

Continued from TEDTALK/1

I mentioned the TMS9995 earlier. Just what exactly is a pipeline microprocessor? Well, the 9995 is not only fast, but it has a distinct advantage over others in it's class, even the intel 80386. Those processors rely on expanded address lines and increased instructions to increase throughput. There was a deeper approach, one that TI envisioned in the 9995. A pipeline microprocessor is one that incorporates special hardware that allows it to have more than one part of the microprocessor running at the same time. These CONCURRENT functions provide that while one instruction is being decoded inside the chip, another is being fetched from memory. Still another is being executed after it has been decoded. At best, with top-down code, and very little jumps, the microprocessor can achieve a throughput 3 times, or more, depending on the level of pipelining, over a regular processor running at that speed. For example, if we put test code into a 9995 and a 9900 running at 12 MHZ, the worst case is that the two run even. But the 9995 can pipeline, and with the pre-fetch and post-store the 9995 can LOOK like it's running 16, 20, or even 24 MHZ. And with the reduced instruction set in the control ROM, the 9995 has a distinct advantage over an 80386, it's MUCH cheaper to produce. The control ROM is a hard-wired design, while the 80386 has to be programmed externally. It is an easy device to interface to a memory system, and with no-wait state static RAM, the memory-9995 combination (up to 4 megabytes) can be phenomenal.

Currently, I am working on a software project. It's a new DOS for the TI, somewhat reminiscent of COMMAND DOS that ryte data released some years ago. However, there is no image file required because the DOS I have resides in a E/A supercart, and the utilities that it needs are extracted from the E/A GROM—that way, I can restore the lower memory expansion to a defined state very quickly without reading from a disk drive. The DOS is completely self contained, and will provide a choice for you on the master title

screen. I am a college student, doing projects to complete my final years of undergraduate study in computer science. This project was inspired by a need for a better operating environment for the TI as well as a need for me to see if it could be done. Well, I have succeeded! The DOS uses the DSRLNK utility to attach to the low level device drivers. It gives you the familiar A> DOS prompt, and will mimic DOS to a degree, but with one delightful exception—the DOS is being written by me, and I can have it do whatever I want it to! I will no longer be a slave to incomplete DOS commands or ambiguous and useless syntax, often the product of overpaid software developers. The commands are clear and precise, and the DOS is very short, only about 5k at this writing. Since most of the DOS is already present in our machines, in places like the E/A GROM, the disk controller ROM, the RS232 interrupt routine—all of these put together with the right glue can make a great DOS, and all I did was to provide the necessary glue for the parts, and it works! It has a batch file load and execute, D/F 80 loader (compressed/uncompressed), program file loader, dos utilities (FORMAT, COPY, RENAME, DELETE, ASSIGN) and screen control commands (WAIT, BEEP, CLS, GOTEXY, PRINT, ECHO ON/OFF) and "smart" control keys, as well as a 255 character input queue for type-ahead. Many of the commands are internal, and they reside only in the supercart. Other commands can be created from object code, which you can create from any one of the compiling languages, or the assembler (i prefer the assembler) and by simply typing the name of the file at the command prompt, the file will be loaded and executed.

I hope to have some sort of language compiler for DOS, such as a basic/pascal compiler, to facilitate creation of programs and utilities. My plans include a file transfer utility (terminal emulator), windowing, an 80-column editor, and multiprogramming. If for no other reason, then to gain experience and to enjoy doing it on my \$49.99 TI99/4A. Of course, I wouldn't dream of charging anyone for this DOS, and I've had some interesting

suggestions for names. "F-DOS" by our own editor, BOB DEMETER, for FRINGMAN-DOS, since my "other" hobby is SCUBA DIVING, "XIOS" for eXtended Input Output System, and whatever...I am using version 1.24, which is relatively complete. I would just like to add the bells and whistles, plus write a manual on it's use.

