OS-9 Newsletter

Basic09 . . .

PDS Database

Tutorial
by Scott Honaker & Rodger Alexander

I've received several inquiries regarding the recent "updates" to the the the *PDS Database*. Last month's "Correction" only seemed to confuse several readers. Overall, the complete database program is complete, so this month we are going to print out the complete listing of *Search* and *Search_DB* Procedures since both associated routines have been heavily modified since their first posting in the Newsletter a couple of months ago.

The SEARCH routine is divided into two Procedures (*Search_DB* and *Search*). The first routine determines the search parameters by prompting the user for the Field to be searched and the SearchString to be matched. These two parameters are then passed to the second routine which actually performs the searching task by SEEKing and GETing the specified field from each record and then comparing the SearchString against each record.

The first revision was caused by the "NEXT" option from the Main Menu. When using this option, it is assumed that the search will continue from where it left off from the previous SEARCH request. The basic parameters used for setting an initial Search beginning with record number 1 must be avoided. This was done by moving those parameter setting commands to the first routine (*Search_DB*). Also, for the "NEXT" option to work, the CURRENT parameter needs to be incremented by one. This was done simply by adding the following line: **temp=current+1**

What is listed below are the two **revised** SEARCH routines. Notice how the double DATA READ routines are handled within the same procedure. You should replace the previous Search Routine Procedures printed in the April

Newsletter with the *Search_DB* and *Search* Procedures below.

```
PROCEDURE Search_DB
REM database Search Routine
PARAM DB_Path:BYTE; Top:INTEGER; Current:INTEGER; offset:BYTE;
length:BYTE; SearchString:STRING[20]
DIM field:BYTE; fieldnumber:BYTE; FieldName:STRING[12];
TempName:STRING[12]
fieldnumber=8
PRINT CHR$(12) \ PRINT
PRINT "Personal Database System - Search Database" \ PRINT
PRINT "Search on which field:" \ PRINT
PRINT "1) First Name"
PRINT "2) Last Name"
PRINT "3) Address 1"
PRINT "4) Address 2"
PRINT "5) City"
PRINT "6) State"
PRINT "7) Zip Code"
PRINT "8) Phone Number" \ PRINT
INPUT "Enter choice: ",field
FOR x=1 TO fieldnumber        */fieldnumber = 8, the number of fields in the
                                first DATA line /*

READ TempName
IF x=field THEN
FieldName=TempName            */desired FieldName is stored /*
ENDIF
NEXT x                        */all TempName's are read in first DATA line
DATA "First Name","Last Name","Address 1","Address 2","City","State"
,"Zip Code","Phone Number"
PRINT CHR$(12) \ PRINT
```

```
PRINT "Personal Database System - Search Database" \
PRINT
PRINT \ PRINT
PRINT "Enter "; FieldName; " to find"; \ INPUT
SearchString
PRINT
FOR x=1 TO field          */ all of the first DATA line
                          variables have been read/eliminated /*
READ offset               */ preventing a crash when the
                          second data line is read /*
NEXT x
READ length
DATA 0,10,25,45,65,80,82,92,106
Current=1                 */ forces the Search Procedure to
                          begin with the first record /*
RUN
search(DB_Path,Top,Current,offset,length,SearchString)
[]
```

NOTE that the second to the last line of the above procedure resets **current = 1**. The last line then runs the following procedure forcing the search to begin with the first record. However. after a record is found and the NEXT match is requested from the main menu. the above *Search_DB* procedure is not called. instead the *Search* procedure below is run directly from the main menu with the **current** value intact from the previous search.

```
PROCEDURE Search
TYPE address=FName:STRING[10];
LName:STRING[15]; address1:STRING
[20]; address2:STRING[20]; city:STRING[15];
state:STRING[2]; zip:
STRING[10]; phone:STRING[14]
PARAM db_Path:BYTE; Top:INTEGER;
current:INTEGER; offset:BYTE; length
:BYTE; SearchString:STRING[20]
DIM CompareString:STRING[20]
DIM rec:address
DIM temp:INTEGER
IF current=Top THEN
END
ENDIF
temp=current+1
FOR match=temp TO Top
SEEK #db_Path,(match-1)*SIZE(rec)+offset
GET #db_Path,CompareString
CompareString=LEFT$(CompareString,length-offset)
IF SearchString=CompareString THEN
current=match
END
ENDIF
NEXT match
END
[]
```

The PACK routine presents an interesting challenge. The idea of course is to rid the database data file of the unwanted records. Since Basic●9 cannot easily change the length of the file on the disk we have to find another more practical way to accomplish effectively the same thing. The most obvious solution would be to rewrite each valid record to a temporary file, delete the old file and then rename the temporary file to the original file name. Very simple, very direct and very easy to do. But, it is costly in regards to disk space, especially if you are running on a floppy drive system. This method requires free storage space equal to the size of the data file to be packed. If the original data file were 200 sectors in length, there would not be enough room on a double sided disk to accomplish the pack routine.

A second method would be to keep track of the locations of the "deleted" files with some kind of an index file.. Then when you add a new record, you would simply write it over one of the deleted files. This is not a true "pack" routine. but it does keep you datafiles down to a minimum size and very fast. Tandy's **"OS-9 Profile"** uses this method. Unfortunately, you will run into some problems keeping track of the files when you SORT your records. Either you don't sort at all or you will need to sort prior to each PACK command. Pretty sophisticated stuff!

