interrupts. Normally a program that uses interrupts is of three sections; initialization, main body, and interrupt handlers. Each section has its unique interrupt related responsibilities.

### **Initialization**

The initialization section of a program is that which is executed first. Its purpose is to prepare the computer and the rest of the program for subsequent execution. Fig. 6-2 shows a skeleton initialization section which is to use IRQ ACVC interrupts caused by the vertical border (Vbord).

First interrupts are disabled until the program is ready for them and the s stack is established. Next (lines 120-170) the unused PIA interrupts are turned off and (lines 180-210) any of their pending interrupts are bled off. Next (220-250) the ACVC Vbord IRQ interrupt generation is enabled and (260-270) any pending ACVC interrupts are bled off. Next the secondary jump table is set up to point to the IRQ interrupt handler (INTH). At lines 320-340 other initialization unique to this program is done, such as clearing tables, etc. Finally, IRQ interrupts are enabled and the MPU is directed to execute the main body section of the program.

```

100 START   ORCC #50           LOCKOUT INTS
110         LDS #30+STACK    ESTABLISH S STACK
120         LDA #34          TURN OFF
130         LDX #FF00        INTERRUPTS
140         STA 1,X          FROM
150         STA 3,X          ALL
160         STA $21,X        PIA S
170         STA $23,X
180         LDA ,X          BLEED OFF
190         LDA 2,X          INTERRUPTS
200         LDA $20,X        FROM
210         LDA $22,X        PIA S
220         LDA #$EE        ENABLE ACVC
230         STA $90,X        IRQ INTRPTS
240         LDA #8           FROM
250         STA $92,X        VBORD
260         LDA $92,X        BLEED OFF IRQ
270         LDA $93,X        AND FIRQ INTRPTS
280         LDA #$7E        SET UP SECONDARY
290         LDX #INTH        JUMP
300         STA $10C         TABLE
310         STX $10D
320         - - - - -        REST OF YOUR
330         - - - - -        PROGRAM
340         - - - - -        INITIALIZATION
350         ANDCC #$EF       ENABLE IRQ INTRPTS
360         JMP MAINBOD      GOTO MAIN BODY
370 *****
380 STACK   RBM 30          S STACK AREA

```

Fig. 6-2. Skeleton ACVC IRQ Initialization

### Main Body

The main body of the program may be a section that performs much processing or simply waits until it is interrupted. Its unique characteristic is that it is interruptable; its program flow may be peppered with interrupt processing.

## Interrupt Handler

Normally there are as many interrupt handlers as there are types of interrupts that will be generated. Fig. 6-3 is the skeleton IRQ ACVC interrupt handler referred to in Fig. 6-2. Its requirements are to release the ACVC interrupt, identify its source, and to return. Between these is done processing unique to your program. When interrupts occur at a high rate, the processing of an interrupt handler should be kept short and simple so it can return before the next interrupt. Otherwise, some interrupts may be lost (never processed).

```
100 INTH  LDA  $FF92      RELEASE INTRPT
110      - - - - -      PROGRAM
120      - - - - -      UNIQUE
130      - - - - -      PROCESSING
140      RTI             RETURN
```

Fig. 6-3. Skeleton ACVC IRQ Interrupt Handler

## Chapter 7

# CONCLUDING DETAILS

This chapter provides more information about the CoCo 3 program execution environment, advanced programming guidelines, and other items not covered in other chapters.

### RESET INITIALIZATION

Reset initialization is that process which the computer performs after it is turned on or the reset button is depressed. This process is composed of the hardware's response to a reset and a reset initialization program. The reset process is begun by activating the Reset signal.

#### Hardware Reset Response

The hardware reset response is activated by activating and then deactivating the Reset signal. This signal is connected to the MPU, PIAS, ACVC, and cartridge connector.

When reset, the PIAS' internal registers (control, data, and data direction) are cleared. When reset, the ACVC's internal registers are cleared. This causes the MMU to be disabled, the ROM/RAM mode to be selected, the ROM mapping to be 16K internal/16K external, and other (less important) conditions to be selected. All of these conditions may be determined from the Cross Reference.

