Owner's Manual

TURBO MASTER CPU™

4.09 MHz Processor Accelerator for Commodore C64

SCHNEDLER SYSTEMS
25 Eastwood Road
P.O. Box 5964
Asheville, North Carolina 28813

Telephone (704) 274-4646

TABLE OF CONTENTS

(Table of Contents continued on next page)

Introduction; Initial Comments

It is highly recommended that you read at least Chapters 1 and 2 of this manual before operating your Turbo Master CPU (TM) processor accelerator cartridge. However, Installation and "Quick Start" procedures begin on the next page.

You will find that Turbo Master CPU is very easy to use. Chapters 1 and 2 are all you really need to know, although you certainly will want to read Chapter 3 for a description of the additional features of Kernal ROM 1, like the DOS Wedge. GEOS procedures are detailed in Chapter 4.

Much of the length of this manual reflects our attempt to place in your hands all of the information you might reasonably need, and does not reflect complexity in operation.

Chapter 1: Overview and Basic Operation

1.0 Brief Description

Turbo Master CPU is a sophisticated plug-in processor accelerator cartridge for the Commodore 64 and 64C and C128 (in C64 mode) computers which combines an on-board 65C02 microprocessor clocked at 4.09 MHz (compared to the standard 1.0225 MHz) for four times faster processing speed, and Turbo DOS routines in EPROM for five to six times faster disk Load and Save (when used with a Commodore 1541 disk drive or close compatible). A DOS "wedge" is also contained in the EPROM for convenient access to the disk drive. The primary feature is the four times faster processing speed (which results in nearly twice the processing power compared to an IBM-PC/XT clocked at its standard 4.77 MHz!). Thus your programs will actually run faster. The Turbo DOS and wedge routines are provided as a convenience. Turbo Master CPU should not be confused with so-called accelerator cartridges which merely speed up disk access, without affecting processing speed.

The 65C02 is an enhanced version of the 6502/6510 as used in the C64. While the 65C02 has several additional instructions and addressing modes, it is fully downwardly compatible with the 6502. In addition to the 65C02 microprocessor, the Turbo Master CPU cartridge includes its own 64K of fast static RAM, a 32K EPROM and numerous support chips. In fact, Turbo Master CPU is practically a complete 64K microcomputer, lacking only a keyboard and screen.

The Turbo Master CPU cartridge simply plugs into the cartridge expansion port. There are no special jumpers or connections to be made. During operation of Turbo Master CPU, the 6510 microprocessor in the C64 is completely bypassed. The VIC (video) chip, SID (sound) chip, keyboard and serial bus port in the C64 all operate normally.

1.1 Installation

To avoid possible damage to both your computer and your Turbo Master CPU cartridge it is essential that your computer be turned OFF when plugging in the cartridge.

Install the two white plastic feet on the underside of the cartridge case if not already done. They snap right in. Carefully line up the edge connector and plug the cartridge into the computer's expansion port. Be sure it is seated firmly. That's all there is to it. The "Quick Start" procedure is described next, followed by a description of the switch functions in Section 1.3.

1.2 Quick Start Procedure

Here's a test and demonstration procedure to get started before moving on to a more detailed description. This procedure is for a Commodore 64. *If you are using a Commodore C128 or 128D, refer first to the special procedures of Section 1.4 below.*

1) Install the cartridge by plugging in as described just above. This first time, before turning ON the power, flip Switch No. 1 to the left to DISABLE, and each of the other three switches to the right (ROM 1, SOFT, 4 MHz).

2) Turn ON power to your computer. You should see the usual start-up screen message, and operation should be normal in all respects. If this doesn't happen, try the RESET switch one time. If that doesn't work, then you have a problem. Contact us.

3) Now flip Switch No. 1 to ENABLE. You should see the Turbo Master CPU start-up message (with white rather than light blue characters). If this doesn't happen, try the RESET switch one time. Again, if that doesn't work, then you have a problem. Contact us.

Note: Notwithstanding Steps 1-3, which are for initial testing purposes only, you will usually turn ON your computer with Switch No. 1 set to ENABLE.

4) Turbo Master CPU ROM 1 initializes with Turbo disk Load OFF, Turbo disk Save OFF, and DOS Wedge ON. If you have a 1541 or compatible disk drive, use the keyboard to enter @6. This will toggle ON the 1541-compatible Turbo disk Load routine.

Notes: The other feature, Turbo disk Save, is toggled ON and OFF with the @7 command. The RUN/STOP-RESTORE key combination returns to the default with both Turbo Disk Load OFF and Turbo Disk Save OFF, and the @6 and @7 wedge commands are again required to turn ON the features. RUN/STOP-RESTORE does not affect the status of the DOS wedge feature.

5) Be sure your disk drive is ON, and insert the supplied DEMO/UTILITIES disk. To ensure the DOS is connected, enter either @ or > to read the disk command (error) channel #15.

6) To view a disk directory enter either >$0 or @$0 or >$ or @$, whichever you prefer. This will appear only slightly faster than normal speed since the Turbo Load routine is not used for directory listings.

7) LOAD the first program file on the disk, which happens to be named "SPEEDSWITCH DEMO". Since this is the first program and since the DOS wedge is enabled, there are several ways: /0:* or /SPEEDSWITCH DEMO or /0:SPEEDSWITCH DEMO or LOAD "0:*",8 or LOAD "0:SPEEDSWITCH DEMO",8 or LOAD "SPEEDSWITCH DEMO",8. This happens to be a rather short program file, so the Load speed increase will not be especially apparent. As an added feature, you will see the Load start and ending (plus one) addresses reported in hex.

8) As a reminder to yourself as to what normal speed is like, flip Switche Nos. 3 and 4 to MANUAL 1 MHz. Use the keyboard and enter LIST to see the program listing scroll by. Now flip Switches Nos. 3 and 4 to SOFT 4 MHz. Enter LIST again. There should be no doubt in your mind you are in turbo mode!

9) Now RUN the little program. It doesn't do much except display patterns on the screen and switch between the fast 4.09 MHz clock speed and the slow 1.0225 MHz clock speed. (The technique employed for software speed selection is discussed in Chapter 6.) As an experiment, flip Switch No. 3 to MANUAL and note that Switch No. 4 then controls the speed regardless of the program's attempt to effect software speed selection. When you are done, be sure to leave Switches 3 and 4 flipped the same way. (The disk drive routines will not work when MANUAL 4 MHz is selected.)

1.3 Switch Functions

Four toggle switches vary the operation of Turbo Master CPU. Here are their functions:

No. 1 ENABLE Turbo Master CPU and its 65C02 microprocessor are operating. The on-board 64K fast static RAM is active, as is the on-board EPROM. The regular 6510 microprocessor in the C64 is not active. The Kernal and Basic ROMs inside the C64 are not active.