Guess we'll take the easy way and simply rewrite the files to a temp file called **"Scratch.DAT"**

```
PROCEDURE Pack_DB
TYPE address=FName:STRING[10];
LName:STRING[15]; address1:STRING
[20]; address2:STRING[20]; city:STRING[15];
state:STRING[2]; zip:
STRING[10]; phone:STRING[14]
PARAM DB_Path:BYTE; Top:INTEGER;
DB_Name:STRING
DIM new_path:BYTE
DIM rec:address
DIM temp:STRING[2]
ON ERROR GOTO 100
CREATE #new_path,"Scratch.DAT":UPDATE
SEEK #DB_Path,0
count=0
FOR x=1 TO Top
SEEK #DB_Path,(x-1)*SIZE(rec)
GET #DB_Path,rec
IF LEFT$(rec.FName,1)<>"!" THEN
count=count+1
SEEK #new_path,(count-1)*SIZE(rec)
PUT #new_path,rec
ENDIF
NEXT x
CLOSE #DB_Path
DELETE DB_Name
SHELL "rename Scratch.DAT "+DB_Name
DB_Path=new_path
Top=count
END
```

```
100 PRINT CHR$(12)
PRINT "Error number: "; ERR
INPUT "Press enter to continue...",temp
END
```

Officially, our Basic09 _PDS Database_ is complete, but there are a few loose ends that should be addressed. The one procedure that should be rewritten is the SORT Procedure. It appears to work fine and very fast, but that is because it is being done in ram and not on the disk. You gain speed by doing it all in ram, but it can only handle a few files. As soon as this database becomes significantly large....CRASH!

Solution: Take a look at the SORT Procedure and simply modify it to write to disk. You should have gained enough experience from typing in all of these procedures that modifying the SORT Procedure should not be too difficult. GO FOR IT!

On a personal note, I still want to write a PACK routine that uses an index to track deleted files and writes over them when new files are added. I would also like to use index files for SORT routines the way Ashton Tate's dbase does. And how about permitting the user to define the number of fields, field lengths and the field descriptions???? Some ideas to ponder and maybe motivate you to expand on our database with your own custom modifications.

Chicago Fest
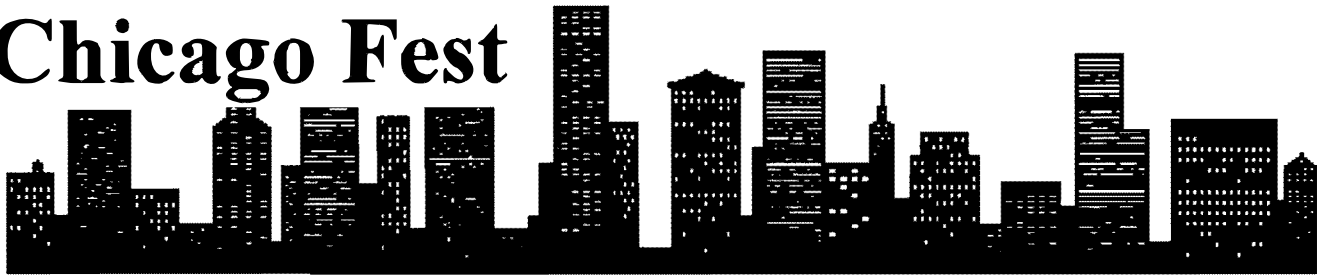
The 1st Annual Last CoCoFest

There appeared to me to be about 200 or so people at the fest, but that really is just a guess. The crowd seemed to be slightly smaller than the Atlanta Fest, but seemed far more willing to spend money. The group seemed very much OS-9 68000 oriented, rather than Disk Basic oriented.

Compact Disk

The seminars were pretty good for the most part. Frank Hogg's seminar on Compact Disk Interactive was probably the most interesting. He passed out copies of the premier issue of a magazine called CDI World. Lots of info on CDI in this magazine. A little thin though, in terms of total pages, but it was just the first issue. Frank gave everyone a general overview of what CDI is, how it works, what makes it different from Commodore's CDTV, and what makes it different from computers with CD ROM drives. Principally it is a consumer unit, somewhat akin to a Compact Disk player, only with video tossed in. The interactive part is what makes it so neat. The control unit allows you to control your progress through the CDI title you are watching.you can play games, just like Nintendo. But you can also get CDI titles.

Kevin Darling

As always, Kevin Darling's seminar was very well attended. Topics discussed included the Level II update. No firmword, but under the threat of dismemberment, Kevin is looking into releasing the upgrade in some form. No promises though. Most of Kevin's talk was centred around OSK stuff, K-Windows in particular. K-Windows is starting to get everything but the kitchen sink in it. VERY nice

package. Kevin was also demoing some stuff on the MM/1, but he was having some problems with a borrowed hard drive, so that was cut short.

Interactive Media's MM/1

Paul Ward's seminar was a summary of what IMS is up to, and where they are trying to go. IO boards should be 100% caught up by the end of June. Same thing with the software updates. The MIDI boards were available at the show. Mike Knudsen has put together a fantastic package. IMS seems to have its act pretty much together, but there is still room for improvement. But the MM/1 is turning into a really nice machine. Paul is expecting shipments to go out with complete kits within 30 days from now on. No more getting the manuals, then the CPU board, then the IO board, etc. Everything should be arriving in one shipment. Some really nice software is showing up, both commercial and pd. All the basic tools are there, word processors, spelling checkers, spread sheets, databases and the games are starting to appear.

Potpourri

Those were the seminars I attended. Others I dropped in on, or didn't stay for the whole thing. Marty Goodman gave a couple of hardware seminars. Glen Dahlgren of SunDog Systems gave a seminar on creating games. There were a couple of others, but I don't recall what they were off hand. the seminar that Chris Burke gave on the 6309. He covered some of the instructions, briefly summarized the Power Boost Package and mentioned the book on the 6309 that is coming out. He also mentioned that he would be making more patches to OS-9 to take advantage of the 6309.

