SUPERBUS

CYBERIA inc.
COMPUTERS
for personal, professional and business use

2330 Lincoln Way
Ames, Iowa 50010
515-292-7634
(800) 247-3900

Dear SUPERBUS Customer:

Thank you for your recent SUPERBUS purchase. SUPERBUS is a versatile device for various applications.

Should you have questions about operating the SUPERBUS system after reading the manual, please feel free to call Cyberia Inc., Monday thru Friday between 9am and 5pm CST. Our staff at Cyberia will answer any questions you may have.

Sincerely,

Dave Martin.

Dave Martin
Software Development

SUPERBUS LIMITED WARRANTY

Cyberia Inc. warrants that the SUPERBUS ROM sets, circuit boards, floppy disks, manual and special connectors will be free from defects in workmanship and material under normal use and service for 90 days from the date of purchase. If any SUPERBUS product is found to be defective within the 90 day warranty period please return them to Cyberia Inc., 2330 Lincoln Way, Ames, IA 50010 for replacement. This warranty does not cover defects resulting from, but not limited to, accidents, abuse, malfunctions or misapplications, or improper installation.

THE SUPERBUS LIMITED WARRANTY is in lieu of all other warranties, expressed or implied, including warranties of merchantability or fitness for a particular use. In no event will Cyberia Inc. be held for any indirect or consequential damages, such as, but not limited to, loss of anticipated profits or income, interruption of service or other incidental damages.

LICENSE AGREEMENT

By accepting the SUPERBUS products, the buyer also agrees not to copy or reproduce, or to allow any second party to copy or reproduce in any manner whatsoever the SUPERBUS diskettes, ROM sets, their program contents or the SUPERBUS manual without specific, written license for that purpose from Cyberia Inc., 2330 Lincoln Way, Ames, IA. 50010, U.S.A.

Cyberia Inc. reserves the right to change specifications of SUPERBUS at any time. Cyberia not only grants but encourages the purchaser to make and use two working copies of the utilities disks and to place the original in a safe environment for safe keeping.

** SUPERBUS is a trademark of Cyberia Inc.

DISKETTE REPLACEMENT POLICY

If after ninety (90) days and during the first year after the date of delivery, a defect in the diskette should occur, the diskette may be returned to Cyberia Inc., accompanied by the Replacement Order Form and payment of the applicable replacement fee and Cyberia will replace the diskette provided that you have previously returned your Warranty Registration Form to Cyberia.

DISKETTE UPDATE POLICY

Updated diskettes would be available to the users who purchased prior versions of a product. For an updated version you should send your old diskette accompanied by the Update Order Form and payment of the applicable replacement fee as outlined in Cyberia's then current User Support Plan and Cyberia will replace the diskette and provide necessary documentation provided that you have previously returned your Warranty Registration Form to Cyberia.

WARRANTY REGISTRATION FORM

Please type or print legibly and return to Cyberia Inc. within 15 days from the date of purchase.

Name_____

Company_____

Title_____ Department or PO Box_____

Address_____ City_____

State_____ Zip_____ Country_____

Computer Brand_____ Cyberia Program Name_____

Version Number_____ Date of Purchase: Mo____ Day___ Yr__

Is your computer owned by:

Self:_____
Company:_____
Other:_____

Other Cyberia products used:

_____

_____

I learned about the product through:

_____ Computer Dealer _____ Advertisement

_____ Another User _____ Other

Primary use of computer:

_____ Office _____ School

_____ Home _____ Other

_____ Home and Office

Please list the magazines you subscribe to regularly:

_____

_____

_____

I have read the enclosed Customer License Agreement and agree to its terms.

Signature_____

REPLACEMENT ORDER FORM

Use this form to order a replacement for a program diskette found defective in material or workmanship as described in the Limited Warranty Statement. Please enclose the faulty diskette and, if it is past the ninety day Limited Warranty period, please also enclose a check or money order for $20.00. (U.S. dollars)

Please type or print legibly below the same informtion as you supplied on your Warranty Registration Form.

Name_____

Company_____

Title_____ Department or PO Box_____

Address_____ City_____

State_____ Zip_____ Country_____

Computer Brand_____ Cyberia Program Name_____

Version Number_____ Date of Purchase: Mo____ Day___ Yr__

Enclosed is my check or money order in the amount of $_____.____

MAIL TO:

Cyberia Inc.
2330 Lincoln Way
Ames, IA 50010

NOTE: Prices subject to change without notice.

UPDATE ORDER FORM

Use this form to order an updated version of a program diskette, provided that such updates are available. Please enclose your old diskette and a check or money order for the applicable updating fee. Please contact Cyberia Inc., 2330 Lincoln Way, Ames, IA 50010, telephone 515-292-7634 for information about the updating fee for your program.

Please type or print legibly below the same informtion as you supplied on your Warranty Registration Card.

Name_____

Company_____

Title_____ Department or PO Box_____

Address_____ City_____

State_____ Zip_____ Country_____

Computer Brand_____ Cyberia Program Name_____

Version Number_____ Date of Purchase: Mo____ Day___ Yr__

Enclosed is my check or money order in the amount of $_____.____

MAIL TO:

Cyberia Inc.
2330 Lincoln Way
Ames, IA 50010

NOTE: Prices subject to change without notice.

Table of Contents

Table of Contents

Table of Contents

Table of Contents

Index of Illustrations

I. HOW TO USE THIS MANUAL

This manual consists of four parts; the Manager's Guide, the User's Guide, the Programmer's Guide and the Appendices.

The Manager's Guide explains the higher functions of the SUPERBUS utilities and how to manipulate the system in an orderly fashion. It assumes little about the programming knowledge of the system manager, other than general PET / CBM acquaintance. The Manager's guide is designed to be read while working at a computer so that you can try out various SUPERBUS utilities.

The User's Guide is aimed at users other than the system manager. It explains little about the SUPERBUS system, just enough so that the user can work with SUPERBUS without being baffled.

The Programmer's Guide is a technical guide which assumes working knowledge of BASIC 4.0 and its operating system. The primitive functions of SUPERBUS are explained here. A skilled programmer could design and program practically any networking feature or system with the information presented in this Guide.

The Appendix details the steps involved in installation and some hints on possible failure conditions due to hardware malfunction.

Everyone using this manual should skim the Table of Contents first to get an idea of the various sections which may apply to you. If you have problems, check the miscellaneous chapter in the appendix. The index should prove to be very useful also.

If you are planning to use the resource-sharing features of SUPERBUS only, you should read chapters I through IV of the Manager's Guide, and the Appendices. Later, in your spare time, you are encouraged to at least skim the entire Owner's manual to fully understand the power of SUPERBUS.

If you are planning to use SUPERBUS in an school environment, you should read the entire Manager's Guide, the entire User's Guide and the Appendices. If you are thinking of expanding the utilities in any way, including modifying existing utilities, you should read the entire Programmer's Guide. If you are not planning on expansion, you can read the Programmer's Guide at your leisure.

If you are planning to use SUPERBUS in a multi-user software development system, read everything right away.

SUPERBUS 4.4

Part 1

Manager's Guide

Cyberia, Inc.
2330 Lincoln Way
Ames, Iowa 50010

Designed for use with PET / CBM computers

with BASIC 4.0 operating systems

I. INTRODUCTION

This part of the manual is intended to be a guide for the manager of a SUPERBUS system. It contains information concerning manipulation of several security-related programs. It is advised that the manager's guide be shown only to those who are in a managerial position.

The manager's guide addresses basic applications of SUPERBUS and makes the system especially easy to utilize at two extremes. These are a simple device-sharing network and a fully integrated learning environment. Accordingly, there are two types of SUPERBUS systems. The first type is called a standard system. This type does not involve a great deal of security but emphasizes the efficient sharing of resources. The second type of system is called an enhanced system in which SUPERBUS takes care of all the petty details so that a number of different users may use the system without fear of a security control breakdown.

You will discover more about the properties of various SUPERBUS configurations as you read on. For the time being, you should follow through the guide, choose your type of system, and learn all about it. Your system should be installed (but not turned on) at this point. If it is not, ask your dealer to install it or do it yourself following the instructions in Appendix I.

II. HOW TO START UP THE SYSTEM FOR THE FIRST TIME

To start the entire system, all power to the computers, disk drives and printers should be off, and the utility disk should not be in the disk drive. NEVER switch the power to the disk drive on or off while there is a disk inside. If you must turn the disk drive power on or off, remove the disk first. The sudden current to the disk drive can damage the disk.

Now, if possible, turn all devices on at once. If this is impossible, turn on disk drives and printers first, then turn on all computers as quickly as possible. When the 8050 disk drive powers up, it needs a few seconds to 'wake up' before it can operate, and if you do not have ribbon cable connectors, each time power is interrupted, the disk drive will have to 'wake up' again. If a computer tries to use a disk drive that has not yet awakened, the system locks up and you will have to start over. Each computer automatically waits about six seconds after you turn it on before trying to access the disk drive.

After a few seconds, screens should light up and display the usual identification and bytes free message. On each computer, a symbol will appear in the upper right-hand corner of the screen. The symbol will be either '?' or '*', depending on what the computer is doing. A computer that is communicating with a device such as a disk drive or printer displays a '*', and computers that are waiting in line to use a device display a '?'. A computer that is neither using nor waiting in line for a device does not show anything in the upper right-hand corner.

Meanwhile, the red light in the middle of the disk drive should be blinking and the drive may be making some noise. It does not matter into which drive the utility disk is inserted, but for the time being, insert it in drive 0. When the disk is properly in place, the drive starts making spinning noises and soon menus appear on the computer screens.

"Wait," you may say, "my little light isn't blinking, the disk isn't making noise and all computers are waiting in line with a '?'." If this is the case, then you probably have a printer hooked up to the system. Some printers do not initialize properly, and monopolize the system when they are first turned on. Turn off the offending printer. NOW the disk should begin to spin and load up menus. If you still get all '?' or one of the computers has '*' yet nothing is happening, start all over, making sure that the disk drive gets turned on first. Contact your dealer if you still cannot get anything to happen.

When all of the disk spinning is done, the computers should all be displaying the title "SUPERBUS Options" and a menu of nine options. THIS IS THE 'SYSTEM MENU' (see FIGURE 1-I). If you had to turn off your printer to get the system

going earlier, turn it on now. Some printers have to wait for SUPERBUS to start up before they can be turned on.

S U P E R B U S Options

CONFIGURE options

ENRICHED to normal convert

NORMAL to enriched file convert

INSPECT disk file

AUTHORIZATION editor

FIX Wordpro, Wordcraft or Visicalc

LEAVE utility system

(FIGURE 1-I)

Now you should choose one of the computers as 'your' computer. The rest will not be used in these examples, but they certainly could be. You should have the entire system to yourself the first time, otherwise you may be doing things that confuse others who want to use the system.

A. BACKUP YOUR DISK

The next thing you should do is make a copy of your utility disk. To do this, press 'L' (LEAVE ...) from the system menu. On the next page (shown in FIGURE 1-II), choose option 'm' to enter SUPERBUS manager's BASIC (the other options on this page will be discussed later). You will see a series of messages followed by a message that reads 'ready.' The white cursor will be blinking on and off on the next screen line.

Make sure that your utility disk is in drive 0 (the right hand drive) of your disk unit. Insert a brand new disk in drive 1 (the left hand drive) and type the following line exactly as shown:

```
backup d0 to d1
```

RE-CHECK this line before you press RETURN. After you press RETURN, the disk drive will take seven to eight minutes to make a copy of your utility disk. After it is done, the following message should appear:

```
00, ok,00,00
ready.
```

If this message does not appear, there was an error detected while copying the utility disk. Contact your dealer.

After the above message appears, remove your original utility disk and put it in a safe place. You can use this as a backup disk if anything bad happens to your copy. Now put your newly made disk in drive 0 and type '\off' to return to the system menu.

Remember that you must NOT have a write protect tab on the utility disk that you are using. Several utility programs modify data on the disk, and will not function correctly if they can not write to the disk.

Press: p to enter normal PET BASIC
(normal PET emulation)

r to reset computer
(same effect as power reset)

m to enter manager's BASIC

(FIGURE 1-II)

III. THE USES OF THE SUPERBUS SYSTEM

As mentioned earlier, there are two main types of SUPERBUS systems, namely, standard and enhanced (see FIGURE 1-V and FIGURE 1-VI). A standard system is good for a business environment in which previously written programs are extensively used, and the power of SUPERBUS lies in its resource-sharing capabilities. A standard system is also good for any environment where the users have strong responsibility and direction (e.g. an advanced computer instruction class or a software development agency). Security is still built into SUPERBUS, but it is not enveloping the users completely. In a standard system, security features would be best used to prevent accidental destruction of work.

However, in an enhanced system, the security features are largely geared toward not letting users anywhere near data that is not theirs, while protecting their own data. In an enhanced system, each user is assigned an individual identification code. The system manager determines the files that each individual user may access. The enhanced system is capable of doing anything that the standard system can do, but it is slightly more complicated and requires more attention. The following is a list of possible uses of SUPERBUS systems and the suggested type (enhanced or standard):

Application	System type
Small businesses	standard
Schools	enhanced
Laboratories	standard
Demonstrations	enhanced
Software development	either

IV. THE STANDARD SUPERBUS SYSTEM

The utility disk of SUPERBUS is initally set up as a standard system, but system type can be easily changed. If you would like to enhance your system, refer to the Chapter V, THE ENHANCED SUPERBUS SYSTEM.

Most programs will require little or no fixing to be compatible with the standard SUPERBUS system. They simply have to be integrated into the system so that SUPERBUS can find them. This section describes both how to fix programs when this is needed and also how to integrate other programs into the system.

A. HOW TO FIX VISICALC, WORDPRO AND WORDCRAFT

Fixing VISICALC, WORDPRO or WORDCRAFT by following the instructions in this section produces a disk separate from the utility disk that will run your fixed program when your computer is turned on. This separate disk, the destination disk, does not have to be blank or formatted. Your destination disk may have programs or data of any kind on it. For instance, if you had been using WORDPRO before you purchased SUPERBUS, you probably have a disk with your WORDPRO program and files on it. This could be your destination disk.