Now for some more TechTalk. If you are confused as to why computers like the c 64 and the apple all have DOS commands built in...well, the designers of those computers anticipated a disk system, and available to most users, so the operating system and BASIC language all had the DOS commands either in the disk unit itself, or in a disk BASIC which loaded in on powerup. Since TI did things a little differently, they preferred to make DOS a separate thing, with a disk manager module to handle disk tests and formatting. It seemed a little annoying that in order to rename a file from BASIC, you had to either load the program and save it under another name, or if it was a DATA file, you had to OPEN it and read all of the data, then re-save the data to disk under another OPENed file name. This could be terribly inconvenient to users, but consider what the others have...the c-64 must send all of it's DOS commands through a command channel, and the disk drive will run itself. It essentially is another computer, a 6502 based one, to be exact, that only accepts commands from a serial line and performs all of the disk commands. Imagine.. a computer so STUPID that you need TWO computers to run any disk software...and you would be paying for TWO computers also. Commodore doesn't tell the average users that they are essentially using TWO computers instead of one. Apple computers are also based on the 6502 series of microprocessors. Apple used an old method of running it's computers...just write a DOS and put it on disk, and when the computer is powered up, the DOS is loaded. Funny thing, though. Although Apple boasts of 64k of RAM, much of that is used to hold the resident DOS, and BASIC. If you want to load a program which needs the space allocated by DOS, you are out of luck, since your program might make DOS calls to perform disk

functions. And if DOS were overwritten, then when your program is finished, it must go back and load it all over again. And 6502 is not exactly the processor I would waste terribly expensive memory on, since it has a very limited instruction set, and things I take for granted now, like memory-to-memory word moves, multiplication, division, and subroutine branching would be terrible to implement on an apple of commodore 64. I just don't know how they have survived this long...

Our little TI, on the other hand, has a wonderful method for handling new devices. The GROM header, present on all ROM in the expansion box, and all command modules, is the link between the unknown and the known. It allows us to plug in new devices at any time in the future, and the operating system will immediately recognize the device, as if it were there from the beginning. This is what will keep our TI computers alive. The method of access is very similar to the IBM pc method. Each peripheral card has a certain address in the serial addressing fields. The operating system can turn on a card singly, look at what occupies a pre-defined memory area (>4000 to >5FFF for us) and can determine if the device exists. With the IBM, certain logical names are assigned to a physical device address, such as COM1:, TTY:, A:, LPT1:, and so on, and can be changed according to the user's wishes. This requires a small modification to DOS to accommodate the new device, and from then on, a new sub-version to dos is created. If the device is removed, an error will be issued since DOS can no longer locate the installed device.

The GROM header in the TI provides a standard table for finding a device quickly and efficiently. All of the devices use a pre-decoded 8k block of memory, and 8k is plenty for most devices. Since we are not limited to 64k of total address space (via memory paging in the MYARC or HORIZON ram cards), larger programs may occupy that memory and give our TI's a greater running capability. The IBM uses a segment register that is pre-decoded to

(CONTINUED ON PAGE 5)

The Computer Programmer

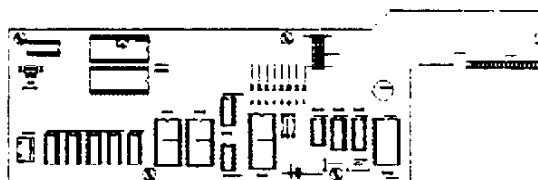
One who passes himself off as an exacting expert on the basis of being able to turn out, after innumerable debugs, an infinite series of incomprehensible answers calculated with micrometric precision from vague assumptions based on debatable figures taken from inconclusive documents of problematical accuracy by persons of dubious reliability and questionable mentality for the purpose of annoying and confounding a hopelessly defenseless department that was unfortunate enough to have asked for the information in the first place.

ANNOUNCING THE "INTERNAL BOARD"

The "Internal Board" AKA the "Zenoboard" is now in production. The PC board is being manufactured by one of the best quality manufacturers around. As promised, this board will allow you to build 32K memory, a clock circuit and add your extended basic and speech synthesizer to the interior of your console. Orders are now being accepted at a cost of \$17.50. Documentation will consist of approximately 8 pages of schematics, builders notes, parts list, software for the clock and parts placement overlay.

Specifications:

- * 32K STATIC RAM
- * battery-backed clock
- * speech synthesizer
- * extended basic
- * 3 additional, switched grom sockets
- * any circuit configuration can be used
- * requires no additional power
- * move your extended basic and speech synthesizer inside
- * eliminates nearly all lock-ups due to extended basic cartridge
- * in addition to soldering components connectors, only 12 additional wire connections have to be made to build a complete board
- * switches lights may be added to turn off any or all circuitry
- * grom reset switch
- * compatible with all other known hardware/software
- * circuit designs have been tested



Orders and technical questions/comments can be directed to:

Eric Zeng 414 Highland Rd., Pgh, PA, USA 15235 (412)371-4779

The boards are being manufactured in quantities of 100. Don't delay and get caught waiting for the subsequent manufacturing runs!

Small modifications must be made to the plastic on the inside of the console. Hand soldering skills are recommended.

Please allow sufficient time for delivery.

Overseas shipping may require additional invoicing.

