

---

---

**C o C o X T**

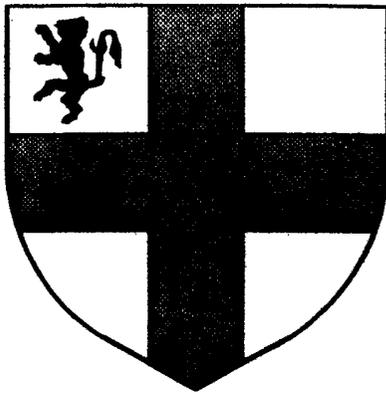
**&**

**C o C o X T - R T C**

**Hard Disk Interface and OS9 Software for the  
Tandy Color Computer  
Version 2.4**

---

---



**Copyright 1989 by Burke & Burke**

PUBLISHED BY BURKE & BURKE

COPYRIGHT NOTICE

All rights reserved. No part of this manual may be reproduced, copied, or transmitted in any form without prior written permission from Burke & Burke.

This entire manual, any accompanying hardware or computer programs, and any accompanying information storage media constitute a PRODUCT of Burke and Burke. The PRODUCT is supplied for the personal use of the purchaser. Burke & Burke expressly prohibits reproduction of this manual, the accompanying computer programs, and the printed circuit artwork.

Burke & Burke expressly requires, as a condition of providing this PRODUCT and the associated LIMITED WARRANTY to the PURCHASER, that one copy of the PRODUCT be purchased from Burke & Burke for every copy used.

Burke & Burke does not in any way transfer ownership of any computer programs to the PURCHASER. The PURCHASER is granted a limited license to use Burke & Burke computer programs distributed with the product, but only on a single computer.

LIMITED WARRANTY

Burke & Burke warrants the PRODUCT against defects in material or workmanship for a period of ninety (90) days from the date of purchase by the original owner. This warranty is limited to repair or replacement of PRODUCT which proves defective during this period, at the sole expense and discretion of Burke & Burke. This warranty specifically excludes software defects and defects caused by negligence, abuse, accident, or tampering.

DISCLAIMER

Every effort has been made to ensure the accuracy of this manual and the quality of the PRODUCT it describes. Burke & Burke makes no warranties, whether expressed, statutory, or implied, of any kind whatsoever, as to the merchantability of the PRODUCT or its fitness for a particular use, except as set forth above as the LIMITED WARRANTY.

Burke & Burke does not assume any liability for any damages, whether special, indirect, or consequential, resulting from the use of the PRODUCT. The PRODUCT is sold on an AS-IS basis.

TABLE OF CONTENTS

1.0	Introduction
1.1	About the CoCo XT
1.2	About the CoCo XT-RTC
1.3	About Hard Disks
1.4	Summary
2.0	Hardware Installation
2.1	Packing List
2.2	Before you Start
2.3	Summary of Installation Procedure
2.4	Removing the Mounting Bracket
2.5	Removing the Housing Top Cover
2.6	Connecting the Hard Disk Controller
2.7	Interface Cable Installation
2.8	Changing the Keep-Alive Battery
2.9	Replacing the Housing Top Cover
2.10	Multi-Pak Installation
2.11	Connecting the Interface Cables
2.12	Checking your Work
3.0	OS9 Software Installation

- 3.1 Summary of Installation Procedure
- 3.2 Choosing a Device Driver
- 3.3 Creating a Device Descriptor
- 3.4 Formatting your Hard Disk
- 3.5 Creating a Bootable Hard Disk System
  
- 4.0 Using the CoCo XT with OS9
  - 4.1 Protecting Data with PARK
  - 4.2 Keeping Track of Real Time
  - 4.3 Managing Directory Trees
  - 4.4 Customizing your System
  - 4.5 Hard Disk Backups
  
- 5.0 Using the CoCo XT with BASIC
  - 5.1 What is HYPER-I/O?
  - 5.2 The "XT" Device Driver
  - 5.3 Building a Device Descriptor
  - 5.4 Formatting a Hard Disk
  - 5.5 Creating Mass-Storage Areas
  - 5.6 Protecting Data with UNLOAD
  - 5.7 Accessing the Real-Time Clock
  
- A.0 Technical Summary
  - A.1 Theory of Operation
  - A.2 Parameters for Popular Hard Disks
  - A.3 Hard Disk Controller Information

## 1.0 Introduction

Congratulations! You have just purchased the first truly affordable hard disk solution for the Color Computer: Burke & Burke's CoCo XT.

This section of the manual will introduce you to some of the capabilities of the CoCo XT and CoCo XT-RTC hard disk interfaces.

### 1.1 About the CoCo XT

The CoCo XT interfaces a standard PC compatible hard disk drive and controller to any Color Computer. Burke & Burke chose this type of hard disk system because it offers a combination of high quality, high performance and low price.

The CoCo XT uses the same type of hard disk controller that is used in many "clone" personal computers. The controller includes a 2000 character disk buffer, which eliminates the need to stop the computer while accessing the hard disk. This is often referred to as "no halt" operation.

The CoCo XT is a hard disk interface. The housing of the CoCo XT holds both the interface board and a hard disk controller board that you supply. In order to use the CoCo XT, you will also

need:

- Hard disk controller (Western Digital WD1002-WX1 or equivalent)
- Hard disk drive(s) & cables
- Case and power supply for hard disk drive(s)
- CoCo, Multi-Pak Interface, and Floppy Disk System.
- RS-DOS (1.0, 1.1, 2.0, or 2.1) and Burke & Burke HYPER-I/O (to use hard disk from BASIC)
- OS9 Level I or Level II operating system (to use hard disk from OS9)

Here are a few of the features of the CoCo XT Color Computer hard disk interface:

- Adapts Western Digital WD1002-WX1 or WD1002-27X (RLL) hard disk controller for use in CoCo 1, CoCo 2, or CoCo 3.
- Controller supports 1 or 2 hard drives of any configuration, up to 1024 cylinders and 16 R/W heads (120 megabytes per drive).
- Unit installs in any Multi-Pak slot.
- CoCo XT direct microprocessor interface is 30% faster than

SASI / SCSI Color Computer hard disk systems.

- Fully assembled and tested circuit board uses all prime parts.
- 100% integrated solid-state design for high reliability.
- Compatible with all versions of CoCo OS9 (Level 1 and Level 2).
- Custom DOS can be installed in hard disk controller BIOS ROM socket and accessed at CoCo \$C000-\$FFFF address range.
- Full OS9 support.
- OS9 memory requirements and performance can be customized by selecting from several hard disk driver routines.
- DDMAKER utility creates hard disk device descriptors for OS9.
- FIX and EZGEN utilities simplify development of custom system configurations.
- SYSDEV utility allows the hard disk to be used as OS9's default drive.
- PARK utility protects R/W heads during transportation of hard

drive equipment.

- Includes device driver for Burke & Burke's HYPER-I/O for full BASIC hard disk support.

## 1.2 About the CoCo XT-RTC

The CoCo XT-RTC has all of the features of the CoCo XT, as well as a battery-backed real-time clock. One battery lasts over 10 years, and you can replace it easily at home as your CoCo becomes a teenager.

The crystal-controlled CMOS clock circuit provides 24 hour time and date, with a resolution of 1/100 second. It automatically adjusts for leap years.

Your computer can automatically set the system time from the real-time clock whenever you boot OS9, and you can display the time from the real-time clock at will. You can set the real-time clock using OS9's SETIME utility and the CLKSET utility provided with the CoCo XT-RTC.

Machine-language subroutines are provided to transfer information between the real-time clock and Disk BASIC strings.

## 1.3 About Hard Disks

You don't need to know how a hard disk works in order to use one. If you are

interested in technical details, Appendix A explains hard disk operation in detail. This section will tell you what you need to know about a hard disk in order to set one up on your CoCo.

The capacity of a hard disk is often described in "megabytes" or "MB". Each megabyte is one million characters of information, so a 20 MB drive will hold roughly 20,000,000 characters of information.

The location of every character of information stored on a hard disk can be described by 4 parameters: "head", "track", "sector", and "byte". In turn, the capabilities of your hard drive are set by the ranges of legal values for these parameters.

For example, a 20 megabyte hard disk might allow these ranges of parameters:

Head	-	0	-	3
Track	-	0	-	613
Sector	-	0	-	31
Byte	-	0	-	255

You will need to know the number of heads and number of tracks supported by your hard disk in order to configure the CoCo XT software correctly. There are always 32 sectors and 256 bytes when using the CoCo XT with a Western Digital hard disk controller. Note that some disk manufacturers refer to heads as "surfaces", "platters", or "R/W heads"; tracks are also

called "cylinders". The number of heads and number of tracks are normally supplied by the hard disk vendor. If you do not have this information, refer to Appendix A for values to be used with several popular hard disk drives.

The CoCo XT supports two other hard disk parameters: write precompensation and head parking track. Each of these parameters is expressed as a "track number". See Appendix A if you want to know more about these parameters -- you will need to know the correct values for your hard drive when configuring the CoCo XT software.

#### 1.4 Summary

The CoCo XT will allow you to economically interface a standard, PC compatible hard disk system to the Tandy Color Computer. The resulting hard disk system will combine rapid hard disk access with "no-halt" operation.

The CoCo XT-RTC augments the features of the CoCo XT with a battery-backed real-time clock.

You will also need a controller, drives, cables, and a hard disk power supply in order to use the CoCo XT interface. You must know the parameters of your hard drive in order to set up the CoCo XT system.

Chapter 2 will provide instructions for installing your CoCo XT hard disk interface and hard drive. Configuration under OS9 is described in Chapter 3.

Chapter 4 provides some hints on using a hard disk under OS9.

RS-DOS users should read Chapter 5, which discusses Burke & Burke's HYPER-I/O software package. This package will allow you to access your hard drive from Disk BASIC.



## 2.0 Hardware Installation

This section will tell you how to connect the CoCo XT (or CoCo XT-RTC), your hard disk system, and your Color Computer.

### 2.1 Packing List

If you have purchased the CoCo XT, the package should contain all of the following items:

ITEM	QTY	Description
----	---	-----
1	1	Black Anodized Aluminum Housing
2	1	CoCo XT Circuit Board (pre-installed in Housing)
3	1	User's Manual
4	1	CoCo XT Software Disk
5	1	1/16" Allen Wrench

If you have purchased the CoCo XT-RTC, the package should contain all of the following items:

ITEM	QTY	Description
----	---	-----
1	1	Black Anodized Aluminum Housing
2	1	CoCo XT-RTC Circuit Board (pre-installed in Housing)
3	1	User's Manual
4	1	CoCo XT-RTC Software Disk
5	1	1/16" Allen Wrench
6	1	Lithium Battery (BR2325) (pre-installed)

If you do not find one or more of these items in your package, please check again. If you are quite sure that something is missing, contact Burke & Burke.

## 2.2 Before you Start

This section will help you get ready to begin setting up your CoCo XT hardware.

Each of the sections below describes a step in the installation procedure. You may want to read through all of the steps before you begin the installation, and then follow along step-by-step as you actually install your controller.

### 2.2.1 Tools

You will need to use the 1/16" allen wrench supplied with your CoCo XT interface in order to open and close the CoCo XT housing.

A small flat-blade screwdriver should be used to remove the mounting bracket from your hard disk controller.

