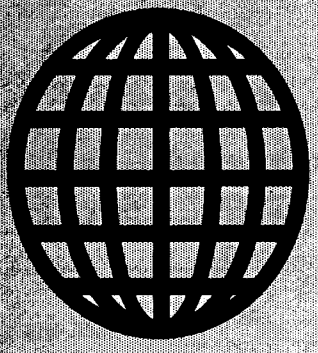
Providing Support Around The World



The Computer Journal

Issue Number 68 July/August 1994 US$4.00

Small System Support

PC/XT Corner

Z-System Corner II

Mr. Kaypro

Real Computing

Support Groups

Little Circuits

Multiprocessing Part 4

Centerfold - Pertec/Mits 88 - 4 PIO

The Computer Corner

ISSN # 0748-9331

TCJ - For Having Fun With Any Computer!

The Computer Journal

Founder
Art Carlson

Editor/Publisher
Bill D. Kibler

Technical Consultant
Chris McEwen

Contributing Editors
Herb Johnson
Charles Stafford
Brad Rodriguez
Ronald W. Anderson
Helmut Jungkunz
Ron Mitchell
Dave Baldwin
Frank Sergeant
JW Weaver
Richard Rodman
Jay Sage
Tilmann Reh

The Computer Journal is published six times a year and mailed from *The Computer Journal*, P. O. Box 535, Lincoln, CA 95648, (916) 645-1670.

Subscription rates within the US: $24 one year (6 issues), $44 two years (12 issues). Send subscription, renewals, address changes, or advertising inquires to: *The Computer Journal*, P.O. Box 535, Lincoln,CA 95648.

Registered Trademarks

It is easy to get in the habit of using company trademarks as generic terms, but these trademarks are the property of the respective companies. It is important to acknowledge these trademarks as their property to avoid their losing the rights and the term becoming public property. The following frequently used trademarks are acknowledged, and we apologize for any we have overlooked.

Apple II, II+, IIc, IIe, Lisa, Macintosh, ProDos; Apple Computer Company. CP/M, DDT, ASM, STAT, PIP; Digital Research. DateStamper, BackGrounder ii, Dos Disk; Plu*Perfect Systems. Clipper, Nantucket; Nantucket, Inc. dBase, dBASE II, dBASE III, dBASE III Plus, dBASE IV; Ashton-Tate, Inc. MBASIC, MS-DOS, Windows, Word; MicroSoft. WordStar; MicroPro International. IBM-PC, XT, and AT, PC-DOS; IBM Corporation. Z80, Z280; Zilog Corporation. Turbo Pascal, Turbo C, Paradox; Borland International. HD64180; Hitachi America, Ltd. SB180; Micromint, Inc.

Where these and other terms are used in The Computer Journal, they are acknowledged to be the property of the respective companies even if not specifically acknowledged in each occurrence.

TCJ The Computer Journal
Issue Number 68 July/August 1994

EDITOR'S COMMENTS

Welcome to number 68 of The Computer Journal. It has been a small struggle to get this issue out, with summer in full bloom and far too many activities calling for one's attention. So pull up a comfortable chair and sit back and relax from the heat. TCJ is here to entertain you!

As usual we start this issue with letters from our readers. I am still trying to catch up on past letters and diminish the pile. I fear I am failing. With eight pages of letters, I still have just as many waiting in the queue for next time. Let us hope the mail person forgets to deliver any more letters to the editor.

Rick Rodman is next with comments on his TINY-TCP and what is happening in the high power world of 32 bit systems. Seems the MCA bus has died, long live the (expletive deleted) ISA bus.

Since Jay Sage is taking a long earned vacation, and I have had many requests for some beginner Z-System support, a brave and daring sole has offered to put his little neck on the old chopping block, by writing for the new Z-System Corner II. Ron Mitchell from Canada graced our pages last issue with a report on the Trenton Computer Fest. He got bit by the writing bug and starts this series with an introduction to some of the CP/M features. Although slanted to the beginning reader, Ron makes some great beginnings that all will enjoy reading. Thanks Ron!

Dave Baldwin is working his way to doing some Z80 construction projects by introducing more Little Circuits and how to keep them from being problems. Many readers are probably unaware that true CMOS circuits have different interfacing problems. Dave gives some solutions.

Our language discussion has moved into Ronald Anderson's Small Systems Support section this issue. Ron has had plenty of years experience using them all, and re-tells his experiences and concerns. Ron also comments on 6800/6809 languages and some ideas in that area.

Mr. Kaypro is not a singular person this issue. Charles Stafford answers some question about figuring out what to do with a stock Kaypro. JW Weaver explains how and why he "Opened and Closed" his Kaypro for repairs. Lastly in the group is an article that had been hiding in my file system for sometime about using the extra function keys on the Kaypro Keyboard. Jack Wyatt has offered up the information and I will be putting his library file on several BBS's.

Frank Sergeant's next article will be dealing with parallel ports and stepper motors. I tried to get all his material in this issue, but just no room. So this time we get the "Bit of Everything" from Frank. To warm you up and maybe get the questions going, I selected the Pertec/Mits 88 - 4 PIO S-100 card for the centerfold. This 4 port parallel card is great for learning about address selection and simplicity of talking to parallel controller chips. When we take what you learn from this centerfold and add Frank's information on using parallel ports for steppers next issue, things will start stepping along for sure.