DISABLE Turbo Master CPU is disabled. The C64 and its 6510 microprocessor operate normally. This switch setting provides a means to revert to completely normal C64 operation without actually unplugging the cartridge. The 65C02 microprocessor in Turbo Master CPU is halted. This is not a "power" switch as Turbo Master CPU remains powered even when switched to DISABLE.

You can switch between ENABLE and DISABLE while your computer is turned ON, but don't expect your program to continue running where it left off when you switch. Usually a RESET will be necessary to properly start operation with whichever processor you have switched to.

No. 2 ROM 1 The "feature" Kernal ROM described in Chapter 3 is active.

ROM 2 The "compatibility" Kernal ROM is active.

When the computer is first turned ON or reset, the screen message indicates which ROM is active. You can switch between the two ROMs so long as you don't switch while one of the "feature" ROM routines is actually executing. If you switch from ROM 2 to ROM 1 without resetting you will need to re-enable the ROM 1 features as described in Chapter 3, Section 3.1.

No. 3	SOFT	Processor clock speed selection between 1.0225 MHz and 4.09 MHz is via a software switch. The software switch is Bit 7 (the high bit) of memory location $00. When cleared to "0", the fast 4.09 MHz clock speed is selected. When set to "1", the slow (normal) 1.0225 MHz clock speed is selected. Refer to Chapter 3 for more.
	MANUAL	Processor clock speed selection between 1.0225 MHz and 4.09 MHz is manually via Switch No. 4. Memory location $00 has no effect.
No. 4	4 MHz	If Switch No. 3 is set to MANUAL, then 4.09 MHz (fast) speed is selected. Switch No. 4 has no effect when Switch No. 3 is set to SOFT. Note: If Switch No. 3 is set to MANUAL and Switch No. 4 is set to 4 MHz then the serial bus (disk) routines will not work. Therefore, Switches Nos. 3 and 4 should normally be switched together. Either both to the right or both to the left.
	1 MHz	If Switch No. 3 is set to MANUAL, then 1.0225 MHz (slow) speed is selected. Again, Switch No. 4 has no effect when Switch No. 3 is set to SOFT.
RESET		A pushbutton is provided as a convenient hardware reset switch. This switch is deliberately positioned so as to be somewhat difficult to reach in order to avoid inadvertently pressing it. The RESET pushbutton functions regardless of whether Turbo Master CPU is ENABLEd or DISABLEd.

1.4 Special Procedures for C128 Operation

Beginning with our Revision "B" circuit boards, Turbo Master CPU is compatible with the C128 and 128D computers in the C64 mode. (Prior to Revision "B", stray screen characters would appear.)

However, when using the accelerator with a C128 or C128D (in C64 mode) it is of critical importance that the computer be in C64 mode before you enable the accelerator. Always power up with the accelerator switch No. 1 set to DISABLE. Enter C64 mode either with the GO 64 command, or by holding down the Commodore key during reset or power up. Only then should you switch the accelerator to ENABLE. The same procedure applies if you use either RESET pushbutton to do a hardware reset. If you do a hardware reset, either just before or at the same time switch the

accelerator to DISABLE, and repeat the above procedure. A soft reset via SYS 64738 can be done at any time without regard to the above procedures.

Note that the special ENABLE/DISABLE procedure is not necessary when the accelerator is used in a regular C64 or C64C. You can power up with the accelerator set to ENABLE.

Many C128 users (and all 128D users) will have a 1571 disk drive. The fast loader included in the accelerator is compatible with the 1571 only in its 1541 mode. To ensure the 1571 is in its 1541 mode, use the wedge command @UØ>MØ.

1.5 Power Supply Considerations

Although a number of low-power CMOS chips are used, Turbo Master CPU does consume substantial power. The power consumption is still within the capabilities of a standard Commodore C64 power supply. However, Commodore C64 power supplies are notoriously weak, which is why there is an aftermarket in replacement Commodore C64 power supplies. As an example, the Commodore 1764 RAM Expansion Unit for the C64, which also consumes substantial power, comes with its own "deluxe" replacement power supply.

For these reasons, we recommend the use of a replacement power supply for your C64. Either a regular commercial one, or the power supply which comes with the 1764 RAM expander.

Chapter 2: Miscellaneous Questions

2.0 Compatibility Questions

Turbo Master CPU closely emulates a C64 and is highly compatible with most software. This includes programs written in Basic, machine-language programs, programs which use bank switching to access RAM under ROM, programs which move screen memory, and bit-mapped graphics screens.

With a minor "patch", Turbo Master CPU is compatible with GEOS, and significantly speeds up GEOS screen operations. GEOS procedures are described in Chapter 4.

There are some programs that set up a bit-mapped graphics screen for 80-column text. Turbo Master CPU works as-is with most of these. However, there are a few where the screen is not visible until a special universal "patch" program is run, described in Chapter 5.

With some programs which have their own built-in fast loader or fast boot it is necessary to switch to MANUAL 1 MHz to Load, and then back to SOFT 4 MHz.

More specifically, an easily circumventable compatibility issue concerns serial bus timing for disk and printer access. The serial bus routines in the Kernal ROM depend on proper timing, and these routines are written with the assumption the processor clock speed is the standard 1.0225 MHz. For most programs this presents no problem whatsoever since the Kernal ROM (actually EPROM) in Turbo Master CPU is slightly patched to switch (using a software switch described in Chapter 3) to the slow 1.0225 MHz clock rate for serial bus access, and back to the fast 4.09 MHz clock rate when each serial bus access is concluded. However, there are some commercial programs which have their own built-in fast disk load routines which bypass the Kernal serial bus routines. These programs do not take into account the faster processor speed, and generally fail to work. The solution is to use the switches on the Turbo Master CPU cartridge to manually select the 1 MHz speed during the program load, and then back to the software-selected 4 MHz speed when the program is loaded. One example of such a program is "Pocket Writer 64". Once "Pocket Writer 64" is loaded, it uses the standard Kernal routines for loading and saving text files.

The fast disk Load and Save routines in the Turbo Master CPU EPROM of course know about the fast processor clock rate, and no special measures need be taken to use them.

Finally, the most serious potential impediment to compatibility is those commercial programs which use "illegal" or undocumented 6502 op-codes, usually as part of a protection scheme. The use of undocumented 6502 op-codes as a programming practice is condemnable since there is no guarantee by the microprocessor manufacturers that the undocumented op-codes will function the same way in later microprocessor chips with the same part number. Fortunately such programs are rare. (Assembly-language programmers using Turbo Master CPU can employ the documented and thus fully legal opcodes of the enhanced 65C02 instruction set. They really are great! Refer to Chapter 7 for more.)

2.1 Cassette Tape; Other Cartridges

Cassette tape cannot be used with Turbo Master CPU. The slow cassette tape is inconsistent with anything "turbo." Any attempt to Open Device #1 returns an illegal device number error.