### 2.2.2 Protecting Your Investment

A hard disk controller is vulnerable to static electricity until it has been installed in the CoCo XT housing. A static spark can permanently damage your controller, so you may want to take the following precautions during installation:

- Work in a humid environment.
- Avoid working on cold, dry days.
- Wear cotton clothing. Never mix rubber or nylon clothing with wool clothing when working with sensitive electronic equipment.
- Ground yourself while working. One way to do this is to hold onto the exposed metal "shield" of your CoCo's cassette or video cable while working.

### 2.3 Summary of Installation Procedure

The mounting bracket is removed from the hard disk controller.

The top cover of the CoCo XT housing is removed, and the hard disk controller's edge connector is inserted into the mating connector on the CoCo XT or CoCo XT-RTC circuit board.

The threaded fasteners along the top edge of the CoCo XT housing securely fasten the hard disk controller to the housing.

The hard disk interface cables may be installed on the hard disk controller board once the top cover of the housing has been removed.

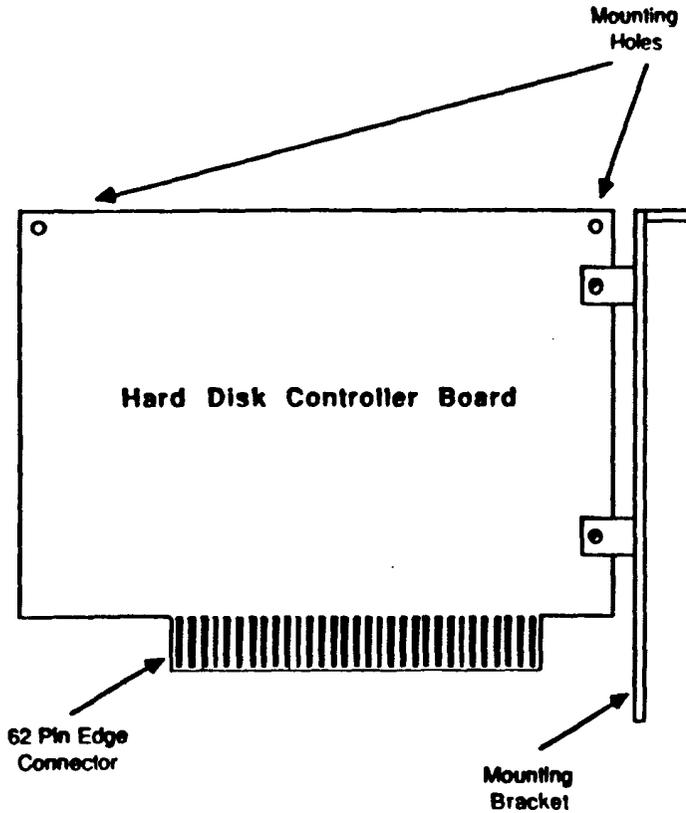
The CoCo XT-RTC battery may also be installed or replaced once the top cover of the housing has been removed. The battery is pre-installed in new CoCo XT-RTC units.

When all installation procedures are complete, the top cover of the housing is replaced. The assembled unit may now be installed in the CoCo Multi-PAK.

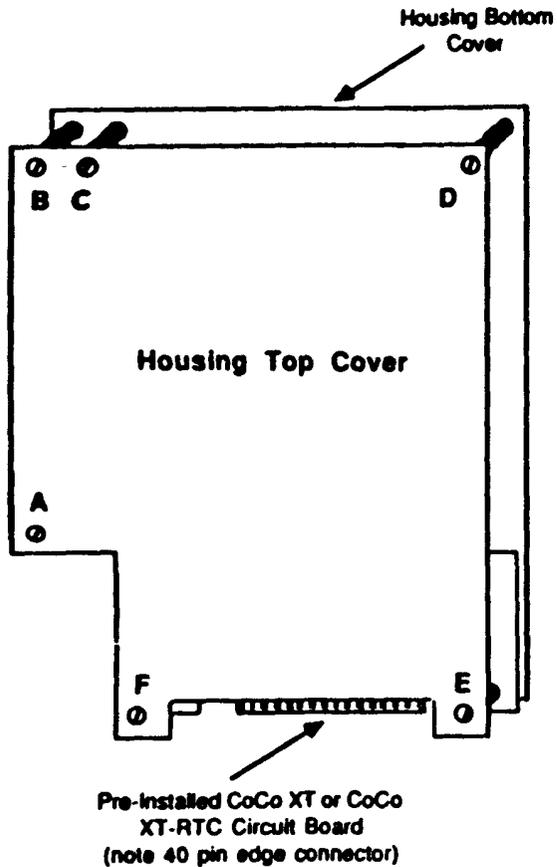
#### 2.4 Removing the Mounting Bracket

TAKE STATIC PROTECTION PRECAUTIONS DURING THIS STEP.

Figure 2.1 shows an outline drawing of a typical hard disk controller board. The board dimensions are different for each



**Figure 2.1 Typical Outline of Hard Disk Controller Board**



**Figure 2.2 Front View of CoCo XT Housing**

type of controller, but all controllers have a mounting bracket that is located as shown in the Figure.

The bracket is attached to the controller board by 2 machine screws. Remove these screws and the mounting bracket. You will not need these items unless you wish to install the hard disk controller in another type of computer in the future.

### 2.5 Removing the Housing Top Cover

The CoCo XT housing consists of a "top cover" and a "bottom cover". Each cover is an anodized aluminum plate. The plates are held together by threaded spacers and machine screws. The housing is shipped fully assembled with the CoCo XT board installed.

To remove the top cover:

- 1) Position the housing as shown in Figure 2.2. The 40 pin edge connector should be at bottom right.
- 2) Using the 1/16" allen wrench, remove the screws marked A, B, C, D, E, and F in Figure 2.2. Save these screws for use when replacing the top cover.
- 3) Lift the top cover from the remaining housing and set it aside. This will expose the component side of the CoCo XT circuit board.

4) Remove the 3/4" threaded spacers supporting screws C and D, by gently turning them counter-clockwise several turns. Set these two spacers aside for use when installing the hard disk controller.

## 2.6 Connecting the Hard Disk Controller

TAKE STATIC PROTECTION PRECAUTIONS DURING THIS STEP.

The CoCo XT board is exposed once the housing top cover has been removed. Note that the board is sandwiched between 2 threaded spacers at its lower right and lower left corners.

Note that the top edge of the board is not supported in any way. Take care not to flex the CoCo XT board during installation, or you may damage it.

The following steps connect the hard disk controller to the CoCo XT and securely fasten both units to the back cover of the housing. This procedure assumes that you have removed the housing top cover.

1) Remove the hard disk controller from its protective packing material.

2) With the housing arranged so that the component side of the CoCo XT board is up, grip the black plastic body of the edge connector on the CoCo XT board

firmly in your left hand. Arrange your grip so as to prevent flexing of the CoCo XT board and to prevent motion of the housing.

- 3) Grip the hard disk controller firmly in your right hand, with the component side up and the gold edge fingers facing you.
- 4) Align the gold edge fingers on the controller board with the gold pins on the CoCo XT edge connector.
- 5) Push the edge fingers firmly into the edge connector, being careful not to flex the CoCo XT circuit board.
- 6) Release your grip on the CoCo XT board and the hard disk controller. The mounting holes (Figure 2.1) on the hard disk controller should align with the mounting posts which secure the spacers at positions C and D (Figure 2.2).
- 7) Place the mounting posts through the mounting holes on the hard disk controller. You may need to adjust the position of the hard disk controller in the CoCo XT edge connector slightly in order to get the posts to line up with the mounting holes.
- 8) Fasten the controller to the housing back cover by replacing the  $3/4$ " threaded spacers at positions C and D.

## 2.7 Interface Cable Installaion

The hard disk controller is connected to a hard disk by two interface cables. These cables are supplied with either the hard disk or the controller (not included with CoCo XT).

The best type of interface cables to use with the CoCo XT are shielded cables. These cables are covered with a layer of copper foil, which is in turn protected by a rubber coating. Shielded cables minimize interference with other electronic equipment such as radios and television sets.

Each interface cable has a different connector at each end. The end of the cable that looks like many small, square holes goes to the hard disk controller (end "A"). The other end, which looks like a long narrow slot, goes to the hard disk drive (end "B").

It is extremely important that you connect the cables exactly as described below. Failure to do this can cause permanent damage to the controller or hard disk.

The steps for connecting the cables to the controller, once the controller has been installed in the housing, are:

- 1) Locate the larger of the 2 interface cables (this cable has 34 wires).

2) Locate "pin 1" of the "A" end of this cable. This pin is usually identified by a painted or recessed arrow head on the connector, and should be on the end of the cable where there is a colored stripe.

3) Locate 2 rows of 17 vertical pins each on the top edge of the hard disk controller board.

4) There will be a small "1" printed near one of these pins. Locate this pin and insert the "A" end of the cable over ALL of the pins so that pin 1 of the "A" end mates with the vertical pin marked with a "1".

5) Locate the smaller of the 2 interface cables. This cable should have 20 wires.

6) Locate "pin 1" of the "A" end of this cable, in the same manner as you located pin 1 of the larger cable.

7) There will be 2 rows of 10 vertical pins each at two different places on the hard disk controller. If you have only one hard drive, you will want to use the header that has the lowest "J number". For example, if the headers are marked J2 and J3, connect your interface cable to J2 in the same manner as you connected the larger cable. BE SURE TO CONNECT PIN 1 TO PIN 1.

8) If necessary, bend the cables so that they exit the top of the housing and do not interfere with any of the threaded spacers that hold the housing together.

## 2.8 Changing the Keep-Alive Battery

The CoCo XT-RTC is shipped with the 3 volt keep-alive battery installed.

The type of keep-alive battery used in the CoCo XT-RTC is often called a "button cell", because it looks like a large silver coat button. The polarity of the cell is clearly marked with a "+" and a "-".

The battery is inserted into a circular holder on the circuit board and is held in place by a spring-like metal clip. This clip is also the positive battery contact, and is marked with an engraved "+". The negative battery contact is in the bottom of the circular battery holder, and the holder itself is marked with "NEG" to indicate the negative contact.

To replace the battery, once the housing is opened:

- 1) Supporting the CoCo XT's battery holder with your right thumb, gently lift the spring metal clip with your with forefinger.

- 2) Turn the CoCo XT board on its side

and tap it lightly against a hard surface. The old battery should fall out of the battery holder. If it doesn't, use a small screwdriver to pry the old battery loose.

3) Remove the new battery from its package.

4) Grip the rim of the battery between your left thumb and index finger with the "+" side facing up.

5) Supporting the CoCo XT's battery holder with your right thumb, gently lift the spring metal clip with your with forefinger.

6) Insert the battery into the holder so that the "+" side is directly under the metal clip. The battery should be centered within the holder.

7) Gently release the metal clip. It should now make firm contact with the "+" side of the battery.

## 2.9 Replacing the Housing Top Cover

When you are ready to replace the housing top cover, perform these steps:

1) Position the top cover over the threaded spacers so that the holes in the top cover line up with the holes in the threaded spacers.

2) Make sure that the hard disk interface cables do not interfere with any of the threaded spacers that secure the top cover to the remainder of the housing.

3) Replace screws A, B, C, D, E, and F.

The assembled unit may now be installed in the CoCo Multi-PAK.

### 2.10 Multi-PAK Installation

The CoCo XT may be installed in any slot of your Multi-PAK. Many hard disk users install their hard disk controller in slot 3, the position normally used by the Tandy hard disk controller.