The fearless writer and hardware hacker Brad Rodriguez sticks his neck out by making his 6809 boards capable of using PC compatible cards. Since I fully endorse this concept and want to do the same with a Z180 project, we will all be following just how well this option works. I'll be checking out this concept myself with my new PT68K motherboard. So check out the drawings and get those soldering irons burning (solder not skin, please), Brad's next installment is here.

Last issue I published the GiMiX SS-50 mother board as the centerfold. Well why not, GiMiX is dead right, WRONG! Got a postcard advertisement from GMX and called to find out the full story. That story unfolds in Support Groups for the Classics. Also check out some of the European Zed-Fest happenings.

Dragging up the rear of this issue is none other than ME? Well editors do have some privileges and being last is my choice. What I talk about is some changes in the PLC market, getting cheaper by the day. I also explain a little about polyFORTH and just set the stage for a later more detailed explanation.

Slow and Late

If you have called or left messages expecting a quick response, sorry that doesn't happen around here. I have plans to get a live person on the other end of our 800 number, however those plans keep stretching out into next year. So till they happen, I have to fit all this into what little time is left in a rather long day.

You can help considerably by leaving complete messages on the answering machine. Please explain your problem and leave phone numbers to get back to (with hours). I have been on site lately and thus not home for several days at a time. That only means the backlog of calls and mail stacks ever higher. Now the best help possible would be to renew on time, early would be even better! Your label has the last issue you will get, unless you renew early. Please spread the word about TCJ as still far too many collectors and users of these wonderful older systems don't know about us, yet!

With that being it, it is time for you to enjoy this collection of hard work from our dedicated writers. Enjoy and Thanks for supporting TCJ. Bill Kibler.

READER to READER

Letters to the Editor

All Readers

MINI Articles

Mr. Kibler:

I would like to thank you for having a forum for older computer equipment. Being a new subscriber, folks like me need a place where we can get together and help each other.

I'm glad that you have the services of Mr. Ron Anderson. He is one of the best on the subject of 6800/6809 SS-50 bus systems. I'm also sure that he would be an expert on many other systems. He had a column in the old *68 Micro Journal*, called 'FLEX USERS NOTES', for about 10 years until the 68MJ folded (too bad). I'm going to look forward for his column from now on in *The Computer Journal*. Mr. Anderson is now helping me with a problem with a SwTPC 6809, which I want to thank him very much. God bless him.

I remember Mr. Stan Veit very well. I first met him when he shared a portion of a store on 5th Ave. & 3rd street in New York. I spent many hours looking over the different equipment. IMSAI, KIM, Altair SwTPC, Just to name a few. I was going to buy an IMASI, but I purchased a few magazines, looked them over and when I read that some cards for the S-100 Bus were not compatible, I took a closer look at the SS-50 Bus. These were SwTPC, GIMIX, MSI, SSB. I decided to go with SwTPC, which I purchased in Mr. Veit's new store at 31st Street off of Madison Avenue. It was a SwTPC 6800 CPU with 8K memory, CT-64 Terminal, AC-30 cassette interface, PR-40 40 column printer, and SwTPC cores, Disassembler, & 8K Basic 2.0. All this was in kit form. When I finished building it and had one problem, which a fellow who worked for Mr. Veit, I think his name was Carl found the problem, I was now able to use my system. I soon up-graded to a full 48K memory, added 5 inch diskettes with Flex 2.0, also percom CIS-30 (300-600-1200) cassette interface, and from Mikbug to SWTbug Monitor. The system was bought in 1977 and is still operating today. In fact I'm using it to write this letter with a mini word processor I've written in CSS Basic 4.3. Mr. Veit's store was like a club house for all computer fans. He even started a computer club which had meetings once a month after hours in the store. Everything was just great. Then one day a notice on the store. The store was going to close. This was a black day. I felt very bad. That was the last time I saw Mr. Veit before I found the *Computer Shopper* magazine. I'm glad that he is doing well and I'm happy that he has written another book about computers. It was a book for beginners. In case you see this, Mr. Veit, what happened to the super SwTPC technician you had in the store, Carl? I would also like any body who lives in the San Antonio area in Texas to find out what happened to.

Southwest Technical Product Company
(SwTPC)
219 West Rhapsody
San Antonio, Texas 78216

SwTPC has since stopped making computers, but I have heard that they are still in the electronic business. I don't know if they are still using the SwTPC name. I would like to get the name, address, telephone, if possible. The owner and president was Mr. Dan Meyer.

I'm looking forward to receiving every issue of *TCJ*. Thanks.

Sincerely, John J. Fiorino, Brooklyn, NY.

Well thanks John for for all those nice words about Mr. Anderson and TCJ. It is slow getting the word out about us, but many people are finding and using our services to help keep their old systems running.

Mr. Veit's book covers all the goings on in his store and many of the old systems. I suggst you get it and relive some of those old memories from his view point. And I do think in his book he mentions that SwTPC went to other electronic ventures. As to them still being in business I do not know (as of this writting-send letters if you know!)

Reader To Reader is set aside for people like you that need and have questions on old systems. If you have a good old timer like story to tell about way back when, I am looking for articles on just such topics. One of these days (maybe this issue,) I will recount my first experiences with IMSAI and later Micropro. So you see John, lots of us have experienced the computer revolution from the beginnng, what we haven't done however is told others just what it was like, "in the old days."

Thanks again John! Bill.

