encounters the RETURN in line 1020, it recalls the most recent place to RETURN to from the top of the stack, thus returning to line 2050. It also takes this information off the top of the stack when it recalls it. The process of recalling the information off the top of the stack and removing it at the same time is called POPPING the stack. The inverse operation, that of storing something on the top of the stack, is called PUSHING. The sort subroutine calls the print subroutine N-1 times, each time PUSHING the RETURN location, and the RETURN statement (line 1020) POPS the RETURN location each time. When the sort routine is done, the RETURN at line 2060 POPS off the only remaining RETURN location - the location where the sort routine was originally called - and returns back to line 160 to the REM and on to line 180, etc.

One other interesting thing about GOSUB and RETURN is that when the RETURN is executed, any incomplete FOR/NEXT loops started in the subroutine are terminated anyway. FOR/NEXT uses the same stack that GOSUB/RETURN uses to remember which variable is the FOR variable and what the limiting value and step are. So BASIC "cleans up" the stack by popping off any FOR/NEXT information still pending above the RETURN location for the subroutine.

4.6 ON expression GOSUB line#, line#, line#, ... [Program Flow 7]

This multiple-way decision making instruction is very useful, and should be very easy to understand since we have already learned the GOTO statement, the ON/GOTO statement, and the GOSUB/RETURN statement pair. ON/GOSUB is to GOSUB what ON/GOTO is to GOTO! Its syntax is the same as that of ON/GOTO except that you replace the GOTO with a GOSUB. Otherwise, it works like a GOSUB.

One of the many uses of the ON/GOSUB instruction is to select one of several subroutines, each of which implements a different "command" as selected by the user when he presses a single key. For example: assume that the user may enter one of A, S, D, or R to cause the program to do one of four things.

```
100 .... display menu ....
110 get c$: if (c$="") goto 110
120 com = instr("asdrASDR",c$): if (com<1) goto 110
130 if (com>4) then com = com-4
140 rem          a      d      s      r
150 on com gosub 1000, 2000, 3000, 4000
160 go to 100: rem get next command
...
```

Line 110 waits for a key to be pressed. Line 120 checks to see if the character is one of a, s, d, or r or their capital. If not (com is 0), it goes back to wait for a valid character. Line 130 makes sure com is from 1 to 4 - com is 1 for capital A as for lower case a, etc. Line 140 just shows which line is GOSUBed to for which command, and line 150 takes care of dispatching to one of the subroutines. Whichever one is executed, it RETURNs to line 160 so that the next command can be executed.

The next issue will continue with Chapter 5.

Any questions arising from this tutorial should be sent directly to the author, whose address is given below. Also, you may obtain disks containing the entire tutorial directly from the author. It comes in either a dot-matrix (4023, 4022, 2022, etc.) version, or a letter quality version (6400, etc.). Each version (1 disk) costs $11, or you can obtain both versions for $18 (both disks). Handling is included. Write to:

Warren D. Swan
1 N 114 Woods Avenue
Wheaton, IL 60188

B-128 SERIAL BUS SOFTWARE

by: Dennis Jarvis and James Springer

The B-128 Fast Bus is now available! With this software and Gary Anderson's hardware, namely his SERIAL BUS INTERFACE AND! 24K RAM/ROM CARTRIDGE you will now have a full blown serial bus on your B-128 just as it is available on other Commodore computers. Along with the normal slow serial bus we also support 'Fast Bus' just as it is available on the C-128 computer.

With the B-128 Fast Bus you will have complete access to all serial bus devices such as the 1525, 1526 etc. printers as well as the 1541, 1571 etc. disk drives along with various serial interfaces for Centronics printers (CARDCO and XETEC to name a few.)

If you should be using the 157X or 158X disk drives you will be able to use the fast bus capabilities of these drives on your B-128. Currently the 157X and 158X series of disk drives are the only devices that support Fast Bus.

Just to give you an idea of the speed possible using Fast Bus, the following table shows the LOAD times for various CBM disk drives, comparisons were performed using a 100 block file:

	1541	1571	1581	4040	8050
LOAD	1 min 30 sec	6 sec	4 sec	14 sec	14 sec

As you can see, Fast Bus makes a great difference in file handling speed.