You must use a Multi-PAK, and not a "Y" cable, with the CoCo XT for two reasons:

- The hard disk controller requires +12 volts (10ma), which is not provided on the CoCo 2 or CoCo 3.
- The CoCo XT interface decodes the BIOS ROM on the hard disk controller into the address range \$C000-\$DFFF, which is the same address as the Disk BASIC ROM.

Many hard disk controllers have a jumper that disables the BIOS ROM. If you disable the BIOS ROM and provide 12 volts to the CoCo XT, it may be used without the

Multi-PAK.

To install the CoCo XT in your Multi-PAK:

- 1) Turn off all power to every part of your computer (including printers, disk drives, etc).
- 2) Grip the assembled CoCo XT with both hands, with the top cover away from you and the 40 pin edge connector pointing down.
- 3) Align the CoCo XT with the desired slot of the Multi-PAK. The housing should extend to the right.
- 4) Firmly insert the CoCo XT into the slot until the housing is resting about 1/4" above the top of the Multi-PAK.

If you have difficulty with step 4, make sure that the CoCo XT has been assembled correctly. Also, verify that the housing is properly aligned with the Multi-PAK. You may have to tilt the CoCo XT forward or backward slightly in order to get its 40 pin edge connector to mate with the socket in the Multi-PAK.

### 2.11 Connecting the Cables to the Drive

Once the CoCo XT has been installed in the Multi-PAK, you are ready to connect the interface cables to your hard drive.



### 3.0 OS9 Software Installation

The OS9 software supplied with the CoCo XT allows you to treat your hard disk like a very large floppy disk. The increased storage capacity of a hard disk lets you take full advantage of the OS9 tree-structured file system.

Imagine having instant access to all of your commands and utility programs without having to swap floppy disks. Imagine having a 100,000 record data base. All of these things are possible when you use the CoCo XT hard disk interface with OS9.

### 3.1 Summary of Installation Procedure

OS9 is a modular operating system. This type of operating system is easily expandable. To add something new to OS9 you just "describe" it to the operating system.

In OS9 jargon, a hard disk is a type of "device". There are several broad classes of devices, but hard disks, RAM disks, and floppy disks fall into a special class called "Random Block File" (RBF) devices. This class of devices can store data and programs.

OS9 needs to know 2 types of information before it will let you use your hard disk as an RBF device:

- How much information can the disk hold, and how is it organized?
- What kind of hard disk controller are you using?

The first question is answered by something called a "device descriptor". OS9 needs a device descriptor for each hard disk that you use, even if they are identical.

The second question is answered by a "device driver". OS9 only needs one hard disk device driver to be able to use any number of hard disks.

The steps that you must follow to tell OS9 about your hard disk are:

- 1) Choose a device driver. CoCo XT allows you to customize your system by selecting from several different device drivers.
- 2) Create a temporary device descriptor for formatting the hard drive. The DDMAKER utility included with CoCo XT allows you to do this quickly and easily.
- 3) Install the hard disk driver and descriptor in your OS9 boot file.
- 4) Format your hard disk. You do this with the OS9 FORMAT utility.

5) Create your final system configuration by modifying the temporary hard disk device descriptor with EZGEN.

6) Boot the new system.

These steps are not difficult. The sections ahead provide you with step-by-step instructions.

### 3.2 Choosing a Device Driver

The device driver is actually an assembly language program that OS9 runs whenever it needs to access the hard disk. As more features and capabilities are added to the driver program, it uses up more of the Color Computer's memory.

Burke & Burke supplies several device drivers on the CoCo XT (or CoCo XT-RTC) software release disk. You should select the driver that best meets your memory and performance requirements.

The capabilities of each driver are coded into its name according to the format below:

BB<system code><options><optimization>

All driver names begin with BB, which stands for Burke & Burke.

The possible system codes are:

Code	System Type
X	OS9 Level 1, Version 1
1	OS9 Level 1, Version 2
none	OS9 Level 2

The possible options are:

Code	Option Type
F	Formatting capability
T	Double-speed block transfers (Level 1, CoCo 3 only)

The possible optimizations are:

Code	Option Type
hdisk	Driver for one or two hard disks, any type.
hd	Driver optimized for one hard disk, 4 heads, 32 SPT

Not all possible combinations of drivers are provided. You may choose from the following device drivers:

BBXThd.dr	BBXhd.dr	BB1Thdisk.dr	
BB1hdisk.dr	BB1Fhdisk.dr	BBFhdisk.dr	
BBXThdisk.dr	BBXhdisk.dr	BBXFhdisk.dr	
BB1hd.dr	BB1Thd.dr	BBhd.dr	BBhdisk.dr

Level 2 users may choose between

BBFhdisk (1-2 drives with full formatting capability), BBhdisk (1-2 drives, no formatting), and BBhd (1 drive, 4 heads, 32 sectors per track, no formatting).

Many more options are provided for users of OS9 Level 1 due to the memory limitations of Level I machines. For example, if you use BB1Fhdisk on a Level I machine you may not have enough free RAM to run large programs like the OS9 "C" compiler.

### 3.3 Creating a Device Descriptor

The device descriptor tells OS9 what type of hard disk you have. OS9 defines a standard descriptor format for most of the needed information, and permits additional information to be specified at the end of the descriptor.

#### 3.3.1 CoCo XT Device Descriptor Format

The CoCo XT device descriptor differs slightly from the standard OS9 RBF device descriptor in order to incorporate additional features. The differences are:

Module Offset	Mnemonic	Function
\$0016	IT.DNS	Slot number - . (e.g 2 for slot #3)
\$0025	---	Precompensation cylinder / 4
\$0026-\$0028	---	LSN of first sector of park track

### 3.3.2 Running DDMAKER

The DDMAKER utility creates device descriptors. You describe your hard disk drive to DDMAKER by answering a few questions, and DDMAKER automatically creates a device descriptor file suitable for use by the OS9GEN or CONFIG system configuration utilities.

The steps for creating a device descriptor with DDMAKER are:

- 1) Copy DDMAKER from the CoCo XT software release disk to your CMDS directory.

- 2) At the OS9 prompt, type:

```
OS9: ddmaker hd
```

This starts the DDMAKER utility, and tells it to create a temporary device descriptor for a hard disk named "/hd". The descriptor will be stored in the file "hd.dd", in the current

commands directory.

3) Answer the questions that DDMAKER asks you.

4) Verify your answers. DDMAKER will not create a device descriptor file until you tell it that all questions have been answered correctly.

5) DDMAKER will create a file named "hd.dd" in your current commands directory.

The paragraphs below will help you to answer DDMAKER's questions. Each paragraph heading is a question that DDMAKER will ask you, and the accompanying text provides a more detailed description of both the question and possible answers.

The responses for many of DDMAKER's questions will be found on your hard disk data sheet. If you do not have a data sheet, Appendix A gives typical responses for several popular hard drives.

### 3.3.1 Enter name of device DRIVER module (e.g. BBFhdisk):

Respond by entering the name of the hard disk device driver that you have decided to use.

If you have not formatted your hard disk yet, answer this question as follows: BBFHDISK for level 2; BB1FHDISK for level

1, version 2; BBXFHDISK for level 1, version 1.

### 3.3.2 Slot # (1-4):

Respond by entering the number of the Multi-PAK slot in which your CoCo XT interface is installed.

### 3.3.3 Drive # (0 or 1):

Your response to this question indicates which hard drive to access when the descriptor name is used. If you have only 1 hard drive, you should always respond with "0".

If you have 2 hard drives, you must pick a unique name for each drive and decide for yourself which is "0" and which is "1". Then run DDMAKER twice, responding "0" for the 1st drive and "1" for the 2nd drive.

### 3.3.4 Step rate code (0-7):

The step rate code indicates how fast the mechanical parts in your hard drive operate. The hard disk controller uses this information to avoid giving instructions to the hard disk too quickly.

All hard disk controllers that are compatible with the CoCo XT understand the following step rate codes:

Code	Step Rate
0	3 milliseconds
4	200 microseconds
5	70 microseconds

Certain hard disk controllers understand additional codes as follows:

Code	Step Rate
1	45 microseconds
2	60 microseconds
3	18 microseconds
6	30 microseconds
7	18 microseconds

Consult your hard disk controller documentation before using any of the additional step rate codes.

Some hard drives have a feature called "buffered seeking". These drives accept instructions very quickly, and let OS9 know when they have finished everything. The data sheet for this type of drive usually gives a parameter called "minimum step pulse width". You must use a step rate code that is greater than or equal to the minimum that your drive will tolerate. Usually a step rate code of "4" will work with this type of drive.

You must use a step rate code of "0" if your drive does not support buffered seeking. This step rate is still about 10 times faster than the step rate of standard

CoCo floppy disk drives.

3.3.5 Tracks (cylinders) / surfac  
(1-1024):

Enter the number of tracks per surface indicated on your hard drive data sheet. For the ST-225 hard drive, a suitable response would be 612 tracks.

If you are using a Western Digital RLL controller, enter 1.5 times the actual number of tracks on your drive. For the Adaptec 2072 RLL controller, enter the actual number of tracks.

3.3.6 1st Write Precompensation Track:

Enter the precompensation track indicated on your hard drive data sheet.

The number you enter will be divided by 4 and stored in the device descriptor as an integer. The driver will multiply this integer by 4 to obtain the actual precompensation track. This loss of precision does not adversely effect hard disk operation.

If your hard drive does not require write precompensation, enter the number of tracks per surface, less 1.

For the ST-225 hard drive, a suitable response would be track 256.

3.3.7 Number of surfaces (heads) (1-16):

Enter the number of heads or surfaces provided by your drive. For the ST-225 hard drive, a suitable response would be 4 heads.

3.3.8 Number of 256 byte sectors / track (1-255):

If you are using a Western Digital hard disk controller, and the capacity of your drive is less than 70 Megabytes, enter "32" for the CoCo XT or CoCo XT-RTC. Use "48" if the capacity of your drive is more than 70 Megabytes (requires special jumpering).

If you are using the Adaptec 2072 RLL controller, enter 48.

3.3.9 Park track:

This is the track at which the heads should be "parked" during shipment. If you are not sure what your drive's park track is, use the number of tracks plus 1.

For the ST-225 hard drive, a suitable response would be track 670.

3.3.10 Verify writes (0=yes, 1=no):

Enter "1" if you want the default condition to be no verify; "0" if you want

the default condition to be verify.

Note that "hd" optimized drivers do not support write verification.

### 3.3.11 Minimum file segment size (1-255):

Unless you are creating a lot of very big files, Burke & Burke recommends that you specify "32" as the minimum segment size.

When OS9 needs to store information on the hard disk, it records the location of that information in something called a "segment list". Each file has its own segment list, which may have as many as 48 parts. Each part is called a "segment", and is a group of consecutively numbered disk sectors that are used by the file.

File size may be limited by the minimum number of sectors in a segment. If the minimum file segment size is 1 sector, it is possible that no file could be bigger than  $48 \times 1 \times 256 = 12,000$  characters -- about 5 pages of text. Of course, OS9 will try to use more than 1 sector per segment, but it can only do this if it is lucky enough to find adjacent sectors. The minimum segment size places an "absolute minimum" upper limit on file size.

The minimum segment size is also used whenever OS9 creates a new directory. The directory is always allocated the number of sectors given by the minimum segment size

Making the segment size large allows OS9 to store all directory entries in a single segment of the directory file, which can improve response time by several seconds.

Reducing the minimum segment size causes less space to be wasted on the hard disk. Increasing the minimum segment size allows you to have larger files and increases the efficiency of directory searches.