Dear Bill,

Received your Issue Number 65 today, and figured it was time to write you a thank you letter. At a time when so many slick and glossy computer publications are turning to drawn-out marketing dross for Windows, MS-DOS, and network products; it is a pleasure to receive your publication.

Where else would I read about 68000 hardware and operating systems like

PT68K and SK*DOS? Where else would I find a Sinclair schematic, or S-100 or CP/M? Where elese would you read about National's 32K chips? (Gee, not even in Rick Rodman's column!) And where else would I find our poor Sanyo support group listed?

Keep it up guys! I just hate it when my 2-3 foot high stacks of slick computer publications slip and slide all over the place. When the revolution comes, they're going to be first to be pitched!

In case you didn't get (or read) California Digital, Inc's latest yellow brochure, you and your readers might find their offering licensed copies of CP/M with completely documented manuals for $9 (tax & shipping extra) to be of interest. When I phoned their toll free line (800) 421-5041, I found that $25 was their minimum credit card order, but I could order through the mail (17700 Figueroa St., Gardena, CA 90248).

You will still need a CP/M-80 computer with a disk format compatible to the one that you wish to install CP/M on. This is a job that separates the men from the boys, but if completed successfully will leave the installer with more understanding of his computer and operating system that 99.9% of the present generation of computer users.

Yours truly, Victor R. Frank, K6FV, Editor of *Sanyo PC Hackers Newsletter*.

Dear Bill,

Received the CP/M that California Digital is Selling. The disk is 5.25" SSDD and for the Xerox 1800 computer. At $9 plus tax and $3 shipping, I guess the binder and documentation is worth that. Section 6 of the manual is entitled CP/M2 Alteration, and they *do* have a sample BIOS and describe the Disk Parameter Tables, their DISKDEF Macro Library, and example of loaders sufficiently that a programmer *could* install it on a different system, provided he had the MAC assembler, a working CP/M-80 computer, and was sufficiently motivated.

I am not so motivated at the present time. Enclosed is a screen dump of the directory of the CP/M disk supplied.

CPM.SYS, DDT.COM,
PIP.COM, SUBMIT.COM,
XSUB.COM, ED.COM,
ASM.COM, LOAD.COM,
STAT.COM, DUMP.COM,
DUMP.ASM, BIOS.ASM,
DEBLOCK.ASM, DISKDEF.LIB,
DSKMAINT.COM.

Yours truly, Victor R. Frank, K6FV.

Thanks Frank, and I always enjoy getting your newsletter. Your newsletter gives me great pleasure to see how others are handling being cut-off by vendors once they move to greener pastures (sort of like most other computer magazines).

You hit our position correctly, that being the only way to become fully computer literate is to do it your self and those $9 CP/M versions certainly will require plenty of learning and work. After the second or third one I did, it seemed nothing could be that difficult again. You really learn what's going on, give it a try.

Thanks again for your support of classic systems. Bill Kibler.

Dear Bill,

My subscription has almost run its course, so I want to renew now so as not to miss an issue. I am enclosing a check for another year!

I am also more than a little dismayed at the angst expressed in the last issue over a "standardized" language for the magazine. I don't think that you are going to be able to reslove the issue in a equitable fashion. The support for Pascal (it appears to be strongest from European readers) would have been easy to predict. Personally, I hope that it is NOT what you pick. I am partial to C, however, that is a bag of worms too. When you consider that most readers are working with 64K memory, C (even Small C) will be a hard sell. It is too severely limited in that amount of memory. I don't know what the best answer is.

I would also like to attempt to dissuade you from your stated goal of supporting IBM PC and XT style computers. I see this as a dilution of the material that I am most interested in: S-100 and Z80 based systems. The PC and XT architecture is pitiful, and the Intel 808x/8018x instruction sets are difficult to work with. IBM certainly did the computing world NO favors with the introduction of such mediocrity. Please don't waste time and space on systems such as these!

I'm a collector of old systems, mostly Z80 based. I have several Cromemco systems, a CompuPro, an S.D. (that I built mostly from kits), a C. C. I. (NorthStar CPU, disk jockey controller), and a number of Intercontinental Micros. Besides these S-100 computers, I have a Kaypro 2, a number of Xerox 820's, a Xerox 820 with double density upgrade (that thinks it's a Kaypro), a Sun 2 (multibus), and an MAI Basic Four (68010 based running a UNIX deivative). I'm sure that I've forgotten something. some of my current projects are: general repair and restoration, replacing old 8" disk drives with 5.25" 1.2M drives, linking these old machines with an RS-232 network, and replacing 4 MHz CPU's with faster (6 MHz and up) ones.

I'm also wire wrapping an S-100 memory card for my Cromemco's that uses some of the new 32K x 8 static memory and EPROMs, and supports some form of bank select.

In the future, I will be working on adding hard disks to some of these systems (both MFM and IDE), and I want to try to figure out a way to attach a "clone" Hercules video adapter (that can now be purchased for between $10 and $15) to a Z80 computer.

I hope this will give you some idea of where my interests lie, and will be of help in your direction of *The Computer Journal*. Good Luck!

Thanks, Roger Hanscom, Alexandria, VA.

Well thanks for the letter, Roger. I sure would like an article on your Xerox con-

version to a Kaypro. I know many of our readers have Xerox boards (I got a couple systems) and making them think they are kaypros would give us a bit more software. Your collection is a bit like mine and keep following Rick Rodman's articles on Tiny TCP so you don't reinvent the linking of computers together.