Let's take a moment to clarify the difference between the normal Serial Bus and Fast Bus. Unlike the IEEE-488 Bus which is used with the 4040, 8050 etc. where 1 byte of data is sent out at a time. The normal serial bus sends out the byte one bit at a time using the DATA line to carry the data and the CLK line to indicate when the data bit is valid and ready to be read off of the bus.

Fast Bus operates almost the same way except there is one major difference. Instead of relying on software to shift the data byte out and manipulating the CLK and DATA lines, all the software has to do is store the data byte into the Serial Data Register (SDR) of a CIA chip and the hardware takes over handshaking the byte out on the bus. This method uses the DATA line for one data bit and the SRQ line as the CLK line. The big advantage to this system is that the Operating System doesn't have to spend all that time to:

1) Place the data bit on the DATA line.
2) Toggle the CLK line to indicate data valid.
3) Looping through steps 1 and 2 eight times.

There are several other steps that are performed in shifting out data in the aforementioned manner but they have been omitted for simplicity.

Along with the additional speed available with the 157X disk drives, you also have the ability to Read and Write to various 5 1/4 Double Density MFM types of disks such as OSBORNE, KAYPRO II, TRS-80, CP/M 86, and MS-DOS disks.

These aren't the only formats that the 157X drives can read/write. There are well over 300 different MFM types of disks that the 157X can read.

Now let's move on to how the Serial Bus is implemented on the B-128 computer. After you have installed the serial bus interface device and a RAM expansion cartridge in the address range of $2000 - $2FFF (both are available from Anderson Communications Engineering), you are now ready to start using your B-128 Fast Bus.

Place the software disk into Device 8 Drive 0 and enter the following Basic command in Direct mode: BLOAD "B-128 FAST BUS.1" and press RETURN, then enter: BANK15:SYS 8192, and press RETURN.

At this point you will see the title screen come on to inform you that the Fast Bus is ready for action. Before going any further take a look at your Serial Bus interface as you need to locate a couple of things. First plug your serial bus devices into the serial interface and then locate the switch marked IEEE/SERIAL. This switch is the key to the power and versatility of your new B-128 Fast Bus system. For now place the switch in the IEEE position then do a DIRECTORY on your disk drive. As you can see you're now obtaining the Directory from the IEEE disk drive. Now flip the switch to the SERIAL position and do another DIRECTORY. Now the Directory is coming from the serial disk drive! That is the procedure to follow when using SUPERBASE or SUPERSCRIPT II. The following guidelines should be used to load up and run programs which you will want to access the serial bus with:

1) BLOAD "B-128 FAST BUS.1"
2) BANK15:SYS 8192
3) Make sure the interface switch is in the IEEE position
4) Load and start your program the usual way

One capability that hasn't been mentioned yet is about the most powerful. When you have the interface switch in the IEEE position you can access BOTH IEEE and Serial Bus devices. Think of it, you can take a SEQ file for example onthe IEEE Bus, print the file out to your IEEE printer, your serial printer and make a backup copy of the file on your serial disk drive AT THE SAME TIME. And all of this from a simple BASIC program! It may sound complicated but it's really not and here's why. Whenever you have the interface switch in the IEEE position we use the device address to select which bus we are going to use. The device numbers are designated as follows:

Device Address		
4 - 23	-	IEEE Bus
24 - 30	-	SERIAL Bus

Simply put device number 4 would be an IEEE printer for example and device number 24 would be a Serial printer.

On the program disk you will find some example programs for copying MFM and GCR disks from BASIC! Along with other examples as the file handling as mentioned above. Also on the disk you will find a text file called Memory Map which gives a detailed description of our memory usage of the B-128 as well as the entry points in our massive Jump table.

One thing to remember, we have attempted to make this software package as compatible as possible, but we are positive there are many programs that this software WILL NOT work with. Currently our version 1.0 is compatible with SUPERBASE and SUPERSCRIPT II, but there may be other versions out there.

B-128 SERIAL BUS HARDWARE

by: Gary L. Anderson

The Commodore Serial Bus can now be connected to the B-128! The hardware consists of an external interface that plugs onto the B-128 internal user port pin field via a ribbon cable. A Commodore Serial Bus cable with 6 pin DIN connector then plugs into the 6 pin DIN receptacle on the external interface to complete the connection. See my ad in this issue.