### 3.3.12 Enter format interleave factor:

This response indicates how fast your computer can process information from the hard disk.

Enter "9" if you are using a driver that does not have the "T" option under Level 1 OS9. Enter "22" if you are using "T" driver or a Level 2 OS9 driver.

### 3.3.13 Please verify drive n ('/hd') parameters:

DDMAKER will display all of the hard disk parameters that you have entered, and then prompt you with:

O.K. ([Y/N])?

If you find all of the parameters to be correct, enter "Y" and DDMAKER will create the device descriptor file. If you find any errors, enter "N" and DDMAKER will

exit. You may then run DDMAKER again to enter the correct parameters.

### 3.4 Boot File Installation

You must use OS9GEN or EZGEN to create a bootable disk that includes your device driver and descriptor before attempting to format the hard disk. You must use a driver with the "F" option for formatting, even if you do not want to use an "F" driver for normal operation.

Here is the OS9GEN procedure for Level 2 systems with 2 floppy drives:

```
OS9: OS9GEN /D1
      /d0/OS9BOOT
      /d0/cmds/BBFhdisk.dr
      /d0/cmds/hd.dd

OS9: mkdir /d1/cmds
OS9: copy #20K /d0/cmds/shell
      /d1/cmds/shell
OS9: copy #20K /d0/cmds/grfdrv
      /d1/cmds/grfdrv
OS9: copy #20K /d0/cmds/format
      /d1/cmds/format
```

This procedure assumes that a copy of your system disk is in drive /d0, and that both the hard disk driver and the descriptor created by DDMAKER are stored in directory /d0/cmds.

Level 1 users can follow this procedure:

```
OS9: OS9GEN /D1
      /d0/OS9BOOT
      /d0/cmds/BBIFhdisk.dr
      /d0/cmds/hd.dd
```

OS9:

### 3.5 Formatting your Hard Disk

The OS9 FORMAT utility is used to format your hard disk.

OS9 keeps track of what sectors are available on a hard disk by grouping the sectors into units called "clusters". CoCo Level 1 OS9 Version 1.00 can only keep track of 4096 clusters, so each cluster must contain several sectors. For example, Version 1.00 OS9 allocates 19 sectors per cluster for a 20 MB hard drive.

Beginning with Level 1, Version 1.01 OS9 there is no limit to the number of clusters on a hard disk. Level 2 OS9 also does not limit the number of clusters on a hard disk.

You must first gain access to the hard drive by booting the OS9 system that you created in Section 3.4. Then put a copy of your original OS9 System Disk in drive /d0, and enter:

```
OS9: format /h0
```

The OS9 formatter will ask you several questions, and will format the hard drive.

One of the questions the formatter ask is whether or not you want a Physical Verify of the hard drive. You should always answer "Y" to this question. Answering "Y" tells OS9 to find any hard drive defects and to mark them as unusable.

When the drive is completely formatted, you can test it out by copying some of your floppy disks to the hard drive. This will also help in later steps of hard disk installation, since the utility programs can be started directly from the hard disk instead of from floppy.

Burke & Burke recommends that you copy the OS9 system disk the CoCo XT software disk to the hard drive as follows:

```
(put copy of original OS9 system
disk in floppy drive 0)
OS9:chd /d0
OS9:chx /d0/xmde
OS9:load DSAVE
OS9:dsave -s20 /d0 /hd ! shell
.
.
OS9:
(put copy of CoCo XT disk in
floppy drive 0)
OS9:chd /d0
OS9:dsave -s20 /d0 /hd ! shell
.
.
OS9:
```

### 3.6 Creating your Final System

When you boot OS9, it reads a file called OS9BOOT from the floppy disk. This file includes all of the device drivers and descriptors that allow you to use printers, disks, serial ports, etc.

In section 3.4 you created a bootable OS9 hard disk system in order to format your hard drive. You can continue to use this system, but you may want to create a different system that includes a different device driver or uses different names for the hard drive.

If you wish to use a different device driver, you must run DDMAKER again and create a descriptor that specifies the new driver name. You can use EZGEN to install the new driver and descriptor.

Suppose you used BBFHDISK to format your hard drive, but in your final system you want to use BBHD and also want to name the hard drive "/h0". The EZGEN steps for these changes would be:

```
(put your hard disk boot floppy
in floppy drive 0)
OS9:chx /hd/cmds
OS9:ezgen -d /d0/os9boot
.
.
EZGEN> -link bbfdisk
EZGEN> -update /d0/modules/bbhd.dr
EZGEN> -link hd
EZGEN> -update /d0/cmds/h0.dd
EZGEN> -quit
.
.
OS9:
```

If you wanted to change the name of the hard disk from "/hd" to "/h0", but didn't need to change the device driver, you could use these EZGEN commands:

```
(put your hard disk boot floppy
in floppy drive 0)
OS9:chx /hd/cmds
OS9:ezgen -d /d0/os9boot
.
.
EZGEN> -link hd
EZGEN> -rename h0
EZGEN> -quit
.
.
OS9:
```

Many other custom configurations are possible. Refer to the OS9 and EZGEN manuals for further information.

#### 4.0 Using the CoCo XT with OS9

Once you have installed your hard disk hardware and software, the hard disk is instantly available to you whenever you boot OS9.

The fact that a hard disk holds very much more data than a floppy disk creates some remarkable opportunities as well as a few special problems. For example:

- What can you do to minimize the risk of damaging your hard disk?
- If you have the CoCo XT-RTC, how do you tell OS9 to use the real-time clock?
- How can you get OS9 to look for commands on the hard disk rather than a floppy disk?
- What do you do with millions of characters of on-line storage?
- How do you keep track of everything that you store on the hard disk?

This section will give you the answers to these questions and others that you may have about using your hard disk.

#### 4.1 Protecting Data with PARK

A hard disk is a precision instrument. When your hard disk is operating, the read-write heads ride above the magnetic recording media on a cushion of air that is only a fraction of an inch thick. If this air cushion were not present, the heads might scrape against the rotating disk media -- destroying the hard disk (and even worse, your precious data) in the process.

The scraping action is sometimes called a "head crash", since the heads of the disk essentially "crash" into the media. Most head crashes occur during either equipment transportation or sudden power failures.

This section discusses steps that you can take to protect your valuable investment from head crashes.

#### 4.1.1 Equipment Transportation

When you turn off your hard disk, the protective air cushion goes away. Most hard disks provide a special area of the disk (a "parking track") that may be used to protect the heads when no air cushion is present.

When a hard disk is transported -- whether across the country or just across the room -- without first positioning the heads at the parking track, it is subjected to mechanical shocks and vibrations strong enough to cause a head crash. YOU CAN PROTECT YOUR HARD DISK BY ALWAYS PARKING

## THE HEADS BEFORE MOVING THE DISK.

Some hard disks automatically park their heads when power is removed. This feature is called "automatic parking".

If your hard disk does not automatically park its heads, you can force the heads to park by using the PARK utility provided with the CoCo XT.

If your hard disk is named "h0", you can park it with the command:

```
OS9: park /h0
Drive '/h0' has been parked
```

If you have several hard disk drives, you should LOAD the park command before executing it. Otherwise one of the drives may become unparked when OS9 attempts to load in the PARK command from disk.

Many experienced computer users routinely park their disks every time they turn off the computer. This is a good practice that Burke & Burke strongly recommends.

### 4.1.2 Power Failures

Power failures (or sudden power surges) may cause the heads of your disk to crash. Most auto-parking disks are protected during power fluctuations, but disks without the automatic parking feature may be damaged.

You don't usually expect a power failure, so you don't have time to run the PARK utility. If you really want to protect your system from power failures, consider purchasing an "uninterruptable power supply" (UPS).

A UPS is about the size of an overnight bag. It contains a battery-operated power converter that can keep your computer running for about 15 minutes even when your building's electrical circuits are dead. This gives you plenty of time to run the PARK utility and shut down your system. At this writing, suitable UPS's are available from a number of manufacturers for under \$300.

## 4.2 Keeping Track of Real Time

The CoCo XT-RTC software release disk includes three utilities that make it simple to use your real-time clock.

### 4.2.1 CLKGET

The first utility is called CLKGET. This utility reads the correct time from your CoCo XT-RTC and copies it to OS9's system clock.

The OS9 system clock is updated automatically by a software routine that is built into OS9. If you do a lot of floppy disk I/O, OS9 can miss the opportunity to

run this routine. This will cause your system clock to "lose" time. The real-time clock continues to keep the correct time regardless of floppy disk I/O; you can always get the correct time back by running CLKGET.

The system clock does not lose time during CoCo XT hard disk accesses.

A typical application for CLKGET is to store the command line:

```
clkget;date t
```

in your OS9 "startup" file. This will automatically set the system time from the real-time clock, and display it, whenever you boot OS9.

You can also use clkget to automatically correct your system time every few minutes. To do this, use the commands:

```
clkget -d &  
date t
```

in your OS9 "startup" file.

#### 4.2.2 CLKSET

The second utility, CLKSET, is the inverse of CLKGET. This utility lets you set your CoCo XT-RTC real-time clock to the current system time.

You will not have to run CLKSET ver often, since the on-board lithium battery allows the clock to keep accurate time for 5-10 years. Here are a few conditions during which you might wa

- When you first install the lithium battery on the CoCo XT-RTC circuit board.
- Twice per year to adjust for daylight savings time.
- If you move to a different time zone.

Otherwise, the clock will just keep on ticking!

The normal procedure for setting your real-time clock is to set the system time via the OS9 SETIME utility, and then to use CLKSET to copy this time into the real-time clock.

#### 4.2.3 RTC

RTC is the third clock utility provided with the CoCo XT- utility displays the correct time and date to you by reading it directly from the CoCo XT-RTC's real-time clock.

The output of RTC has the following format:

```
yy/mm/dd HH:MM:SS
```

You can use RTC to determine whether or not the system clock has lost time by issuing the OS9 command:

```
OS9: rtc;date t
```

This will cause OS9 to display the actual time stored in the real-time clock, followed by the current system time and date.

### 4.3 Managing Directory Trees

OS9 allows you to create layer after layer of sub-directories on a hard disk or floppy disk. The entire set of sub-directories may be thought of as a tree, with each sub-directory representing a branch.

You will find your hard disk much easier to use if you take care to grow it's "directory tree" in a way that will be manageable now and in the future -- especially if you are running a multi-user system.

This section presents a few techniques that the experts use to manage their hard disks. You don't have to follow these guidelines, but they have been used successfully on mainframe computers for at least 15 years.

#### 4.3.1 Keep the Root Clean

The highest level of directory on your hard disk is called the "root" directory. All sub-directories and data files on the hard disk can be found from the root.

Many first-time hard disk users tend to store data or programs in their root directory. This works pretty well for floppy disks, but can get out of control on a hard disk.

Experience has shown that only a few essential data files (e.g., "startup") should be stored in the root directory. Everything else should be stored in sub-directories.

#### 4.3.2 Files and Sub-Directories Don't Mix

Another common tendency is to store sub-directories, programs, and data files all in the same directory. This can make it difficult to determine whether a name corresponds to a file or a nested sub-directory.

Managing your hard disk will be an easier task if you put only one type of object (e.g., programs, data files, or sub-directories) in each sub-directory.

If you must mix different types of files in a directory, be sure to follow the OS9 convention of capitalizing directory names and using lower case for file names.

### 4.3.3 Divide and Conquer

If you have several distinctly different types of files, it is a good idea to put all files of the same type in a single sub-directory. For very broad classes of files you should put this sub-directory in the root; for more limited classes the sub-directory may be nested.