Sorry no C.O.D., US currency only!

SHIPPING			QTY.	AMOUNT
	USA	OTHER	PC BOARD \$17.50 X	
FIRST PC BOARD	\$2.50	\$8.00	SHIPPING SEE CHART	
EACH ADDITIONAL BOARD	\$.50	\$6.75	8 PAGE DOCUMENTS \$1.00 X	
SHIPPING ADDRESS:			-TOTAL-	
NAME:			ERIC ZENG	
STREET:			414 HIGHLAND RD.	
CITY:			PGH. PA. 15235	
STATE:				
ZIP:				

EDITORS NOTE: You will supply ALL chips and parts for the above project They are NOT included in the \$17.50 price.

(CONTINUED FROM PAGE 3)

page in banks of memory, which is essentially the same way the HRD or MYARC does it, so memory expansion is no problem. The safe area in the TI is the first ROM bank, which is the invaluable interrupt routine and powerup routines. the SUPERCART is the only save RAM alternative for a kernel or DOS, since it is battery backed and it remembers all the changes you have made to DOS. In the CRU, the only area you could use for your UMI bit-twiddling is the >400 to >1000 area, which is not decoded presently and could be wired to something (I will let you imagine that). It would not be a difficult task to interface an IBM card to the TI, provided you had the correct cross-wiring, and a ROM to control the new device. A few of us in the Chicago users group will attempt this. The price of IBM cards is falling like a rock, and I don't see any interfacing pitfalls.

*****EOG*****

UP-COMING CHICAGO TI-FAIRE

Saturday, November 4th

This year's 7th annual Chicago Faire will be held on Saturday, November 4th, starting at 9:00 a.m. The Faire is one of the largest and best attended events in the TI world. Preceding the faire on Friday night will be a mixer which will include snacks and a cash bar, and following the Faire on Saturday night will be a banquet for which reservations will be required. Admission will be \$5.00 and the Faire will be held in Rolling Meadows, ILL at the:

**Holiday Inn
3505 Algonquin Road
Rolling Meadows, IL 60008**

Faire info Hotline:

(312) 869-4304

For those of you interested in attending, arrangements can be made at our September and October meetings for car-pooling.

OFFICERS

**PRESIDENT....JOHN POWELL 786-3270
VICE-PRES.....CARL CLARK 1-398-6226
LIBRARIAN..BRYANT PEDIGO 255-7381**

DISCLAIMER

This newsletter is brought to you through the efforts of the officers and members of the HOOSIER USERS GROUP. Every member is encouraged to submit articles.

If you have an article you would like to share with the other members mail it to:

John Powell
327 W. Southern Ave.
Indianapolis, IN 46225

Opinions expressed are those of the author and not necessarily those of the HOOSIER USERS GROUP.

MONTHLY MEETING LOCATION

**ST. ANN'S SCHOOL
2839 S. McCLURE
INDIANAPOLIS, IN
MEETINGS OPEN AT
2:00 PM
SEPTEMBER 17 1989**



INSURANCE SALES

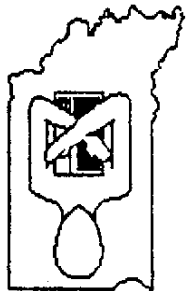
"Mr. Jones is awfully busy but our computer can spare you five minutes."

89/10 781DE
Dan H. Eicher
4410 Cardinal Drive
Indianapolis, IN 46237

TIME DATED
September 17 1989
MATERIAL



HOOSIER USERS GROUP
P.O. Box 2222
Indianapolis, IN 46206-2222
Forwarding and Address
Correction Requested



APPLICATION FOR MEMBERSHIP

Below you will find an application for membership to the Hoosier Users Group. Active membership entitles you to the Newsletter, up and download on the HUGbbs, attendance and voting rights at regular club meetings, access to the HUGger Library of Programs, special club activities and special guest speakers for one year. Subscribing members will receive the **NEWSLETTER** only.

Make check or money order payable to **Hoosier Users Group**. Send completed application to:

HOOSIER USERS GROUP
P.O. Box 2222
Indianapolis, IN 46206-2222

(Cut on dotted line)

Check One:

Active Member

New: \$20 _____
Renewal: 15 _____

Subscribing Member

New: \$10 _____
Renewal: 7.50 _____

Amount Enclosed: \$ _____

_____ D _____
S _____

Name: _____ Today's Date: _____

Address: _____ Apt. # _____

City: _____ State: _____ Zip: _____

Phone: (_____) _____ - _____

Interests/Comments: _____