The design of the external interface was done with the goal of maximum implementation of serial bus features. The hardware will allow the B-128 to act as a serial bus controller, like the C-64 or C-128, and a listener/talker, like a printer or disk drive. It will operate in both slow bus and fast bus modes. Also a serial bus power-on reset and manual reset was included in the design. A three position function switch, accessable from the outside of the case, was included for user selection of manual reset (a momentary position) and two positions for selecting custom features determined by software. The interface comes in a rugged plastic case for protection. A BASIC test program listing is included to self test the interface.

I have heard a report that the high profile CBM-256 main circuit board is missing the ground traces going to the user connector pin field. With this being the case, the interface will not work in the high profile. One would have to go inside the case of the high profile and make the repairs.

As in all computer systems the hardware requires software to control it and presently a software package exists to make use of this hardware interface. Two software experts, Dennis Jarvis and James Springer, have teamed up to write a software package that activates the B-128 Serial Bus Interface as a serial bus controller and runs slow bus and fast bus with present available Commodore Serial Bus peripherals. The hardware and software are available together as a package from CBUG or available seperately, the software from CBUG and the hardware from Anderson Communications Engineering.

Gary L. Anderson
2560 Glass Rd. NE.
Cedar Rapids, Ia. 52402

WHATS NEXT

by: Dennis Jarvis & Jim Springer

Now that you have obtained your new FAST BUS software/hardware, what can you expect in the future? Plenty, some of our future projects include a program to transfer files between MS-DOS disks and COMMODORE GCR disks. This will enable those of you with MS-DOS type machines at work to bring your text file home and use your word processor such as SUPER SCRIPT, etc. to edit your MS-DOS text files then when done place them BACK onto your MS-DOS disk and load them back up correctly with your MS-DOS machine.

How about a program which will allow you to transfer files from CP/M-86 to other CP/M diskettes? A good MFM sector editor? These are just a few of the items we are looking at for future projects. TRANSACTOR which is a TECH NEWS journal for Commodore computers will now start supporting the B series of computers more, and more. We will be releasing this package thru TRANSACTOR, to enable the C-128 computer to keep up with the B!! This package will only include the code to enable them to use BASIC to access the disk drives much the same way the B can now access the disk drives. Once this has occured we will begin releasing thru the PUBLIC DOMAIN various small programs for the 1571 disk drive. We strongly recommend that you obtain a subscription to this journal as we've been seeing the B being talked about more and more by such well known authors as LIZ DEAL, GARY ANDERSON and others. Perhaps we can all make an effort to show the world that the B series is NOT just another HAS BEEN computer. With such packages as this one and others the B will quickly become a formidable computer. Now when you see a program for the 1571 disk drive such as Miklos Garamszeghy's 'Exploring The World Of MFM On the 1571 Disk Drive' (TRANSACTOR Volume 7, Issue 4) you can, with SOME effort on your part, convert these programs over to the B-128 computer. Keep a look out in the CBUG news letter as we will be, from time to time, sending in various programs and tips, etc.

Well that's it for now we hope you enjoy your new FAST BUS capability.

Enjoy,

Dennis J Jarvis
and
James D Springer

UNIVERSAL TRANSFER

<<Following is one of several user instruction files from the Jarvis/Springer FAST BUS disk being introduced in this issue. It is included in THE ESCAPE to show just how powerful these programs are.>>

by: Dennis Jarvis & Jim Springer

This program will give you the capability of transfering various files back and forth over the IEEE, and SERIAL bus with almost NO effort on the USER. With this program you can do all of the following options all at the same time;

1) Print a file to up to 2 printers
2) Copy a file to another disk drive
3) Print a file to the screen

You can perform any or all of the aforementioned items at the same time!

Let's go ahead and use this program to do all of aforementioned items, if you don't have one or more of thos devices don't worry about it.

Go ahead and load up the UNIV. TRANSFER program and run it. When it's up and running you will see the following statement, if our B-128 Fast Bus is in the computer:

IF THE DEVICE IS ON THE SERIAL BUS ADD 20 TO THE DEVICE NUMBER

IF YOU DO NOT WANT TO USE A SPECIFIC DEVICE ENTER 0

What this is telling you is that all SERIAL bus devices start off at 24 and end at 30. All IEEE devices start at 4 and end at 23. The following table will make it clearer to you.