Here is a typical "good" root directory for a single-user system:

startup	CMDS	SRC	USR
DEFS	LIB	ASM	

This directory is very clean, except for the "startup" file. Note that the upper case / lower case naming convention has been followed.

If you looked at nested levels of the root's sub-directories, you would discover the following:

- The CMDS directory contains only executable OS9 commands.
- The SRC directory contains several layers of nested sub-directories. Ultimately we would reach a set of sub-directories that contain only "C" source files (1 complete "C" program per sub-directory).
- The USR directory contains a few data files that are needed to

convert the system to multi-user mode. These are normally not used.

- The DEFS directory contains only data files that provide information about OS9 to all "C" and assembly language programs.
- The LIB directory contains several layers of nested sub-directories. Ultimately we would reach a set of sub-directories that contain only common libraries for assembly or "C" or language programs.
- The ASM directory contains several layers of nested sub-directories. Ultimately we would reach a set of sub-directories that contain only assembly language source files (1 complete assembly language program per sub-directory).

#### 4.3.4 Multi-User Systems

It is much easier to set up a multi-user system on a hard disk than on a floppy disk. It can become difficult to manage a multi-user system if it's directory tree is not set up carefully.

One effective technique for organizing hard disk files in a multi-user system is to create a separate sub-directory for each authorized user. All of these sub-directories are located in the hard disk's USR directory. Every user's

directory looks very much like the root directory of a single-user system.

Here is a typical "good" root directory for a multi-user system:

```
startup      CMDS      USR
DEFS         LIB
```

It is important to note that the root directory of a multi-user system is similar to the root directory of a single-user system, but there are no SRC or ASM directories.

If you looked at nested levels of this root's sub-directories, you would discover the following:

- The CMDS directory contains only executable OS9 commands. These commands are shared by all users.
- There is no SRC directory.
- The USR directory is significantly different from that of a single-user system. There are a few data files with names like "motd" and "passwd", but there is also a sub-directory here for each authorized system user.
- The DEFS directory contains only data files that provide information about OS9 to all "C" and assembly language programs. The information in this directory is available to

all users.

- The LIB directory contains several layers of nested sub-directories. Ultimately we would reach a set of sub-directories that contain only common libraries for assembly or "C" language programs. This information is available to all users.
  
- There is no ASM directory.

Let's take a closer look at theUSR directory of this multi-user system:

motd	passwd	ROOT	LARRY
MOE	CURLY	GUEST	

You can see a directory for each authorized user. This system has five authorized users, named ROOT, LARRY, MOE, CURLY, and GUEST.

Now lets look in MOE's sub-directory:

startup	CMDS	SRC
DEFS	LIB	ASM

Does this look familiar? It's the same as the root directory of a single-user system (without the USR directory).

You would see a similar listing if you looked in any of the other user's sub-directories. It is important to note that each user's directories are completely independent of every other user. For

example, CURLY's "SRC" directory is stored at a completely different place on the hard disk than LARRY's "SRC" directory, and these directories contain completely different source files.

#### 4.4 Customizing your System

This section presents a few tips and "tricks" that might help you to make better use of your hard disk system if you are very familiar with OS9.

Many of these tips involve making permanent modifications to OS9, and should not be attempted by beginners.

##### 4.4.1 Changing the Default System Device

When you boot OS9, it determines the initial settings for your working directory and execution directory by consulting a "system configuration table". OS9 Level 1 is shipped with "/d0" and "/d0/cmds" stored in the configuration table.

The system configuration table is stored as an OS9 module named INIT. You can change the initial directory settings to use a drive other than "/d0" by modifying the INIT module in your OS9BOOT file. OS9 will then automatically set your default directories to the hard disk every time you boot.

To make this change, follow the

procedure below. The procedure assumes:

- Your boot disk is in floppy drive "/D0"
- The EZGEN utility is stored in the cmds directory on the hard disk
- The current execution directory is the hard disk cmds directory.
- You want to change the initial OS9 directories to "/h0" and "/h0/cmds".

If your system is configured differently, you will have to modify the procedure accordingly.

(place hard disk boot floppy in drive "/d0")

```
OS9:ezgen -kd /d0/os9boot
```

```
.
```

```
.
```

```
EZGen> -link init
```

```
EZGen> -rename /h0 10
```

```
EZGen> -quit
```

```
.
```

```
.
```

```
OS9:
```

In the example above, the LINK command tells EZGen which module you want to modify (in this case, INIT). The RENAME command indicates that you want to change a name within this module; the "10" at the end of the command tells EZGen where in the module

to find a pointer to the name.

For the INIT module, offset 10 is the pointer to the name of the system device. Other offset values can be used to change other system characteristics. For example, using an offset of 12 will change the name of the "console" device -- normally /TERM. These offsets apply only to the INIT module.

Note that some OS9 programs (like the "C" compiler) have names like "/d0" or "/d1" built into the program. These programs will not be able to find certain information that is on your hard disk unless you modify them to refer to "/h0" instead.

#### 4.4.2 The Hard Disk as "/D0"

You can overcome most of the difficulties associated with "built-in" program parameters by setting up your hard disk as drive "/d0".

When you buy OS9, it is set up to use "d0" as the name of floppy disk #0. You cannot name your hard disk "d0" until you give floppy disk #0 a new name.

Here is a summary of the procedure for setting up your hard disk as drive "d0":

- 1) Change the name of floppy disk "d0" to something else, like "f0", by using EZGen.

2) Change the name of your hard disk from whatever it was (e.g., "h0") to "d0" -- again by using EZGen.

To make this change, follow the procedure below. The procedure assumes:

- Your boot disk is in floppy drive "/D0"
- The EZGEN utility is stored in the cmds directory on the hard disk
- The current execution directory is the hard disk cmds directory.
- You want to change the name of your hard drive from "/h0" to "/d0".

If your system is configured differently, you will have to modify the procedure accordingly. Note that the changes are not effective until the next time you boot your system.

If you have changed the name of the default system device to something other than "/d0", you must change it back to "d0" before performing this procedure.

When you have finished the procedure, your hard disk will be called "d0" and your floppy disk will be called "f0".

(place hard disk boot floppy in

```
drive "/d0")
OS9:ezgen -d /d0/os9boot
.
.
EZGen> -link init
EZGen> -rename /d0 10
EZGen> -link d0
EZGen> -rename f0
EZGen> -link h0
EZGen> -rename f0
EZGen> -quit
.
.
OS9:
```

#### 4.4.3 The Hard Disk as "/DD" (Level 2)

Many programs written for OS9 Level 2 use a "built-in" drive name of "DD". As shipped, Level 2 OS9 treats "DD" as another name for floppy disk #0. Level 2 users may want to change this so that the hard disk is used as both "DD" and "H0".

If you already have a floppy disk "/DD" descriptor, you must replace it with a hard disk "/DD". Otherwise, you can just add the "/DD" descriptor to your boot. The procedure in this section works for either case. It assumes that your hard disk is named "/H0".

```
(place hard disk boot floppy in
drive "/d0")
OS9:ezgen -d /d0/os9boot
.
```

```
.  
EZGen> * Do these 2 steps only if  
EZGen> * you already have a /DD  
EZGen> -link dd  
EZGen> -delete  
EZGen> * Always do these steps  
EZGen> -link h0  
EZGen> -save h0  
EZGen> -rename dd  
EZGen> -insert h0  
EZGen> -$del h0  
EZGen> -quit  
.  
.  
OS9:
```

#### 4.5 Hard Disk Backups

Hard disks are a very reliable information storage medium, but it is wise to make a complete backup of your hard disk at least once per month.

Burke & Burke provides a simple hard disk backup utility with the CoCo XT hard disk interface. This utility is called HDB.

HDB will copy all of the allocated sectors on your hard disk out to multiple floppy disks. Information is saved to the floppies in a manner that allows the HDR hard disk restoration utility to replace it EXACTLY as it was originally stored on the hard disk. This means that special files like OS9Boot and HYPER-I/O mass-storage areas will be preserved intact.

To use HDB, enter the command:

```
OS9: hdb /h0 /d0
```

You can substitute whatever device names you like; this example will back up hard drive /h0 to floppy drive /d0.

The HDR utility copies information from floppy disks created with HDB back to a freshly formatted hard drive. HDR will destroy any other information stored on the hard drive.

To use HDR, enter the command:

```
OS9: hdr /d0 /h0
```

You can substitute whatever device names you like; this example will restore hard drive /h0 from floppy drive /d0.

Other hard disk backup utilities are faster or easier to use than HDB, but may "lose" the OS9 boot file or other special files. If you use Burke & Burke's XT-ROM boot ROM with your hard disk, be sure to save the hard disk boot to a floppy (using BootPort) before using any non-HDB backup utility.

Vertical line of text on the right side of the page.

## 5.0 Using the CoCo XT with Disk BASIC

The CoCo XT and CoCo XT-RTC Color Computer hard disk interfaces are designed for compatibility with both OS9 and Color Disk BASIC.

Color Disk BASIC alone does not support hard disks. Burke & Burke has developed a special program called HYPER-I/O that allows you to use many different types of device, including the CoCo XT hard disk interface, with Disk BASIC. HYPER-I/O is not included with the CoCo XT interface, but is available from Burke & Burke.

The HYPER-I/O user's manual gives more information about HYPER-I/O and how to configure your system. The sections below provide you with information specific to using HYPER-I/O with the CoCo XT.

### 5.1 What is HYPER-I/O?

HYPER-I/O is a machine language program that modifies BASIC to provide a "dynamic disk interface". This software interface makes it very easy to expand HYPER-I/O by installing device drivers for hard disks, RAM disks, high-density floppies, or other types of mass storage devices.

HYPER-I/O works with versions 1.0, 1.1, 2.0, and 2.1 of Tandy Disk Extended Color BASIC, and with CoCo 2 or CoCo 3 CDOS. You

can use HYPER-I/O on the CoCo 1, CoCo 2, or CoCo 3.

You can have HYPER-I/O burned into a 27128 EPROM along with BASIC, making all of HYPER-I/O's features available the instant your computer is turned on.

The HYPER-I/O HARD DISK INSTALLATION MANUAL provides valuable information about configuring your hard disk for use with BASIC. The information that is unique to the CoCo XT hard disk interface is repeated here for convenience.

## 5.2 The "XT" Device Driver

HYPER-I/O is shipped to you with a Burke & Burke hard disk driver already installed. This driver is named "XT", and is stored in a file called "XT.DR" on the BASIC portion of the CoCo XT software disk.

If you ever need to replace the XT device driver with a newer version, you must follow these CONFIG steps:

- 1) Save all of your device descriptors to disk.
- 2) Remove all of your hard disk device descriptors.
- 3) Remove the old "XT" device driver.
- 4) Install the new "XT" device driver

using CONFIG.

5) Reinstall the hard disk device descriptors

### 5.3 The Hard Disk Device Descriptor

HYPER-I/O is shipped to you with device descriptors for a Seagate ST-225 hard drive already installed. These descriptors are named "/H0" through "/H3".

If you are using a 20 Meg hard disk, and your CoCo XT interface is installed in Multi-PAK slot #3, the pre-installed device descriptors will let you get up and running immediately. Otherwise you must edit the device descriptors to tailor them to your system.

The XT device driver handles all types of hard disks that are compatible with the CoCo XT interface. The driver uses a table of information (called a device descriptor) to determine the unique characteristics of your hard disk. You must fill in a hard disk device descriptor before HYPER-I/O will allow you to access the hard disk.