When reset, the MPU clears the DP register, sets the I and F bits of the CC register, and loads the PC register with the contents of addresses FFFE and FFFF (typically 8C1B). This directs the MPU to execute the reset initialization program.

## Reset Initialization Program

The reset initialization program exists in ROM. The first small part starts at address 8C1B. The remainder is in internal ROM addresses C000-DC09. In Fig. 7-1 is the flowchart of the reset initialization program. In general, the operations performed are as follows.

- a. Set up all PIA and ACVC control registers.
- b. Copy part of itself to RAM addresses 4000-436C.
- c. Copy ROM to RAM.

Notice that copying part of itself to RAM will destroy any program that was in addresses 4000-436C. Three variations of reset initialization program operation depend on which keyboard keys were depressed during its execution. The keyboard keys and the results are listed below.

- a. None - Select normal color set and go to BASIC's initialization.
- b. F1 - Select alternate color set and go to BASIC's initialization.
- c. CTRL + ALT - Display picture of Hawkins, Harris, and Earles.

The new BASIC commands are provided by Super Extended BASIC. The reset initialization program adds them to the original BASIC (after it and Super Extended BASIC are copied into RAM) by installing jump instructions into selected areas of the original BASIC. The jump instructions direct the MPU to Super Extended BASIC which performs the new commands. Thus, the new commands are available to BASIC only in the all-RAM mode. However, even though the new BASIC commands are not available in the ROM/RAM mode, an assembly language program can use subroutines of Super Extended BASIC in ROM.

Fig. 7-2 is the memory map as set up by the reset initialization program and as used by BASIC. The original BASIC (Color, Extended Color, and cartridge) uses the executive set of PARS which map 64K of virtual memory to 32K of RAM and 32K of RAM or ROM (depending on whether in the all-RAM or ROM/RAM mode). Virtual addresses 0000-7FFF point to RAM in either memory mode. Virtual address 8000-FEFF point to RAM in the all-RAM mode or to ROM in the ROM/RAM mode.

Super Extended BASIC uses the task PAR set. These PARS are set up so virtual memory covers Super Extended BASIC, Color BASIC, BASIC's lower 8K of RAM, and the secondary stack and high resolution graphics display buffer areas. Apparently, Super Extended BASIC modifies some task PARS to point to the high resolution text and Get/Put buffer areas as required. As you can see in Fig. 7-2, BASIC may use most of the RAM of a 128K computer.

Addr	Step	Operation
8C1B	1	Set CC I and F bits
ROM	2	Init0 < \$0A
C000	3	Set CC I and F bits
ROM	4	S < \$5EFF
	5	Load all palette regs with color code 18 (green)
	6	Load all PAR s      Exec - 38,39,3A,3B,3C,3D,3E,3F Task - 38,30,31,32,33,3D,35,3F
	7	Init0 < \$CE
	8	Copy rest of this program to RAM \$4000-\$436C
	9	Go to step 10 (in RAM)
4000	10	S < (S) - 1
RAM	11	Timer 1 & 0 < \$FFF
	12	Load ACVC control regs FF98-FF9F with 0,0,0,0,0F,E0,0,0
	13	Load all PIA regs for normal operation
	14	Clear all SAM registers and set SAM F1
	15	Clear DP register
	16	Write zeroes to keyboard (FF02)
	17	F1 key depressed ? If so set F1 flag
	18	CTRL and ALT keys depressed ?    Yes - go to step 19 No - go to step 26
	19	Clear BASIC (71) and init (5EEF) warm start flags
	20	Select ROM/RAM mode
	21	Load palette regs 10 < 9 and 11 < 3F
	22	Move picture from C405 to buffer (E00)
	23	Select G6R low res display mode
	24	Point SAM buffer to E00
	25	Go to step 25