IEEE PRINTERs	4 thru 7	(normally but these COULD be disk drives also)
IEEE DISK DRIVEs	8 thru 23	
SERIAL PRINTERs	24 thru 27	(normally but these COULD be disk drives also)
SERIAL DISK DRIVEs	28 thru 30	

NOTE: If you do not want to use a certain device such as the SERIAL BUS printer then just enter a device address of zero which will inform the program to bypass using that device.

First the program will ask you to enter the DEVICE number of the device to get the SOURCE file off of, the default is currently set up for a SERIAL BUS disk drive with a device address of 8. You must enter a numeric value between 4 and 30 inclusive. If the value is less than 4 or greater than 30 an error message will occur. Then it will ask you to enter the drive number to copy from which has a default drive number of 0. This will be repeated for the destination disk also remember: if you do not wish to use the destination device enter a zero.

Next you will be asked to enter the device and secondary addresses of your printers, after you have selected an active printer you will be asked to enter it's secondary address value. Normally this value will be zero, but certain printers give various options using the secondary address so we passed this option setting on to you the user. If in doubt enter 0.

Finally you will be asked if you wish to print the file out to the screen if you do enter y, if not enter n. That's all there is to it.

Next the computer will check to make sure that there were no mistakes such as entering a device number that does not exist and other such checks. After these are performed either an error message will be printed such as "YOU HAVE SELECTED BOTH PRINTERS AS HAVING THE SAME DEVICE ADDRESS" and other errors. If an error is not critical the program will continue and print out a summary of where each device is located and which bus it's on (IEEE, or SERIAL) and ask for your final aproval. If you answer yes then the program will ask if you need to see a DIRECTORY of the SOURCE drive

And if you do enter 'Y' for yes you will soon see a directory go passing by. If you see the filename you want just press the RUN/STOP key which will cause the DIRECTORY command to ABORT.

Now you will be asked to enter the SOURCE, and DESTINATION filenames. You need not worry about the file type the program will figure that one out for itself! At this point the program will check to make sure the SOURCE filename exists and if it does not it will restart the FILENAME entry section of the program to let you reenter it. If the filename exists on the source drive it will check to make sure it DOES NOT exist on the destination drive. If it already exists it will ask you if you wish to scratch it off of this disk. If you answer yes it will SCRATCH that file from the DESTINATION diskette. If you answer no it will restart the filename entry section of this program.

NOTE: The above references to the DESTINATION files only hold true if you selected a device number to COPY TO.

Finally the program starts to send out the information to the devices selected, if at any time you wish to 'DISABLE' a device because of such things as a PAPER JAM or for any other reason you can think of just turn that device OFF, and you will receive the following error message:

A BUS ERROR HAS OCCURED WITH DEVICE: XX ON THE ?? BUS WOULD YOU LIKE TO

1) IGNORE THE ERROR
2) LOG OFF THAT DEVICE

NOTE: XX is the unit device number
?? is the bus the error occured on (SERIAL or IEEE)

If you want to continue on as if nothing had happened then select option number 1. If you no longer wish to send data to that device select option number 2.

Well that's about it go ahead an use this program and play around with it, make any changes you wish.

FROM COL. WRIGHT

Dear Norm:

Here are the instruction and catalog files from disks 1 and 2. Tony has just provided me with a disk that will give me enough room on the "NATIVE" side to provide these to you in future releases.

I am currently working on a Z-80 emulator. It is working pretty well right now and runs most of the Z-80 stuff for the C-128 and a few others as well. This is what I plan on putting on Disk 003.

I also have a Small C compiler that I am working on. Right now, I am not having too much luck getting it to run. It may take a while, but, hopefully by the end of the summer, I should have it up and running.

I have also found the DRI (Digital Research Institute) GEOS or GIOS, (I can't remember which). I think this is the CP/M-86 version of the popular GEOS system. I will work it, but it may take a "long long" time as it is all in source code and over 4 disks worth of code. I am not to good with source code and we may have a problem since it seems we really don't have a good handle on how this machine processes CP/M.

I am also working on a "Tutorial" disk for the club. I have "accumulated" several books on CP/M over the last few months, unfortunately none on CP/M-86, and am experimenting with what works and what doesn't. It's slow, since I have to test everything out first then try to put it into words. Hopefully, I will have lessons 1 and 2 ready by July.