Begin by chosing option 'F' (FIX ...) from the system menu on the utility disk in drive 0. If you do not see the menu, reset your computer and it should come up. The disk spins and the title of the screen display reads 'Superbus Fixer'. The following prompt also appears.

```
Please indicate what program you would
like fixed:       1) VISICALC
                  2) WORDCRAFT
                  3) WORDPRO
Press 1, 2 or 3: *
```

Press the number associated with the program you would like to fix. You need not press the RETURN key. If you select 3 (WORDPRO), then you will see the following display:

```
Is the ROM chip that came with your
WORDPRO system labelled 50.54 (y/n): *
```

Answer this question. The later versions of WORDPRO 4+ and 5+ have ROMs labelled 50.54.

The screen displays the following menu:

```
There are two ways in which you may
integrate this program into SUPERBUS.

Case 1:  Independent disk.  Good for a
         low-security environment.
         Completely compatible with any
         previous work done.

Case 2:  Integrate with SUPERBUS security
         system and associate with an
         authorization.

Refer to the manual for more details.
Enter case number (1 or 2): *
```

(FIGURE 1-III)

Case 2 is usually associated with the enhanced system, so press '1' to indicate a standard system fix. The screen will display:

```
Please insert the disk that has the
programs to be fixed in drive 1 and press
any key when ready: *
```

Insert your source disk that has the original copy of the program to be fixed into drive 1. If possible, you should use the original copy that you purchased. Press any key when you are ready. Some file names will go by. If you have more than one copy of the original program on your source disk, you will see the following display:

```
You must choose which copy of your
program to fix.  Here's a list:

 1 wordpro 4 plus
 2 wordpro 4+ b/u

Use which copy: *
```

The names next to the numbers will probably be different. If you're not quite sure which copy to fix, choose the first one. Otherwise, press the number of the copy.

```
Enter value for OPNFLG, 't' or 'f' (refer
to manual): *
```

This message will only appear if you are fixing WORDPRO. OPNFLG (read open flag) is a SUPERBUS parameter that will be discussed in detail later. For now, press 't' (true) if you are not going to be using the sequential file output feature of WORDPRO. If you do plan to use this feature, press 'f' (false). There is a side effect of choosing 'f' for this feature, however. When running your fixed WORDPRO, the computer will pretend as though a file that you want to save already exists even though it doesn't. This can be misleading if you are not aware of it.

You now see the following message:

```
Please remove the UTILITY disk from drive
0 and put your destination disk in its
place. Press 'n' to new this disk, or
RETURN to continue: *
```

Remove the utility disk from drive 0 and replace it with the destination disk. If you do not, you may lose important files on the utility disk. If the disk you want to be your destination disk has not been formatted, or if you do not care for its contents, press 'n' to erase it. If you want to keep the contents, press RETURN.

At this point, the computer starts working on the file. It may seem to you as though the computer has crashed, but it has not. After about two minutes of silence, things will come back to life and you will see the following message:

```
Please swap the UTILITY disk with the one
in drive 1 now and press RETURN: *
```

Follow the instructions. After some more disk spinning, you should see the following message:

```
All done! Insert the utility disk in
drive 0 and press RETURN.
```

If you are fixing WORDCRAFT, you will not see this message. Instead, you will see this:

```
Please remove the UTILITY disk from drive
1, insert your source disk and press any
key to continue: *
```

Follow the instructions. You will see an 'all done' message similar to the one earlier.

At this point, your destination disk should be 'fixed'. Press RETURN for the menu. To see how to use this disk, refer to section D, entitled HOW TO BOOT A FIXED DISK.

If a problem arises and you fear your computer has crashed or the disk has not been fixed, reset your computer and start over, following all of the instructions very carefully.

B. HOW TO SET UP A 'TURNKEY' DISK FOR OTHER PROGRAMS

It is likely that you may want to use programs other than WORDPRO, WORDCRAFT or VISICALC. This section explains how to set up a disk that will run a program (e.g. JINSAM or MAGIS) that does not need converting. Any program or set of programs can be set up on a turnkey disk. It is important to understand that this procedure will in no way harm the programs already on your disk.

To begin, choose option 'C' (CONFIGURE ...) from the system menu. You will see this display:

```
1.. Enter "configure" system program to
    create boot files

2.. Switch system type (standard or
    enhanced).

3.. Set up a turnkey disk.

4.. Install a disk with a boot file that
    enters BASIC upon booting.

Choose which type of configure operation
to perform: *
```

(FIGURE 1-IV)

Choose option number 3 to set up a turnkey disk. You will then see this message:

```
Please insert the disk that you wish to
make into a turnkey system into drive 1
and press RETURN: *
```

This process of creating a turnkey system can only be done on a disk that already has programs on it. Presumably, the disk has a collection of non-SUPERBUS programs. So, insert your disk with the programs into drive 1 and press

RETURN. The disk spins for a while, and soon an 'all done' message should appear. At this point, your disk has been changed into a turnkey system and you should press RETURN to get back to the system menu.

However, you may see the following warning message:

> WARNING: This disk already has a SUPERBUS boot file on it. Re-check the disk and press RETURN if you want to commence (writing over the existing '.b') or press any other key to abort: *

If you see this message, follow the instructions. The program is just making sure that you do not destroy any data in the process of making your boot file.

To use your turnkey system, refer to Section D, entitled HOW TO BOOT A FIXED DISK.

C. HOW TO FORM A DISK TO ENTER BASIC

When a SUPERBUS computer is turned on, it automatically tries to load a program from the disk drive called '.b'. When resetting or turning on a computer, if neither of the disks in the disk drive have a '.b' program in the correct format, the computer will not know what to do and repeatedly displays a 'file not found' error message. This section describes how to install a '.b' program on any formatted disk that will cause a computer to simply enter PET BASIC when it is booted from that disk.

To make a disk bootable to enter BASIC, first choose option 'C' (CONFIGURE ...) from the system menu. You will see FIGURE 1-IV. Choose option number 4 at this prompt, and you will see the following message:

> Please insert the disk that you wish to make bootable to enter basic in drive 1 and press RETURN: *

Follow these instructions. An 'all done' message will appear and you should press RETURN to get back to the system menu. As in the previous section, you may see a warning message. Use good judgement in writing on top of boot files that already exist. For instance, if you do this to the utility disk, it will no longer boot properly and the utility programs will be practically useless.

D. HOW TO BOOT A FIXED DISK

Booting a fixed disk, a turnkey system or a plain bootable disk is quite simple. All you have to do is insert

the disk in the disk drive and reset the computer. The bell will sound (if you're using a 12-inch screen computer) and after about six seconds the Commodore basic message will appear, the boot file will be loaded, and your computer continues normally.

You can use a fixed disk any time you want, even when you turn on your system in the morning: you need not start with your utility disk. There is one problem you may have: the 'file not found' syndrome. If the computer keeps printing '?file not found' and nothing good is happening, either the disk was never fixed properly (has it ever worked before?) or there is a connection problem between your computer and the disk drive. Check your cables.

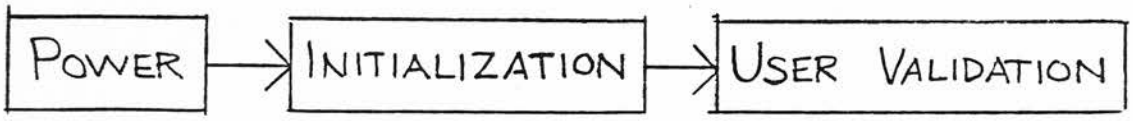
When a computer is turned on, it searches in the most recently used drive for the boot file. If both disks in the drive have boot files, it is possible that your computer will get the 'other' boot file, not the one that you wanted. To solve this problem, just open the drive that holds the disk which you do not want to boot from. This will make the disk unreadable until you close the door again, so the computer is forced to take the boot file from the other drive.

E. A FINAL NOTE ON THE STANDARD SYSTEM

When you use the utility disk, you will see several unexplained options. After reading the rest of this manual (though it's not vital to do so), you may want to make use of the other utility programs. Until then, it shouldn't be necessary to operate any of the others.

(FIGURE 1-V) A flow chart for a standard type SUPERBUS utility disk from power-up.

(FIGURE 1-VI) A flow chart for an enhanced type SUPERBUS utility disk from power-up.



(FIGURE 1-VII) The User Authorization File, which contains all the authorization information for the security / authorization system.

SYSTEM TYPE	NUMBER OF AUTHORIZATIONS	AUTHORIZATIONS

(FIGURE 1-VIII) An individual authorization in the User Authorization File.

USERNAME	PASSWORD	TIME TO HOLD BUS	LEVEL	ROUTER DATA	STATUS

V. THE ENHANCED SUPERBUS SYSTEM

This section of the manual deals with the enhanced SUPERBUS system and its operation. To use SUPERBUS in an enhanced system, a conversion of the utility disk is necessary. This section describes how to convert to enhanced and standard systems (if you already have an enhanced system).

The only difference between enhanced and standard systems is the authorization system (refer to FIGURE 1-V and FIGURE 1-VI). This allows the system manager to determine what each user of the system can do. This chapter is much larger because it includes documentation on the user editor and login programs. Also it describes utilities that would normally apply only to an enhanced system, such as format conversion programs, configurations, and the file inspection program. In a standard system, one does not have to perform any of those types of functions. An enhanced system is needed to provide such capabilities.

Remember that all of the 'how to' sections in Chapter IV (The Standard System) apply to the enhanced system as well, with the exception of program 'fixing'. You still may set up independent fixed disks, turnkey disks, and so on while running an enhanced system.

A. HOW TO CONVERT THE UTILITY DISK

SUPERBUS is initially configured for use in a standard system. This section explains how to configure for use in an enhanced system. For conversion to an enhanced system, press 'C' (CONFIGURE ...) from the system menu. The screen displays FIGURE 1-IV.

Choose option number 2 to convert the system type. You will see the following display:

```
Your system is currently of the STANDARD
type.  Type  'convert'  (without  the
quotes) to convert your system to an
ENHANCED one: *******
```

(The words STANDARD and ENHANCED would be exchanged if you were running on an ENHANCED system wanting to convert to a STANDARD system.)

Type the word 'convert' without the quotes. This will in no way harm any programs on either of the disks in the disk drive. If you type anything but 'convert', you will return to the configure options menu. If you type 'convert', the disk will spin a little and you will see this message:

```
Your disk has been converted!  Press any
```

key to return to the system menu: *

In case of any conversion problems, you will see this message:

> ERROR in conversion - check your utility disk and the write protect tab on it and try again or contact your dealer - press any key to continue: *

If you have just converted to an ENHANCED system, make sure to read the rest of this chapter.

B. THE SYSTEM MANAGER

The system manager refers to the individual(s) in control of a SUPERBUS system. Until now, the definition has not been made because in a stardard system, the system manager phrase can apply to practically anyone with any knowledge of the system. In an enhanced system, however, the system manager has several important responsibilities and powers that must be closely guarded and explicitly defined to avoid confusion.

The remainder of the Manager's Guide applies only to the system manager of an enhanced SUPERBUS system. The system manager should be familiar with all the programs, concepts and situations described in the following pages. In a hierarchical system such as the enhanced system, the system manager must have enough knowledge to perform the functions for which the other users do not have the power.

C. THE SECURITY / AUTHORIZATION SYSTEM

The security / authorization system is the basis of the enhanced SUPERBUS system. This section describes various aspects of the security system provided. After you gain sufficient knowledge of this system, you may want to design your own security system utilizing SUPERBUS' basic security functions. The security system described here is one good application of the features provided by SUPERBUS.

1. Authorizations.

What is an authorization? At the most primitive level, an authorization is a record in the user authorization file describing how the security system should handle specific situations. An authorization is also a username - password pair that functions as a 'key' to various parts of the system (refer to FIGURE 1-VII and FIGURE 1-VIII).

The system manager is responsible for adding, deleting (removing), changing and monitoring authorizations. The 'edit' utility in the system menu is available to perform

these functions. This utility will be explained in detail later.

Authorizations are needed only when you are logging in (identifying yourself to) the computer. After a user is validated, only the characteristics associated with the authorization are remembered by the computer; the username and password are not.

2. Types of file Security.

There are two types of file security formats, normal and enriched (refer to FIGURE 1-X and FIGURE 1-XI). Format refers to both the way a file is stored on a disk, and how the computer expects files to be stored. The format types have little relation to the system types (standard and enhanced), and can be used any time, with either. However, standard systems tend to use the normal format more often than enriched format and enhanced systems tend to use the enriched format. As a rule of thumb, remember (standard system - normal format - no security) and (enhanced system - enriched format - active security).

Normal format is seldom used for access restricting purposes. As the name suggests, normal format files are no different than files that a normal PET computer (without SUPERBUS) would store or expect to recieve. This 'normal format' option was included to make SUPERBUS as compatible as possible with already existing products. For instance, when a SUPERBUS computer is first turned on, it tries to load and run program '.b' from the disk drive, as mentioned earlier. It expects this program to be stored on the disk in normal format rather than enriched format. The SUPERBUS parameter called CORFLG determines the format that the computer will expect files to be stored in and the format it will use for storing. Parameters can usually be changed at any time - and will be discussed in the Programmer's Guide.

Enriched format is usually used when security on a file is desired. There are two ways to specify the security characteristics of a file: by level and key. The SUPERBUS parameter LEVEL is usually associated with an authorization. There are 256 possible levels, 0 through 255. Like the CORFLG parameter that determines format, LEVEL determines both in what level a file will be stored and in what level it will expect to find them. The important security feature of LEVEL is that a computer will refuse to work with a file when the file's level does not match the computer's current level. Since LEVEL is a feature associated with the enriched format, if a computer is running in normal format, the levels are ignored. The computer can not expect files stored in normal format to have levels.

It is like a fingerprint scanner for access to a building. The scanner has a recognized fingerprint stored in its memory permanently. When someone requests access, the machine checks the person's fingerprint. If the fingerprint

matches, it will let the person in. If it does not match, it will refuse access. The only way to fool a fingerprint machine is to be born with the same fingerprints as someone else, which is unlikely. The system manager of a SUPERBUS system, however, can assign groups of authorizations to the same level. Do you see now why it's difficult to trace a level back to an authorization if more than one person has the same level? It's the same as two people having the same fingerprint. The machine is just concerned with fingerprints, not people. Similarly, SUPERBUS is concerned with levels, formats and keys, not individuals or authorizations.