I agree that PC/XT design is bad, but the cost is had to pass up. I think since you sent your letter I have added Frank Sergeant for our XT support. After you have read a few of his articles, I doubt you will still feel I would be "selling out" other support for that platform. We have a tendancy to support what ever is being done by my writers. I do get to redirect some activity, but not all. My keeping, or trying to keep a few regulars has worked very well and not demissed my old time readers and supporter.

I have a number of projects looking for homes, and a few coming to pages soon (like IDE on S-100). Others are just waiting for more input (Z180 on XT bus) or consensus (like languages). I think languages has mellowed out a bit, with the general feeling that comments and explanations are more important than a standard language. And, yes C is very problematic on small computer systems.

What we need really now is more subscribers like yourself. We know they are out there and just haven't heard of TCJ yet. Can you spread the word? Thanks again. Bill.

Hi ya!

David Greelish, CompuServe 100116,217

Alright, now we're in touch "virtually"! How goes it? Did you get my phone message? I received *TCJ* #67 last week and I am still enjoying it. About my classified ad - never mind. I would like to run a Market Place ad instead. When is your deadline for next issue? Got an interesting ad the other day which I thought I'd go ahead and forward to you. Your readers who indulge in single board projects should relish this > davethez@netcom.com I have 200 Apple Mac 512 motherboards and 70 Mac 128 motherboards for sale. I'm asking $10 each for the 512 boards, and $7 each for the 128 boards.Dave Zampino620 Alamo Ct, #4Mountain View, CA 94043415/ 967-6628

What a deal huh?! Hope to hear from ya soon.

Take care, David, HCS

Wow David. What a deal. If I had the time I'd buy some, but just got a PT68K instead, oh well...If it seems I am a bit late returning calls, I have been on site for several weeks and several more to go. So you and any reader a bit upset with my slow phone response, just call my employer and complain, I am sure he won't mind. However if every reader got some one else to subscribe, I might be able to do this full time and thus answer your calls more quickly. Baring that or a lottery win, I'll catch up when I can and settle that ad problem. Thanks David for the word and your great Historical Computer society magazine. Bill.

From France VIA Compuserve friends:

"Emmanuel Roche is disassembling the "NCR DecisionMate V" GSX CRT driver. This driver uses four I/O ports: 10H, 11H, 0A0H, and 0A1H. From the code, Emmanuel thinks that ports 0A0H and 0A1H give access to the NEC uPD7220 GDC and that ports 10H and 11H give information about the hardware (color or monochrome screen). Could someone confirm and give the requested technical information? (No! Emmanuel doesn't have an NCR available: he merely disassembles the driver to see how DRI did it... (and maybe to port it to the Epson QX-10, which uses the same uPD7220...))"

Emmanuel Roche can be contacted at:

8 rue HERLUISON
10000 TROYES
FRANCE

From Emmanuel, a early mini-article:

Intel HEX file format

The intel HEX file format was designed for use with the Teletype ASR-33, the standard I/O device for mini-computers at the time of creation of the Intel 8080 micro-processor (1973). Hence it has no lower-case (like the TTY), and uses only ":", digits, and upper-case letters. Gary Kildall dixit: "The translator output for an Intel 8080 microcomputer is a "hex format" file, contianing a sequence of absolute records which give a load address and byte value to be stored starting at the load address." Each record is composed of the following items:

: nn aaaa tt d1 d2 ...dn cc

A one-character-long header: a colon (":") indicates the begining of a record.

A two-digit-long data Record Length (RL): this is a two-digit hexadecimal value indicating the number of data bytes in this record. The oldest HEX file I have used 28 bytes (1Ch). The mid-1977 revision of the Intel Assembler proably took the opportunity of changing this to the present standard of 16 bytes (10H) of MAC. The last record of a hex file is denoted by a Record Length of 00H.

A four-digit-long Load Address (LA): this is a four-digit hexadecimal value (most significant byte first), which specifies the address from which the data bytes in the record are to be stored. In the case of the last record, this value represents the address to which the PC (Program Counter) of the micro-processor (and hence DDT and SID, if you load the HEX file under them) will be set (or the address of the first instruction to be executed, if you prefer). That's why the MAC manual recommends that you end your assembly language programs with: "END 0100H", which is the normal begining of the TPA for CP/M.

A two-digit-long Record Type (RT): this is a two-digit hexadecimal value, which indicates the type of record:
00H = absolute record for CP/M
01H = end-of-file record for ASR-33 Teletype autostart paper tapes!

Up to sixteen pairs of two-digit hexadecimal data values: the data bytes.

A two-digit-long Checksum (CS): this is a two-digit hexadecimal value representing the two's complement of the sum (ignoring overflow and carry) of all the previous bytes in the line (in binary form). If the byte values are added, starting immediately after the colon, and, continuing through to the end of the record, including the checksum byte, then the sum should be zero when computed with an 8-bit counter. The checksum byte is included as an error detection mechanism, and is used by PIP to check correct transmission of HEX files, using only upper-case ASCII characters: no need for sophisticated protocols for binary transfers!

If there is any error, or if you know more than me, be kind to tell me.

ROCHE Emmanuel, France.