--- Colin Mckay;Maximus-CBCS v1.02

* Origin: Micro80 Computer Club of Ottawa BBS
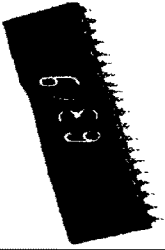
Power Boost (6309)
from Burke and Burke

(c) Marty Goodman 1992

At the Chicago CoCo Fest put on by CoCoPRO! I had a chance to look at the initial release of Burke and Burke's POWER BOOST. This impressive hardware-software enhancement for CoCo 3's running OS9 depends on patches to OS9 made by Chris Burke that allow OS9 to use the enhanced instruction set of the Hitachi 6309 to speed a number of system operations.

The 6309 was made by Hitachi to be an exact instruction for instruction and pin for pin duplicate of the 6809 in low power CMOS technology. Tho we don't know for sure at this time, all evidence would indicate that the 6309 is a micro-coded type chip, unlike the original 6809, which was one of the last complex "random logic" ("hard wired") central processor chips made. Hitachi's engineers apparantly had a fair amout of space left in the micro-code ROM after meeting Motorola's specs for the chip. They apparantly secretly added extra registers and instructions that allow for much faster data transfers, 16 by 16 to 32 bit product multiplications, and more. Burke and Burke's product takes advantage of the enhanced data transfer instructions of the 6309 to greatly speed some common OS9 system operations. Specifically, I observed a MDIR E and a read of a megabyte of data off a Burke and Burke hard drive to be speeded by about 40%. This is a quite visible improvement! In another demonstration, Chris Burke showed how his patched version of OS9 could take about 20 seconds off the time it took to cobble a boot disk. He made the demonstration dramatic by cobbling to a RAM disk, where the cobbler time went from about 22 seconds to about 2 seconds.

The patcher program currently being sold is quite elaborate and professional. As it is installed, it one by one looks for over fifty separate modules of OS9, identifies, them, and automatically patches them for use of the 6309. It displays to the user which modules it found to patch and which ones it looked for but could not find or could not patch. The operation of the patcher program is quite "smart" and automatic.

There's a lot more to come, however. Chris Burke has been working on his project for just one month. He currently does not have patches for hard drive software drivers other than his own that go with the Burke and Burke CoCo XT. However, he hopes to have out patches for other hard drive systems that use SCSI and SCSI-like host adaptors, such as Disto and Kenton / RGB. He is also working on patches to the OS9 assembler, as I understand it.

Additionally, Chris' current patches make use ONLY of the extra instructions that one has acess to in the 6809 emulation mode of the 6309. It is Chris Burke's hope to soon have out improved patches that not only deal with more modules, but also take advantage of the native 6309 mode of the 6309, in which even ordinary 6809 instructions are executed in fewer machine cycles, and therefore more quickly. Chris reports that in his preliminary testing, utliizing this mode should add roughly an extra 15% speed increase over the speed increase he already has achieved. Chris' conservative estimation is that he may achieve an avarage speed increase in many situations of around 50% over a CoCo 3 running ordinary OS9 on a 6809 once all his planned patches are completed. There's no need to wait to buy the Power Boost, for Burke and Burke will have a very generous upgrade policy when new patches are available.

The Hardware Hurdle

The biggest hurlde in installing the Power Boost product is that one must remove the SOLDERED IN 68B09E chip from one's CoCo 3 and replace it with a socket in which one can insert the 63B09E that Burke and Burke supplies with the Power Boost package. This requires one skilled in repair of printed circuit boards, or at least able to do a competent "destructive removal" of the old chip. Destructive removal (a technique I have described several times in the past in my CoCo Consultations column in Rainbow, and which I expect to be described in detail again in an umcoming Rainbow article on the Power Boost product) is sufficiently easy to learn that anyone who has a fine soldering iron, a fine needle nosed pliers, a fine diagonal cutting pliers, and a decent solder sucker should be ablle to complete the task. But it does take time, patience, and skill with soldering equipment.

Owners of 1 meg upgrades will have to be a bit MORE skillful and creative in installing the Power Boost product. They will have to desolder their "CPU board" from the existing 68B09E chip, then either solder it to the 6309 chip and add extra pin extentions to the 6309 so it will go into the socket they install on the CoCo, OR (and this is what I recommend) they will have to make up a clever satellite board that lies under the main 1 meg upgrade CPU board and permits the 6309 to be put into a socket on that extra board. The CPU board would be "up on stilts" above this extra board, and the extra board will have pins below that allow it to be plugged into the socket that is installed into the CoCo 3. Such extra little boards would be most professionally made using flush mount machine pin pin strips, tho they can be finagled with the less esoteric and more common ordinary machine pin sockets and pin strips.

Chris Burke actually did to a little bit of experimenting with Disk Basic and the 6309. However, given the fact that most RS DOS based software spends very little time using RS

DOS code, patches to Disk Extended BASIC will be of little value. Chris notes that if a CoCo 3 running RS DOS with a 6309 is put into 6309 mode, execution is speeded by about 15%, but printer baud rate will be thrown off and disk I/O will cease to work reliably due to changes in certain critical delays and timing loops. He noted that the ultra fast data move instructions of the 6309 could be used to speed up considerably the graphics commands in Disk Basic, and even provided a few patches that do speed BASIC a little, but not dramatically.

I hope this article contributes to the understanding of just what Power Boost is and can do. Chris Burke occasionally drops by Delphi OS9 and CoCo SIGs under user name CoCoXT, and may be available to answer further questions here from time to time.

-- Marty --

Installing the 6309

Whether you've bought a PowerBoost kit from Burke & Burke, or just want to upgrade the processor in your Color Computer, you may want to install a socket for the processor. This article describes how to install a processor socket in your Color Computer without unsoldering the old 6809 chip. By following these instructions, you can actually install the new processor socket directly on top of the old processor!