The HYPER-I/O manual includes a detailed discussion of device descriptors. Follow the descriptor editing instructions in the CONFIG section of the HYPER-I/O user's manual. You can find the correct values for most of the fields of the device descriptor on the data sheet supplied with your hard disk. If you do not have a data

sheet, consult Appendix A for the parameters of many popular hard disks.

The PHY DEV field should be set to 0 for your first hard drive and to 1 for your second hard drive.

The INDEX DATA LSN field of the device descriptor should be set to \$000000. The MSATool utility will change this value automatically when it creates an "index file" on your hard disk. If you do not specify \$000000, MSATool will attempt to create the index file using the specified LSN (logical sector number).

The DRIVER OPTION BYTE of the device descriptor tells the XT driver the Multi-PAK slot in which the CoCo XT interface is installed. Set this byte to the slot number minus one (e.g. use a DRIVER OPTION BYTE of 2 if your CoCo XT interface is installed in slot #3).

Each HYPER-I/O device descriptor can be used to create up to twenty-six identically sized Mass Storage Areas. You can create several different sizes of MSA's on your hard disk by using a separate device descriptor for each size.

#### 5.4 Formatting a Hard Disk

If your hard disk is already formatted for use with OS9, you may use it from HYPER-I/O without performing any low-level or logical formatting. The OS9 hard disk

format is 100% compatible with HYPER-I/O.

FORMATTING YOUR HARD DRIVE FROM HYPER-I/O WILL ERASE ANY OS9 FILES ALREADY STORED ON THE DRIVE.

If your hard disk is already formatted for use with a PC compatible computer, you do not need to perform a low-level format. You will still need to perform a logical format using the HYPER-I/O HDFMT utility.

#### 5.4.1 Low-Level Formatting

Low-level formatting is the process of dividing the hard drive into tracks and sectors. This process erases all data stored on the hard disk and prepares it for use in your computer system.

The XT device driver does not include hard disk formatting capability. Burke & Burke provides a separate low-level format utility called XTFMT on the CoCo XT release disk.

The XT device driver must be installed, and at least one device descriptor must be installed for each hard drive that you wish to format, before you can run XTFMT.

To perform a low-level format:

1) Insert a backup of your CoCo XT release disk in drive 0 and enter:

RUN "XTFMT"

2) The prompt, "ENTER DEVICE NAME (/..):" will appear. Enter the device descriptor name of the hard disk that you want to format, for example:

/H0

Note that if you have several device descriptors for the same hard disk, you only need to specify one of the names in order to format the entire disk.

3) The prompt, "ENTER INTERLEAVE FACTOR:" will appear. You should respond with:

11

for optimum performance.

4) The formatting operation takes about 5 minutes for a 20 megabyte hard drive. When low-level formatting is complete, BASIC's OK prompt will appear.

#### 5.4.2 Logical Formatting

Logical formatting is the process of organizing the hard disk into groups of sectors that OS9 and HYPER-I/O can allocate to data files. This process deletes all data files that may already be stored on the disk.

The logical format operation is

performed by HYPER-I/O's HDFMT utility. To perform a logical format:

1) Insert your HYPER-I/O work disk in drive 0 and enter:

```
RUN "HDFMT"
```

2) The prompt, "ENTER DEVICE NAME (E.G. /H0):" will appear. Enter the name of a device descriptor for the hard disk that you want to format, for example:

```
/H0
```

Note that if you have several device descriptors for the same hard disk, you only need to specify one of the names in order to format the entire disk.

3) HDFMT will display its status to you as it performs the logical format. It will then ask you for a "VOLUME NAME:" You may enter any 0-32 character string, which will be stored in the first sector of the hard drive in order to allow you to identify the drive in the future. Respond with [ENTER] if you don't care about the volume name.

4) When logical formatting is complete, BASIC's OK prompt will appear.

### 5.5 Creating Mass Storage Areas

You can use the HYPER-I/O MSATool

utility to create multiple Mass Storage Areas in which to store BASIC and machine language programs on your hard disk. Consult the HYPER-I/O user's manual for instructions.

### 5.6 Protecting Data with UNLOAD

HYPER-I/O modifies BASIC's UNLOAD command. The HYPER-I/O version of this command closes all open files on the active MSA and then "parks" the hard disk. This protects the delicate read/write heads and polished recording surfaces of your hard disk from damage during shipping or transportation.

If you ever need to transport your hard disks, be sure to park them with the UNLOAD command before turning off the computer.

Many experienced computer users routinely park the heads of their hard disks each time they turn off their computer.

If you have two hard drives, named "H0" and "H1", you can park both of them with the sequence:

```
OPEN DRIVE 1, "/H0":UNLOAD1  
OPEN DRIVE 1, "/H1":UNLOAD1
```

Be sure not to enter any other commands that access the hard drives before turning off the computer. Any access to the hard drive will unpark the heads.

See the description of the OS9 PARK command in the OS9 section of this manual for further information on disk head parking.

### 5.7 The CoCo XT-RTC Real-Time Clock

A library of BASIC compatible machine language routines is provided with the CoCo XT-RTC. These routines allow you to read and set the CoCo XT-RTC's real-time clock.

The CoCo XT-RTC disk also includes source code for routines that access the real-time clock. You can edit the file CLKIO.SRC to create your own clock I/O routines for custom software.

The library routines are stored in a set of files called "RTCLIBx.BIN". The file that you use depends on the slot used by your CoCo XT-RTC interface:

Slot #	Library File
-----	-----
1	RTCLIB1.BIN
2	RTCLIB2.BIN
3	RTCLIB3.BIN
4	RTCLIB4.BIN

The library files are completely relocatable, and require 512 bytes of machine language program storage. To load the library at a particular address, include the command:

LB=xx:LOADM "RTCLIB3",LB

where xx is the desired load address.

You access the library routines via BASIC USR function calls. The sample program at the end of this section shows how to use the library routines to provide a continuous time and date display on the CoCo's screen.

The library contains the following routines:

CLKGET	Returns current date and time to a string variable
CLKSET	Sets date and time from a string variable
BGTCLK	Reads current date and time to a buffer
BSTCLK	Sets date and time from a buffer

Each of these functions is described in detail below. The descriptions assume that the BASIC variable LB has been set to the load address of the library.

#### 5.7.1 CLKGET (LB+0)

This routine returns the current date and time in a 17 character BASIC string variable. The format of the returned string is:

MM/DD/YY HH:NN:SS

where MM=month, DD=day, YY=year, HH=hour, NN=minute, SS=second.

The STRING\$ function should be used to define a string variable for use by CLKGET, since assigning a 17 character literal to the string variable will cause the string to be allocated inside of the BASIC program rather than in BASIC's string table. If the string is stored in the BASIC program, the program will be modified every time CLKGET is called.

CLKGET is called using the sequence:

```
LOADM      "RTCLIB3.BIN"  
.  
.  
.  
DEFUSRO = LB+0  
LET AS  = STRING$(17,32)  
LET X   = VARPTR(AS)  
LET Y   = USRO(X)
```

The value of Y will be 0, and the date and time string will be in AS, after successfully executing this sequence. If Y is not zero, CLKGET could not return the time for some reason.

#### 5.7.2 CLKSET (LB+3)

This routine sets the date and time from a 17 character BASIC string variable. The format of the string is:

MM/DD/YY HH:NN:SS

where MM=month, DD=day, YY=year, HH=hour, NN=minute, SS=second.

CLKSET is called using the sequence:

```
LOADM      "RTCLIB3.BIN"
DEFUSR1 = LB+3
.
.
.
LET A$ = "03/05/88 12:37:20"
LET X = VARPTR(A$)
LET Y = USR1(X)
```

The value of Y will be 0, and the original date and time string will still be in A\$, after successfully executing this sequence. If Y is not zero, CLKSET could not set the time for some reason.

### 5.7.3 BGTCLK (LB+6)

This routine allows machine language programs to read the real-time clock. BGTCLK reads the current time into the buffer pointed to by the 6809 microprocessor's X register. All 6809 registers are preserved.

Call this function via the sequence:

```
ldx      #TIMBUF
jsr      >(LB+6)
```

At the end of the call the 9 byte buffer

will contain the following information:

Address	Data	Range
TIMBUF+0	BCD 1/100 sec	(\$00-\$99)
TIMBUF+1	BCD seconds	(\$00-\$59)
TIMBUF+2	BCD minutes	(\$00-\$59)
TIMBUF+3	BCD hour	(\$00-\$23)
TIMBUF+4	---	
TIMBUF+5	BCD date	(\$01-\$31)
TIMBUF+6	BCD month	(\$01-\$12)
TIMBUF+7	BCD year	(\$00-\$99)
TIMBUF+8	scratch variable	

#### 5.7.4 BSTCLK (LB+9)

This routine allows machine language programs to set the real-time clock. BSTCLK sets the time from the buffer pointed to by the 6809 microprocessor's X register. All 6809 registers are preserved.

Call this function via the sequence:

```

*   Be sure to set up TIMBUF
*   ahead of time!
ldx  #TIMBUF
jsr  >(LB+9)
    
```

Before the call the 9 byte buffer must contain the following information:

Address	Data	Range
TIMBUF+0	BCD 1/100 sec	(\$00-\$99)
TIMBUF+1	BCD seconds	(\$00-\$59)

TIMBUF+2	BCD minutes	(\$00-\$59)
TIMBUF+3	BCD hour	(\$00-\$23)
TIMBUF+4	---	
TIMBUF+5	BCD date	(\$01-\$31)
TIMBUF+6	BCD month	(\$01-\$12)
TIMBUF+7	BCD year	(\$00-\$99)
TIMBUF+8	scratch variable	

### 5.7.5 Sample #1: Setting the Real Time

This program is stored on your CoCo XT-RTC release disk as "SETIME.BAS".

```
10 '
20 '   SET THE COCO XT-RTC REAL-TIME CLOCK
30 '
40 CLEAR 255,&H7DFF:      'ROOM FOR LIB
50 LB = &H7E00
60 LOADM "RTCLIB3.BIN",LB: 'SLOT 3 LIBRARY
70 DEF USRO = LB+3:      'DEFINE CLKSET
80 LINE INPUT
   "ENTER TIME (YY/MM/DD HH:MM:SS): ";A$

90 X = VARPTR(A$): 'ADDRESS OF TIME STRING
100 Y = USRO(X):    'SET THE TIME
110 IF Y<>0 THEN PRINT
   "ERROR SETTING TIME":GOTO 80

120 END
```

Note that this program does not check the user's input, and is designed for use when the CoCo XT-RTC is installed in Multi-PAK slot #3.

### 5.7.6 Sample #2: Displaying the Time

This program is stored on your CoCo XT-RTC release disk as "SHOWTIME.BAS".

```
10 '
20 '   DISPLAY OF COCO XT-RTC REAL-TIME
30 '
40 CLS: CLEAR 255, &H7DFF:   'ROOM FOR LIB
50 LB = &H7E00
60 LOADM "RTCLIB3.BIN", LB: 'SLOT 3 LIBRARY
70 DEF USR1 = LB+6:         'DEFINE CLKGET
80 A$=STRING$(17,32):       'CREATE STRING
90 X = VARPTR(A$):         'ADDRESS OF TIME STRING
100 Y = USR1(X):           'GET THE TIME
110 PRINT @14, A$;
120 GOTO 80
```

It is very important that you recalculate the VARPTR of A\$ every time you call any library routine. This is necessary because BASIC may attempt to save memory by changing the value of the VARPTR without telling you.