Thanks Emmanuel for the work you are doing and the two care packages you sent me. The explanation on HEX format should help some of our readers, I know I hadn't thought about "adding the checksum should produce Zero", before. I think we will have to get you to summarize your work and findings on GSX. What GSX provides is the graphics tools needed to have windows on a CP/M system. I have not called Novel, but suspect as you have that GSX and any CP/M material is still not public domain. As such we can't really publish all the dissassembly work you have done. Unofficially it could get passed to some of our Z-System people who could (maybe) clone a version for use with ZCPR?

What is most interesting about the article you copied to file was their feeling how CP/M was going to be the end all operating system. GSX was to be the tool to help make it all easier and better. Too bad Bill Gates blew their hopes away and we got stuck with an inferior operating system design, of course IBM should really get most of the blame for that.

I promise you a letter back as soon as I get some time free, until then thanks for your work! Bill.

Greetings! Some comments about issue #65:

PAGES: I always enjoy your *cinema-verite* approach to publication, but I think things went too far here where my issue at least is missing page 19 and 20 and some others. I know pages sometimes wander-off, but I couldn't find these anywhere...

8048 emulator. In the letters, Tilmann REh (I think) suggested my 8048 emulator might make a good article. I'm flattered but I suspect it wouldn't; that is, the answer to the question "how it's designed and built up" is "with great difficulty". It's around 30 ICs, has I'm sure the usual amateur-night drawbacks (i.e., it works for *me*), and in any case you would absolutely need an 8048 EPROM burner first - which I cunningly bought instead of building. However I'd love to upload this and numbers of other projects to some suitable BBS that can be reached via the phone system, but I honestly don't know of such a one. Suggestions?

Walnut Creek SIMTEL MSDOS CDROM. This was mentioned two or three times at least in issue #65, and sounds wonderful, but I can't figure-out where you can get it! I realize if I had only been paying attention for the last 10 years I'd know all these things, but please print a source....

Best wishes, J. G. Owen, South Huntington, NY.

It took me sometime to fit this in, but thanks and hoped you enjoyed the make up copy of #65. Yes our printer sometimes skips a page here or there, but feel good that at least your pages were there. I got a big surprise the other day when a reader returned the cover to issue #67. It seems that was all he got, no pages, none! I usually stop whatever I am doing and ship out makeup issues if I hear about them. Unfortunately I can't promiss it will not happen again.

I say this often, but please consider writing about your project, even if it seems a bit off the mark. Our readers want to know what others have tried and especially where you encountered problems. Yes your project may not be the best, but tell us why you consider that. Suppose I woke up one night at two in the morning and considered I had just come up with a solution to a design problem in my sleep. But suppose a major flaw or nightmare had been forgotten. Before I start the expensive and time consuming design phase, I happen to read your article descrbing all the problems and failures you had to overcome. Since my design was almost like yours, I would know then, that it was a nightmare and NOT the end all solution I was looking for. Many wonderful sages have stated that we learn more by our mistakes than anything else we do. So please tell us about your mistakes as well as your successes.

I am really puzzled how we could have missed explaining about Walnut Creek CDROMS. They are a business that publishes CDROMS and can be reached at (800) 786-9907. They are and have been for over a year promising a CP/M CDROM in a few weeks (just called and got the same story, again). Currently only clone machines can read CDROMS, although some have been talking about CP/M systems using SCSI being able to read them. We are all waiting (for what I am not sure, but why buck the masses).

So thanks for the letter and what about that article. Try uploading to the DIBS or JW BBS's listed in Support Groups. Bill Kibler.

Dear Bill,

I'm writing to you about the latest *TCJ*, another good issue. I'd like to make some general comments on the issue.

ZX-81

I liked the Sinclair ZX retrospective, and like many TCJ'er I was a ZX user. When they hit the retail stream I was working at a small university teaching a class called "Computers Can Be Fun". It was an adult extension class, and I was

teaching Basic on a mainframe. It was an aggravation for the adult students to get terminal time. I saw the ZX and decided that they would work for the class. I got the local store to sell them at cost. We distributed the computers to each student along with the course material. With each student having their own computer it was no problem for them to get the home work done.

One of the students was an 70 year old grandmother that was living in a nearby adult care center. During the third week of class I got a call from the manager of the center. The other residents were complaining that my student was monoplozing the lounge TV to do her homework. I went to our Instructional Resource Center and got her a TV to use for the class.

I taught three semesters using ZX's, I created a "More Computers Can Be Fun" that I taught on the ZX. Everyone that took the class thought that owning their own personal computer was really cool.

Tiny-TCP

I wish Rick Rodman luck getting Tiny-TCP to work. I also saw code on the Source CD. It looked like it would take alot of work to get it to be useful. It needs the bi-directional FTP and a simple character Telnet. I hope that Rick can get all that to work.

Basic Stamps

I've been using 8052-Basic controllers for some time now. I like being able to quickly program them in a higher level language. The setup that I use has the program area in a battery backed CMOS RAM. The system thinks it is writing to a PROM when it saves the programs.

I've been trailing the Parallax Basic Stamp. They are based on the Microchip Technology PIC family. It consists of the PIC microprocessor with a basic interpreter in ROM and a serial EEPROM for program storage. While the program space is limited there is lots that these systems can do. The PIC processor has 8 digital I/O lines and a serial port for program download or data interchange. Not bad for $40.

For the true *TCJ* hacker you can program the PIC using a $70 programmer. There were a series of articles in Electronics Now on how to build the programmer. The PIC supports a RISC like Instruction set which makes it very easy to program.