The 6809E TSC Pin

Pin 39 of the 6809E processor is called the TSC (Tri-State Control) pin. TSC was added to the 6809E processor to support DMA controllers, multi-processor configurations, and other shared bus configurations.

When pulled to ground (less than .5 volts), TSC allows the 6809E to operate normally. When pulled to a logic high level (more than 3 volts), TSC places the 6809E's address and data busses, as well as the R/W line, in a high-impedance state - nearly the electrical equivalent of unplugging the 6809E from its socket.

In the Color Computer, there's no DMA controller and the video controller uses a "hidden DMA" technique to access memory without slowing down the 6809E. Under these conditions there's no need for TSC; the Color Computer ties it directly to ground.

Using TSC to Advantage

By disconnecting TSC on your CoCo's 6809E from ground, and reconnecting it to +5V, we effectively remove the 6809E from the computer's electronic circuitry. This allows us to stack a 2nd processor (or processor socket) on top of the old one, taking care to connect the 2nd processor's TSC pin to ground.

The Color Computer then ignores the old processor, and takes commands only from the new one.

Preparation

You'll need a 40 pin IC socket, a pair of needlenose pliers, a pair of small diagonal cutters, a low-power fine-tip soldering iron, and about 3 inches of wire-wrap wire. And, of course,

the 6309E chip.

Begin by bending pins 5, 6, 33, 36, 38, and 39 of the socket inward, underneath the body of the socket. You can actually clip off pins 5, 6, 33, 36 and 38 if you like, but be sure you just bend pin 39.

Now solder a 1" length of wire-wrap wire between pin 39 and pin 1 of the socket. Run the wire across the bottom of the socket, and don't bend pin 1 or get a lot of extra solder on it. Set the socket aside in a safe place when you're done.

Unplug the Color Computer, and open it up (voiding the warranty, of course). Locate IC1, which should be marked MC68B09EP. This is the old 6809E processor. Cut pin 39 of the processor, using the diagonal cutters. DO NOT JUST CUT THE CIRCUIT BOARD TRACE GOING TO PIN 39. YOU MUST CUT PIN 39 ITSELF. If you'r not sure where pin 39 is, look at the circuit board. You should see the numbrers 10, 20, 21, 30, and 40 printed in white on the circuit board around the processor. Find the number 40, which marks pin 40 at one corner of the processor. The next pin down is pin 39. Be sure you cut all the way through the metal pin, so that the top half of the pin sticks out of the black plastic body of the processor while the bottom half sticks out of the circuit board. Make sure that the two halves of the pin don't touch each other.

Stack the socket (prepared as described above) on top of processor IC1. The bent-under socket pins must rest on top of the processor, without touching any processor pins. The other socket pins must each make firm contact with the corresponding processor pins. When you're happy with the way the socket sits on top of the old processor, solder together each pair of touching pins. This locks the socket firmly in place, in addition to establishing good electrical connections.
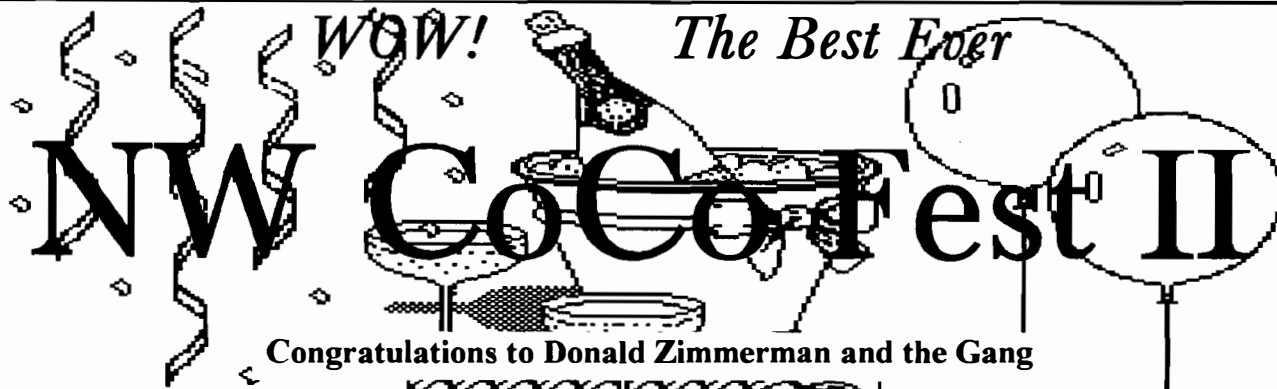
Finally, plug the 63B09E into the socket. Put the Color Computer back together, and it's ready for use.

Technical Notes

The bent-under pins include TSC, BS, BA, LIC, AVMA, and BUSY. You have to bend over the last 5 of these because the TSC line doesn't place them in high-impedance state: if you connected them up to the 6809E, you could cause a battle for control between it and the new 63B09E. It's OK to just bend these pins over because they're all outputs and they aren't connected to anything in the CoCo.

If the system doesn't work right after installing the new processor, make sure you've bent under the correct pins and that all of the pins you didn't bend have good solder joints. If you absolutely can't get the computer to work, you can either reconnect the old processor (pin 39) and try again some other time, or desolder both the socket and the old processor to install a fresh socket directly on the CoCo's circuit baoard.

The tests I've done indicate that this "piggyback" technique works reliably when done correctly, but it's certainly better from an engineering perspective to remove the old processor and install the socket in its place. The technique described in this article is about 1/10 the work, though, and unless you're an expert at repairing circuit boards the chances of hurting the computer with this technique are much lower than if you actually remove the old processor.