In general, Turbo Master CPU *cannot* be used with plug-in ROM cartridges, *nor* with the 1764 RAM Expansion Unit. Turbo Master CPU *can* be used with Schnedler Systems data acquisition and control interface board products.

2.2 Commodore SX-64 Computer

In general Turbo Master CPU *will* work with the portable version of the C64, the SX-64. Some artifacts may appear on bit-mapped graphics screens due to a bus timing disturbance caused by a ribbon cable internal to the SX-64 connecting the main circuit board and the expansion socket.

We do not actually recommend use with the SX-64 because we do not know the rated capacity of the power supply built into the SX-64. Commodore does not recommend use of a RAM expander with the SX-64, and perhaps that is the reason. An SX-64 is now a rather rare and valuable computer, and it seems unwise to risk damage.

2.3 Turbo Master CPU Kernal ROMs Brief Description

Two versions of the C64 Kernal ROM are provided, switch selectable. ROM 1 in particular is described in detail in Chapter 3, but here is a brief introduction.

ROM 1 is a "feature" ROM. The cassette tape routines have all been removed and replaced by fast disk Load and Save routines and by a DOS wedge. Turbo Master CPU initializes with the DOS wedge active, and the fast disk Load and Save inactive but present. The DOS wedge commands have the same syntax as Commodore's DOS 5.1, but the wedge code itself

is different. Also, SHIFT/RUN-STOP generates the text string "/0:* [CR] RUN" which, assuming the DOS wedge is enabled, performs a Basic Load and Run of the first program on your disk. Care has been taken such that the "feature" ROM 1 code is highly compatible with most programs.

ROM 2, which is a "compatibility" ROM, is nevertheless provided. The Kernal cassette tape routines are present, but are disabled. ROM 2 does not have any extra features compared to a standard C64 Kernal ROM. It does have minor patches so that the serial bus works properly. There are some programs which, while not using cassette tape for I/O, do call Kernal ROM subroutines which would normally be considered as limited to cassette tape operations. For these ROM 2 is necessary. (One such program is the version of the Micromon machine-language monitor which has previously been supplied with some Schnedler Systems products. As a convenience to previous customers, a replacement Micromon compatible with ROM 1 is included on the supplied Turbo Master CPU DEMO/UTILITIES disk.)

2.4 JiffyDOS ROM Option Brief Description

A JiffyDOS ROM option is available for the Turbo Master CPU accelerator which allows you, if you already have or are getting JiffyDOS, to preserve the JiffyDOS benefits when using the fast processor. JiffyDOS differs from simple "turbo" disk load and save in that it also speeds up disk operations from within programs. The included JiffyDOS kernal ROM was developed by Creative Micro Designs, P.O. Box 789, Wilbraham, MA 01095, phone (413) 525-0023. The ROM has been specially modified to run in the accelerator, and is functionally identical to their regular JiffyDOS C64 kernal ROM replacement.

With the JiffyDOS option, you will have to be using a disk drive with a corresponding JiffyDOS ROM installed in order to get the benefits of JiffyDOS, and the option is not otherwise available. The JiffyDOS system is available from Creative Micro Designs, above. You will get the faster processor speed in either event.

JiffyDOS is installed in the "ROM 1" switch position. If you have this option, disregard all description of the features of "ROM 1" in this manual, and instead refer to your JiffyDOS manual. "ROM 2" is basically a straight kernal ROM for compatibility purposes, with no disk enhancements.

2.5 Other Disk Drives

Due to ROM space limitations, we could only include disk load and save code for 1541 disk drives and compatibles. The fast load and save are also compatible with the 1571, but only in its 1541 mode. (To ensure a 1571 is in its 1541 mode, use the wedge command @UØ>MØ.)

Other disk drives, like the 1581 and MSD drives do work with the fast processor, but not with the included fast disk load and save. It is for this reason (to avoid accidentally locking up the computer in the event a non-compatible disk drive is used), that turbo disk load and turbo disk save default to OFF.

With GEOS, which has its own "turbo" disk routines, all three drives 1541, 1571 and 1581 are supported. GEOS procedures are detailed in Chapter 4.

There is another way to have fast disk speed with drives other than a 1541, and that is to purchase a JiffyDOS system from Creative Micro Designs (contact them at the address and telephone above), and order the JiffyDOS option in the Turbo Master CPU accelerator.

2.6 What About Sound

The effect on programs that play music or otherwise make sounds depends on how the program is written to establish its timing.

In simple programs that use software delay loops to establish the time interval from one note to the next (e.g. FOR I=1 TO 1000: NEXT), the effect of Turbo Master CPU is to greatly speed up the tempo, often to the point where notes run into each other. The actual pitch (frequency) of the individual notes will be the same, because pitch is determined by values poked into the SID chip registers.

Some programs use IRQ interrupt timing or the jiffy clock to establish the time interval from one note to the next. With these the notes may play the same regardless of the processor speed. Refer to Chapter 6, Section 6.5 for more.

Chapter 3: "Feature" ROM 1 Description

3.0 Features Provided

ROM 1 provides three features in a highly compatible manner:

(1) Turbo Disk Load for use with a 1541 disk drive, or close compatible;

(2) Turbo Disk Save, also for use with a 1541 disk drive, or close compatible; and

(3) DOS wedge for convenient disk access, and also for convenient toggling ON and OFF of Turbo Disk Load and Save.

Upon reset, the initial status is:

Turbo Disk Load	OFF
Turbo Disk Save	OFF
DOS Wedge	ON

In addition, with "ROM 1", SHIFT/RUN-STOP generates the text string "/0:* [CR] RUN" which, assuming the DOS wedge is enabled, performs a Basic Load and Run of the first program on your disk.

For fastest processing speed, and even if all three features are disabled, the "ROM 1" switch position should always be selected unless "ROM 2" is absolutely needed for compatibility. See Chapter 6, Section 6.4 for the technical reason.

3.1 Enabling and Disabling Features

The features of ROM 1 can be enabled and disabled either by added DOS Wedge commands or by SYS commands given in the following chart. (Either way, enabling and disabling is actually effected by the called routine changing Kernal vectors.)

Feature	Toggle from Wedge	Enable	Disable	Toggle
Turbo Load	@6 or >6	SYS 64669	SYS 64649	SYS 64614
Turbo Save	@7 or >7	SYS 64703	SYS 64688	SYS 64632
DOS Wedge	@Q or >Q (Quits)	SYS 64134		

Obviously the DOS wedge must be enabled (the default condition) before @6 and @7 will work.

Also, the *RUN/STOP-RESTORE key combination returns to the default* with Turbo Disk Load and Save OFF, and commands as above are again required to turn ON these features. RUN/STOP-RESTORE does *not* affect the status of the DOS wedge.