With your interest in Forth, you might want to consider building the "Forth Stamp". The level 0 Forth may be enough to imbed in the controller. There is a limit to what you need to do inside an embedded controller. The PIC's come in a number of configurations. The PIC 16C56 (the chip on the stamp) has 1K*12 of EPROM and 32 bytes of RAM. The 16C57 has 2K*12 of EPROM. I don't know how much space a Forth kernel would take up.

Anyway, it's a idea. I'm happy with the Basic version, but I'm sure you can think of tons of applications for the Forth version.

Back Cover

On the cover it says "This is your LAST ISSUE". At this point I'll bid you a fond farewell. I've been a reader for a number of years, from Art Carlson era thru Chris's to yours. While it has been enjoyable, I no longer have any of the older systems. I've been thru three Zorbas, two Xerox 820s, a pair of Model 100's, and NEC 8041 CPM laptop. I'm pretty much MS-DOS 28/38/486 systems with a plethora of embedded controllers.

I wish you luck in your future endeavers, it has been a pleasure.

Sincerely, Foster Schucker

Well Foster, thanks for that excellant mini article. Sorry to see you go when it sounds like you could write a few very good articles about your past and present work. I am still having trouble getting my embedded people to stop working and grind out an article for us. That is why prople like you could do so much for TCJ now.

Your course sounds just like what we need now in TCJ. I'am looking for someone to provide some Basic fun projects that also teach. I guess maybe I am a few years too late for your class notes!

Well I think Rick found all the bugs and by now should have Tiny-TCP talking amoung CP/M and MS-DOS systems. A Forth Stamp sounds great and I know just the people to do it, but I am afraid they are too busy with other projects to find time to do the port.

Thanks Foster for your support and sorry to see you go, as I think you will not find this kind of support elsewhere. Bill.

Dear Mr. Kibler,

Enclosed find $24.00 for my subscription renewal.

As you might gather from my address, I live in the middle of nowhere as far as computing is concerned. It's 26 miles to the nearest Radio Shack, and a whole lot further to any place that knows about computers. Nobody knows anything about anything other than IBM or Apple anymore anyway, so I am pretty much on my own.

I have subscribed to *TCJ* since issue 20-something and still find it valuable. I especially like the support for the old 8-bit systems (which I still use) such as Kaypro. I have one — started as a 2/84 and got converted to a 10/84 with TurboROM, custom BIOS, and other goodies. I also have a Morrow MD-3 which has undergone considerable hardware and software modification. Hardware and low-level systems programming information is hard to come by for these older systems, and TCJ is about the only source I know of. The Support Group column will be especially valuable in the future. (I am sending a separate letter to J.W. Weaver as I have quite a bit of hardware technical info on the above two systems.) I have also learned much from the Advanced CPM column in the past. I would like to see more of that kind of programming information.

Again, thanks for the support for the older systems.

Sincerely, Jim Sinning, Williamsburg, IA.

Thanks Jim for your renewal, but I didn't think Iowa was really that far removed from civialization. All kiding aside, it can be very hard even in big cities to find people who have any true knowledge about computers, especially non-MS-DOS units. I find so often that those who say they are experts only know MS-DOS from a users perspective. Ask a serious how does it really work question and watch them start asking "why would you want to know anyway?"
Our User Group section is a bit in flux, JW has had some hardware and back problems, so I have been mostly doing double duty these days. I am interested in finding a better way or changes to that section to help people out better. For now I feel this Reader to Reader section might be just what most people want and need to find support. What do you think?

Thanks and how about a mini article on all those upgrades you did? Bill Kibler.

Dear Bill,

Enclosed please find a check for $44 for a two-year subscription renewal. I'm not doing much with 8-bit iron besides pushing it around and trying to make enough room to set some of it up again just now, but *TCJ* is one of few computer magazines for which I manage to find the time to read the whole thing.

The speculations (by you and Rick Rodman) about networking CP/M systems are personally interesting; I work for a company (The Wollongong Group) that has been in the TCP/IP networking business for around 10 years, and there are two or three of us here who have joked about doing TCP/IP for CP/M and Apple][systems, but haven't really done anything with it. (Well, that's not quite true; I've personally accumulated quite a bit of hardware, including a couple of S-100 Ethernet card sets from LRT!) I do think it has serious hack value, but fear that it's proably easier to put minifloppy drives on our CP/M systems and use a cheap PC/XT clone and something like 22DISK or Bridger Mitchell's DosDISK in conjuction with Sneakernet if all we want to do is move an occasional file.

Of course, given the low prices I see used PC/XT and PC/AT clone systems, as well as older Macintoshes, going for down there in Sillycon (sic) Valley, and the spotty availability of usuable CP/M systems, I'd venture to say that CP/M systems are destined to be the domain of antiquarians, hackers, and committed folks who already have working systems and aren't anxious to change. Sad, but I do have to admit that it was easier to get my mother the non-computer-literate to use a Mac versus an Apple][with Applicard CP/M and Wordstar.

Back to TCP/IP for CP/M, my opinion is that it's proably possible to do something along the lines of a network client (e.g. Telnet or FTP user program) with an embedded TCP/IP stack (i.e. the TCP/IP is statically linked with the client program); the early MIT PC/IP software for MS-DOS was done this way, as were the earliest versions of our MS-DOS TCP/IP product (although we've long since converted to using a TSR-based TCP/IP stack). You could probably even have the TCP/IP stack stay loaded all the time, but given the limited memory available in many CP/M systems might not want to.