WOW! The Best Ever
NW CoCoFest II

Congratulations to Donald Zimmerman and the Gang

This year's Pacific Northwest CoCo Fest was an unprecedented success. On Friday night, there was well over 40 people in attendance competing for prizes in the CoCo Trivia Game, a Color Computer version of *Wheel of Fortune*. Unfortunately, Vana was not there. But Bob van der Poel was and spoke to us briefly about the future of the 68K OS9 machines and the MM/1 in particular. Frank Hogg sent a video tape of his presentation at the Chicago CoCoFest in which he demonstrated CDI (Compact Disk Interactive) on a CoCo. However, the video quality did not lend itself well to a group showing. Contact Donald Zimmerman for information on obtaining a copy of the tape.

On Saturday, attendance swelled to well over 80. **Andre LaVelle**, the west coast representative for **CoCoPro** was there with four tables stacked high with CoCo hardware and software. The Computer Bank Charity had two tables stacked with all sorts of CoCo and Tandy computer equipment. Saturday morning featured an auction of everything left. One lucky bidder got 3 CoCo's, 2 disk drives, bunch of software and a printer for $30. The final bidder took everything that was left (about 1000 pounds) for $20.

Mr. & Mrs. Chris Burke of Burke & Burke were there selling all of their products including the new *6309 Power Boost*. Chris also donated a DMP 100 printer to the Computer Bank Charity which was sold to Jeff Brittan for $22. Chris spoke to us about his *Power Boost* Kit and the soon to be released Scanner software for the already available XT Interface Board. Upgrades for the Power Boost package

will be made available in the future and OS-9 Drivers/Descriptors for IDE Hard Drives with the XT Interface will be coming out next year. Great things coming from Burke and Burke and their participation at this year's CoCoFest was the icing on the cake.

Gonzales Data Systems set up an **MM/1** display and were very helpful with demonstrations and just letting us 6809'ers get our hands on this new machine to see what was so special about it. WOW! OSK looked and feels exactly like OS9, but **F A S T ! ! ! !** Gonzales Data Systems is the Pacific Northwest representatives for Interactive Media and will deliver a completely assembled machine in perfect working order to you for only $1100. Think that's expensive? You should see what this machine can do!

Eversoft Games, Ltd. from Arlington, Washington were there displaying and selling their games: **DuoDeck, Classic Solitaire, La Belle Lucie, Deception Path**. All day Saturday there was a constant "group" or should I say "groupies" surrounding the Eversoft Games table. Their graphic displays in their games was indeed impressive and I'm sure they made a lot of new friends and sold a bunch of games.

Bob van der Poel was the keynote speaker at the luncheon on Saturday. He spoke to the 80+ CoCo Nuts in attendance about the CoCo Community and what the CoCo has meant to him and how it has affected his life. Most of us know Bob through his software, such as Telewriter 128, the VED text editor for OS9 and many more. But Bob is also a tremendous speaker. Very dramatic, very entertaining. On a one

to one basis, Bob is also very personable, making each individual he spoke to feel like they are one of Bob's personal friends.

The SWAP MEET went on all day and was even part of Friday night's activities. Everything you could want for the CoCo was there. I bought an external 1200 baud modem for $5. There were boxes of 512K RAM cards for $35. Brand new CoCo-3's for $80. A used CoCo-3 for $40. Disk Drives, Disk Controllers, Multipaks, Printers, Monitors, and every possible Software package you could imagine.

Wes Gale of Gale Enterprises from Vancouver, B.C. gave a presentation on Telecommunications, Bulletin Boards and Networking. Specifically he dealt with FidoNET and all the advantages that can be realized by using Electronic Mail service. However, I was disappointed that he did not display or demonstrate any of his software (*QwikGen* or *Zap*).

In the afternoon small seminars were held on various topics: **CoCo in a PC Case** with Rodger Alexander, **C-Programming** with Bob van der Poel, **Basic Programming** with Mark King,, **Computers in the Future** with Microsoft's Scott Honaker, **Where to Find CoCo Stuff** with Terry Laraway, **Burke & Burke** with Chris Burke.

It was a Great CoCoFEST. The CoCoFest MUGS were better than last year. The T-Shirts were really COOL!. Almost everyone got some sort of gift like a mouse pad, etc. Even the CoCoFest Paper Sacks were super. We went home completely exhausted and completely satisfied.

Thanks Donald!

H6309

by Hirotsugu Kakugawa, (kakugawa@csl.hiroshima-u.ac.jp)

** INTRODUCTION **

The CPU 6309 by HITACHI has secret features which are not written in its manual. The purpose of this article is to introduce them.

The features were originally reported in a magazine, Oh!FM (1988 Apr.), which was written in Japanese. I have not tried all of the features reported in the article, but I'll report the features as far as I know.
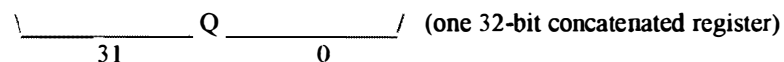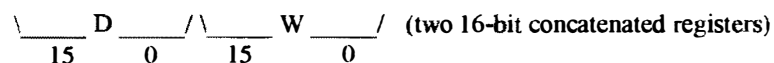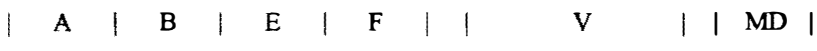
HITACHI says in its 6309 manual that the 6309 is compatible with the 6809, but some OS-9 hackers found that it has secret features:
1. More registers (additional two 8 bit accumulators, 8 bit register, and a 16 bit register),
2. Two modes (6809 emulation mode and native mode),
3. Reduced execution cycles in native mode,
4. More instructions (16 bit x 16 bit multiplication, 32 bit / 16 bit division, inter-registers operation, block transfer, bit manipulating operations which are compatible with the 6801, etc)
5. Error trap of illegal instruction, zero division.

I substituted the 6309 for the 6809 in my personal computer, and I changed OS9/6809 Level II such that the 6309 executes in native mode. I had to change the interrupt handling routine in the kernel. I implemented illegal instruction trap; I was really happy because most bugs are caught by a trap handler.