Fig. 7-1a. Reset Initialization Flowchart



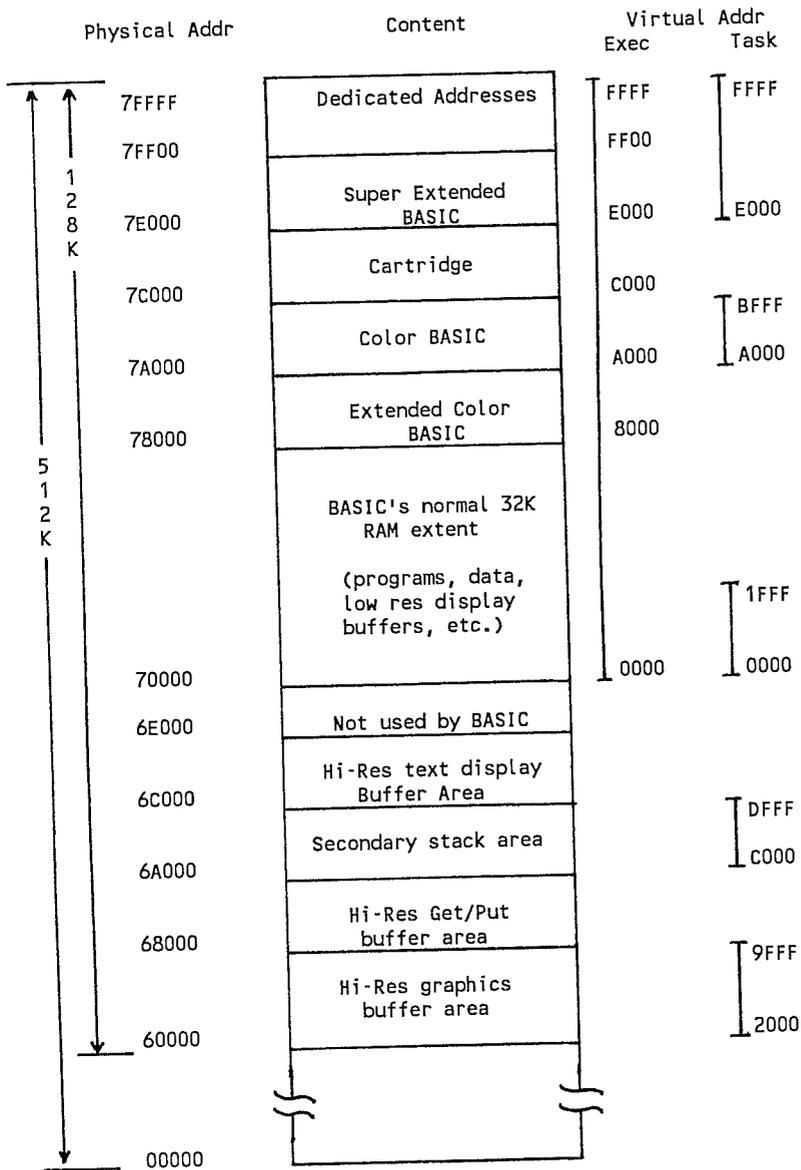


Fig. 7-2. CoCo 3 Memory Map

## **ADVANCED GRAPHICS**

This section provides information which may be useful when planning and programming advanced graphics effects. The topics include colors, animation, and horizontal scrolling.

### **Colors and Animation Speed**

For any graphics application there are requirements such as the number of colors simultaneously displayable, the horizontal and vertical resolutions, and the speed at which the content of the screen buffer is to be updated by the MPU. More colors and higher resolution require a larger buffer area. However, the content of a large buffer area can not be changed as quickly as that of a smaller buffer area. Thus, a compromise is often reached that results in a manageable buffer size and an adequate color set and resolution. For example, graphics applications that require high speed moving objects are more successfully implemented in a display mode with a smaller buffer size. Of course, the smaller buffer size limits the number of simultaneously displayed colors and the resolution.

The number of simultaneously displayed colors can be increased considerably by blending pixels of different colors. This is done by setting the even numbered pixels of a row to one color and the odd pixels to another. With this technique a 2-color display mode can generate three colors, a 4-color mode can generate ten colors, and a 16-color mode can generate 136 colors. However, the horizontal resolution should be high enough (160 or more) to cause the different colors of adjacent pixels to blend. The colors appear to blend most successfully on a TV monitor.