SUPERBUS will also refuse access to a file stored in normal format if the computer is in the enriched format. That is like a back-up to the fingerprint machine. The doors to the building will lock automatically if the machine gets disconnected, and the only way to get in is with a maintenance key provided by the city. With SUPERBUS, only a computer operating in normal format can access a file stored in normal format.

Also associated with enriched format files are keys. Keys are like passwords and can be assinged to any enriched-format file. Once a key is assigned to the file, the file cannot be accessed without it if the computer is running in enriched format. The file can be accessed without the key in normal format, but you need a program to do it accurately. Many utility programs use this feature. This topic will be discussed in the Programmer's Guide.

3. The Authorization Editor - 'edit'

This section explains how to operate the authorization utility called 'edit', which is supplied on the utility disk. This utility allows the system manager to manipulate authorizations, and ultimately decide who can use the system and for what purpose. Choose option 'A' (Authorization editor) from the system menu, and a display like this one will appear:

1 ADD an authorization

2 DELETE an authorization

3 CHANGE an existing authorization

0 EXIT / save changes, if any

Choose an option: *

First, let's look at the existing authorizations. Press '3', and a roster list will appear that looks like this:

```
No.  Name              Level   Status
 1   system              255    - on  -

Enter the number to change: ***
```

Before the computer asks you what authorization to change, it shows you a list of those already existing. If there are more than one page of authorizations, 'edit' will give you options to move through the pages of the roster.

Number 1, 'system', is a special authorization provided for the system manager. It is the only record that cannot be removed, and is always record #1. It is provided with level 255, the highest possible level. Normally, this level (255) is reserved for the system manager's programs and system utility programs.

Type the number '1' and press RETURN. This indicates that you want to change or inspect the system manager's authorization. A display like this one will appear:

```
Password -- exists --

Time to hold bus = infinite.

Level for router = 255

Router = system menu

Format of router = enriched
Device for router = last used

Status = ON (usable)

Press '+' to edit the next record, '-' to
edit the previous, or a letter to change
the associated datum: *
```

The first option allows the manager to change the password to an authorization. The manager can never look at the passwords to authorizations. If someone forgets a password, the manager should change it via 'edit'. There are three basic types of passwords: press 'P'.

You will see three options and a choice prompt. The types of passwords should be self-explanatory: a blank password that will be then determined by the first user of the authorization, no password required or a simple password. If you do not really want to change the password to 'system',

press RETURN alone. Usually in utility programs, you may press RETURN alone when the cursor is flashing to return to a previous section.

You will see the same display once again. The next option, time to hold bus, is a very special feature of SUPERBUS. This can be set to anywhere from 1 to 255 seconds, or infinite. This determines how long a computer will tie up the network with a device, such as a disk drive or printer, with no communication with this device before giving up, closing the device and returning to BASIC. This is a VERY effective way to ward off malicious system-grabbers. For instance, if a user began to get a directory listing and stopped it in the middle of the listing, the computer would no longer be communicating with the disk drive. Depending on how long that particular user was allocated, the computer would eventually give up on him/her and dump into BASIC.

The time option is only used if the router is restricted SUPERBUS BASIC, which will be described soon.

The next option, level, means different things in different cases. In the case of the system authorization and any others that have routers other than restricted SUPERBUS BASIC, it determines at which level the computer should run the router (presumably, the same level that it was stored at). It was previously mentioned that system programs and utilities are usually stored at level 255. The level associated with 'system' reflects that.

In another case, level determines at which level a user shall enter BASIC. When the router is set to restricted SUPERBUS BASIC, instead of running a menu program, the user of the authorization will enter BASIC at this predetermined level. The user will then be able to access only files stored at the level associated with their authorization and may not harm the system in any fashion. This is an excellent feature for classroom applications.

The 'router' is the name of the program that the authorization will run immediately after being validated. If the router is set to restricted SUPERBUS BASIC, the user will run NO router program, but instead, enter a highly secure BASIC, where they may not harm the network or disks. The user will be able to use all commands except 'sys', 'poke' and 'wait'. More about restricted basic will be discussed later.

The format and device of the router should be self-explanatory. The format should be set to the format the desired router was stored in, and the device should indicate which device to load the router from (usually a disk drive #8).

The status flag is special. If it is set to off, the owner of this authorization will not be able to use the

SUPERBUS enhanced system, and if they try, will be given an explanatory message referring them to the system manager. This is useful when the system manager feels an authorization should be temporarily deactivated.

If the 'system' authorization is deactivated, ALL OF THE USERS except the system manager will be unable to log in! Therefore, the system manager can disable the entire system by deactivating his/her own authorization. This feature is good for an environment in which access should be restricted.

Press RETURN twice to get back to the main 'edit' menu. The other options should seem clear now. This program will be the center of authorization manipulation for the system manager. Check the roster list often to make sure your system has not been infiltrated.

When you are finished using the authorization editor, press RETURN a number of times until you reach the main menu, then press '0' (zero).. This will store the changes in the user authorization file and return you to the system menu.

4. The Login Program.

The login program validates users and routes them to the correct places in the system. It is meant to be entirely self-explanatory, but there are a couple of options available to the system manager that are not described in detail.

To run the login program, press 'L' (LEAVE ...) from the system menu, and then choose option 'r' for reset. This has the same effect to the computer as turning it off and on. The disk will spin a bit, and you should see this prompt:

```
Type the name of your authorization and
press RETURN: ****************
```

A word about the login program. After a short period of no activity, it will clear the screen. It does this to save the screen phosphor from burning unevenly by having the same thing written for a long time. It will recover when the user presses any key.

Type your username, 'system'. The computer will respond with a password prompt:

```
Type your password and press RUN/STOP:
************************************
```

The password to 'system' is 'manager' unless you have already changed it. Type in the password and press RUN/STOP instead of RETURN. You will notice that what you are typing does not appear - that is to ensure that no one sees your

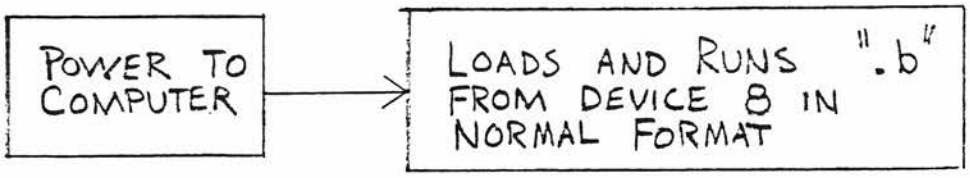
password as you type it. The reason that you have to press RUN/STOP instead of RETURN is that only system programs can trap the stop key when it is pressed without stopping the program that is currently running. Without forcing the user to press STOP, anyone could write a program that looked like the login program that simply steals passwords.

You will then see an option to change your password. It is recommended that you do so often, in case anyone finds out what it is. Do so now and REMEMBER what your password is. Do not use initials or anything else that is easy to guess.

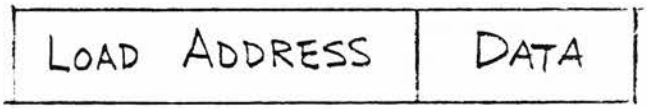
The next screen page that you will see provides some special options. They were included mostly for advanced system debugging and are available only to the system manager. One of the options allows you to 'mimic' a restricted user of the system. If you choose this option, you will be asked for a level and then you will be dumped into restricted SUPERBUS BASIC at that level. The time limit will always be infinite when this method is used.

If you press RETURN once again, you will be sent to the system menu. This is the procedure you will have to go through each time you want to use the computer.

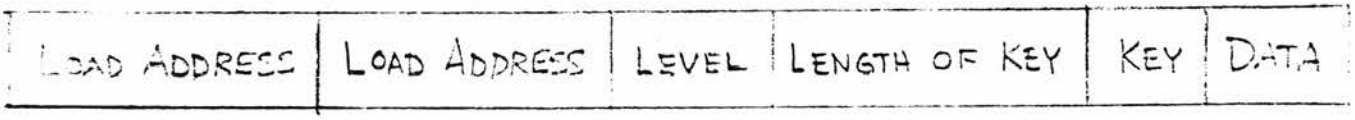
(FIGURE 1-IX) A flow chart for a SUPERBUS computer not using the utility disk.



(FIGURE 1-X) A normal format file. Computers that are unequipped with SUPERBUS deal with normal format files, as do SUPERBUS computers running in normal format.



(FIGURE 1-XI) An enriched format file, with its improved error recovery and security features.



5. The File Inspector.

This section deals with the file inspector program included on the utility disk, called 'inspector'. The file inspector reports the characteristics associated with any set of disk files. This output may be received on the screen or in report form on the printer.

As mentioned earlier, the two major characteristics of an enriched format file are the level and key (if one was assigned). The inspector program reports the load address, the level, and the key of an enriched format file, and there is also an option provided to cross-reference the level of an enriched format file to an authorization in the user authorization file. However, with a normal format file, the inspector program just reports the load address, since there are no other known characteristics associated with the file.

Choose option 'I' (Inspect ...) from the system menu. You will see the following display:

```
There is a section in the Manager's
Guide of the SUPERBUS manual about this
program.

In which drive are the files that you
wish to inspect: *
```

Enter the number of the drive that holds your disk, one or zero. You then see the following display:

```
Press RETURN alone to consider all
files, or enter a file specification to
consider: ****************
```

If you wish to inspect a single file or a group of files with similar names, enter the name of the file specification here, including any wildcards. Your entry is used in a directory search, so anything that is legal in a directory search is legal here. If you want to consider all files on the disk, press RETURN. The following display appears:

```
1. Filetype (seq, usr, prg, or all) : a
2. Format (normal, enriched, or all): a

3. Minimum level: 0
4. Maximum level: 255

Enter the number of the field to change,
or press RETURN when done: *
```

Here you may select characteristics that a file must posess to be considered for inspection. For instance, on the filetype entry, if you enter 's' for sequential, then only sequential files will be considered. In the filetype and format cases, 'a' for all indicates that all files in that category will be considered. The minimum and maximum levels do not apply if you selected 'normal' for the format field, because normal format files have no levels.

The disk will spin for a while, and then you will see the following display:

```
1. Output to screen or printer: screen

2. Page size: 66

3. Cross reference enriched format
   files to an authorization (y/n): n

4. Show keys to files (y/n): n

Press the number of the field to alter,
or press RETURN to continue: *
```

Here you see the output options that determine how the information will be reported. You may change any of these options. Output may be sent to the screen or a printer. If you specify a printer, you must also specify a device number for that printer.

If you choose to cross reference to authorizations, the owner of an enriched format file will be identified if possible. If more than one authorization in the user authorization file is at the same level, files at that level will not be cross referenced.

You may also choose to display the keys to files if they exist, however, if you choose this option, be VERY CAREFUL with the output. If you send it to a printer, lock away the listing safely so that no unauthorized persons can steal the keys to files.

After pressing RETURN to continue, the disk spins, and your report is generated. On a printer listing, you will see the following header line:

```
FILE NAME       TYP SIZ LOAD RN FR LV KEY              XREF
```

TYP is the file type, SIZ is the size of the file in blocks, LOAD is the load address of the file (if it is a program), and RN is the value of RUNFLG for a program (if RUNFLG = 1, the program automatically runs upon loading). FR

is the format that the file was stored in (n for normal, e for enhanced), LV is the level of the file if in enriched format, and KEY is the key to the file if it exists. XREF is the cross-referenced authorization name.

6. Manager's BASIC.

Manager's BASIC is the kind of BASIC that the system manager usually uses. It provides three extra security-control commands in addition to the five utility commands provided with the enhanced system. Choose option 'L' (LEAVE ...) from the system menu. You will see FIGURE 1-II on your screen. Choose option 'm' to enter SUPERBUS manager's BASIC. You will then see the following display:

```
System security has been disabled.

\off to return to the system menu.

\s, \l and \f are now active.

ready.
```

You are now in manager's BASIC. You may execute any BASIC commands, including the five extra SUPERBUS commands. You may find documentation on \g, \t, [and] in the User's Guide. This section discusses the three extra commands and the \off command as it applies to manager's BASIC.

When you first enter manager's BASIC, file security is turned off. This does not mean that the computer is in normal format. This means that you may scratch (delete) files, copy files, backup disks and perform other disk functions regardless of the format, level or key of the file. However, you must be at the proper level and specify the proper key (if there is a key) when you wish to load, save or open an enriched format file.

The \s command turns security on or off. Entering '\s1' (one, not the letter 'l') turns file security on and '\s0' (zero, not O) turns it off.

When you enter manager's BASIC, the computer is running in enriched format at level 255 (the system level). You may change the format of the computer with the '\f' command: '\f1' (one) sets the computer to enriched format, and '\f0' (zero) sets the computer to normal format. You may change the current level with the \l (the letter 'l') command. For instance, '\l23' sets the computer to level 23 and '\l255' sets the computer to level 255.

You may access any file on the disk if you are in normal format, but if you try to load an enriched format program, your computer will probably crash. You may access all files without fear of crashing with the BASIC commands 'dopen' and 'open' when your computer is in normal format. This is

possible because a normal CBM does no security checking, and when a computer is is normal format, it is basically a normal CBM.

The '\off' command returns you to the system menu when your computer is at level 255. If your computer is at another level, you will receive a 'wrong level' error message and you must set your level to 255 with the '\l' command before '\off' will function. The '\off' command automatically sets your computer to enriched format before returning to the system menu, so you do not have to reset format before issuing the '\off' command.

When you enter manager's BASIC, those three commands are installed. When you are done, you must either reset your computer or choose option 'r' from the leave utility system options. If you do not, anyone will be able to execute these security-penetrating commands until the computer is reset!

7. Restricted BASIC for the System Manager.

This section explains some aspects of restricted BASIC and the users of restricted BASIC that the system manager should be familiar with. A restricted user may not execute the \l, \s, \f, poke, sys or wait commands. This effectively prevents them from harming the system in any fashion. A restricted BASIC user always runs in enriched format, and security is always turned on. This way, the computer is truly user-proof.

A restricted user goes through almost the same login procedure that the system manager does. After the system manager creates an authorization for a restricted user (see section 3, 'edit'), the user may attempt to log in to the system. The user must know his/her username and password (if one was assigned) to be validated by the 'login' program. Anyone may use an authorization that has no password required if the username is known, and only the system manager may change the password.

After validation, the user of an authorization that does have a password is given the option to change his/her password in the same way that the system manager may change his/her own password. This gives users a degree of protection.