The best bet for a network connection is probably a serial port. IP can run over that using SLIP (Serial Line Internet Protocol), which is a very simple (read "small and light") encapsulation for IP. I've also thought that it would be interesting to do an Ethernet connection, but the LRT card sets are not so widespread (so far as I know); a more likely prospect is to use an IBM PC-style parallel-port-to-Ethernet adapter, but the only one of these I'm familiar with (Xircom) supplies their own driver to drive the parallel port and may not document what would have to be done to use the adapter with a generic parallel port on a non-MSDOS system.

If either you or RICk Rodman has an Internet archie client available, you may want to look for something called "tinytcp." It's just that: C source for a tiny TCP/IP implementation that was originally developed for embedded system to download its boot image over an IP-based network. It's not very robust, but it might be a good starting point for a small network kernel for CP/M. If you can't find it, I can probably dig it off one of my diskettes around here and e-mail it to one or both of you.

One other note: along with your decision to expand the magazine's scope to PC/XT-class systems, you might want to take a look at some of the palmtop systems like the HP 100LX, Zeos Pocket PC, Sharp PC-3000 (I think), and Poqet PC. These are effectively PC/XT-class systems that fit in the palm of your hand! I have to admit that my 100LX is a near-constant companion; I even have Turbo C 2.0 installed on it so I can hack anywhere!

Enough of my ramblings for now. If you want to contact me, feel free to send me e-mail via CIS where I am 76400,2655, or to my work (Internet) mailbox at <frank@twg.com>. Best wishes for you, yours, and the rest of the *TCJ* writers and readers in the new year!

Sincerely, Frank McConnell, Mountain View, CA.

As one can guess this has been in the "to publish" pile for some time now. Still, thanks for the adivce Frank, we went that way with the project on TCP as you thought we should. I too have considered using a parallel port Ethernet device and have been kicking myself for sometime about not buying one for $99 at a computer sale. I agree however that what happens inside might be hard to figure out, let alone program to. Maybe you could help us out and do a little e-mailing to see if there is some standard or protocol like structure these vendors might give us.

The number and type of machines available to play with is almost limitless these days. That reasoning is why I have broadened our scope. I get questions about used mainframes to embedded single chip systems. Our support is based on using

one person for a group of machines, and getting as many platform independent articles to back them up with. So far it seems to be working, especially since several other magazines have since taken up the banner of platform independence. I however got a big laugh the other day when CCInk said they support CP/M systems. Talk about stretching the truth a bit. All they do is advertise a Z180 system for sale. Not much support with that.

Might I ask you to contact Rick and get him to e-mail you his work, such that you might test and port it to another S-100 system? How about that challenge? Thanks. Bill Kibler.

Hi Bill -

Here's my check to renew my subscription for 6 more issues of *TCJ* starting with issue number 63. Overall I think you are doing a very good job with the magazine, but I would like to see Bill Tishy's "ZBest Software" articles to return in the magazine. Maybe you could talk him into coming back and doing a Z software update column once or twice a year.

I am also looking for a copy of a book published by Prentice-Hall in 1984 called "CP/M Techniques" by Ken Barbier. This was the second of three books he wrote on CP/M and I would like to get a copy for my collection. If anyone has a copy in good condition for sale or for trade I like to hear from them. They can write to me at the address above.

Thanks for your time and keep up the good work.

Sam Vincent, Box 82, Hyde Park, PA 15641.

Well here too is another long over due request. I hope you have found that book by now, Sam, since you wrote in October of last year. But anyone with a copy for sale or trade please drop him a line. As to digging out old authors, I try, but many are burnt out or doing other projects that leave them without free time. Basically I am trying to get some fresh authors who are re-exploring these old systems and software packages. Since many of our readers are exploring CP/M and classic systems for the first time, I think writers like our new Ron Mitchell might provide more help. Don't forget that Reader to Reader is always here (although usually not so slow at publishing letters) for you to ask questions.

Thanks Sam for supporting TCJ. Bill.

Dear Bill,

Your package of *The Computer Journal* and flyers arrived today. And while I am not interested in other than the T/S (timex Sinclair) computers I am still interested from the standpoint that computers are actually generic if you look at the makeup of the computers. There are not too many parts of a computer that is not used on other computers. You get down to the circuit boards, the ROM and a few things such as that which are not used in other computers.

No mention was made in the ZX81 article about the way to convert the ZX81 into a FORTH computer. That is done by removing the ROM chip and replacing it by way of a twister socket (or special socket to enable you to have both the ROM and EPROM that contains FORTH so that you can switch from ZX81 to FORTH - with the power off of course) so that an EPROM with FORTH language programmed in it can be plugged in. The version of FORTH is TREE FORTH. Before I moved here from Cedar Rapids Iowa a member of the group had used a modified ZX81 on a kite using FORTH to move a camera both vertically or left and right and trip the shutter to take panoramic shots. Last experiments were with a model airplane radio transmitter to feed data to the on board kite computer. He had three stepper motors to do the work.