As a matter of information, the three features are respectively enabled and disabled by redirection of the following three Kernal vectors:

ILOAD	$0330,$0331	Kernal LOAD routine vector
ISAVE	$0332,$0333	Kernal SAVE routine vector
ICRNCH	$0304,$0305	Basic Crunch Tokens vector

When a particular feature is not enabled, the corresponding vector is pointed to the appropriate standard Kernal ROM routine for full compatibility. The DOS wedge does not wedge into the Basic CHRGET routine and thus does not slow the execution speed of Basic programs (as does Commodore's DOS 5.1).

3.2 DOS Wedge Commands

The DOS Wedge commands are essentially the standard ones, with the addition of @6 and @7 to toggle Turbo Disk Load and Save OFF and ON. DOS wedge commands accessing the command (error) channel or the disk directory can begin with either @ or >, whichever you prefer. The two are equivalent. The wedge works in Basic direct mode only, not from within Basic programs.

Here is a summary:

/filename	Performs a normal Basic program Load. Equivalent to LOAD "filename",8.
↑filename	Does a Load then Run of a Basic program. Equivalent to LOAD "filename",8 followed by RUN.
←filename	Saves a Basic program in memory to disk. Equivalent to SAVE "filename",8. (Note: ← is left arrow, not CRSR left.)
%filename	Does a non-relocating (machine-language) Load without disturbing Basic's memory pointers. This command is similar to LOAD "filename",8,1 but not exactly the same since the regular Basic LOAD command does change Basic's pointers.

Notes: In each of the above commands quotes are optional and leading spaces before the command and before the filename are ignored. Each can be entered from a directory listing by typing over the number of disk blocks in the left-hand column. Instead of filename, the preferred form is 0:filename and should be used to minimize the Save-with-Replace bug in the Commodore 1541.

Continuing on with more commands:

@$ or >$	List Disk Directory to screen, leaving Basic program in memory intact. Preferred forms are actually @$0 and >$0. Pattern matching can be employed to obtain partial directory listings.
@ or >	Read disk command (error) channel #15. Normally used if red light on drive is flashing. Several DOS commands automatically perform this command upon their conclusion.

@command or >command — Sends a command to disk via channel #15. Here are some examples:

@N0:diskname,id	Format a disk (long new).
@N0:diskname	Clear disk directory (short new).
@I0	Read disk BAM to initialize drive
@V0	Validate a disk. Use whenever a file is left unclosed, as indicated by an * next to the file type. Never scratch such files.
@S0:filename	Scratch (delete) a file.
@R0:newname=0:oldname	Rename a file.

@#9 or >#9 Changes the wedge default to access disk device 9 rather than the default device 8. Change back with @#8 or >#8. Only disk devices 8 and 9 can be selected.

@Q or >Q Quit the wedge. Use SYS 64134 to re-enable.

@6 or >6 Toggle Turbo Disk Load ON and OFF.

@7 or >7 Toggle Turbo Disk Save ON and OFF.

3.3 Wedge Memory Usage

The DOS Wedge is highly compatible with software, and you will usually want to leave it enabled. Its code resides entirely in EPROM in space made available by removed cassette tape routines and so does not occupy any RAM memory. The wedge does not use any RAM in areas like the cassette tape buffer $033C - $03FB (decimal 828 - 1019). The only RAM variable the wedge takes as its own is BUFPNT ($A6,$A7), which in a standard C64 is used by the cassette tape routines. The wedge also uses the Kernal variable MEMUSS ($C3,$C4) as a temporary variable, but this should not result in any conflict.

Also, as already noted above, the DOS wedge does not wedge into the Basic CHRGET routine and thus does not slow the execution speed of Basic programs (as does Commodore's DOS 5.1). The wedge is wedged

into the Basic Crunch Tokens vector ICRNCH ($0304,$0305) and the wedge code is called only as each line is entered in Basic direct mode.

3.4 Turbo Disk Load Notes

The fast disk Load code likewise resides entirely in EPROM space made available by removed cassette tape routines and so does not occupy any RAM memory. This code is self-contained (i.e. the standard Kernal Load routines are left absolutely intact and unmodified) so that when the fast disk Load feature is OFF for program compatibility purposes you are assured of reverting exactly to the standard routine.

The fast disk Load routine works only with disk devices 8 and 9. It does not blank the screen. You can expect about 5-6 times the Loading speed, except with very short programs. Faster fast loaders are available, but they generally are not as compatible. The Load starting and ending (plus one) addresses are reported in direct mode. If a directory load is detected, the routine reverts to the standard Load.

The fast Load code is compatible with most printers and printer interfaces. However, like all software-based fast loaders, there may be some printers and interfaces which need to be turned OFF or unplugged before the fast Load code will work. (Fast loaders redefine the individual lines on the serial bus.)

Due to ROM space limitations, the included fast disk load and fast disk load are compatible with 1541 disk drives and compatibles only. The fast load and save are compatible with the 1571 only in its 1541 mode. (To ensure a 1571 is in its 1541 mode, use the wedge command @UØ>MØ.) Refer to Chapter 2, Section 2.5 above for more.

To avoid accidentally locking up the computer in the event a non-compatible disk drive is used, ROM 1 defaults with the fast Load code OFF. Section 3.1 above details how to turn it ON and OFF.

3.5 Turbo Disk Save Notes

In the same manner, the fast disk Save code resides entirely in EPROM space and does not occupy any RAM memory. This code is likewise self-contained so that when the fast disk Save feature is turned OFF for program compatibility purposes you are assured of reverting exactly to the standard routines.

The fast disk Save routine works only with disk devices 8 and 9. It is to be used only with a 1541 or close compatible disk drive. It does not

blank the screen. You can expect about 5-6 times the Saving speed, except with very short programs.

If you do a Save-with-Replace (i.e. SAVE "@0:filename",8 or ←@0:filename) the routine reverts to the standard Save. So if you have Save-with-Replace disk drive problems, don't blame it on Turbo Master CPU.

Printer and printer interface compatibility is as above. Some printers and interfaces will need to be turned OFF or unplugged before the fast Save code will work.

Fast Save as a feature is less important than Fast Load because you normally will do far more Loads than Saves. While we believe the code to be reliable, there is inherently more risk when saving to a disk, but almost none when loading from a disk. Accordingly, ROM 1 defaults with the fast Save code OFF. Section 3.1 above details how to turn it ON and OFF.

3.6 Switching Between ROM 1 and ROM 2

When the computer is first turned ON or reset, the initial screen message indicates which ROM is active. You can switch between the two ROMs so long as you don't switch while one of the "feature" ROM routines is actually executing. When you switch however you do not get a new message at that time; nevertheless the change takes place.

If you switch from ROM 2 to ROM 1 without resetting you will need to re-enable the ROM 1 features as described in Section 3.1 above.