After the login process is completed, the user is dumped into BASIC in a completely secure environment. Users log out by issuing the '\off' command.

Any user running in enriched format may read or load enriched format files stored at level 0. For this reason, level 0 files are called LIBRARY files. If you have a program that you want all of the restricted users to be able to run, store it at level 0. A user authorized at level 0 is thus called the LIBRARIAN.

D. INTEGRATING PROGRAMS INTO THE ENHANCED SYSTEM

There are a few ways to integrate non-SUPERBUS programs into the enhanced system. The first method is to simply copy the program onto the utility disk and define an authorization (using the authorization editor discussed in Section D, subsection 3) that points to this program. This way, when someone logs in with the new authorization, they will run the program associated with it. Say you wanted to integrate a program called 'poker'. First, you would want to enter the authorization editor from the system menu.

After adding your authorization and setting the password, you will have to change the authorization to fit your needs. Just set the router name to 'poker', and the router format to normal. This way, when the user logs in, they will run program 'poker' on the utility disk in normal format.

However, the program is not there yet. Return to the system menu and enter manager's BASIC. Put the disk with 'poker' on it in the empty disk drive and use the BASIC 4.0 'copy' command to copy the program onto the utility disk. You do not have to do any format conversions, because you indicated earlier that the program 'poker' is in normal format, and non-SUPERBUS files are normal format.

Reset your computer and try your new authorization. Notice that if you drop into BASIC, the computer will automatically reset, preventing you from harming the network. Remember that your 'poker' program is in normal format, so if you wish to change it, you must be in normal format. It is very important that you understand this: since the 'poker' program is a router (you defined it as one in the authorization editor), it can execute any BASIC command, including sys, poke and wait. It is up to the programmer of 'poker' to make sure that nothing hazardous happens to the network because of 'poker'.

If you want 'poker' to be in enriched format, you will have to choose a level for it and convert 'poker' to enriched format. There are two ways to do this - either with the convert to superbus utility from the system menu, or by yourself in manager's BASIC. If you do convert 'poker' to enriched format, any restricted BASIC user at the same level can edit it. By storing it at level 0 (the library level), all restricted BASIC users can run it.

E. FIXING WORDPRO, VISICALC, WORDCRAFT FOR THE ENHANCED SYSTEM

The fixing process for the enhanced system is quite similar to the process for the standard system. Review Chapter IV, Section A before continuing in this section to get a 'feel' for the fixing process.

After entering the fixer program from the system menu and choosing the program to fix, you will see FIGURE 1-II. Choose case 2, and you will see the following message:

```
Please insert the diskette that has the
program to be converted in drive 1 and
press any key when ready: *
```

As in the standard fix, you may have to choose from a list of programs that could be fixed. You will see the following prompt:

```
Enter the name of the authorization to
associate with: **************
```

At this point, enter the name of the authorization that you wish to automatically run the fixed program upon logging in. If the authorization does not already exist, you will be asked if you wish to create it, and you will have to indicate at what level to set the authorization.

If you are fixing WORDPRO, you will see a message referring to OPNFLG. Refer to Chapter IV, Section A for an explanation of OPNFLG.

After about ten minutes, your program will be fixed and may return to the system menu. Log in to your new authorization and make sure it works. You may want to look at the authorization with the authorization editor to see exactly what happens when you log in.

SUPERBUS 4.4

Part 2

User's Guide

Cyberia, Inc.
2330 Lincoln Way
Ames, Iowa 50010

Designed for use with PET / CBM computers

with BASIC 4.0 operating systems

I. INTRODUCTION

This part of the SUPERBUS manual is intended to be shown to users of restricted SUPERBUS BASIC. It explains features of BASIC as they apply to the user, such as security and the extra commands. Users of restricted SUPERBUS BASIC should also have a PET / CBM guide for BASIC 4.0 reference.

The BASIC extensions are documented in this part of the manual, and a few of them are not documented elsewhere. There is a stand-alone chapter in this manual that explains how to log in. That chapter should be presented to everyone using the enhanced system.

II. LOGGING IN

Logging in is the process of identifying yourself to the computer. After you identify yourself, the computer knows how it should treat you.

You should have received an authorization assignment from your system manager. The system manager is the individual or group that is in charge of the SUPERBUS network of Commodore computers. Included in this assignment is an authorization name (commonly referred to as a username) and perhaps a password.

To start, find a computer that is not being used. A computer that is not being used will have a blank screen. Sit down at a free computer and press the RETURN key. An asterisk will appear in the upper right hand corner for a few seconds and the following prompt will appear:

```
Type the name of your authorization and
press RETURN: ****************
```

Type the name assigned to you by the system manager into the computer and, once again, press the key marked RETURN. This key is used to tell the computer that you are done typing a line.

At this point, the screen may clear and the asterisk may re-appear. If this is the case, sit back and wait: you have successfully identified yourself to the computer.

You may see the following message:

```
Make up a password so that no one else
can use your authorization and type it
below:


Type your new password here:
**********************************
```

Some authorizations require a 'password' to be entered before the computer will consider them to be identified. The computer is simply asking you what your password should be. Make up a password that is not easy to guess and type it into the computer, pressing RETURN when you are done. You may notice that the letters you type do not appear on the screen when you type them. This is because your password is secret. Only the computer and you should know the password.

After you press RETURN, you will see the following

prompt:

```
Now type it again to verify:
**********************************
```

The computer is asking you to type the same word again. It doesn't want to misunderstand you, and if you type it twice, there should be less chance of error in typing it. After you have typed your password again and pressed RETURN, the computer checks to see if you typed the same thing. If you didn't, the computer will inform you and make you type your password twice again.

After you have successfully entered your password, remember it, but do not write it down. The next time you log in, the computer will demand that you type the same password. If you cannot remember it, contact your system manager, who can clear it up for you.

After you have chosen your password and logged in again, you will be asked if you wish to change it. If you do not, just press RETURN alone. If you do, you will have to type the password twice, just like before.

III. SUPERBUS BASIC

After you have logged in, you will see a message like this:

```
Welcome to SUPERBUS BASIC.

You may not execute the poke, sys or
wait instructions.  Use \g to set
graphics mode and \t to set text mode.


You are running at level 1.

You may read or write to files stored at
this level, and you may read files
stored at the library level.


Type \off when you are finished.

ready.
```

This is the introduction you will receive every time you log in. The level that you are running at may differ from this example.

A. SAVING YOUR PROGRAMS

After your program is written, you will probably want to save it on the disk so you don't have to type it back in when you return. First, you have to choose a name for your program. This name must be from 1 to 16 characters long.

Following is an example of the BASIC 4.0 dsave command. The command 'dsave' stands for 'disk save'. In this Guide, the computer's responses will be shown in **boldface print.** Do not type the boldface print into the computer.

```
10 ?"this is a short program
dsave"myfilename
00, ok,00,00

ready.
```

If the 00, ok,00,00 message appears, your file was saved properly, and you can load it back into your computer later with the dload instruction.

B. LOADING YOUR PROGRAMS

Loading a program puts a previously saved program into the computer's memory so that you can use it. You will want to use the BASIC 4.0 'dload' command, which stands for 'disk load' to retrieve your program. Following is an example.

```
dload"myfilename

searching for 0:myfilename
loading 00, ok,00,00

ready.
```

If the 00, ok,00,00 message appears, your program was successfully loaded into your computer. If a different message appears, something went wrong while trying to load your program. If you do not understand the message, contact your system manager.

Your system manager may give you a list of additional programs that you can load into your computer. Always use the 'dload' command to load a program into your computer.

C. LOGGING OUT

When you are done using the computer, you should log out so that others can use the computer. To log out, issue the following command:

```
\off
```

Be careful when issuing this command - if you have not saved your current program at least once before, it will be lost!

D. GRAPHIC AND TEXT MODE COMMANDS

In the PET / CBM User's Guide, you may notice two instructions which set text and graphic modes. Those instructions cannot be executed in this type of BASIC, but two commands have been installed to replace them. The commands are: \g to set graphics mode (equivalent to poke 59468,12), and \t to set text mode (equivalent to poke 59468,14). Following is an example.

```
10 ?"UI
20 ?"JK
30 \g
40 \t
50 goto 30
```

E. FILES WITH KEYS

It is possible to put a key on a file. That way, even if someone finds out the password to your authorization, you still have a degree of protection, for keyed files cannot be harmed unless the key is specified. To specify a key to a file, simply tack the '< ' character followed by your key of zero to fifteen characters to the file name. Here is an example:

```
10 rem this is a keyed file
dsave"mine< frodo
00, ok,00,00

ready.
dload"mine< bilbo

searching for 0:mine< bilbo12,wrong
key,00,00

?file not found error12,wrong key,00,00

ready.
```

If you forget the key to your file, contact your system manager. Your system manager can find out the key to the file if necessary.

E. ARRAY READ AND WRITE COMMANDS

SUPERBUS also implements two more commands. These commands are extremely fast and powerful input / output instructions. With these commands, you can read and write entire arrays to a disk file with a single instruction. The disk read command is '[', and the disk write command is ']' (notice how the bracket points to the right - write).

You must specify the logical address of the file to write to or read from before you can read or write your variables. Following is an example.

```
10 dim a(100),b$(3)
20 dopen#1,"myfile",w
30 ]1,a(0)-a(100),b$(2)-b$(3)
40 dclose#1
```

Note that you can have multiple array specifications on one line separated by a comma. You may also read and write scalar variables (such as a$ or zz%) with the array read/write instructions, if you wish. However, the array read and write instructions store the variables in an unusual format, so if you write the array with the ']' instruction, be sure to read it with the '[' instruction.

F. FURTHER INFORMATION

For further information on the SUPERBUS system, please contact your system manager.

SUPERBUS 4.4

Part 3

Programmer's Guide

Cyberia, Inc.
2330 Lincoln Way
Ames, Iowa 50010

Designed for use with PET / CBM computers

with BASIC 4.0 operating systems

I. INTRODUCTION

In writing the Programmer's Guide of this manual, some assumptions were made. It must be assumed that the reader is familiar with PET/CBM BASIC 4.0. If you are not confident of your knowledge, the manual that comes with all Commodore 4.0 computers will make a splendid review.

The advanced programming chapter of this manual assumes a working knowledge of the PET/CBM 4.0 operating system and 6502 assembly programming.

The Programmer's Guide is NOT for the user of restricted SUPERBUS BASIC. The previous part, the User's Guide, was provided for that purpose. Most of the functions discussed in the Programmer's Guide will not work in restricted SUPERBUS BASIC, because of their security requirements. As with the Manager's Guide, this part of the manual should not be shown to someone who does not have access to perform the indicated functions.

II. SUPERBUS PARAMETERS

This chapter serves as an introduction to SUPERBUS operation at the most primitive level. We do not mean primitive in the powerless sense, but rather in the undeveloped sense. All of the SUPERBUS programs on the utility disk utilize the SUPERBUS features by manipulating various SUPERBUS parameters. By understanding the function of the SUPERBUS parameters, you may create powerful programs and tailor advanced systems to your own needs.

In the remainder of this guide, when a parameter is designated as a flag, the logical value true translates to 1 (one) decimal, and false translates to 0 (zero) decimal. Unless specified otherwise, all examples in the remainder of this chapter run in SUPERBUS manager's BASIC. To enter SUPERBUS manager's BASIC, choose option 'L' (LEAVE ...) from the system menu and then choose option 'm'. Be aware that when you enter manager's BASIC, the computer is initially running in enriched format at level 255.

In the examples in this guide, the computer's responses to the user's commands will be shown in **boldface print.** The user should not type the boldface text into the computer.

A. SECURITY PARAMETERS

1. CORFLG

Located at $EF80 (61312 decimal), default = false

As described previously, a SUPERBUS computer can run in either normal or enriched format. The flag called CORFLG determines the format that the computer is running in, where true represents enriched format and false represents normal format.

When a computer is running in normal format, that computer may read from or write to ANY FILE ON THE DISK. Be careful when running in normal format, and be especially careful not to grant normal format access to the wrong person. The integrity of your system is at the mercy of any uncontrolled user running in normal format.

A computer running in normal format may access any enriched format file. However, one must remember that normal format and enriched format are NOT compatible, therefore, a computer running in normal format will probably crash if it is instructed to load an enriched format program. However, with care, one may open an enriched format file and manually deal with the extraneous data. See FIGURE 1-X and FIGURE 1-XI to see an image of the two formats.

The inspector program on the SUPERBUS utility disk uses this feature to determine characteristics of an enriched format file. It opens the file while running in normal

format and examines the characteristics of the file.

On the other hand, a computer running in enriched format refuses to access normal format files. An enriched format computer will also refuse to access enriched format files stored at a level other than the current level of the computer, except for level zero, the librarian level. Any computer running in enriched format is permitted read access to files stored at level zero. A computer running in enriched format at level zero is granted no special privileges.

An enriched format file that has a key associated with it cannot be accessed by an computer running in enriched format without specifying that same key. Files with keys do not display the key in a directory listing. A key may be specified by appending the '←' symbol to the filename, followed by the key of 0 to 15 characters. Following is an example.

```
10 rem program with key
dsave"test0←frodo
00, ok,00,00

ready.
dload"test0

searching for 0:test0
?file not found error62, file not
found,00,00

ready.
dload"test0←niffleheim

searching for 0:test0←niffleheim12,wrong
key,00,00

?file not found error12,wrong key,00,00

ready.
dload"test0←frodo

searching for 0:test0←frodo
loading 00, ok,00,00

ready.
```

The first two commands enter a simple BASIC program and store it on the disk as file 'test0←', in enriched format at level 255 with a key of 'frodo'. The next command fails because the file name is 'test0←', not 'test0'. The following command fails also because the wrong key was specified. Finally, the key is matched with the last command and the file is loaded properly.

The save-replace feature of the 8050 disk drive cannot be used to write an enriched format file on top of a normal format file, since normal format files cannot be accessed while running in enriched format.

With enriched format relative files, the characteristics of the file are written to the first record. Be careful not to make the record size of a relative file too small, or the enriched format characteristics will become lost between records. Due to technicalities involving relative files, an enriched format relative file must have a key associated with it.