### **Smooth Horizontal Scrolling**

Ideally one would like to horizontally scroll the image one pixel at a time. Unfortunately the horizontal offset register provides scrolling of 2-byte increments. This moves the image 4, 8, or 16 pixels for each increment when the display mode is capable of 16, 4, or 2 colors, respectively. For single pixel scrolling, the solution is to have 4, 8, or 16 buffers, depending on the number of colors (16, 4, or 2, respectively) provided by the chosen display mode. The content of each buffer is the same as the others but each image is offset by one pixel w.r.t. the preceding buffer. Smooth horizontal scrolling is accomplished by incrementing or decrementing the horizontal offset register content and selecting the appropriate buffer with the vertical offset 1 and 0 registers. It is suggested that HE be set to prevent screen image wrap-around.

## **INTERRUPTS & VIRTUAL MEMORY**

This section provides guidelines for using interrupts in a virtual memory computer. These guidelines are most important for programs that use interrupts and change the virtual memory mapping as they perform different processes.

## Interrupt Jump Tables

The NMI, IRQ, and FIRQ interrupts may occur at almost any time during a program's execution. Therefore, the primary jump table (and secondary if used) must always be mapped by a virtual address extent. That is, a PAR of the PAR set in use must point to the physical page(s) in which the interrupt jump table(s) resides. If both PAR sets are used, a PAR of each must point to that physical page.

The primary jump table exists in physical page 3F of RAM. Also, PAR 7 of both sets are initialized to point to that page. This jump table's virtual address (FEEE-FEFF) is constant because it is determined by the content of the interrupt vector table (in ROM). Thus, it would seem that PAR 7 of both sets must always point to physical page 3F. Fortunately, this is not the case if bit 3 of Init0 is set. If it is clear, this is the case. When set, the virtual address extent FE00-FEFF always accesses those memory locations of physical RAM page 3F as if PAR 7 were pointing to it (even in the ROM/RAM mode). With this bit set PAR 7 can be pointed to another physical page, but the primary jump table and other unused cells will always be available in virtual addresses FE00-FEFF. Notice that the working space of virtual page 7 is limited to E000-FDFF.

If the secondary jump table is used it must also always be mapped. This can be done by including it in a part of your program (the controller) that is always mapped by both PAR sets. That is what BASIC does. In Fig. 7-2 the virtual extent (which includes the secondary jump table) 0000-1FFF is mapped by the executive and task PAR sets.

## Interrupt Handlers

The interrupt handlers must be as ever-present as the jump tables. A program that changes the virtual memory mapping should have a segment (the controller) that is always mapped. Within this segment should be the interrupt handlers.

## ODDS & ENDS

This section provides information about other differences between the CoCo 3 and its predecessor. These differences are relatively minor and are not addressed in other areas of this book.

## Keyboard and Joystick Buttons

The new keyboard has four additional keys (F1, F2, CTRL, and ALT). Also, new joysticks, each with two fire buttons, are accommodated. All the keyboard keys and fire buttons are read by PIA 1 as shown in Fig. 7-3. Button 2 is the new joystick button.