If you switch from ROM 1 to ROM 2, the Kernal vectors which ROM 1 changes to enable the features are restored to their standard values the first time each corresponding function is used. Thus, after switching to ROM 2, the first time you enter a line in Basic direct mode the ICRNCH vector is automatically restored, disconnecting the wedge, and normal operation continues without your even being aware of it. Similarly, the Kernal LOAD vector is automatically restored and a normal Load continues the first time you Load a program. There is a slight exception in the case of a Save. If fast Save is enabled in ROM 1 and you switch to ROM 2, the first time you enter a SAVE command nothing will seem to happen. Just enter the SAVE command a second time, and all will be normal. We had room in the available ROM space to restore the standard SAVE vector for the next time, but not to continue on with a normal Save in the same automatic operation. For compatibility reasons, the automatic restoration code is patched into ROM 2 in the very few areas that are absolutely (we think) otherwise unused.

Chapter 4: GEOS Procedures

4.0 GEOS Overview

GEOS can run in Turbo Master CPU, and is greatly improved. The normally sluggish screens are greatly speeded up, especially when using the GEOS wordprocessors and scrolling through text. However, first a small utility program (either "PATCH GEOS" or "AUTO PATCH") must be run using the procedures described in Sections 4.1 through 4.3 below to modify the disk routines in GEOS to compensate for the faster processor speed.

The joystick for pointer control functions normally, and does not require further comment.

When the Commodore 1351 proportional mouse is used, however, the GEOS pointer will jitter wildly at the 4 MHz speed and be unusable unless a modified mouse input driver program is used. Four different modified mouse drivers are provided on the accompanying disk, "TM 1351", "TM 1351(A)", "TM 2B" and "TM L2R2L". The origin, use and characteristics of each are described below.

While Turbo Master CPU is not at present compatible with a Commodore RAM expander (REU), Turbo Master on its own greatly benefits GEOS. Depending on your activity at any given time, you may choose the accelerator with its benefit of the faster processor speed, or the REU for faster file operations. The REU of course does little to speed up screen operations like scrolling.

The GEOS "turbo" disk routines support all three drives, 1541, 1571 and 1581. This compatibility is maintained when the Turbo Master CPU is in use. An efficient system for GEOS includes two disk drives and the accelerator.

4.1 Procedures for GEOS at 4 MHz in Detail

Before booting GEOS, it is essential that you switch the accelerator to MANUAL 1 MHz. The reason for this is that GEOS has its own fast disk loader which doesn't know about the fast processor speed. (You will be able to switch to SOFT 4 MHz after the patch program has run.)

We offer two procedures for installing the patch program.
Procedure 1, described next, involves manually clicking on an icon.
Procedure 2, described thereafter, is more convenient and employs an auto-execute file copied to your boot disk.

The patch programs only modify the GEOS kernal code in memory. They do not alter your GEOS system disk in any way. The patch programs can for convenience be freely copied to any GEOS disk, using the GEOS file copy procedure.

4.2 GEOS Procedure 1 -- Program "PATCH GEOS"

After you have switched to MANUAL 1 MHz simply insert your GEOS system disk and boot normally. Then Close the GEOS boot disk, either by clicking or by using the Commodore/C key combination.

Next, insert your Turbo Master CPU disk, and open it, either by clicking or using the Commodore/O key combination. (As an alternative, note that the "PATCH GEOS" program file can be copied to any application disk, or even to your boot disk.)

The file you want is of course "PATCH GEOS", and has our rather crude attempt at an icon that depicts Turbo MaSTer. This same patch program works with either GEOS 1.3 or 2.0. It may work with 1.2.

To implement the patch, simply double click on the icon, or otherwise open the file. The patch is very fast. Wait for all disk activity to end, and then switch the accelerator to SOFT 4 MHz. Click on OK, and the GEOS desktop is reloaded.

4.3 GEOS Procedure 2 -- Program "AUTO PATCH"

"AUTO PATCH" is another version of the patch program. This version has a GEOS filetype of 14 and, when copied to a boot disk, it accordingly auto-executes before the DeskTop is loaded. The process becomes very convenient.

Note (1). The directory position of "AUTO PATCH" must follow that of "CONFIGURE" so that "CONFIGURE" auto-executes first.

Note (2). Do not leave "AUTO PATCH" on your boot disk if you ever use an REU with that boot disk. The patch overwrites REU code. "AUTO PATCH" can of course later be deleted from your boot disk. To simplify things note that, with GEOS 2.0, both the "System Disk" and the "Backup System Disk" are boot disks, so you can copy "AUTO PATCH" to one of these disks, and use the other when you don't want "AUTO PATCH".

4.4 Mouse Drivers for 4 MHz

Standard GEOS mouse input drivers do not work at 4 MHz, so we have provided slightly modified versions of four popular mouse drivers as summarized below. The modified versions work fine at 1 MHz also, so there is no problem in leaving them on your disks as defaults. The original author credit will be found in "file info" for each of the modified versions.

Original	New Name	Characteristics
COMM 1351	TM 1351	Standard mouse input driver.
COMM 1351(A)	TM 1351(A)	Accelerated mouse driver covers more area when moved quickly.
TWOBUTTONS	TM 2B	Prevents inadvertent double clicks. Left button can only single click. Right button can double click.
64 L2R2L 1351	TM L2R2L	Two features: (1) Right button jumps cursor left or right to opposite screen edge. Useful for quickly scrolling sideways in geoWrite and geoPaint documents. (2) Driver adjusts itself for mouse plugged into either Port 1 or Port 2. Default is Port 1. Call up file info and edit the first character of info text box to either "1" or "2" to specify the port.

By way of technical explanation, the problem with standard mouse drivers at 4 MHz is that the mouse uses the computer's analog potentiometer or "paddle" input. To read this input, a delay is required to allow the reading to settle after the driver program addresses the port. The software loops which establish this delay assume 1 MHz operation, and so provide only one-fourth the required delay at 4 MHz. The reading is erratic, and the screen pointer jitters wildly. The modified mouse input drivers have a couple of program lines inserted at the entry and exit points to ensure the processor is running at 1 MHz while the mouse driver code is executing during each interrupt, and to resume 4 MHz.

4.5 What to Expect from Accelerated GEOS

You can expect the normally sluggish GEOS screens to be greatly speeded up, like when scrolling through geoWrite documents and long geoPaint operations like pattern fill. Even plain typing is improved, because there is less delay between when you strike a key and the character is put on the screen.

GEOS disk operations are not speeded up because GEOS already has its own disk Turbo routines, and these are left alone by the patches.

When sending GEOS high-resolution output to your printer you will not notice any substantial increase in speed, assuming you are using the regular Commodore serial bus and perhaps an interface card. The reason is that printing bit-mapped graphics requires an extreme amount of data to be sent over the serial bus, and that is where the bottleneck is. For slightly higher speed, be sure to use the accelerator ROM 1 switch position. The real solution to the printer serial bus bottleneck is to use a geoCable parallel printer driver and a geoCable printer cable, or equivalent.