** NEW REGISTERS **

The 6309 has some additional registers which the 6809 doesn't:

```
|   A   |   B   |   E   |   F   |   |       V       |   |  MD  |
 --------- --------- --------- ---------   -------------------   ----------


\_____D_____/\_____W_____/  (two 16-bit concatenated registers)
   15      0    15      0

_____Q_____/  (one 32-bit concatenated register)
  31                    0
```

The E and F register
These are 8 bit accumulators. Like the D register is a pair of the A register and the B register, these two registers can be used as a 16 bit accumulator. The pair of the E and the F registers is called the W register. In addition to that, pair of two 16 bit registers, the D register and the W register, can be used as a 32 bit accumulator called the Q register.

The V register
This a 16 bit register can be used only by TFR, inter-register operation, etc. But even if the chip is reseted, contents of this register does not change. Some people may use this register to keep constant value (V for value).

The MD register
This is a 8 bit register to keep the mode and status of the chip. The meaning of each bit is as follow.

Read value
    bit 7 --- 1 is set if zero division happen.
    bit 6 --- 1 is set if illegal instruction is fetched.

Write value
    bit 1 --- The mode for FIRQ interrupt.
        0 -> the the action for FIRQ is the same as that of 6809.
        1 -> the the action for FIRQ is the same as IRQ.
    bit 0 --- The execution mode of 6309.
        0 -> the emulation mode.
        1 -> the native mode.
(When the chip is reset, all bits are 0.)

** TWO MODES OF THE 6309 **

The 6309 has two modes, emulation mode and native mode, as described in the previous section. When the chip is reset, the initial mode of 6309 is the emulation mode.

When the 6309 is in the emulation mode, the chip emulates the action of the 6809. But we can use extended registers and extended operations in this mode. The 6309 executes instructions in the same number of cycles as the 6809 does.

When the 6309 is in the native mode, it executes instructions in fewer cycles. And when the chip is interrupted (IRQ, for example), it pushes extended registers (PC, U, Y, X, DP, W, D, CC, in this order). If you want to use the 6309, you must rewrite interrupt handling routines (for example, the entry of system call of OS9). (And the multiple system map routines for CoCo GRFDRV - kd)

** TRAPPING **

If either of the following two events happen, a trap is caused.
    1. A illegal instruction is fetched.
    2. A number is divided by zero.

The action of the 6309 when a trap is caused is:

    1. Pushes the registers on the system stack.
    (In the emulation mode, PC, U, Y, X, DP, B, A, CC, in this order
    and in the the native mode, PC, U, Y, X, DP, W, B, A, CC in this order)
    2. Reads the trap vector address ($FFF0) and jumps to the vector.
    (Note that $FFF0 was reserved by 6809.)
To check the reason of the trap, BITMD instruction is provided. This instruction is explained in a later section.

** NEW INSTRUCTIONS **

The Register Addressing Mode
To specify registers in TFR and EXG, the 6809

H6309 Continued.......

uses a 4-bit code pattern. New registers of the 6309 are specified by bit patterns in TFR and EXG operations. In addition to that, the bit pattern is also used in instructions of inter-register operations. We call this bit pattern used to specify registers, the "register addressing mode".

Inter-Register Operations

Operations of 6809 are operations between register and immediate value or between register and memory. Therefore, we had to store value of register on memory if operation between two registers was necessary. But the 6309 has inter-register operation.

Block Transfer

Block transfer instructions are provided such as the Z80 has. The TFM instruction requires source address and destination address and block size as its argument. One or two 16 bit registers (X/Y/U/S) are used to specify source and destination addresses. Block size to be transfered is specified by the W register. Four styles are provided:

    TFR r●+,rl+ (transfered in address is increasing order),

    TFR r●-,rl- (transfered in address is decreasing order),

    TFR r●+,rl (poured into the same address, I/O port for instance),

    TFR r0,rl+ (read from the same address, I/O port for instance).

Multiplication And Division

The 6309 has a MULD instruction which performs a 16bit x 16bit multiplication. We can use various addressing modes (immediate, direct, indexed, extend) The result is stored in the Q register.

Division instructions are also provided. The 6309 has two division instructions: 16bit / 8bit, 32bit / 16bit divisions. Various addressing modes (immediate, direct, indexed, extend) can be used. The quotient is stored into the W register; the modulo (remainder) in D.

Bit Manipulation / Bit Transfer

The 6309 provides AIM, OIM, EIM, TIM instructions which are compatible with instructions of the Hitachi 6301 CPU. Instructions called BAND, BOR, BEOR, BIAND, BIOR, BIEOR, LDBT, STBT are provided. Behavior of thses instructions is that a logical operation is performed for n-th bit of a data in a memory (only direct mode is allowed) and m-th bit of a register, then the result is stored in the register. The format of the object is :

    $11. x, (post byte), (operand).

NOTE: 14 pages of instruction tables were included with this article. If you would like a copy of the complete text, please send a stamped self addressed envelope. Also please specify which article you are requesting.

Product Review
by Rodger Alexander

63B09E PowerBoost from Burke & Burke was
one of the most simple upgrades I've ever installed. The package sells for $29 and includes the Hitachi HD63B09EP high performance micro-processor, 1 40 Pin IC socket. Software disk for both OS-9 and Extended Basic, and the manual.

The manual is only 9 pages long but is complete in every detail with especially good instructions regarding the removal of the old 6809 processor chip. The manual wisely recommends that if you are not experienced in soldering or de-soldering, that you get an experienced assistant to help you.

It should be noted that it is not actually necessary to remove the old chip. There is a piggy back method described elsewhere in this issue of the OS9 Newsletter.