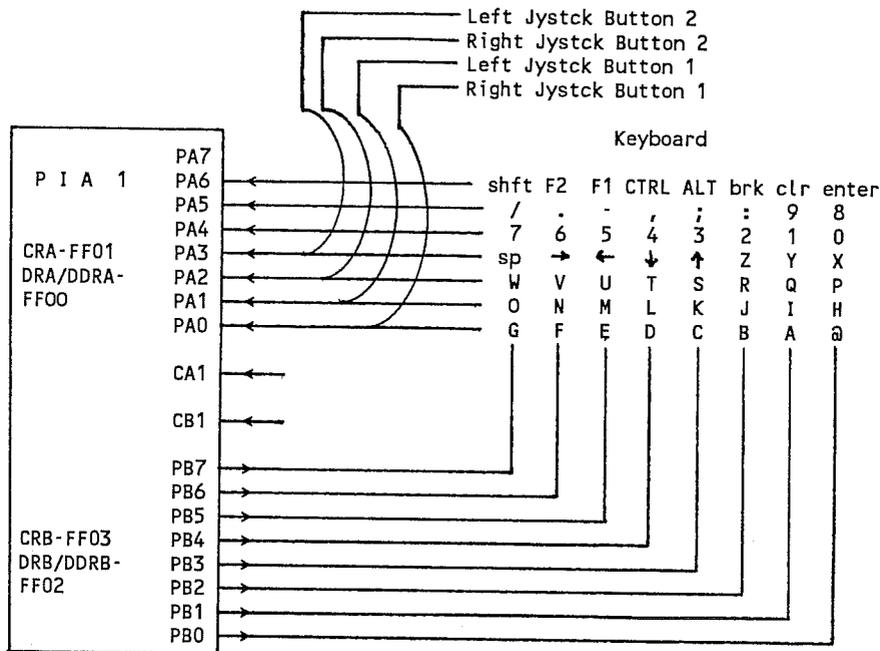


Fig. 7-3. Keyboard and Joystick Button Connection

### A Split Dedicated Address

In the original CoCo the VDG was connected to bits PB7-3 of PIA 2 (accessed via dedicated address FF22). Now the VDG is incorporated into the ACVC, leaving bits PB7-3 of PIA 2 unused. Now, whenever dedicated address FF22 is accessed (read or write) the transmitted byte is split such that bits 7-3 are connected to the VDG in the ACVC and bits 2-0 are connected to side B of PIA 2 (as before).

### Redundant Dedicated Addresses

A redundant dedicated address is one that performs the same as the normal dedicated address with which it is associated. Their use is not necessarily recommended. Table 7-1 shows the dedicated addresses that are accessible by other (redundant) addresses.

### SCS Enable Control

The scs signal goes to the cartridge connector and is activated when the scs Enable bit (2) of Init0 is set and the MPU reads from or writes to virtual addresses FF40-FF5F. This signal is used to select control registers in a cartridge such as those in the disk controller. When the scs Enable bit is clear the scs signal is never activated.

Dedicated Address	Redundant Addresses
FF00	FF04, FF08, FF0C
FF01	FF05, FF09, FF0D
FF02	FF06, FF0A, FF0E
FF03	FF07, FF0B, FF0F
FF20	FF24, FF28, FF2C
FF21	FF25, FF29, FF2D
FF22	FF26, FF2A, FF2E
FF23	FF27, FF2B, FF3F
FF40	FF41-47, FF50-57
FF48	FF4C, FF58, FF5C
FF49	FF4D, FF59, FF5D
FF4A	FF4E, FF5A, FF5E
FF4B	FF4F, FF5B, FF5F

Table 7-1. Redundant Dedicated Addresses

### Lo/Hi MPU Speed

The MPU clock rate, which determines the instruction execution speed, is controlled by dedicated addresses FFD8 and FFD9. Writing anything to FFD8 selects the normal low speed. Writing to FFD9 selects the high MPU speed. In the high speed mode the MPU executes instructions twice as fast as when in the low speed mode. Certain subroutines of BASIC (serial and cassette I/O) which depend of the MPU being in the low speed mode will not operate correctly in the high speed mode.

### Accessing Dedicated Addresses

Of the dedicated addresses, some may be only written to and others either written to or read from. Table 7-2 lists these categories. An interesting phenomena is the PAR and palette registers appear to be of seven bits in the all-RAM mode and of six bits in the ROM/RAM mode. The bit 6 which appears in the all-RAM mode is not used.

Dedicated Address	Access
FF00-FF03	read/write
FF20, 21, 23	read/write
FF22 (bits 2-0)	read/write
FF22 (bits 7-3)	write
FF40	read/write
FF48-FF4A	read/write
FF90-FF9F	write
FFA0-FFBF	read/write