For some reason the Calculator desk accessory crashes the system. We've never investigated why.

4.6 Technical Explanation of Patch Programs

For the technically inclined or curious, here is a bit of explanation regarding the patch programs: The primary incompatibility is, as noted above, the fact that GEOS has its own Turbo disk routines that bypass the disk routines in the Turbo Master CPU Kernal ROMs and don't themselves know to slow the processor down for proper serial bus timing.

In the Official GEOS Programmer's Reference Guide from Berkeley Softworks we noticed two relevant routines. InitForIO is called before using the serial bus, and DoneWithIO is called when done with the serial bus. The patch is put in place by changing the GEOS jump table entries at $C25C and $C25F for InitForIO and DoneWithIO to JMP to added speed switch code at $9EC0 and $9EC7. The speed switch code overwrites REU code beginning at $9EBF within the DoRAMOP routine, memory locations we assumed would be available since an REU cannot presently be used with the Turbo Master CPU accelerator. We would prefer to use other memory locations, but we lack the detailed knowledge of the GEOS code necessary to select a safe location. The patches are placed into memory using GEOS' own InitRam routine, followed by a call to DoDlgBox. A JMP to EnterDeskTop exits the patch program. Here is the actual speed switch code:

```
9EBF-  60          RTS              ;Disable DoRAMOP code
9EC0-  A9 AF       LDA #%10101111   ;1 MHz
9EC2-  85 00       STA $00
9EC4-  6C 00 90    JMP ($9000)      ;InitForIO vector
9EC7-  A9 2F       LDA #%00101111   ;4 MHz
9EC9-  85 00       STA $00
9ECB-  6C 02 90    JMP ($9002)      ;DoneWithIO vector
```

Chapter 5: Bit-Mapped Text Screens

5.0 Background

Refer to this Chapter in those rare situations where a bitmapped screen you expect to show text instead shows an unchanging pattern of light and dark areas, usually either thin vertical bars or wider horizontal bars with steps.

There are some programs that, instead of the standard character set, use a graphics bitmap to set up a screen with up to 80 columns, such as for wordprocessing. Bitmapped mode involves an 8000 byte block for the bitmap itself, and an additional 1000 byte block for defining the two colors available in bitmapped mode. Programs can select different areas of memory for these two blocks of memory, and there is no particular standard.

Turbo Master CPU provides nearly perfect emulation of the C64 hardware, even to the extent of allowing machine language programs to exist and execute in the RAM under I/O and Character ROM $D000 through $DFFF, not to mention in the RAM under the Basic and Kernal ROMs $A000 through $BFFF and $E000 through $FFFF. However, one technical limitation of Turbo Master CPU is that Turbo Master CPU cannot store either a bitmap or bitmap colors in the range $D000 through $DFFF in a manner such that the VIC chip can access and display the information.

One configuration a program might establish is to put the 8000-byte bitmap under the Kernal ROM starting at $E000, and put the 1000 bytes of color somewhere in the $D000 through $DFFF range. Two programs which do this are "Pocket Writer 64" (from Digital Solutions) in the 80-column wordprocessor mode, and "Screen 80" published in Compute!'s Gazette, September 1984. Both of these programs give very nice 80 column text displays even on a C64 provided a good monitor is used. Turbo Master CPU speeds up the text scrolling quite nicely. (GEOS presents no problem whatsoever in this regard since the GEOS bitmap is under the Basic ROM beginning at $A000, and the bitmap colors are at $8C00 through $8FD7. GEOS has merely program code at $D000.)

5.1 Solution

Our solution is a universal patch program that fills the entire range $D000 through $DFFF with bitmap colors you select in advance. The application program will work; it just won't be able to change the colors itself. "Pocket Writer 64", for example, runs fine, except bold text shows the same as regular text.

Here's the procedure:

1) Before loading the application program, flip Switch No. 1 to DISABLE, and RESET if necessary.

2) In normal C64 mode, LOAD then RUN the patch program "SET BITMAP COLOR". You are given the opportunity to select the Pixel ON and OFF colors. The program defaults will give white characters on a black background. After you make your selection, the program is done almost instantly. Don't be surprised at that, and don't expect to see any immediate result.

3) Flip Switch No. 1 back to ENABLE Turbo Master CPU. Use the RESET, but do not turn OFF the power. Otherwise your colors will be lost.

4) Load and Run your application program. Hopefully the screen display will be visible. (Remember, since "Pocket Writer 64" has its own fast loader, you must select MANUAL 1 MHz during the Loading process, then switch back to SOFT 4 MHz.)

Note: Once "SET BITMAP COLOR" has been Loaded in Run in regular C64 mode, the colors remain in the C64's memory until you turn off the power. Nothing you do with Turbo Master CPU ENABLEd will disturb the colors.

Chapter 6: Speed Selection (Advanced Topic)

6.0 Speed Selection Overview

Two different methods of selecting between the standard 1.0225 MHz processor clock speed and the fast 4.09 MHz clock speed are briefly mentioned in the previous Chapters. The two methods are manually via the switches and via a software speed switch. They are described in greater detail here.

Normally you will not have to be concerned with software speed selection in your programs since the Turbo Master CPU Kernal ROM takes care of it automatically when necessary (for serial bus access). But we are giving you the details just in case.

6.1 Manual Speed Switch

Manual speed selection is enabled when Switch No. 3 is flipped to MANUAL. Then Switch No. 4 manually controls the processor clock speed. Refer back to Section 1.3 for a detailed description of these switches.

The usual reason for selecting MANUAL 1 MHz is to allow a program's built-in fast disk loader to operate when that program doesn't know to allow for the faster processor clock speed of Turbo Master CPU. Timing is critical in serial bus operations. Selection of MANUAL 1 MHz is not necessary when using the fast disk Load and Save routines built into Turbo Master CPU. Sometimes you may wish to select MANUAL 1 MHz to slow down screen scrolling.

While it is perfectly acceptable to select MANUAL 4 MHz when a program is merely running, again note that with the MANUAL 4 MHz selection the serial bus (disk) routines in EPROM will not work because these routines are prevented from performing a software speed selection. Therefore, *Switch Nos. 3 and 4 should normally be switched together. Either both to the right or both to the left. Always following this procedure makes it impossible for you to inadvertently select MANUAL 4 MHz.*

6.2 Software Speed Selection

When Switch No. 4 is flipped to SOFT, processor clock speed selection between 1.0225 MHz and 4.09 MHz is via a software switch. The software switch is Bit 7 (the high bit) of memory location or address $00. When cleared to "0", the fast 4.09 MHz clock speed is selected. When set to "1", the slow (normal) 1.0225 MHz clock speed is selected. The speed selection can be accomplished in either Basic or assembly language.

In the examples below both a rigorous approach and two "shortcut" approaches are given. The rigorous approaches will allow your programs to run even in a standard C64. Caution: The "shortcut" approaches may run only with Turbo Master CPU and run the risk of crashing a standard C64 by disturbing the memory bank configuration.