The most interesting part of the installation is the software portion. The disk contains a program called **PBOOST** that runs under extended basic. Simply place the disk in your drive and enter:

RUN "PBOOST"

This patches BASIC to use new microprocessor instructions which will slightly speed up internal operations. However this speed up will only effect RAM operations, not ROM calls. Under OS-9, a directory of the PowerBoost disk reveals a CMDS directory containing three files:

    booster   **fcheck**   **ezgen**

EZGEN is version 1.09 and has become the standard tool for modifying ones bootfile.

FCHECK is an RBF Fragmentation Check utility. It will read and report or update your RBF device (disk drive). The read out is quite impressive.

BOOSTER is the actual utility that accomplishes the task of updating your OS9 modules and kernel to take advantage of the 6309 processor. *Booster* reads each module using a pattern search rather than a CRC comparison. The advantage being that if you have patched one of your modules changing it's CRC value from the original. *booster* won't reject it like *IPatch* or *ModPatch*, but instead will seek the pattern of bytes within the module that needs to be modified. The patching process takes from 2 to 5 minutes.

What I especially liked was that *booster* is capable of linking with *ezgen* to modify your boot disk directly rather than the standard method of using *cobbler* to copy your bootfile from RAM to a formatted disk.

Chris Burke was at the PNW CoCoFest II and was very willing to answer any questions about his new product. As more "patches" are developed, Chris promises a very inexpensive "update disk" will be made available to all registered owners. Unfortunately, his **"The 6309 Book"** is not yet available. This will also contain a disk with patches to the OS-9 Assembler so that machine language and C language source codes can be compiled into 6309's native mode for faster operation.

Software Review Continued.......

Magazine Index System by Bob van der Poel.

This was another purchase I made at the PNW CoCoFest II for $20. At first this appeared to be fairly a plane jane program. A simple text presentation on your 80 column text screen. No bright CoCo Colors or fancy title screens, whistles, buzzers, etc.

What was impressive was that it contained CoCo Magazine articles from 1988 to 1992. These articles arc not just those found in *The Rainbow* but from other magazines as well.

After the initial title screen, an index window is displayed with a listing of all of the index files that come with the program. Additional files from 1981 to 1987 are available for an additional $10.

```
////\\\\
coco1988.mis
coco1989.mis
coco1990.mis
coco1991.mis
coco1992.mis
kissableos9.mis
\\\\V////
```

You select the desired file by using the arrow keys then pressing <ENTER>. Another window pops up that list the options available:

```
Add new data
Edit current file
Search files
Change file
cHange directory
Utilities menu
Quit
```

Again you can use the arrow keys to highlight your choice or type in the capital letter of each choice.

From the Edit option the first article is listed. At the bottom of the screen are the edit options including using the arrows to scan all of the records.

From the Search option you are prompted to enter key words. With multiple key words any file that contains any one of the key words will be displayed. As an extra bonus, you are even asked if you want the search to be extended to all of the magazine datafiles on the disk.

When I was looking for an article on the IRQ fix to the multipak, I entered: **IRQ, PATCH, FIX,** and **MULTIPAK** as my key words. From the "*coco1990.mis*" file alone I found five matches from *The Rainbow Magazine*. When I selected all files to be searched, I found 52 matches and 6 of them were from the ◆*S9 Clipboard*.

The more I used the program, the more impressed I was. How many times have you spent hours trying to find an article that you remember from a few years back. Now you can find that article and any related articles in just a couple of minutes. This is a must have program.
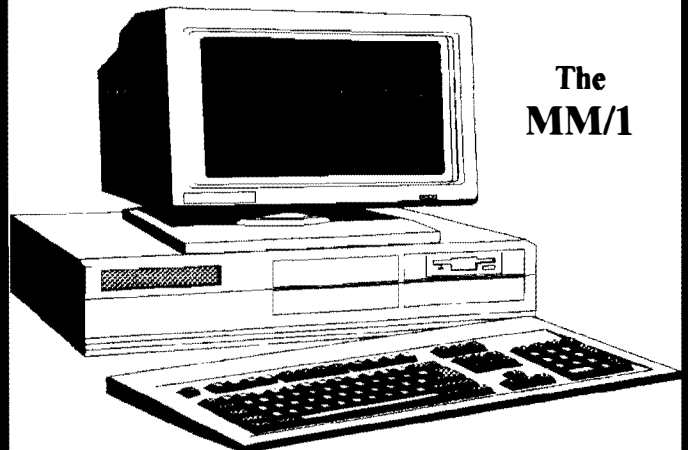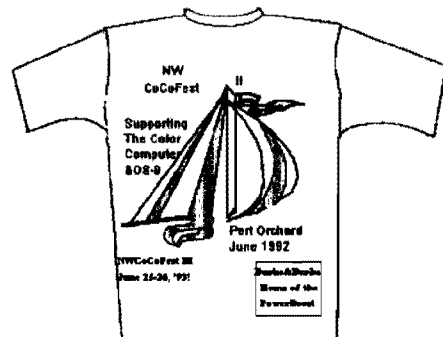
Club Activities Report

Bellingham OS9 Users Group - Longview/Kelso CoCo Club
Mt. Rainier CoCo Club - Port O'CoCo Club - Seattle 68xxx Mug

Bellingham OS-9 Users Group

Attendance was very poor. Only 3 members showed up. So, after a short wait, we moved the meeting to the local Red Robbins Restaurant.

A preview of the CoCo to PC Interface board was shared with a lengthy discussion about binary logic devices such as AND & NOR Gates and the concepts about how addressing is used in computers to access different I/O devices such as the RS-232 Pak.

A final preview of the upcoming CoCo Fest II at Port Orchard on Friday and Saturday. Big news is that Chris Burke is going to be there with his latest "6309 PowerBoost".