Table 7-2. Dedicated Address Accessibility

# APPENDIX A

## High Resolution Text Characters and Video Codes

Char	Video dec	Code hex	Char	Video dec	Code hex	Char	Video dec	Code hex
Ç	0	0	+	43	2B	V	86	56
ü	1	1	,	44	2C	W	87	57
é	2	2	-	45	2D	X	88	58
â	3	3	.	46	2E	Y	89	59
à	4	4	/	47	2F	Z	90	5A
à	5	5	0	48	30	[	91	5B
â	6	6	1	49	31	\	92	5C
ç	7	7	2	50	32	]	93	5D
è	8	8	3	51	33	↑	94	5E
è	9	9	4	52	34	←	95	5F
è	10	A	5	53	35	^	96	60
ï	11	B	6	54	36	a	97	61
ï	12	C	7	55	37	b	98	62
ß	13	D	8	56	38	c	99	63
Ä	14	E	9	57	39	d	100	64
Å	15	F	:	58	3A	e	101	65
Ö	16	10	;	59	3B	f	102	66
ö	17	11	<	60	3C	g	103	67
Æ	18	12	=	61	3D	h	104	68
Ö	19	13	>	62	3E	i	105	69
ö	20	14	?	63	3F	j	106	6A
ø	21	15	@	64	40	k	107	6B
Û	22	16	A	65	41	l	108	6C
ù	23	17	B	66	42	m	109	6D
ø	24	18	C	67	43	n	110	6E
Ö	25	19	D	68	44	o	111	6F
Û	26	1A	E	69	45	p	112	70
£	27	1B	F	70	46	q	113	71
£	28	1C	G	71	47	r	114	72
±	29	1D	H	72	48	s	115	73
°	30	1E	I	73	49	t	116	74
f	31	1F	J	74	4A	u	117	75
sp	32	20	K	75	4B	v	118	76
!	33	21	L	76	4C	w	119	77
"	34	22	M	77	4D	x	120	78
#	35	23	N	78	4E	y	121	79
\$	36	24	O	79	4F	z	122	7A
%	37	25	P	80	50	{	123	7B
&	38	26	Q	81	51		124	7C
'	39	27	R	82	52	}	125	7D
(	40	28	S	83	53	~	126	7E
)	41	29	T	84	54	-	127	7F
*	42	2A	U	85	55			

# CROSS REFERENCE

## Dedicated Addresses

ADDRESS	DESCRIPTION	PAGE
	PIA 1 Access	
FF00	Data reg A / Data direction reg A	44,46,48,52-54
FF01	Control register A	
FF02	Data reg B / Data direction reg B	
FF03	Control register B	
	PIA 2 Access	
FF20	Data reg A / Data direction reg A	44,46,48,53,54
FF21	Control register A	
FF22	Data reg B / Data direction reg B and VDG	35-38,46,53,54
FF23	Control register B	
	ACVC Registers	
FF90	Init0 Register	
Bit 7 - 0 = Hi-Res, 1 = Lo-Res display		23,31,35
Bit 6 - 0 = MMU disabled, 1 = MMU enabled		20
Bit 5 - 0 = Disable, 1 = Enable ACVC IRQ interrupts		41,42
Bit 4 - 0 = Disable, 1 = Enable ACVC FIRQ interrupts		41,41
Bit 3 - 0 = FExx is normal, 1 = FExx is constant		52
Bit 2 - 0 = SCS disabled, 1 = SCS enabled		53
Bit 1 - MC1 ROM mapping		21
Bit 0 - MC0 ROM mapping		21
FF91	Init1 Register	
Bit 7 - Not used		
Bit 6 - Not used		
Bit 5 - Timer decr rate 0 = 63.5 us, 1 = 70 ns		27,43
Bit 4 - Not used		
Bit 3 - Not used		
Bit 2 - Not used		
Bit 1 - Not used		
Bit 0 - 0 = Exec PARs, 1 = Task PARs in use		20
FF92	IRQ Control register (IRQEN)	41-45
Bit 7 - Not used		
Bit 6 - Not used		
Bit 5 - Timer		
Bit 4 - Hbord		
Bit 3 - Vbord	( 0 = disable	
Bit 2 - SerIn	1 = enable )	
Bit 1 - Kybd/Jy		
Bit 0 - Cart		