All enriched format programs (files that are manipulated through secondary address 1 or 0) are checksummed to insure proper storage and retrieval. The last three bytes of the program file on the disk are the low, middle and high bytes (respectively) of the checksum. The file characteristics are NOT checksummed, just the program itself. The SUPERBUS parameter TRYCNT determines how many times to attempt to load a file before giving up because of a checksum error.

The following is a BASIC program that reads the characteristics of an enriched format file while running in normal format. You can type in this program from SUPERBUS manager's BASIC, and run it with the utility disk in drive zero. This program is documented on the page following.

```
10 \f0
20 dopen#1,"menu"
30 gosub 130:lo=v:gosub 130:lo=v
40 gosub 130:hi=v:gosub 130:hi=v
50 la = lo+hi*256
60 gosub 130:lv=v
70 gosub 130:nk=v
75 if nk=0 then 110
80 for i=1 to nk
90 gosub 130
100 next i
110 dclose#1:end
130 rem get character in a$
140 get#1,a$:if a$="" then a$=chr$(0)
150 v=asc(a$):return
```

(FIGURE 3-I)

Following is documentation for the program shown on the previous page in FIGURE 3-I. It is not intended to be entered on your computer.

10 This line sets the SUPERBUS parameter CORFLG to false (zero), which indicates normal format. You may also set normal format by executing the following command: poke 61312,0

20 This line opens the enriched format file called 'menu' on the utility disk in drive zero. If you must have your utility disk in drive one, you may change the line to: 20 dopen#1,"menu",d1

Notice that the computer is running in normal format, but the file that is being opened is in enriched format. This is permissible, but as previously mentioned, loading an enriched format program (with the load or dload command) while running in normal format will probably crash your computer. Using the dopen command, we can look at the entire contents of the file as it is stored on the disk.

30 This line reads the first two bytes of the file by calling the subroutine located at line 130 twice. The first two bytes of an enriched format file are always the same and indicate the low byte of the load address of the file. The low byte of the load address is then stored in the BASIC variable 'lo'.

40 This line reads the next pair of bytes from the file by calling the same subroutine. These two bytes are equal to each other also, and indicate the high byte of the load address of the file. The high byte of the load address is then stored in the BASIC variable 'hi'.

The low and high bytes are repeated so that SUPERBUS knows that it is indeed looking at an enriched format file. Normal format files do not repeat the load address. This repetition is the only means by which SUPERBUS determines whether a file is in enriched format or normal format. It should now be clear why an enriched format program loaded while running in normal format usually crashes your computer - because the normal format computer does not expect the load address to be 4 bytes long instead of 2, and the normal format computer loads the program at the wrong location in memory.

50 This line computes the actual load address of the file and stores it in the BASIC variable 'la'. Most programs begin at 1025 or 1024 decimal. Even

enriched format sequential and relative files have dummy repeated load address so that SUPERBUS can recognize them. The actual load addresses of these files are meaningless, they are just there to fit the format.

60 This line reads a single byte that determines the level that the file was stored at. It then sets the BASIC variable 'lv' to the level of the file.

70 This line reads a single byte that determines how many characters long the key to the file is. It then sets the BASIC variable 'nk' to the number of characters in the key. If the file has no key, then 'nk' will have a value of zero.

75 This line checks to see if there is a key to the file by checking the length of the key. If there is no key, then the next three lines of BASIC are skipped.

80 -
100 Lines 80 through 100 read past the key of the file, but do not store it anywhere. When an enriched format file is stored with a key, an exclusive-or operation is performed on each byte of the key with the level of the file.

110 This line closes the disk file and ends the program execution.

130-
150 This subroutine reads a byte from the open file and assigns its numeric value to the BASIC variable 'v'.

The program shown in FIGURE 3-I can also be found on the utility disk as file 'demo', in enriched format at level 255 on the utility disk.

2. LEVEL

Located at $EF78 (61304 decimal), default = 255

This SUPERBUS parameter determines what level a computer is currently running at, from 0 (zero) to 255. Since LEVEL is only used in relation to enriched format files (when CORFLG = true), the value of LEVEL is ignored if the computer is running in normal format (CORFLG = false). When a computer is in enriched format and tries to access an enriched format file, the current value of the LEVEL parameter must match the level stored with the file for the operation to complete successfully. An exception is level 0 (zero), the library level. Any enriched format files stored at level 0 are readable by any computer running in enriched format.

Level 255 is reserved as the system level. There is nothing special about this level, except that it is the default level on computer power-up. System files and utilities are usually stored at level 255. Following are some examples.

```
10 rem this is a program
dsave"testl
00, ok,00,00

ready.
```

The previous lines enter a short program and save it under filename 'test', on the utility disk in drive zero. This file is stored in enriched format at level 255, the default when you enter manager's BASIC.

```
\115

ready.
dload"testl

searching for 0:testl3,wrong level,00,00

?file not found errorl3,wrong level,00,00

ready.
```

The line with the backslash in it tells the computer to change the SUPERBUS parameter LEVEL to 15. One could replace this command with: poke 61304,255

The next command, dload, instructs the computer to load the file previously saved into the computer. The error message 13,wrong level,00,00 is returned because the level of the computer was just changed to 15, and the original file was saved at level 255. The file is not re-loaded.

```
dsave"@testl
13,wrong level,00,00
00, ok,00,00

ready.
```

The dsave command instructs the computer to remove the old file from the disk and replace it with the program currently in memory. Again, an error message is generated reporting that the command cannot be executed because the computer is at the wrong level.

```
\1255

ready.
dload"test1

searching for 0:test1
loading 00, ok,00,00

ready.
```

Notice that after the computer's level is changed back to 255, the computer successfully loads the program previously stored.

3. SENBL

Located at $EF82 (61314 decimal), default = false

SENBL is a flag that determines whether the additional security features of SUPERBUS are in effect or not. The name SENBL stands for security enable. This parameter only processes commands passing through the command channel (secondary address 15) to the disk drive.

When SENBL is false, no command channel BASIC commands are restricted. You may header a new disk, backup disks, or execute whatever kind of DOS function you want. However, when SENBL is true, the only BASIC 4.0 DOS commands that may be executed are rename, scratch, record#, initialize and copy. If any of the other DOS commands are executed, then an error will be generated.

Access to the files involved is checked before the permitted commands are executed. For instance, when SENBL is true, the computer is in enriched format, and a scratch command is executed involving a file at a level other than the computer, the file will not be scratched. Following are some examples.

```
\s1:\115

ready.
collect d0
14,wrong format,00,00

ready.
rename "test1" to "test2"
13,wrong level,00,00

ready.
\s0

ready.
rename "test1" to "test2"
```

```
00, ok,00,00

ready.
```

The first two commands (seperated by the colon) set SENBL to true (turning on security) and set LEVEL to 15. The next command, collect, is not executed because that command is disallowed while security is turned on. The next command, rename, is allowed when security is turned on but is also not executed because file "test1" is at level 255, not level 15. Then security is turned off and the same rename command is executed successfully, EVEN THOUGH the computer is still at a different level. All DOS commands (not including load, save and open instructions) are allowed when security is turned off, regardless of level or format.

4. DTLEN

Located at $EF7F (61311 decimal), default = 0

DTLEN is the SUPERBUS parameter that determines how many BASIC tokens are to be absolutely disallowed. When a disallowed token is recognized, a syntax error message is generated and the current program ends execution. The table of tokens to be deleted is at DTOK, the deleted token table, which begins at $EF87 (61319 decimal). Refer to FIGURE 3-IV for a list of BASIC tokens and their numeric value.

DTLEN and DTOK can be used to set up an extremely secure system in which users can program in BASIC without harming the system. That is what restricted SUPERBUS BASIC does - it turns on security, sets up the level, sets enriched format and disallows the poke and sys instructions. This way, the user is free to do whatever he/she pleases, but cannot harm disk files because of SENBL, CORFLG and LEVEL, and cannot change those parameters because of DTLEN and DTOK.

Following is a simple example.

```
s=0

ready.
poke 61319,asc("s")

ready.
poke 61311,1

ready.
s=0

?syntax error
ready.
poke 61311,0

ready.
```

The first command functions correctly and sets the BASIC variable 's' to zero. The next command puts an entry in the DTOK table that specifies that the 's' token is to be disallowed. However, DTLEN is still at the default, so no tokens are actually yet disallowed. By executing the next instruction, DTLEN is set to one and the 's' token is disallowed from BASIC, as illustrated by the next command. The following command restores DTLEN to zero, and the 's' token is once again available.

5. KILSTP

Located at $EF84 (61316 decimal), default = true

KILSTP is the SUPERBUS parameter that determines whether the STOP key is to be processed or ignored. When KILSTP is true, the STOP key does not break out of programs, but is processed as any other key would be. It has an ASCII value of 3, and can be recognized by chr$(3).

6. LDSTP

Located at $EF85 (61317 decimal), default = false

LDSTP has no effect if KILSTP is false. When KILSTP and LDSTP are both true and the STOP key is depressed, the current program halts execution and the menu program is loaded and run. The current menu is determined by the MENLEN and MENPNT parameters. This is an excellent feature for menu-oriented systems.

LDSTP also has no effect if the first token of the current program is 'm', indicating a menu. This is to prevent the novice user from repeatedly pressing STOP and locking up the system.

B. UTILITY PARAMETERS

1. MENPNT

Located at $EF7D - $EF7E (61309 - 61310 decimal), default = $E91C (59679 decimal)

This parameter is a two-byte pointer stored in lo-hi format to the beginning of the current menu name in memory. The default location, $E91C, contains the name '.b', which is the first file loaded on power-up. The menu name can be anywhere in memory. This parameter is used in conjunction with MENLEN and MDN.

1. MENLEN

Located at $EF7C (61308 decimal), default = 2

This parameter determines the length of the current menu

name as determined by MENPNT. Since the default MENPNT points to '.b' in memory, the default length is 2 (two characters). This parameter is used in conjunction with MENPNT and MDN.

3. MDN

Located at $EF7B (61307 decimal), default = 8

This parameter determines the device number of the current menu program. Most disk drives have a device number of 8. Also, whenever control of the computer is passed into direct-mode BASIC, all files on the MDN device are closed. Therefore, if you are trying to write a file to the same device as the MDN, your file will be closed every time you see the 'ready.' prompt. This can be cured by temporarily changing MDN with a poke instruction, then you may write to your file without it being closed.

The \off command available from manager's BASIC and restricted BASIC simply sets CORFLG to true and calls the SUPERBUS routine that loads and runs the current menu as determined by MENPNT, MENLEN and MDN.

4. CHNFLG

Located at $EF86 (61318 decimal), default = false

CHNFLG determines whether BASIC variables are to be preserved between program loads. When CHNFLG is true, all BASIC variables will be preserved if a load or dload instruction is recognized and executed while running a BASIC program. CHNFLG has no effect if the load or dload instruction is in direct BASIC mode. A normal PET/CBM computer operates as if CHNFLG were true.

There is a bug in Commodore BASIC with relation to chaining programs - if the chained program is bigger than the original program, the variable pointers point into the middle of the chained program, and problems arise. SUPERBUS fixes this by allowing the user to set CHNFLG = false, so the variable pointers will be cleared after chaining.

When creating boot files with CHNFLG = true, some special programs are added to your boot file to solve this problem.

5. PENBL

Located at $EF83 (61315 decimal), default = 3

PENBL is the parameter that determines whether the current disk error message should be reported automatically after each disk function. A value of 0 (zero) represents no automatic error report. A value of 3 indicates automatic error report in direct mode BASIC, and a value of 255 indicates to always report the error message, even within a

program.

C. SYSTEM PARAMETERS

1. DSKLST

Located at $EF76 (61302 decimal), default = 255 (devices 8-15)

This parameter determines which devices are to be treated as disk drives. There are 8 devices numbers that may be entered as disk devices, device 8 through device 15. Bit 0 of DSKLST refers to device #8, bit 1 refers to device #9, etc., through bit 7, which refers to device #15. The special disk features are applied only to disk devices as determined by DSKLST.

To emulate a PET computer, you may want to disable your disk devices so that SUPERBUS does not treat them unusually.

2. SLWLST

Located at $EF77 (61303 decimal), default = 15 (devices 4-7)

This parameter is arranged in the same fashion as DSKLST, with bit 0 corresponding to device #4 and bit 7 corresponding to device #11. DSKLST is the list of slow devices, that is, devices that may be interrupted for higher priority functions. Printers are typically slow devices.

3. TRYCNT

Locaed at $EF7A (61306 decimal), default = 1

This parameter determines the numbe of times to try reloading an enriched format file if the checksum at the end of the file is inaccurate. After loading the file TRYCNT times without success, SUPERBUS gives up and terminates the command.

4. SECADD

Located at $EF79 (61305 decimal), default = 110

The SUPERBUS security system must have a secondary address of its own to the disk drive. This parameter determines what that secondary address is. To form the correct value for SECADD, add 96 to the secondary address that you wish to use. This secondary address must not be used anywhere else in your programs! If you are not securing data files, you may use 96 as your value for SECADD.

5. ERRVEC

Located at $EF74 - $EF75 (61300 - 61301 decimal), default = $FF69

ERRVEC is a vector pointing to a machine language routine to gain control upon entry to direct mode BASIC. For security, this can reset the computer upon entering BASIC, re-load the menu, execute your own routine, etc. The configure utility on the utility disk has several choices, including BASIC continue and BASIC run.

When a SUPERBUS computer is first turned on, the menu name is set to '.b', as determined by the MENPNT, MENLEN and MDN defaults. On power-up, ERRVEC points to the SUPERBUS routine that loads and runs the current menu. Now you should understand why the computer loads and runs '.b' when you turn it on.

To utilize normal BASIC, ERRVEC should point to $E5F8, the ready vector.

6. DTOK

Located at $EF87 - $EFFF (61319 - 61439 decimal)

DTOK marks the beginning of the disallowed token table. All of the disallowed tokens should be listed starting here. Since you are unlikely to disallow 120 tokens, there is also plenty of free space in this table. It is quite common to place the menu name right after the deleted tokens and point MENPNT to it. There is also room for various machine language routines in DTOK.

7. OPNFLG

Located at $EF81 (61313 decimal), default = true

OPNFLG is the flag that tells SUPERBUS how to handle open files. If OPNFLG is true on a computer, that computer will tie up the bus if it has any disk files (as determined by DSKLST) open. The bus will not be released until the file is closed. In this condition, the error channel is read and deposited into ds and ds$ twice - once after opening the file, and once after closing it.