You can try changing the program as follows:

```
80 A$="--/--/-- --:--:--"
```

This change will cause the current time to be stored IN LINE 80 each time CLKGET is called. To see what we mean, run the modified program and then LIST line 80.



## A.0 Technical Summary

This appendix gives you an overview of how a CoCo XT hard disk system operates.

The table at the end of the Appendix gives typical device descriptor values for several popular hard drives.

## A.1 Theory of Operation

Your CoCo XT hard disk system has four major components:

- CoCo XT or CoCo XT-RTC hard disk interface board
- Hard disk controller board (WD1002-WX1 or equivalent)
- One or more hard disk drives
- Hard disk cabinet and power supply

These components are explained briefly in the sections below.

### A.1.1 CoCo XT Hard Disk Interface

This circuit board converts the Color Computer's 40 interface signals into the 62 signals necessary to drive a PC compatible hard disk controller.

Both the CoCo XT and the CoCo XT-RTC circuit boards include bus conversion logic. The CoCo XT-RTC also includes a battery holder, clock circuit, and quartz watch crystal.

The interface board allows the CoCo to access the hard disk controller at addresses \$FF50-\$FF53. This address range is selected by decoding SCS\* = 0 with A4=1.

The BIOS EPROM on the hard disk controller is decoded into the address range \$C000-\$DFFF. Note that address lines A4, A5, and A6 are inverted into the EPROM.

#### A.1.2 Hard Disk Controller Board

Hard disk controllers used with the Burke & Burke system must have both a 62-contact edge connector and the ST-412/ST-506 hard disk interface. The edge connector plugs directly into the 62 position socket on the CoCo XT interface card; the combined interface card and controller then plug directly into the Multi-PAK's 40 position socket. The height of the combined interface card and hard disk controller is almost exactly the same as the height of the original Tandy floppy disk controller.

The recommended hard disk controller boards for use with the CoCo XT interface

are:

- Western Digital WD1002-WX1  
(1/3 size card, MFM encoding)
- Western Digital WD1002-27X  
(1/3 size card, RLL encoding)
- Data Technology DTC5150-CRH  
(1/3 size card, MFM encoding)
- Data Technology DTC5160-CRH  
(1/3 size card, RLL encoding)
- Adaptec 2072  
(1/3 size card, RLL encoding)

Several other hard disk controllers are compatible with CoCo XT hardware and software, but will not mount in the CoCo XT housing and may exceed the capacity of the Multi-PAK power supply:

Western Digital WD1002S-WX2  
Western Digital WDXT-GEN,  
OMTI 5520A, 5527A  
Everex 391

When purchasing a hard disk system, please be sure that you are getting one of the recommended controllers.

A complete description of the hard disk controller logic is beyond the scope of this manual. The decoding of the hard disk controller registers is summarized below:

Address	Read	Write
---------	------	-------

-----	-----	-----
\$FF50	Data	Data
\$FF51	Status	Force reset
\$FF52	Read jumpers	Start command
\$FF53	Not used	Select mode

The hard disk controller requires both +5 and +12 volts for proper operation.

The hard disk controller converts commands from the Color Computer microprocessor into signals which control a hard disk. The controller also converts data from the Color Computer into an encoded format suitable for magnetic storage.

Two types of data encoding are commonly used: MFM and RLL. MFM is the older and more reliable encoding, and is used with most hard drives in the 5-20 megabyte range. RLL is a more modern technique that stores 50% more data than MFM on the same size hard disk. Since more bits are stored per square inch, a defect or failure on an RLL drive may destroy more data than a similar failure on an MFM drive.

Burke & Burke does not recommend that you use an RLL controller on a drive that was not specifically designed for RLL. On drive that has been designed for RLL is the ST-238R, with a capacity of 30 megabytes.

All hard disk controllers used with the CoCo XT interface are "no halt" controllers. These controllers do not use interrupts and do not slow down the Color

Computer during hard disk access. Data to be transferred between the CoCo and the hard disk is stored in an internal 2K RAM buffer that allows the hard disk and computer to operate at different data transfer rates.

The hard disk controller also includes a "BIOS ROM". This ROM stores a PC-compatible hard disk driver and formatting utility that are not used on the Color Computer. The CoCo XT interface allows the Color Computer to execute instructions out of an EPROM that replaces the BIOS ROM. For example, Burke & Burke's XT-ROM product can be installed to provide an automatic OS9 boot from the hard disk.

The controller will operate any hard disk that has an industry standard "ST-506" or "ST-412" connection. These connections use a 34 wire control cable and a 20 pin data cable that exit from the rear of the hard drive. The maximum data transfer rate for this type of interface is over 600,000 characters per second, but the speed of the CoCo limits actual operation to about 25,000 characters per second under OS9 Level II.

Hard disk controllers used with the CoCo XT read and write 512 bytes at a time from the hard disk. The CoCo XT OS9 device driver packs two 256 byte "logical sectors" into each 512 byte "physical sector" using a caching technique. Whenever an even numbered logical sector is read, the device driver saves the odd numbered sector data and returns it automatically on the next

hard disk access, without accessing the hard drive, if possible.

Other Color Computer hard disk interfaces are designed to be used with SCSI or SASI hard disk controllers. This type of hard disk controller is not compatible with the CoCo XT. The CoCo XT is significantly faster than SASI and SCSI interfaces, which operate at about 15,000 characters per second.

### A.1.3 Hard Disk Drive

The hard disk drive is the heart of your hard disk system. It consists of a stack of rigid aluminum disks. These disks are separated by enough space to allow a small coil of wire (called a read/write head) to slide back and forth between them. There is one read/write head for usable disk surface.

There is a thin magnetic coating on one or both sides of each disk. The quality of this coating determines the number of data bits that can be stored per square inch of disk surface. Drives certified for use with RLL controllers use a higher quality of coating than MFM-only drives.

The disks are mounted to a drive shaft that spins them at 3,600 RPM (60 rotations per second). All of the read/write heads are connected to a servo or stepping motor that can slide them along the radius of the disks.

Data can be stored on both sides of a disk. Each usable side is called a "surface". The disk itself is called a "platter".

Each surface is divided into a number of ring-shaped "tracks". The width of each ring determines how many tracks the drive has. This width is determined by the quality of the stepping motor or servo, since the read/write heads must be positioned precisely over each track.

Each surface is also divided into a number of wedge-shaped "sectors". The number of sectors depends on the type of data encoding used on the drive, and the number of bytes per sector. There are normally 16 sectors of 512 bytes each when using MFM encoding, and 24 sectors of 512 bytes each when using RLL.

The location of any 256 byte block of data on the hard disk can be described by a track number, surface number, sector number, and whether the odd or even half of the sector is used.

When accessing the hard drive, the heads are first positioned over the correct track. Then the head corresponding to the correct surface is activated. Since the disks are rotating at 3,600 RPM, the desired sector will pass beneath the active head 60 times per second. The hard disk controller synchronizes itself with the rotation of the disks, waits for the

correct sector, and reads it off of the disk.

If the desired data has partially passed beneath the head when the controller starts looking for it, the controller must wait for one complete revolution of the disk (about 17 ms) before it can recover the data. A technique known as "interleaving" is used to increase the probability that the desired data is just about to pass beneath the head when the controller starts looking for it.

Interleaving is accomplished by arranging consecutively numbered disk sectors in non-adjacent positions on each track. The amount of interleave is chosen based on the speed of the computer, so that the disk takes as much time to rotate to the next sector as the computer takes (on the average) to process the previous sector. The controller allows any value of interleave to be used. There is no good rule for determining the optimum interleave. Burke & Burke has determined by experimentation that an interleave of 22 sectors (256 bytes per sector) is optimum for use with the CoCo XT under Level II OS9. The optimum Level I interleave is about 12 sectors.

Sectors near the center of the platter are physically smaller than sectors near the rim due to the wedge shape of each track. Some drives use "write precompensation" and "reduced write current" techniques to alter the way in

which data is stored on the small sectors near the center of the platter.

#### A.1.4 Hard Disk Power Supply

A hard disk uses much more power than the CoCo alone can supply. Most of this power is used to operate the motors that spin the platters and position the read/write heads, especially when the drive is coming up to speed at power-up.

Typical half-height hard drives need +12 volts at 2.5 amps and +5 volts at 1.5 amps. This corresponds to a 38 Watt power supply for 1 drive. A 65 Watt power supply can generally run two half-height drives.

Full-height hard drives, especially older models, need +12 volts at 3 amps or more, and +5 volts at 2 amps. A 48 Watt power supply should be used for one full-height drive.

The new 3.5" hard drives have very low power requirements, and can generally be operated from a 25 Watt power supply.

Most floppy disk power supplies provide only about 1.5 amps at +12 volts. Burke & Burke does not recommend using a floppy disk power supply to operate any type of hard disk.

A.2 Parameters for Popular Hard Disks

The hard disk parameters listed in this section were believed to be accurate at the time this manual was printed, but these parameters are subject to change by manufacturers. You should only use the parameters in this table if you do not have access to a data sheet for your hard disk.

Burke & Burke does not warrant the validity of these parameter values or suggest that they be used in place of manufacturer's suggested values.

A.2.1 ATASI

Model	Hds	Cyls	RWC	Prec	Park	Size
-----	---	----	---	----	----	----
AT3046	7	635	0	0	>635	36MB
AT3051	7	703	703	703	>703	40MB
AT3085	8	1024	1024	1024	1024	67MB

A.2.2 CDC (Control Data)

Model	Hds	Cyls	RWC	Prec	Park	Size
-----	---	----	---	----	----	----
9415-21	3	697	697	0	>697	17MB
9415-36	5	697	697	0	>697	29MB
94205-30	3	989	989	0	>989	24MB
94205-51	5	989	989	0	>989	40MB
94155-25	4	615	615	300	>615	20MB
94155-38	5	733	733	733	>733	30MB
94155-48	5	925	925	0	>925	37MB
94155-67	7	925	925	0	>925	53MB
94155-86	9	925	925	0	>925	67MB

A.2.3 CMI (Computer Memories)

Model	Hds	Cyls	RWC	Prec	Park	Size
-----	---	----	---	----	----	----
CM5410	4	256	256	256	>256	8MB
CM5616	6	256	256	256	>256	12MB
CM6426	4	615	256	256	>615	20MB
CM6426S	4	640	256	256	>640	21MB
CM6640	6	640	640	256	>640	21MB
CM6853	8	640	640	256	>640	42MB

A.2.4 EVOTEK

Model	Hds	Cyls	RWC	Prec	Park	Size
-----	---	----	---	----	----	----
5820	8	375	375	0	>375	26MB

A.2.5 MicroScience

Model	Hds	Cyls	RWC	Prec	Park	Size
HH725	4	612	611	611	>612	20ME

A.2.6 Miniscribe

Model	Hds	Cyls	RWC	Prec	Park	Size	
1006	2	306	153	0	>306	5MB	
1012	4	306	153	0	>306	10MB	
3012	2	612	128	128	>612	10MB	
3053	5	1024	---	---	1024	44MB	
3085	7	1024*	---	---	1024*	58MB	
3438 RLL	32	4	940	940	128	>940	30MB
3438 RLL	48	4	615	615	128	>615	30MB
3650	6	809	---	---	>809	42MB	
3675	6	1024*	---	---	1024*	50MB	
8425	4	615	615	128	shock	20ME	
8438 RLL	32	4	940	940	128	shock	30MB
8438 RLL	48	4	615	615	128	shock	30MB

RLL drives are formatted with 32 sectors per track when using Western Digital or DTC controllers, or with 48 sectors per track when using Adaptec controllers.