To switch to the standard 1.0225 MHz clock speed in Basic following the rigorous approach:

```
POKE 0,((PEEK(0) AND 127) OR 128)
```

To switch to the fast 4.09 MHz clock speed in Basic following the rigorous approach:

```
POKE 0, (PEEK(0) AND 127)
```

Programmers will appreciate that the above program lines affect only Bit 7 of Register 0, and leave Bits 0 through 6 undisturbed, since decimal 127 is hex $7F and binary %01111111. You will see examples of the above in the program "SPEEDSWITCH DEMO".

Here's one shortcut approach to switch to the standard 1.0225 MHz clock speed in Basic:

```
POKE 0,175
```

To switch back to the fast 4.09 MHz clock speed:

```
POKE 0,47
```

To understand these two, note that the standard value for register $00 is decimal 47, which is hex $2F, which is binary %00101111. POKE 0,47 thus makes the high Bit 7 a "0". Decimal 175 is hex $AF, which is binary %10101111. POKE 0,175 thus makes the high Bit 7 a "1", leaving the other bits at their standard values. Despite the "caution" above, both of these POKEs are in fact rather safe in the event you run your program in a standard C64.

Here's another shortcut approach to switch to the standard 1.0225 MHz clock speed when using Turbo Master CPU. Its virtue is that it is easy to remember, but should definitely not be used in any program you expect to also run in a standard C64:

```
POKE 0,128
```

Decimal 128 is hex $80, which is binary %10000000.

Here's the corresponding other shortcut approach to switch to the fast 4.09 MHz clock speed when using Turbo Master CPU, likewise easy to remember:

POKE 0,0

Here are rigorous approaches in assembly language:

```
hhhh-  A5 00     LDA $00
hhhh-  09 80     ORA #%10000000  ;Set Bit 7 to select 1 MHz
hhhh-  85 00     STA $00

hhhh-  A5 00     LDA $00
hhhh-  29 7F     AND #%7F        ;Clear Bit 7 to select 4 MHz
hhhh-  85 00     STA $00
```

These next two are a little shorter, but nevertheless are probably safe to run even in a standard C64:

```
hhhh-  A9 AF     LDA #%10101111  ;Bit 7 set to select 1 MHz
hhhh-  85 00     STA $00

hhhh-  A9 2F     LDA #%00101111  ;Bit 7 set to select 4 MHz
hhhh-  85 00     STA $00
```

Finally, here are two which definitely should be used only in programs you intend to run only with Turbo Master CPU. They are quite efficient because they are just a single two-byte instruction each. However, they use instructions available only in the enhanced 65C02 instruction set. These are the ones we use in the Turbo Master CPU EPROM:

```
hhhh-  F7 00     SMB7,$00        ;Set Memory Bit 7 to select 1 MHz

hhhh-  64 00     STZ $00         ;Store zero to select 4 MHz
```

6.3 Technical Details of Software Speed Selection

By way of technical background, in a standard C64 address $00 is a register address assigned to D6510, the 6510 on-chip data-direction register. The bit pattern in register $00 controls the I/O status of the individual lines of the corresponding I/O Port at register address $01. As noted above, register $00 is normally initialized by the operating system to decimal 47, which is hex $2F, which is binary %00101111. Bits 0, 1 and 2 of register $01 (LORAM, HIRAM and CHAREN) are thus all set to output.

LORAM, HIRAM and CHAREN control the memory bank configuration in a C64, and are used by some programs to access RAM under ROM and under I/O. Bits 3, 4 and 5 of register $01 relate to the cassette tape unit. Bits 6 and 7 are unused. Programs should not (and typically do not) disturb register $00, which is why we selected it for the software switch register.

In Turbo Master CPU, register $01 is hard-wired as an output-only port so that Bits 0, 1 and 2 always drive LORAM, HIRAM and CHAREN, except during and immediately following a hardware reset. Following a hardware reset, the hard-wired configuration takes effect upon the first write to register $00 during Kernal initialization. Bits 3 through 7 of register $01 are not connected. (They can with care be used as bit memory locations.) Register $00 is used only as the software speed switch, not as a data-direction register, and thus has no effect whatsoever on the I/O status of the lines of register $01.

6.4 Effect of Serial Bus Access on Speed

When the "ROM 1" switch position is selected, the accelerator in effect maintains the faster processing speed even while the serial bus is active. The accelerator switches to 1 MHz for the absolute minimum time necessary to preserve serial bus timing. Rather than waiting for disk or printer activity to cease before resuming the fast speed, "ROM 1" switches back to 4 MHz after each byte is transferred over the serial bus. With many programs this results in an apparent increase in speed.

When the "ROM 2" switch position is selected, the accelerator switches to the slow 1.09 MHz speed when serial bus access begins, and back to 4.09 MHz whenever a serial bus access ends, specifically when either the Kernal UNLSN or UNTLK routine is called. This is a more conservative approach from the point of view of compatibility, but has the disadvantage that the accelerator runs at 1 MHz all the while a disk channel is open, and does not necessarily switch back to the fast 4 MHz speed after each byte is transferred. For this reason we recommend you use "ROM 1" even if all the features are switched off, and reserve "ROM 2" for use only when absolutely required for software compatibility.

In either case, if your program for some reason does a software switch to 1.09 MHz, be aware that a serial bus access will leave the clock rate at 4.09 MHz.

6.5 Programs That Don't Seem to Run Faster

As noted in Chapter 2, in the context of programs that play music, some programs use interrupt timing or the jiffy clock to establish the time intervals between events. Typically IRQ interrupt-driven routines running in the background are involved. Games are typical. With such programs, some or all of the events may occur at the normal speed regardless of the processor clock speed. So don't assume that something is wrong just because a particular program doesn't seem to run any faster even when you switch to MANUAL 4 MHz.

As one example of this, observe the rate of the cursor blink. It's the same whether the processor speed is 1.0225 or 4.09 MHz! That's because the cursor blink is maintained as a part of the IRQ interrupt routine. An IRQ interrupt in the C64 is normally initiated by a 6526 CIA timer every 1/60 of a second, and this function is unaffected by Turbo Master CPU. "Jiffy" clock updating and the keyboard scan are performed by the same interrupt routine. So Turbo Master CPU does not disturb the Jiffy clock. As another example, the program SPEEDSWITCH DEMO uses the Jiffy clock to determine when ten seconds have elapsed, and this ten seconds stays constant regardless of the processor clock speed.

6.6 Determining Accelerator Presence and Speed

There may be times when you will wish to determine from within a program whether the accelerator is present and, if so, its clock rate. Here briefly are programming techniques.

For a presence test, take advantage of op-code $3A. In a straight 6502/6510 this is an undocumented NOP. Does nothing. In a 65C02, as used in the accelerator it is a DEC A instruction. Decrements the .A register.