ADDRESS	DESCRIPTION	PAGE
FF93	FIRQ Control register (FIRQEN)	43,44
	Bit 7 - Not used	
	Bit 6 - Not used	
	Bit 5 - Timer	
	Bit 4 - Hbord	
	Bit 3 - Vbord ( 0 = disable	
	Bit 2 - SerIn ( 1 = enable )	
	Bit 1 - Kybd/Jy	
	Bit 0 - Cart	
FF94	Timer 1 Register	27,43
	Bits 7-4 - Not used	
	Bits 3-0 - Most significant nibble of timer value	
FF95	Timer 0 Register	27,43
	Bits 7-0 - Least significant byte of timer value	
FF96	Not Used	
FF97	Not Used	
FF98	Video Mode Register	
	Bit 7 - BP 0 = text, 1 = graphics	23,24
	Bit 6 - Not used	
	Bit 5 - BPI 0 = normal color set, 1 = alternate set	14,47,49
	Bit 4 - MOCH 0 = color, 1 = monochrome video signal	15
	Bit 3 - H50 0 = 60 Hz rate, 1 = 50 Hz rate	15,16
	Bit 2 - LPR2 Vertical	
	Bit 1 - LPR1 size of	23,24,27,
	Bit 0 - LPR0 display	28,31,32
FF99	Video Resolution Register	
	Bit 7 - Not used	
	Bit 6 - VRES1	24,27,28,
	Bit 5 - VRES0	31,37
	Bit 4 - HRES2	
	Bit 3 - HRES1	24,28,31,33
	Bit 2 - HRES0	
	Bit 1 - CRES1	24,28,30,
	Bit 0 - CRES0	31,33
FF9A	Border Register	24,27,31
	Bit 7 - Not used	
	Bit 6 - Not used	
	Bit 5 - R1	
	Bit 4 - G1	
	Bit 3 - B1	
	Bit 2 - R0	
	Bit 1 - G0	
	Bit 0 - B0	
FF9B	Not Used	

ADDRESS	DESCRIPTION	PAGE
FF9C	Vertical Scroll Register	25,29,32
	Bit 7 - Not used	
	Bit 6 - Not used	
	Bit 5 - Not used	
	Bit 4 - Not used	
	Bit 3 - SCEN 0 = Enable, 1 = Disable vert scroll	
	Bit 2 - SC2	
	Bit 1 - SC1	
	Bit 0 - SC0	
FF9D	Vertical Offset 1 Register	25,29,32,35, 36,51
	Bit 7 - Y18	
	Bit 6 - Y17	
	Bit 5 - Y16	Display buffer
	Bit 4 - Y15	physical address
	Bit 3 - Y14	bits 18-11
	Bit 2 - Y13	
	Bit 1 - Y12	
	Bit 0 - Y11	
FF9E	Vertical Offset 0 Register	25,29,32,35, 36,51
	Bit 7 - Y10	
	Bit 6 - Y9	
	Bit 5 - Y8	Display buffer
	Bit 4 - Y7	physical address
	Bit 3 - Y6	bits 10-3
	Bit 2 - Y5	
	Bit 1 - Y4	
	Bit 0 - Y3	
FF9F	Horizontal Offset Register	
	Bit 7 - HE 0 = normal, 1 = expanded buffer width	25,26,28,29, 32,37,51
	Bit 6 - x6	
	Bit 5 - x5	Bits x6-0 are
	Bit 4 - x4	the horizontal
	Bit 3 - x3	shift value
	Bit 2 - x2	
	Bit 1 - x1	
	Bit 0 - x0	
FFA0-AF	MMU Page Address Registers	19-21,48,52
FFB0-BF	Palette Registers	11-14,27,30, 31,37,38,39, 49
Clear Set	SAM Registers	
FFC0 FFC1	V0 Mode	
FFC2 FFC3	V1 Register	35,36,38,48
FFC4 FFC5	V2	

ADDRESS	DESCRIPTION	PAGE
FFC6 FFC7	F0	
FFC8 FFC9	F1	
FFCA FFCB	F2 SAM	35,36,48
FFCC FFCD	F3 Vertical	
FFCE FFCF	F4 Offset	
FFD0 FFD1	F5 Register	
FFD2 FFD3	F6	
FFD8 FFD9	MPU Rate	1,54
FFDE FFDF	Memory Map Mode	21,49
	Interrupt Vectors	40,46
FFF2,3	SWI3	
FFF4,5	SWI2	
FFF6,7	FIRQ	
FFF8,9	IRQ	
FFFA,B	SWI	
FFFC,D	NMI	
FFFE,F	RESET	