Rodger had an 18 page print-out of the 6309 specs. The first 4 pages pretty much explained the differences between the 6809 and the 6309 and the advantages of the 6309. The remaining 14 pages were specific register address listings------All greek to me!

-- Rodger Alexander --

Mt. Rainier CoCo Club

John Schliep kicked off the June meeting with several demos of public domain software from the OS9 Library collection. There are over 12 megabytes of OS9 and RSDOS programs available from this source. These include *pmap*, *smap* and *mmap*. John has been delving deep into these utilities and showed us how to use the information that can be obtained from them.

Chris Johnson gave us our final update on the CoCo Fest. By the time you read this it will be all over and plans for CoCo Fest III will probably be getting under way.

Rick Ouzts then took over the keyboard and gave an interesting presentation of the OS9 Play program. This will play digitized sounds thru the CoCo. Actually, sounds may be a little too simplified because we were able to hear Captain Kirk asking Scotty to beam him up to the Enterprise. There are several digitized sound files available on Chris Johnson's BBS

The remainder of the meeting was devoted to small group discussions and problem solving. Our next meeting will be on July 9th.

-- Alan Johnson --

Port O-CoCo

The June 22nd meeting was completely devoted to the final preparations for the CoCoFest II on Friday and Saturday

For the July 20th meeting there will be two topics for this meeting: Getting feedback about the NW CoCoFest II to polish our jem further for NW CoCoFest III. Also we will be writing a thank you note to all those who attended. Since there were almost 75 people attending, that will take some time. The second part of the meeting will be showing a new development in the computer world: The disk magazine. There are currently two of them for the CoCo. Both are RS-disk based. Thus they will be very helpful to the beginner. We will discuss the content, purpose, advantages, disadvantages, and cost.

There also will be a VERY warm financial report on the Fest. All expectations were exceeded. We had far more attendees than planned. They bought all but 5 of the T-shirts and wiped us out of most of the mugs. Pre-event advertising and expenses were VERY low and so we have a very

handsome reserve for next year's event. The collection of the four clubs in the area will also be discussing the idea of forming some kind of an official organization to promote the CoCo and OS-9.

--Donald Zimerman --

Seattle 68xxx Mug

This meeting featured NO COMPUTERS. Rodger Alexander displayed an overhead of his revised *Search Procedures* for the **PDS Database** program that the club has been working on for the past several months and is featured in the *OS-9 Newsletter*. Severeal more suggestion were by those present to improve the program even more.

Donald Zimmerman presented that latest information about the upcoming CoCoFest II in Port Orchard on July 26 & 27. A representative from CoCoPro will be coming with a trailer load of CoCo Hardware and Software.

Scott Honaker made his presentation of the *Pack Procedure* for the **PDS Database**. He explained the two possible methods he considered and several suggestion were made on variations to make the PACK routine more efficient. Scott's version rewrote all valid records to a temp file, then deleted the old database data file and then renamed the scratch file to the original data file name. On a hard drive system this is satisfactory but has a limitation problem when using a floppy drive or ram drive system.

The final portion of the meeting was unofficial small talk about the upcoming CoCo Fest.

-- Rodger Alexander --

The OS9 Underground (R)

The "OS9 Underground" is a magazine dedcated to OS9/OSk users everywhere...
 Each month a new issue will provide you with...
o Articles and Feature Stories
o New Product News
o Hands-on Reviews of New Software and Hardware
o Programs and Tutorials Columns in Basic09 and C
o Free Want Ads section
o BBS Listings and User Group Information as well as
   a Network News column
o Question and Answer Column for OS9/OSK software/
   hardware
o Covers the Coco, MM/1, TC70 and System IV
   Machines

This is a MONTHLY magazine, with a yearly subscription rate of only $12.00!*(12 Issues)*($15.00 Candian, $20.00 overseas)
To subscribe or to get a FREE trial issue mail your request to:

> "The OS9 Underground" Magazine
> Fat Cat Publications
> 4650 Cahuenga Blvd. Ste #7
> Toluca Lake, CA 91602

Washington CoCo/OS9 Clubs

Bellingham OS-9 Users Group
Meets the 4th Wednesday of each month at 7pm
Bellingham Public Library, 210 Central
(206) 734-5806

Mt. Rainier CoCo Club
Meets the 2nd Thursday of each month at 7pm
Fern Hill Library, 84th and Yakima, Tacoma

Longview/Kelso CoCo Club
Call Steve Hammond for meeting information
(206) 577-7316

Port O'CoCo Club
Meets the 3rd Monday of each month at 7:30 pm
Stock Market Grocery, Port Orchard

Seattle 68xxx Mug
Meets the 1st Tuesday of each month at 7:30pm
Gugenheim Hall, University of Washington

Bellingham OS-9 Users Group
Presents....

5 Tutorial Chapters

10 Hardware Projects

OS9 Level Two
and the
Color Computer3

writting by
Rodger Alexander
Scott Honaker

"Updates"
Disk
with 17 files

$5

Free

Advertizing

Space

for

Subscribers

Buy or Sell Listings

Where's yours?????

Washington State BBS List

FAR POINT BBS (Seattle)
RiBBS (Fido NET)
(206) 285-8335

COLUMBIA HTS. BB (Longview/Kelso)
RiBBS (Fido NET)
(206) 425-5804

DATA WAREHOUSE BBS (Spokane)
RiBBS (Fido NET)
(509) 325-6787

BARBEQUED RIBBS (Bellingham)
PC-Board (PC-Net)
(206) 676-5787 - CoCo Conference #5

OS-9 TACOMA BBS (Tacoma)
RiBBS (Fido NET)
(206) 566-8857

COCO EXPRESS BBS (Anacortes)
RiBBS (Fido NET)
(206) 293-1057

OS-9 Newsletter
3404 Illinois Lane
Bellingham, WA 98226-4238