If OPNFLG is false, the bus is not locked up while disk files are open, and the error channel is read after every disk operation.

8. RPETN

Located at $E91B (59675 decimal)

RPETN is the unique identification number of your computer. To find the number from 1 to 18 of your computer, take RPETN divided by fourteen. Following is an example.

```
print "my id # is";peek(59675)/14
my id # is 1

ready.
```

RPETN is the default priority number assigned to the computer, and it CANNOT BE CHANGED.

9. PETN

Located at $EF3E (61246 decimal)

PETN is the current priority number of your computer. RPETN is used as the default value for PETN, and the two parameters are stored in the same fashion. PETN is set to the value of RPETN on power-up, and is not altered by SUPERBUS afterwards. Be extremely cautious when altering PETN, for if two of your computers end up with the same priority, they may collide when they access the bus!

III. CONFIGURATIONS AND BOOT FILES

This chapter of the Programmer's Guide deals with a new method for manipulating SUPERBUS parameters. This method is much faster than laboriously setting each SUPERBUS parameter individually, and is totally secure. For instance, if a program used the poke command to set every parameter in a long list and security was desired, some problems would arise: should the STOP key be enabled before or after the poke and sys tokens are removed? If it is enabled before the tokens are removed, the user could simply press STOP and then do whatever he/she wanted. On the other hand, if the tokens were removed first, the STOP key could not be re-enabled because the poke command has been disallowed!

With the aid of a little machine language programming, situations like these are easily remedied by using configurations and boot files. With configurations and boot files, a manager can also easily define an entire environment for a computer, not just a seemingly random set of pokes. After some environments are defined, the computer can switch back and forth between them quite easily.

A. CONFIGURATIONS

A configuration is a set of values for all of the parameters discussed in chapter II. A configuration of SUPERBUS parameters defines the environment or state of the computer. Manually altering one or more SUPERBUS parameter indeed changes the current configuration of the computer. Although this is an acceptable means to alter the computer's current configuration, it is much more common to change the entire configuration of parameters at once. This way, the manager can be confident that the computer is in the proper environment. Boot files perform this function.

B. BOOT FILES

A boot file is a file that sets the all of the computer's SUPERBUS parameters to those defined in a configuration, thus completely altering the computer's environment. A boot file is non-interactive - it performs its functions upon loading, running automatically, and then chains to another program or drops into BASIC.

Boot files are created by utility programs or can be tailor-made to your specifications using the configure utility. They are machine-language programs, partly in BASIC.

C. CONFIGURE

The first program loaded into your computer when you turn it on is '.b', usually a boot file (though it can be any program). On the utility disk, '.b' is a boot file that sets up the initial environment and then chains to some other machine language programs that implement new BASIC commands.

What does this '.b' on the utility disk have to do? Since a computer powers up in normal format and most of the utility programs are in enriched format, '.b' had better set CORFLG to true and LEVEL to 255, the system level.

Since the system must be secure, '.b' should not let any users accidentally fall into BASIC because of a disk error, so ERRVEC should point to the computer's cold start routine at $FD16. After that is done, falling into BASIC would cause your computer to reset as if it were just turned on. This will effectively keep people from harming the system from BASIC.

If the STOP key were enabled, then a user could screw up the initialization process (though it would not be a security threat) by pressing STOP over and over. Therefore, KILSTP should be set to true. There is no reason for LDSTP to be true, so it should be left false. The other values for the other parameters will make little difference, if any, so they can just be kept at the default.

Now that we know what the computer's environment should be, we are ready to formalize our image into a configuration. Choose option 'C' (CONFIGURE options) from the system menu. You will see FIGURE 1-IV. On this page, choose option #1 to enter the system configure utility. You will see the following display.

```
0) Create a new configuration

1) n.b.normal
2) user boot.data
3) .b.s.data
4) logout.data
5) system menu.data
6) .b.data

Which to use?

   (run/stop to quit): *
```

This is the entry page for the configure utility. It presents you with a list of already existing configurations, and allows you to create a new one. The first one is a configuration for normal BASIC unsecure operation. The second, user boot.data, is a configuration used in conjunction with restricted SUPERBUS BASIC. The next one, .b.s.data, is an enriched format duplicate of .b.data. Number 4, logout.data, is used also in conjunction with restricted SUPERBUS BASIC, and number 5 is the utility program configuration.

Number 6, .b.data, is the configuration that we are

going to try to duplicate. It is the configuration used in the '.b' file on the utility disk. We do not want to use #6, though - it would be better as a learning experience to create our own. Choose option #0 to create a new configuration, and you see this prompt:

File name: **********

You may choose any name, regardless of whether it is already on the disk or not. We will not be saving this configuration, so it does not matter. You now see this display:

1) Edit this configuration

2) Resave this configuration

3) Delete this configuration

4) Create a boot file

RETURN to main options

Exiting without saving will lose your configuration: *

(FIGURE 3-II)

Choose option #1 to edit this configuration. When you choose to make a new configuration, all the parameters of the configuration are default. You now see FIGURE 3-III.

```
a) dsklst: 8,9,10,11,12,13,14,15

b) slwlst: 4,5,6,7

c) level  255          d) secadd 14

e) trycnt 1            f) mdn    8

g) menu   .b           h) dtlen  0

i) corflg f            j) opnflg t

k) senbl  f            l) penbl  3

m) kilstp t            n) ldstp  f

o) chnflg f            p) errvec #5

   RETURN for options or which: *
```

(FIGURE 3-III)

Notice that there are no MENLEN or MENPNT parameters shown on this screen. For ease of operation, the configure utility finds a place to put the menu name and computes MENLEN and MENPNT. All you need to do is enter the menu name under option 'g'.

You may also notice that ERRVEC has a peculiar value. The configure utility has 5 pre-defined commonly used vectors that you may specify by entering #1 through #5. Two of the vectors are special cases and include small machine language programs.

We now should change the display so that the parameters match what we want them to be for '.b'.

First of all, we decided that '.b' should set CORFLG to true. Looking at the screen, you will notice that the value for CORFLG is 'f', which means false. To change this value, press the letter associated with CORFLG (which is 'i'). You will see a screen that partially describes the functions associated with CORFLG, and then a prompt that looks like this:

```
Hit return for no change

New value: ***
```

At this prompt, enter the letter 't' and press RETURN. This indicates that CORFLG should be set to true instead of false. When using the configure utility, you may use 't' or '1' (one) to indicate true and 'f' or '0' (zero) to indicate false for flags. If you use this notation for a parameter

which is not a flag, you may receive an error message.

After you press RETURN, you will once again see FIGURE 3-III on your screen. Choose option 'p' to reset ERRVEC. You will see a display like this in the middle of your screen:

```
#1... Warm start (normal) ($E5F8)
#2... Cold start (reset) ($FD16)
#3... Run program (special case)
#4... Continue (cont) (special case)
#5... Load/run menu ($FF69)
```

You will also see a prompt like the one before to change the value of ERRVEC. The special options #1 through #5 deserve some explanation. The first, warm start, is the normal ready vector. When ERRVEC is $E5F8, the computer's normal ready routine gains control if the computer falls into direct mode BASIC. The second, cold start, resets the computer completely if the computer falls into direct mode BASIC. This is the one we want: type '#2' and press RETURN.

The next two are not really just vectors, but are provided by the configure utility. When #3 or #4 is specified and a boot file is created, configure appends a short machine language routine to the boot file that performs the desired function. The last special option, load/run menu, is the default.

At the prompt for the new value of ERRVEC you may also enter a decimal address to gain control at ready. In this case, you would supposedly already have a machine language program in memory by the time the boot file was loaded.

Looking at the display with all the parameters and their values, everything seems to be appropriate. We have just finished formalizing our concept of '.b' into a configuration.

Press RETURN alone, and you will once again see FIGURE 3-II. We are now ready to create our boot file from our configuration. Choose option #4 to create a boot file. You will see a screen full of text. Read this text and press RETURN when you are done. This is just a reminder of how boot files work, which we have already discussed.

If our configuration is not accurate or we make a mistake creating the boot file, the boot file on the utility disk will be gone, so be very careful! You may wish to go through all the steps but not actually create the boot file. If this is the case, keep reading and operating your computer and you will be notified when you are about to create the file.

You now see a prompt for the name of the boot file. Our

name is '.b', since we are going to try and recreate the '.b' on the utility disk. Enter '.b' without the quotes and press RETURN.

You now see a prompt for a drive number; enter the drive number that the utility disk is in. Next is a prompt for format, normal or enriched. Well, we know that when a SUPERBUS computer is turned on, it looks for '.b' in normal format, so we had better specify that '.b' be written in normal format. Press 'n' to specify normal format. Had we specified enriched format, we would have also been prompted for the level and key to the file.

You now see the following display:

```
Enter the name of the file (in normal
format) to append, which starts at
61321, or RETURN for none.
: ****************
```

Appending a file to your boot file is a very powerful feature. For instance, if you wanted to do some extra processing on entering direct mode BASIC, you could write a machine language program, save it in normal format, point ERRVEC to the beginning of the program (61321 in this case) and enter your filename here. In other words, if you enter a normal format file name here, the contents of that file will be loaded into your computer starting at 61321 decimal when you load the boot file. This feature is used in conjunction with restricted SUPERBUS BASIC.

However, our '.b' needs no special additions, so press RETURN alone here, indicating that you want no programs appended. You now see the following prompt:

```
After booting:
   Load a program
or New, dropping into basic?
: *
```

What should the computer do after it has changed all of the SUPERBUS parameters? We have two choices, either simply clearing memory and entering BASIC, or chaining to another BASIC program. There is no way for you to know this, but for the extra SUPERBUS commands to be implemented, '.b' should chain to 'off.chain'. Press the letter 'l' to indicate that a program should be loaded. Type in the program name, 'off.chain' without the quotes and press RETURN.

You now see a prompt for a device number, the device that 'off.chain' resides in. If you do NOT wish to create your boot file, press the STOP key a number of times until you reach FIGURE 3-II.

If you do wish to create this boot file (replacing the one on the utility disk), press RETURN to specify the last device used. There will be a pause, the disk drive will spin, and the boot file will be created. After it is created, you will see FIGURE 3-II.

It's time to try out the new boot file - press the STOP key, and you will return to the system menu. Choose option 'L' to leave the utility system, and you will see FIGURE 1-II. Choose option 'r' to reset your computer, and watch. Everything should start normally.

If for some reason the computer crashes in one way or another, something went wrong in the creation of the configuration or boot file, and you should retrieve your backup copy (do you know where it is?) of the utility disk. Reset your computer with the backup copy in drive 0 (zero) and the broken utility disk in drive 1 (one). Enter manager's BASIC, and enter the following two commands:

```
scratch d1,".b

are you sure ?y 01, files scratched,01,00

01, files scratched,01,00

ready.
copy d0,".b" to d1,"*"
00, ok,00,00

ready.
```

Your disk should now operate properly. Review the previous section and try to determine what went wrong.

IV. ADVANCED PROGRAMMING

Advanced programming should seldom be required when operating SUPERBUS. However, if the need arises or you are simply a machine language enthusiast, you will find this chapter helpful. Some other SUPERBUS parameters not included in boot files are discussed, various BASIC extensions are documented and some utility programs are explained in this chapter.

A. ADDITIONAL PARAMETERS

1. BASOLD

Located at $EF6E - $EF6F (61294 decimal), default =$0076

When the SUPERBUS relocator utility is first invoked to implement a new BASIC command, the CHRGET patch at $0071 - $0072 is moved to BASOLD. After CHRGET is invoked, SUPERBUS does its own processing and then transfers control to BASOLD. This is a usually a transparent parameter, but it can be altered at will. It is useful, for instance, when DOS or some such BASIC extension is loaded into a computer and the SUPERBUS relocator utility is run. In that case, the DOS commands will not be lost, because SUPERBUS builds a 'chain' of routines to call.

2. BASNEW

Located at $EF70 - $EF71 (61296 - 61297 decimal), default = $FF4A

BASNEW is a vector pointing to the first routine in the chain of BASIC patches. If an individual patch does not recognize the command, control should be passed through the next routine and so on until control is passed through $FF4A, the SUPERBUS patch. The SUPERBUS relocator utility takes care of this twisty road.

3. IRQVEC

Located at $EF72 - $EF74 (61298 - 61299 decimal), default = $E7CA or $F627 (POKE 40)

After SUPERBUS has processed the IRQ (interrupt request) signal, control is passed through this vector. If you change this vector, write a machine languagge program to do it using the 6502 SEI and CLI instructions. If you do not, the timing may be off in which case your computer will go zonkers. The stopper utility used in conjunction with restricted SUPERBUS BASIC modifies this vector to point at a program in memory, and the login and edit programs reset this vector to normal.

4. RUNFLG

Located at $0400 (1024 decimal), default = false

RUNFLG is the only SUPERBUS parameter that is not located in the SUPERBUS extra memory. RUNFLG has no immediate effect when it is set to true.

When a program that was saved with RUNFLG = true is loaded into the computer, two things happen. First, the program automatically runs. Second, DTLEN is set to zero, allowing all BASIC tokens. RUNFLG is always set to true in a boot file, which is why boot files always run automatically and can perform parameter-altering functions.

A program stored with RUNFLG = true can be loaded without automatically running if you use the monitor load instruction rather than the BASIC load. Following is an example.

```
ready.
\f0

ready.
sys4

b*
     pc  irq  sr ac xr yr sp
.;  0005 e455 30 00 5e 04 f6
.l ".b",08
searching for .b
loading 00, ok,00,00

.x
ready.
list

 10 p$=" ... unprintable machine language
program ...":sys1033
 20 dn=peek(212):::::::::::::::::
 30 load"off.chain",dn::::::
ready.
```

B. USER BOOT

Restricted SUPERBUS BASIC has already been discussed. This section deals with the programming problems involved in creating the proper environment and how they were solved.

What should a restricted BASIC user be able to do? What should he/she be restricted from? Well, a restricted BASIC user should basically be able to operate any BASIC function that is not dangerous. If the user is at a predetermined level in enriched format with SENBL true, the only dangerous functions short of smashing the disk drive are poke, sys and wait. In subsection 3 of section B in chapter V of the

Manager's Guide, a time limit for users was discussed. This time limit is handled through IRQVEC.