In the Basic ROM, at $A86F (decimal 43119) there happens to be the sequence $3A $60. $60 is an RTS for either 6502 or 65C02. So, in Basic:

```
10 POKE 780,1 :   REM  SET .A TO 1
20 SYS 43119  :   REM  EXECUTE THE $3A THEN RTS
```

Then a PEEK(780) will return the initial value of 1 for the straight 6510 in the C64, and will return a value of 0 if the 65C02 accelerator CPU was present, regardless of clock speed.

The simplest way to determine speed is to PEEK(0), then look at bit 7. A value less than 128 indicates bit 7 is 0, and thus the software speed

switch is set to 4 MHz. A value greater than or equal to 128 indicates bit 7 is 1, and thus the software speed switch is set to 1 MHz.

The above simple test assumes that Switch 3 is set to "SOFT". The test however is meaningless if Switch 3 is set to "MANUAL". As a universal alternative, the following empirical procedure uses a CIA timer (which runs at a speed unaffected by the accelerator) to gauge how fast the Basic interpreter is running.

```
10 TAH = 56325   :   CRA = 56334
20 POKE CRA,0    :   POKE CRA,17  :  TV=PEEK(TAH)
```

A value for TV less than 60 indicates 1 MHz, while a value greater than or equal to 60 indicates 4 MHz.

Chapter 7: Enhanced 65C02 Opcodes (Advanced Topic)

7.0 Good News for Assembly Language Programmers

Turbo Master CPU allows you to use the enhanced 65C02 instruction set. There are some nice ones. See the next section. Several companies make 65C02 microprocessors. They are similar to each other, but there are slight differences. So it is worth mentioning we are using the Rockwell version, and intend to continue. Of course any programs you write using instructions unique to the 65C02 will not run in a regular C64 with its 6502/6510 processor.

7.1 Brief Introduction to Available Instructions

Here are some of the additional instructions:

BRA Branch Always. (No need for CLC BCC.)
STZ Store Zero. (Quite handy at times.)
PHX Push .X on stack. (Why bother with TXA PHA?)
PLX Pull .X from stack
PHY Push .Y on stack
PLY Pull .Y from stack

There is a whole set of instructions for setting, clearing, and conditionally branching on individual memory bits. You are not limited to Bits 6 and 7. These include TSB, TRB, SMB, RMB, BBS and BBR.

In addition to the entirely new bit addressing mode, new addressing modes are added to existing instructions. Like INC A -- increment the accumulator. There is a straight indirect addressing mode. So instead of clearing .Y to 0 and then doing LDA (ZP),Y just do LDA (ZP).

The text files "65C02 OPS.DOC" and "INSTR MODES.DOC" on the DEMO/UTILITIES disk have a more complete description for those who wish to print them out. A reference source is "R6500 Programming Manual". It is available from Rockwell International, P.O. Box C, Newport Beach, CA 92658-8902, Attention 501-300. The cost is $10.00. Telephone (800) 854-8099 or, from California, (800) 422-4230.

7.2 Assembler Available

There is a Commodore 64 assembler available which handles the enhanced 65C02 instruction set. This is the MAE assembler from Eastern House Software, now published by and available from Schnedler Systems. That is the assembler we used to develop the EPROM programming in Turbo Master CPU.

Chapter 8: Hacking the EPROM (Advanced Topic)

If you don't like our ROM and want to roll your own, here is the information you'll need. The EPROM is a 32K x 8, 150 nanosecond, type number 27C256-15. A Jason-Ranheim "Promenade" can be used.

The memory map as stored in EPROM is as follows:

EPROM Address	Use
$0000 - $1FFF	8K Basic ROM. Mapped to $A000 - $BFFF.
$2000 - $2FFF	4K Unused. Filled with $FF. Not addressable.
$3000 - $3FFF	4K Character ROM. Mapped to $D000 - $DFFF.
$4000 - $5FFF	8K Kernal ROM 1. Mapped to $E000 - $FFFF.
$6000 - $7FFF	8K Kernal ROM 2. Mapped to $E000 - $FFFF.

Here are the bare minimum Kernal ROM patches needed to software switch to 1.0225 MHz for serial bus access, and then back to 4.09 MHz. Note that we used some 65C02 instructions assembled with the MAE assembler:

```
E4B7- F7 00       SMB7, $00     ;Switch to 1 Mhz
E4B9- 4C 97 EE    JMP $EE97
E4BC- 64 00       STZ $00       ;Switch to 4 Mhz
E4BE- 4C 97 EE    JMP $EE97

ED24- 20 B7 E4    JSR $E4B7
ED3A- 20 B7 E4    JSR $E4B7
ED41- 20 B7 E4    JSR $E4B7
EE10- 4C BC E4    JMP $E4BC     ;Note JMP, not JSR
EE2A- 20 B7 E4    JSR $E4B7
```

The following additional patches are to maintain the faster processing speed even while the serial bus is active as described in Chapter 6, Section 6.4 above:

```
EA73- 85 95       STA $95
EA75- 58          CLI
EA76- 18          CLC
EA77- 64 00       STZ $00
EA79- 60          RTS

EDEB- 4C 73 EA    JMP $EA73
EE82- 4C 75 EA    JMP $EA75
```

Index

Quick Reference Chart

Upon reset, the initial status of "Feature" ROM 1 is:

Turbo Disk Load	OFF
Turbo Disk Save	OFF
DOS Wedge	ON

SHIFT/RUN-STOP generates the text string "/0:* [CR] RUN" which, assuming the DOS wedge is enabled, performs a Basic Load and Run of the first program on your disk.

Feature	Toggle from Wedge	Enable	Disable	Toggle
Turbo Load	@6 or >6	SYS 64669	SYS 64649	SYS 64614
Turbo Save	@7 or >7	SYS 64703	SYS 64688	SYS 64632
DOS Wedge	@Q or >Q (Quits)	SYS 64134		

The *RUN/STOP-RESTORE key combination returns to the default* with Turbo Disk Load and Save OFF, but does *not* affect the status of the DOS wedge.

DOS Wedge Commands

/filename	Performs a normal Basic program Load.
↑filename	Does a Load then Run of a Basic program.
←filename	Saves a Basic program in memory to disk.
%filename	Does a non-relocating (machine-language) Load without disturbing Basic's memory pointers.
@$ or >$	List Disk Directory to screen.
@ or >	Read disk command (error) channel #15.
@command or >command	Sends a command to disk via channel #15.
@#9 or >#9	Changes the wedge default to access disk device 9 rather than the default device 8. Change back with @#8 or >#8. Only disk devices 8 and 9 can be selected.
@Q or >Q	Quit the wedge. Use SYS 64134 to re-enable.
@6 or >6	Toggle Turbo Disk Load ON and OFF.
@7 or >7	Toggle Turbo Disk Save ON and OFF.