84XX series drives are fully shock mounted, and do not require parking.

Only the first 1024 tracks of the drive are available when using IBM-compatible hard disk controllers.

A.2.7 Rodime

Model	Hds	Cyls	RWC	Prec	Park	Size
-----	---	----	---	-----	-----	-----
101	2	192	96	0	>192	3MB
102	4	192	96	0	>192	6MB
103	6	192	96	0	>192	9MB
104	8	192	96	0	>192	12MB
201	2	321	132	0	>321	5MB
202	4	321	132	0	>321	10MB
203	6	321	132	0	>321	15MB
204	8	321	132	0	>321	20MB

## A.2.8 Seagate

Model	Hds	Cyls	RWC	Prec	Park	Size
-----	---	----	---	----	----	---
ST-506	2	306	306	0	>306	5Mb
ST-412	4	306	306	0	>306	10MB
ST-419	6	306	306	0	>306	15MB
ST-4026	4	615	615	615	>733	20MB
ST-4038	5	733	733	300	>733	32MB
ST-4051	5	977	977	977	>977	43MB
ST-4096	9	1024	1024	1024	1024	75MB
ST-213	2	615	615	300	670	10MB
ST-225	4	615	615	300	670	20MB
ST-238 RLL	4	940	940	300	940	30MB
ST-238 RLL	4	615	615	256	670	30MB
ST-251	6	820	820	820	>820	42MB
ST-277 RLL	6	940	940	300	940	65MB
ST-277 RLL	6	820	820	300	>820	65MB
ST-125	4	615	615	300	670	20MB
ST-138	6	615	615	300	670	30MF
ST-138 RLL	4	940	940	300	940	30MB
ST-138 RLL	4	615	615	256	670	30MB
ST-157 RLL	6	940	940	300	940	45MB
ST-157 RLL	6	615	615	256	670	45MB

RLL drives are normally formatted with 32 sectors per track when using Western Digital or DTC controllers, or with 48 sectors per track when using Adaptec controllers. The ST-277 MUST be used with 48 sectors per track. This requires jumper changes on Western Digital controllers.

A.2.9 Shugart

Model	Hds	Cyls	RWC	Prec	Park	Size
-----	---	----	---	----	----	----
SA604	4	160	---	---	>160	5MB
SA606	6	160	---	---	>160	7MB
SA612	4	306	---	---	>306	10MB
SA612	4	320	---	---	>320	10MB

A.2.10 Tandon

Model	Hds	Cyls	RWC	Prec	Park	Size
-----	---	----	---	----	----	----
TM252	4	306	306	306	>306	10MB
TM262	4	615	615	615	>615	20MB
TAN 501	2	306	128	153	>306	5MB
TAN 502	4	306	128	153	>306	10MB
TAN 503	6	306	128	153	>306	15MB
TM602S	4	153	128	153	>153	5MB
TM603S	6	153	128	153	>153	8MB
TM603SE	6	230	128	128	>230	12MB
TM703	5	695	695	695	>695	28MB
TM755	5	981	981	981	>981	40MB

### A.3 Hard Disk Controller Information

Not all PC-compatible hard disk controllers are alike, and some will not work with the CoCo XT. The mounting screws in the interface housing, and the hard disk support software, are optimized for use with particular hard disk controllers. Several controllers that work well with the CoCo XT hard disk interface are:

- Western Digital WD1002-WX1 (MFM)
- Western Digital WD1002A-WX1  
(see below; MFM)
- DTC 5150CRH (MFM)
- Western Digital WD1002-27X (RLL)
- Western Digital WD1002A-27X  
(see below; RLL)
- Adaptec 2072 (RLL)
- DTC 5160CRH (RLL)

#### A.3.1 Mechanical Considerations

Western Digital has recently changed the mechanical design of their hard disk controllers, moving the mounting holes and eliminating jumper bank S1. There are three such controllers on the market at this time: the WD1002A-WX1, the WD1002A-27X and the WDXT-GEN.

The WD1002A-xxx controllers may be used if a slight change is made to the CoCo XT housing: one of the mounting screws must be removed, and the threaded insert must be snapped off of the board with a pliers. You must also file a small notch in the controller board, which does not effect its operation in any way. Contact Burke & Burke for further information if you need to use this type of controller.

The WDXT-GEN controller does not fit into the CoCo XT housing. It can be used if you don't mind about 1/4" of controller board hanging out the end of the housing. Many vendors who normally sell the WDXT-GEN will replace it with a WD1002-WX1 for about \$10.

### A.3.2 Controller Jumpering

Most hard disk "kits" are shipped with one of several common hard disk controllers. Each of these controllers has slightly different jumpering requirements.

On an RLL controller, the most important jumpering is the mode select jumper. This jumper determines whether you should set up your hard disk device descriptors for 32 sectors per track or 48 sectors per track. When using 32 sectors per track ^Uwith an RLL controller only^U, the number of tracks per surface (indicated on the hard disk data sheet) must be multiplied by 1.53. For example, an ST-238 drive normally has 615 tracks but you must

specify 940 tracks in order to use 32 sectors per track. If you are using 4 sectors per track, the ST-238 has 615 tracks.

#### A.3.2.1 WD1002-WX1 Controller

This is the original "short card" MFM hard disk controller for the CoCo XT. Jumpering on this controller is:

W4 -- 2-3  
W8 -- 2-3  
W6 -- 2-3  
W3 -- IN  
S1 -- All positions OUT

#### A.3.2.2 WD1002A-WX1 Controller

Same jumpering as the WD1002A-WX1 controller. There are about 3 different versions of this controller.

Some versions do not have the S1 jumper. S1 is not necessary for operation with the CoCo XT.

The locations of the mounting holes have been changed on some versions. The board can still be mounted in the CoCo XT if you file a small notch in the board and snap off one of the CoCo XT housing's mounting supports.

#### A.3.2.3 WD1002-27X Controller

This RLL controller is exactly the same shape and size as the WD1002-WX1, and has identical jumpers. There is an additional jumper, W9, on the controller.

W9 -- IN for 32 sectors per track.  
Use with drives having up to  
669 tracks per surface --  
multiply tracks by 1.53

W9 -- OUT for 48 sectors per track.  
Use with drives having 670  
or more tracks per surface.

#### A.3.2.4 WD1002A-27X Controller

A newer version of the original WD1002-27X. This RLL controller is the same shape as one of the WD1002A-WX1 versions, and must be modified for mounting in the CoCo XT.

There are only two jumpers on this controller board.

W1,W2 -- IN for 32 sectors per track.  
Use with drives having up to  
669 tracks per surface --  
multiply tracks by 1.53

W1,W2 -- OUT for 48 sectors per track.  
Use with drives having 670  
or more tracks per surface.

#### A.3.2.5 Adaptec 2072 Controller

This controller is the same shape and size as the WD1002-WX1. It is an RLL controller that operates only at 48 sectors per track. When using the Adaptec 2072, set up your descriptors for 48 sectors per track regardless of the size of the drive. Do not multiply the number of tracks by 1.53.

The jumpers on this controller should be set as follows:

J5 -- All positions OUT  
J8 -- All positions OUT  
J9 -- All positions OUT

#### A.3.2.6 DTC Controllers

These are the same shape and size as the WD1002-WX1. Normal jumpering of the DTC-5150CRH is:

W1 -- IN  
W2 -- 3-4  
W3 -- 2-3  
W4 -- 1-2  
W5 -- 2-3  
W6 -- All positions OUT  
W7 -- IN

# CoCo XT Housing Installation

1.

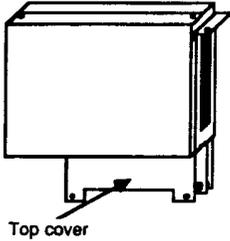


Figure 1 CoCo XT Housing

Figure 1 shows what your new CoCo XT hard disk interface case should look like when you first receive it.

The housing is made of aluminum and has a gold finish. It has two parts -- a top cover, and a bottom cover.

The hard disk interface circuit board is sandwiched between the top and bottom covers.

Machine screws and recessed dimples hold the assembly together.

2.

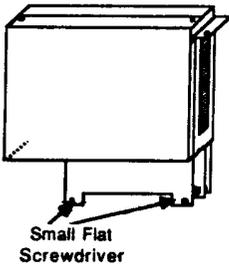


Figure 2 Top Cover Removal

Begin by removing the housing's top cover. This is done by first removing the two screws that hold it in place, as shown in Figure 2. The screws can be removed with a small flat-head screwdriver.

When both screws have been removed, lift off the top cover and set it (and the screws) aside.

3.

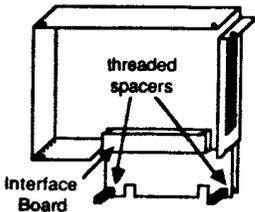
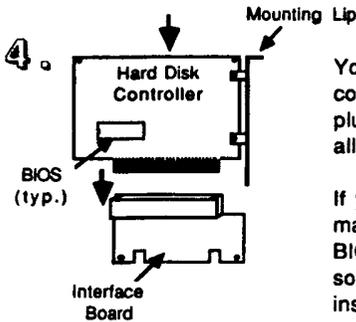


Figure 3 Interface Board Removal

If the interface board has already been installed in your housing, you must remove it at this time.

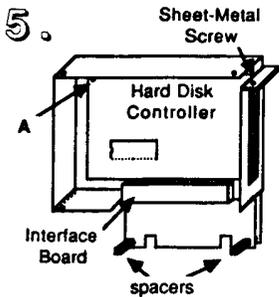
The interface board is held in place by two 1/2" long threaded spacers. Remove the spacers by turning them counterclockwise (fingers work best for this). Then lift the interface board off of the back cover's threaded posts.



Your hard disk controller board plugs into the 62 pin connector on the interface board. **Make sure that you plug the boards together "face up", so that you can see all of the parts that are mounted on each board.**

If you will be using the XT-ROM auto-boot ROM, you may want to install it in your hard disk controller's BIOS socket at this time. If your controller has a soldered-in BIOS, send it to Burke & Burke and we'll install a BIOS socket for \$7.50.

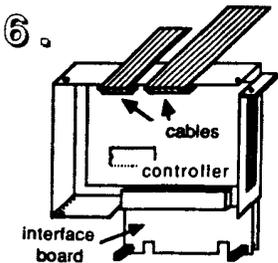
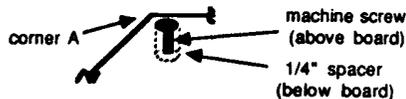
Figure 4. Plugging in the Controller



Insert the combined interface and controller boards into the bottom cover of the housing. The mounting lip on the controller can be secured to the mounting lip on the bottom cover by tightening the provided sheet-metal screw with a small Phillips screwdriver.

Use the 1/2" threaded spacers to secure the boards to the bottom housing. If necessary, use a 1/4" spacer and machine screw to prevent corner (A) of the hard disk controller from touching the back of the bottom cover:

Figure 5. Combined Boards in Housing



Install the 20 pin and 34 pin cables that will connect your controller to your hard drive. When you have correctly connected the cables to the controller, bend them up and over the top edge of the bottom cover.

You may now replace the top cover and fasten it with the screws that you removed in step 2. It's best to put the top edge of the top cover over the cables first, then press the bottom and sides into place.

Figure 6. Cable Installation