We can set up a configuration with the previous tokens deleted, but we have to patch the stopper program into IRQVEC so that we can limit the time accessing the bus.

Configurations do not allow for setting up IRQVEC. The easiest way to accomplish this is to write a short machine language routine to do it, append the program to the boot file, and point ERRVEC to the beginning of the appended routine. This way, when the computer falls into BASIC, IRQVEC will be set up nicely.

This is the way restricted SUPERBUS BASIC was handled. To enter restricted SUPERBUS BASIC from a program, you need to have three files in the disk drive; they are, 'user boot', 'user bootf' and 'user program'. All three files are in enriched format at level 255. The following short program should demonstrate nicely how to enter restricted BASIC from a program:

```
10 poke 61312,1:rem set enriched format
20 poke 61304,255:rem set level = 255
30 poke 61438,5:rem time on bus: 0=infinite
40 poke 61439,5:rem level to enter basic at
60 if peek (58873) = 4 then load "user bootf",8
70 load "user boot",8
```

C. ERROR RECOVERY

Most of the utility programs have an error recovery routine built in. The machine language segment is invoked when the system menu is loaded. When the computer enters BASIC for any reason, memory location 61437 becomes 1, the last executed line number is stored in 61438 - 61439, and the program is rerun. If one of the first lines in the program checks 61437, an error can be detected and reported.

The recovery program used on the utility disk is written in BASIC, and is called 'error recover' in enriched format at level 255 on the utility disk. The machine language part is in file 'error.b' in normal format.

To invoke the routine, set ERRVEC to the beginning of the machine language segment and then run whatever program you like. This can be easily accomplished by appending the machine program to a boot file and pointing ERRVEC to the beginning of the routine in DTOK.

For an example, examine 'configyour' in enriched format at level 255 on the utility disk.

D. THE SUPERBUS RELOCATOR

The SUPERBUS relocator is a valuable utility for machine language programming. It adjusts machine programs to the current top-of-memory so the programmer does not have to find a good place for the program. It saves on memory by packing programs closely, eliminating the need to program in reserved areas such as the cassette buffers.

All of the SUPERBUS BASIC extensions are implemented with the relocator. This way, the programs are relocated each time they are loaded, and do not destroy any patches into BASIC. The relocator builds a 'chain' of routines to process.

To use the relocator, your object code must be in two parts. The first part is a table of information that the relocator needs to be able to operate with your program correctly. The second part is your machine language program, which should fit a general pattern that will be discussed later.

The relocator data part of your program should begin at $0500, and your program should follow the data immediately. The data section is as follows:

$0500 - $0501	lo-hi pointer to the first instruction of your machine language program
$0502 - $0503	lo-hi length of your machine language program
$0504 - $0505	lo-hi pointer to vector for patch minus one
$0506 - $0507	pointer to initialization routine
$0508	length of adjust table
$0509	beginning of adjust table, each entry two bytes in lo-hi format, points to absolute indexed instructions that must be adjusted

The first instruction in your machine language program should immediately follow the last entry in the adjust table.

The relocator makes two assumptions about the general structure of your program. First, the first three bytes of your program must be an absolute jump to the default value of the vector that you wish to patch into. This jump instruction is adjusted by the relocator to form a 'chain' of routines that need to be called, according to what was in the vector before it was patched. Second, the first instruction to execute follows the previous jump instruction immediately. All absolute addressing in your program should be entered into the adjust table.

The initialization routine follows your machine language program. This routine is not adjusted and should not be

included in the length of your program. Following the relocation of your program, control is passed to your initialization routine unless the pointer to the routine is zero.

The example following should be helpful. It is the source code for the '\off' instruction used with the utility disk.

```
LOC   CODE       LINE
0000             ;****************************************
0000             ;
0000             ; \off = load current menu
0000             ;
0000             ; written by dave martin 1/82
0000             ;
0000             ;****************************************
0000             ;
0000             ; basic locations
0000             ;
0000             chrget =$70              ;basic chrget
0000             chrgot =$76              ;basic chrgot
0000             txtptr =$77              ;basic text pointer
0000             corflg =$ef80            ;SUPERBUS corflg
0000             ldmenu =$ff69            ;loads and runs menu
0000             basnew =$ef70            ;basic new vector
0000             ;
0000             relocator parameters
0000             ;
0000                    *=$500            ;start of relocate
0500 0b 05       start  .word quit        ;start of program
0502 24 00       length .word end-quit    ;length of program
0504 6f ef       vector .word basnew-1    ;new basic
0506 00 00       init   .word 0           ;no initialization
0508 02          modcnt .byte 2           ;length of table
0509 17 05       table  .word a1+1        ;alteration
050b             ;
050b             ; main program
050b             ;
050b 4c 76 00    quit   jmp chrgot
050e c9 5c       enter  cmp #'\'          ;activation
0510 d0 f9              bne quit
0512 a0 01              ldy #1
0514 b1 77       check  lda (txtptr),y    ;check next char
0516 d9 2b 05    a1     cmp name-1,y
0519 d0 0d              bne notus         ;doesn't match
051b c8                 iny
051c c0 04              cpy #end-name+1   ;length of name
051e d0 f4              bne check
0520 a0 01              ldy #1            ;command recognized
0522 8c 80 ef           sty corflg        ;corflg true
0525 4c 69 ff           jmp ldmenu        ;load menu, run
0528 a9 5c       notus  lda #'\'
052a d0 df              bne quit
052c 4f 46 46    name   .byte 'off'
052f             end    .end
```

In order to use the relocator with one of your programs, simply load the file called 'relo.b' in enriched format at level 255 on the utility disk. Then load your machine language binary file (starting at $0500) into memory with the dload command. Save the resulting file with the dsave command. When you run this program, your machine program will be relocated to the top of memory, vectors will be patched, and line 20 of the BASIC program will be run.

You can alter line 20 from 'relo.b' to print a message, chain to another file, or whatever will fit in the line. If you change the BASIC part of the relocator, you must keep it exactly the same size. You can use colons to pad the line if needed. Remember that BASIC tokens like print, load and new take only one byte for storage. To make sure that you haven't changed the size of the BASIC part, print fre(0) before and after the change and make sure the numbers match.

Several commands are implemented when you use the utility disk. The four files at the top of the directory with the '.chain' suffix are relocator files that chain to another program. The first, 'off.chain' is chained to by '.b'. The last file, 'sys.chain', chains to 'login' for user validation or 'system menu' if the utility disk is a standard system.

E. CONFIGURINE

In section C, the configure utility was discussed. There is another very similar utility called 'configurine'. This utility is intended to be used in conjunction with other utility programs. It allows a program to determine the requirements for a boot file and then creates the boot file without any user interaction. There are several configurine commands that may be used. The commands are read from a sequential file on the disk. Following is a list of the commands and their function.

format - the format to store the boot file in (1=enriched, 0=normal)

name - the name of the boot file

drive - the drive to store the boot file in

append - the name of the file in normal format to append

llevel - the level to store the boot file at, if enriched format

key - the key of the boot file, if enriched format

load - the file to chain to after

booting, '*' if none

device - the device # to load from ('dn' if none)

commence- to create the boot file (also created at the end-of-file)

report - error report: 0=none, 1=messages, 2=display all input

dtok - a string of one-byte tokens to be disallowed (use dtlen also)

message - end-of-processing message to screen, wait for RETURN, '*' if no pause

All of the SUPERBUS parameter names used in the configure utility may be used also. Each command starts a new line, with a space between the command and its operand. By default, configurine takes its input from 'moritz' in enriched format at level 255. You may alter the file name for input by printing the file name to the screen, followed by a colon, and positioning the cursor on top of the file name. Configurine reads the file name from the screen and uses it as input.

For a detailed example, look at lines 12230 through 12360 in file 'configyour' on the utility disk.

F. FILES ON THE UTILITY DISK

There are several files on the utility disk that have useful functions that have not been mentioned. Their description is included here because they would most likely be used by a programmer.

File Name	Description
.b.s.	A duplicate of '.b' stored in enriched format at level 255 (rather than normal format)
p.chain	Implements \p command (printer fix) and chains to off.chain
off.chain	Implements \off command and chains to g&t.chain
g&t.chain	Implements \g and \t commands and chains to r&w.chain
r&w.chain	Implements [and] commands and chains to sys.chain

File Name	Description
sys.chain	Chains to 'login' if an enhanced system; chains to 'system menu' if a standard system
user boot	Discussed in Chapter IV, section B - 8032 and 9-inch screen versions of user boot program
user bootf	Version of user boot that works with 12-inch 40-column computers
user program	A simple BASIC program that prints greeting messages and falls into BASIC
system menu	A boot file that sets up several SUPERBUS paramaters (including automatic error recovery) and chains to 'menu'
menu	The actual BASIC portion of the system menu
configyour	The 'Configure Options' from the system menu - either sets up a boot file via 'configurine' or chains to 'configure'
configure	User-defined boot file creation (see Chapter III), will run in 32K only
reloc.conf	Program segment required for configure and configurine utilities
runfix	Data file required for configure and configurine utilities
confix	Data file required for configure and configurine utilities
chnfix	Used in conjunction with configure/configurine - chnflg problem fix
chnflg	Used in conjunction with turnkey systems - chnflg problem fix
\p.hi.b	Used in conjunction with boot files to enter BASIC (created via 'Configure Options') - automatic printer fixer
\p.hi.b.chn	Used in conjunction with turnkey systems - automatic printer fixer
conv to normal	Convert to normal format utility
conv to enriched	Convert to enriched format utility
inspector	The File Inspector Utility

File Name	Description
inspect	Data file used in conjunction with inspector
edit	Authorization editor
fixer.in	First part of fixer program
fixer.out	Second part of fixer program
fixer.craft	Wordcraft special fixer processing
fixer.boot	Stored as '.b' on fixed disks - implements \p (automatic printer fixer) and chains to '.b.cont'
fixer.boot.cont	Stored as '.b.cont' on fixed disks - waits for user to set up printer
wp4.b	Special appended fixer program that fix some WORDPRO 4 bugs
wp4+.b	Appended WORDPRO 4+ (50.54) bug fixes
wp3.b	Fixes some WORDPRO 3 problems
relo.b	The SUPERBUS Relocator skeleton (see Chapter IV, Section D)
logout	Used in conjunction with restricted SUPERBUS BASIC - sets up some SUPERBUS parameters and chains to login
configurine	Configure-type utility (See Chapter IV, Section E)
supermon30923	Handy-dandy SUPERMON loads at 30923 decimal
demo	Discussed in Chapter II, Section A, subsection 1
\p	Relocates \p command at the top of memory
\off	Relocates \off command at the top of memory
\g&\t	Relocates \g and \t commands at the top of memory
r/w	Relocates [and] commands at the top of memory
\system	Relocates \s, \f and \l commands at the top of memory
user.boot.b	Machine language part of 'user boot' for 8032 and 9-inch screen computers

File Name	Description
user.boot.bf	Machine language part of 'user bootf' for 12-inch 40-column screen computers
error recover	BASIC program used with utility programs to report errors
error.b	Machine language part of automatic error recovery
u←	User Authorization File

The rest of the files on the disk are configure data files. To find the format and levels of these files, use the inspector utility and print out the results.

COMMAND	TOKEN	
	HEX	DEC
APPEND	D4	212
BACKUP	D2	210
CATALOG	D7	215
CLOSE	A0	160
CLR	9C	156
CMD	9D	157
COLLECT	D1	209
CONCAT	CC	204
CONT	9A	154
COPY	D3	211
DATA	83	131
DCLOSE	CE	206
DEF	96	150
DIM	86	134
DIRECTORY	DA	218
DLOAD	D6	214
DOPEN	CD	205
DSAVE	D5	213
END	80	128
FOR	81	129
GET	A1	161
GOSUB	8D	141
GOTO	89	137
HEADER	D0	208
IF	8B	139
INPUT	85	133
INPUT#	84	132
LET	88	136
LIST	9B	155

COMMAND	TOKEN	
	HEX	DEC
LOAD	93	147
NEXT	82	130
NEW	A2	162
ON	91	145
OPEN	9F	159
POKE	97	151
PRINT	99	153
PRINT#	98	152
READ	89	137
RECORD	CF	207
REM	8F	143
RENAME	D8	216
RESTORE	8C	140
RETURN	8E	142
RUN	8A	138
SAVE	94	148
SCRATCH	D9	217
STOP	90	144
SYS	9E	158
VERIFY	95	149
WAIT	92	146

(FIGURE IV)

Memory Address		Description
Decimal	Hexadecimal	
		VECTORS
61294-61295	EF6E-EF6F	BASOLD-old chrget vector
61296-61297	EF70-EF71	BASNEW-new command vector
61298-61299	EF72-EF73	IRQVEC-after irq vector
61300-61301	EF74-EF75	ERRVEC-ready trap vector
		PARAMETERS
61302	EF76	DSKLST-list of disk drives in system
61303	EF77	SLWLST-list of slow devices in system
61304	EF78	LEVEL-access level of computer
61305	EF79	SECADD-security secondary address
61306	EF7A	TRYCNT-times to retry load
61307	EF7B	MDN-menu device number
61308	EF7C	MENLEN-length of menu name
61309-61310	EF7D-EF7E	MENPNT-pointer to menu name
61311	EF7F	DTLEN-# of disallowed tokens
		CONTROL FLAGS
61312	EF80	CORFLG-superbus/pet file format
61313	EF81	OPNFLG-hold bus on open disk files
61314	EF82	SENBL-security enable
61315	EF83	PENBL-print enable for disk errors
61316	EF84	KILSTP-kill stop key
61317	EF85	LDSTP-load menu on stop
61318	EF86	CHNFLG-chain/clr variables
		MISCELANEOUS
61319-61439	EF87-EFFF	DTOK-deleted tokens and free space
1024	0400	RUNFLG-auto run of BASIC program
		SUPERBUS ROUTINES
59648	E900	VGET-Get control of bus
59651	E903	VCLR-Release control of bus
59654	E906	VLISTN-Listn/talk routine
59657	E909	VSECND-Secnd/tksa routine
59660	E90C	VCIOUT-Ciout routine
59663	E91F	VACPTR-Acptr routine
59666	E912	VUNLSN-Unlsn/untlk routine
59669	E915	VCLEAR-Clear slow devices
59672	E918	VBASIC-Initialize basic patch

SUPERBUS 4.4

Part 4

Appendices

Cyberia, Inc.
2330 Lincoln Way
Ames, Iowa 50010

Designed for use with PET / CBM computers

with BASIC 4.0 operating systems

I. INSTALLATION

A. CONTENTS

SUPERBUS consists of a circuit board, a user port edge connector, a manual, and a utility disk. Each SUPERBUS circuit board contains two ROM (read-only memory) chips, two RAM (random access memory) chips, five integrated circuits and a ribbon cable. A single SUPERBUS circuit board and user port board are required for each computer to be used on the SUPERBUS network.

Holding the circuit board so that the ribbon cable hangs straight down, the two SUPERBUS ROMS are the two large 24-pin integrated circuits labeled SUPERBUS F ROM (the left hand one) and E ROM (the right hand one). The F ROM is the same for all BASIC 4.0 SUPERBUS models. See FIGURE 4-I.

The SUPERBUS E ROM is dependent on the computer model in which SUPERBUS is installed. It also contains the computer's priority ID number (1 through 18). No two computers may have the same ID number with SUPERBUS. Currently there are three computer models available: the 8000 series, the 12-inch 4000 (also called the FAT 40), and the 4000 9-inch screen series (this includes all upgraded to BASIC 4.0 2001 computers). This ROM is labeled as <SUPERBUS> on the first line, <x000-yy> on the second line, and <E ROM ID-xx> on the third line.

For example, one labeled <4000-12> will work with any 4016 or 4032 large screen computer while a <4000-9> will work with any 4016 or 4032 small screen computer. The last line indicates the unique priority identification number for this SUPERBUS circuit board. Any two computers with the same ID number can collide and cause a system crash to occur.

B. CIRCUIT BOARD INSTALLATION

The SUPERBUS circuit board plugs directly into the main computer board at connector J9 (which is located at the rear of the main logic board on the right hand side) while the ribbon cable connects to the SCREEN EDITOR E ROM slot of the computer main logic board as outlined in FIGURE 4-II.

Installation Procedure:

1) Be sure that the computer is turned off, unplugged, and all connectors at the rear of the computer (such as IEEE, etc.) are removed.

2) Carefully remove the case screws from both sides. They are the Phillips type

screws under the overhang of the white casing.

3) Swing back the top and put the cover support arm in place.

4) Locate the SCREEN EDITOR ROM (ROM E) at :

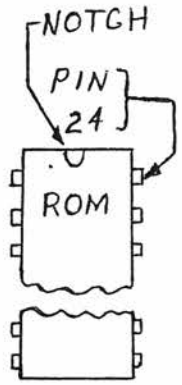
UD-8 for the 9-inch screen series
UD-7 for the 12-inch screen series

on the main computer logic board. Note the location of pin number 1, and then carefully remove this ROM from the socket by prying upwards with a screwdriver or standard IC chip remover. CAUTION--DO NOT force tool under the DIP SOCKET!

5) Locate the I/O ROM (ROM F) at:

UD-9 for the 9-inch screen series
UD-6 for the 12-inch screen series

on the main computer logic board. If this ROM is mounted in a DIP socket, carefully remove the ROM from its socket as before.



If the ROM is SOLDERED in place, clip the power pin (pin number 24) as close to the circuit board as possible and bend the pin slightly upward. Be certain that pin 24 is properly identified before clipping this pin. See adjacent sketch.

6) Connector J9 on the main computer logic board is located at the back right hand corner in front of the rear cassette port. This connector consists of two rows of 25 pins each. The SUPERBUS circuit board plugs onto pins 1 through 22 of the left hand row where pin 1 is the one closest to the front of the computer. See FIGURE 4-II.

7) Orient the SUPERBUS circuit board so that the IC components are on the bottom side of the board and the ribbon cable is extended toward the front of the computer. Now place the SUPERBUS row connector labeled J9 over pins 1 though 22 of the left hand row and firmly press the connector down until seated. It is normal for the SUPERBUS circuit board to angle upward to the left.

8) Insert the ribbon cable connector with the proper polarity into the DIP socket that previously held the SCREEN EDITOR ROM (ROM E). Pin 1 of the ribbon connector is marked with a blue dot. See FIGURE 4-I.

This completes the installation of the SUPERBUS circuit board.

C. EDGE CONNECTOR INSTALLATION

The standard SUPERBUS is supplied with a single user port edge connector with pins 3 and C connected together. One of these boards also contains a resistor and a wire that is attached to a smaller edge connector. This smaller edge connector is to be attached to the cassette port located at the right hand rear corner of the computer. Be careful to mount this connector with the side marked "TOP" on the top side. The user port edge connector is attached to the user port connection located in the middle of the three openings at the rear of the computer.

***** CAUTION **** **** CAUTION ****

The resistor used to hold the SERVICE REQUEST LINE high is about 6.0 kohms. Under NO CIRCUMSTANCES should more than one such power connection be used. Failure to follow this CAUTION can lead to SERIOUS DAMAGE to EVERY COMPUTER on the bus.

**** CAUTION **** **** CAUTION ****

Do not re-install the computer's casing screws until the system has been checked out. Continue with the above steps until all computers contain SUPERBUS boards.

D. RECOMMENDED DRIVE MODIFICATION

While not strictly necessary, it is recommended that pin 9 (interface clear) of a disk drive's IEEE connector be covered with a piece of electrically insulating tape. This prevents the drive from resetting every time a computer is turned on and will speed the initial loading process since the disk will not initialize every time a computer is powered up. This is extremely important to prevent loss of files that happen to be open when a computer is reset.

CYBERIA INC. has special IEEE cables of variable length and connectors available at modest cost that disable the interface clear line at each computer.

E. TROUBLE SHOOTING

SYMPTOM: When the computer is turned on, the screen is filled with random characters.

CAUSE: One or both of the ROMs have not been installed correctly. Check for pins not in the socket and the orientation of the ROMs.

SYMPTOM: The screen is cleared, but the computer "crashes" and does not respond.

CAUSE: The SUPERBUS circuit board is not installed correctly.

CAUSE: The SUPERBUS E ROM does not match the computer it is installed in.

SYMPTOM: When accessing the disk, two computers crash.

CAUSE: The two computers have the same SUPERBUS ID. Contact your dealer.

CAUSE: One of the computers does not have properly installed edge connectors.

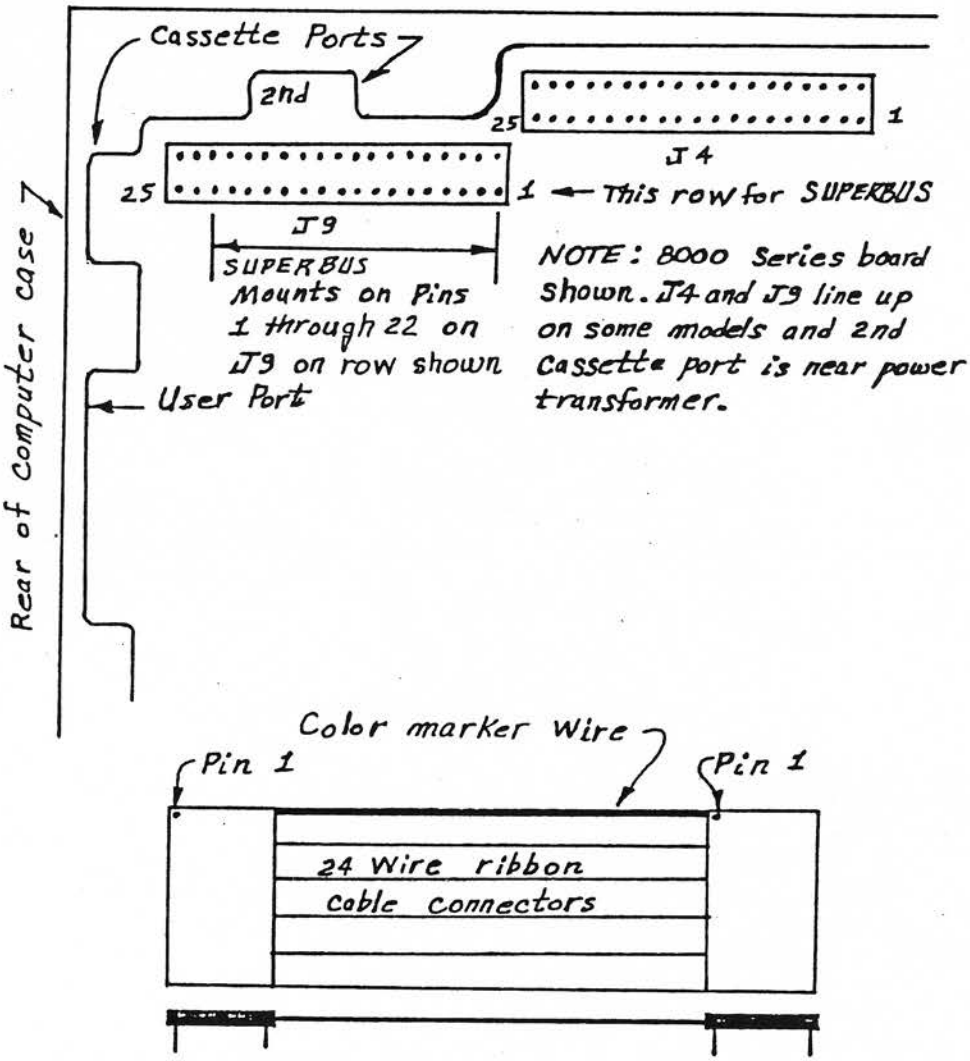
SYMPTOM: All computers come up with a ? in the upper right hand corner indicating that each is waiting for the bus.

CAUSE: No five volt power is applied to the service request line. (Make sure you have your connectors hooked up properly).

CAUSE: Several computers are turned off and the service request line voltage is less than 3.4 volts. Either turn on all computers or disconnect those not being used.

CAUSE: The 6522 IC controlling either the user port or cassette write line is defective. This can be found by disconnecting one computer at a time.

CAUSE: The 4022 printer is connected to the bus and is turned on. Turn off the printer and the system should start. Start the system according to Part I, Chapter II.



(FIGURE I and FIGURE II)

II. SUPERBUS DISCONNECTION

If you wish to use SUPERBUS on only a few of the computers hooked up, you must either disconnect those computers not being used from the IEEE-488 bus or turn them on and let them sit there. If too many computers are plugged in and turned off, it will drag the bus lines and nothing will happen.

If you actually want to remove SUPERBUS from your computer, simply reverse the installation steps. If you had to cut a pin on your ROM, bend the pin back in place.

III. MISCELLANEOUS

Several points need to be clarified here that are not mentioned elsewhere.

A. MODEMS

Modem use is not recommended with SUPERBUS, but it is possible. Several modems use the service request line (pin 10) of the IEEE bus. The use of this line conflicts with the SUPERBUS controller, and all the computers will probably lock up. To remedy this problem, mask off pin 10 on the modem with some electrically insulating tape.

After covering pin 10, the modem should work. If the computers all work but the modem doesn't, the modem program in the computer is dependent on the service request line and must be rewritten. BASIC programs will work nicely.

B. HARD DISK SUPERBUS

Cyberia, Inc., does not sell hard disk versions of the SUPERBUS utility disk. However, the utility programs can be easily transferred to a hard disk by your dealer.

There is one limitation, however - the fixer utility requires two drives to operate. If you are planning to use WORDPRO, WORDCRAFT or VISICALC, your dealer must fix these programs before transferring them to a hard disk.

C. COMMODORE 4040-2031 UTILITY DISKS

The utility disk on the Commodore 4040 is 25 blocks short of being full. If you need more space on the disk, you can make a new disk with just a few necessary utilities. This new disk will have the login, system menu, user boot, inspector and edit utilities. Several files must be on this disk, they are;

```
.b
.b.s.
off.chain
g&t.chain
```

```
r&w.chain
sys.chain
login
user boot
user bootf
user program
system menu
menu
edit
inspector
inspect
logout
\system
u←
```

The other utilities need not be copied. Make sure and keep the original copy of the utility disk in case you ever need it.

D. RIBBON CABLE CONNECTORS

The ribbon cable connectors available from Cyberia, Inc., are optional. They have several advantages. First, they are available at practically any length. They also mask off pin 11 (interface clear) of the IEEE bus, so that when a computer is power reset, the disk drives and printers do not reset. They are generally cheaper than standard IEEE cables.

The ribbon cable connectors come in two parts, the connector boards and the cables themselves. The connector boards connect onto the back of each computer's IEEE port. This port is the one closest to the power switch.

There are two sockets in the connector boards, each marked with a blue dot in the corner. The ribbon cables connect between these sockets, with the blue dots on the cables aligned with the blue dots on the connector boards. The cables should extend in the most convenient direction. If they don't seem to, try the other end of the cable.

After you have all of your computers linked to each other, you will need a PET to IEEE cable to connect your computers to your disk drive. Plug the PET end onto the vertical IEEE port of one of the connectors. The PET end of the cable should face TOWARDS the computer on the connector.

E. THE SUPERBUS USER PORT

The PA0 pin of the user port is used to power the service request line of the IEEE bus. All of the other pins are free for your use.

F. AUTOMATIC PRINTER FIXER

In previous versions of SUPERBUS, printers often had to be fixed by a BASIC program or they would lock up the entire

bus. With this release of the utility disk, printers are fixed automatically when you turn them on. Printers are checked through the computer's interrupt routine, and when an offending printer is detected, it is fixed.

There is a command not documented elsewhere in this manual that determines whether automatic printer fixing is enabled. The command is '\p'. The '\p1' (one, not the letter 'l') command turns on the automatic fixing capability and '\p0' (zero) turns it off.

When automatic printer fixing is turned on, previous interrupt routines are not lost. The old pointers are saved and the old routines are still executed. When the fixing is disabled, the previous value of IRQVEC is restored.

There is a file on the utility disk called '\p' in enriched format at level 255 that contains this program. For your own applications, you may find this useful.

G. MEMORY SIZE

All of the utility programs on the utility disk except one have been written to run in 16K RAM. This program is 'configure', which should not be necessary to operate most of the time. If you must create a boot file, you can use the configurine utility described in the Programmer's Guide.

INDEX

-- The Rest Of This Manual Is Intentionally Left Blank --