About speed - heard one man give the answer about using a faster computer. I don't know what he had but a friend was trying to sell him his faster bigger capacity computer so he could make a step up. So the man used it a few days, very few days. He returned it and said that if he had to use that computer he would not get anything done creatively. He said that he had heard where he could get two or three computers like his slow one and had bought them. The reason that he did not want a fast machine was that it gave him no time to think. He wanted to give the computer a task to do and then sit and think while it did it. He said that ninety percent of his good ideas came while he was waiting for the computer to do a task.

I am neither a programmer nor a hardware expert but I dabble. And I am a young 68 years old. I do things and I go through the manuals and everytime I learn more.

I am using a Timex Sinclair 2068 which is very similar to the T/S 1000 or ZX81 but has color (sometimes) and is faster and has 64K of memory (48K of memory available to the programmer). But it never had a disk interface designed and sold by the maker. But it has several that were availalbe through venders, in all there are some 800 to 1500 total disk interfaces made for the T/S 2068 and there is one still being produced. I have three of the disk interfaces, and while they all have a capacity of either 360K or 720K and can use both the 5.25 and 3.5 inch disk drives the different disk interfaces are not capable of being compatible to each other. Because they are cheaper I have stayed with the 5.25 full height disk drives and my choice of drives is the Tandon TM 100-2A which are double sided 40 track drives (360K). Using one disk interface I can check the rpm of the drives and tweak them to be in specs for best operation on the interfaces I use.

The three disk interfaces that I have are the AERCO, Larken and the Oliger. The interfaces are all hardware and all except for the Larken don't need any disk to get going. The software is in an EPROM on the disk interface and is transparent till you use a disk command. The Larken needs a disk to load the FORMAT program. Why? I don't know unless the designer couldn't or didn't want to fit it into the 8K EPROM. And the disk commands are somewhat simi-

lar. To read a directory with the AERCO you would use the command CAT"", and on the Larken you would use RAND USR 100:CAT "", and on the Oliger you would use CAT/

But there is a software DOS that came from the UK known as Millenia K that uses the Oliger interface. That requires two disks, the first disk prepares the computer to accept the SPDOS that is on a different disk. And that DOS is stuck into the top 12K of the computer's memory leaving 36K of usable programming space for the user.

The AERCO has another DOS called RP/M which is a version of CP/M and I need to find a book on CP/M to find out how to use it. The Journals that you sent give me leads to find that information.

I don't remember which one wrote about the MSDOS calling a disk 360K but that is not actually the capacity of a disk. With my interfaces, they all FORMAT 5K to a track so forty tracks is 5 X 40 or 200 and two sides is 400K but I normally with the Larken and the Oliger FORMAT to the 42 tracks which is 415K plus 5K for track 0 and the directory. Of course there are disks that are not coated that far. Currently I am using used disks, that is they were programmed with commercial software and erased because of revisions (I think) and I get them in person for $15.00 a hundred. Last year at the Dayton Computerfest I bought some for a nickle each. Failure rate is less then the disks I used to buy from MEI Micro for $29.00 a hundred including postage.

It is now 06 15 1994. There was a necessary delay in this letter. I had to gather up what I had for the next issue of *ZXir Qlive Alive!* and send it to get out the next issue. It is quarterly (thank goodness) so that is not pressing me too hard but still it takes a little time to get it all ready. I try to mail on the 15th of the month preceding publication and we publish April, July, October and January. We started with an issue in April so I always think of that as a starting point. Now that is out of the way I am free till the next issue is due.

Usually the summer is completely devoid of computer action but today I had two calls about T/S computers: one from Oregon and one from Arizona. So there is action after all.

About the material that I am sending off to be put into the next issue: I use an Oliger disk interface for most of my T/S 2068 work. But the person on the other end does not have an Oliger disk interface but does have a Larken disk interface and I also have one. The Larken uses a dock board which other DOS do not so if I have the proper EPROM on the Larken dock board I can use the Oliger interface and it will produce a disk readable by a Larken disk interface. I also have a simpler word processor than the other party uses that I have set up so that it LOADs from the Oliger disk DOS and automatically SAVEs when I use the SAVE command to the Larken disk interface. Since I have designated the LOAD drive as drive 0 (or A) and the SAVE as drive 3 (or D) I don't have to manually change disks. I do however must have a list of file names. It is not automatic but it works. But once I have it on a Larken disk I can make a copy very easily and send a copy on and still have a copy here. The copy is made 5 tracks at a time including track 0 (directory) as it FORMATs the disk being copied to. Start it and go do something else for a few minutes. If the screen shows BAD DISK then you know that you have to try another disk or another pass to copy.

The Larken did have a RAMDISK available that was expandible to 256K and I have two that I use on different computers. That is nearest thing that the T/S 2068 has to a hard disk. 256K is not alot of capacity but with 48K of programmable RAM that is quite a bit since you never have a 48K program. You only put most important stuff there. And LOADing the programs is so fast and silent. For personal reasons I use 360K (or 40 track) disks in preference to the 720K or 80 track disks. Don't need that many programs on a disk to confuse me. I use 5.25 drives but do have one 3.5 720K drive. For a while I was trying to use a drive that I finally found out was a 3.5 single sided 180K drive and no way to get at the other side of the disk. I have converted some disks to flippies for the novelty since one uses only single sided drives in the 5.25 size.

Fred Stern's letter in the readers input section reminds me that there are several user's groups still going. And still one magazine for the T/S computers. The magazine is *UPDATE MAGAZINE* and is published the same months as *ZXir Qlive Alive!* and that means quarterly. It is $18 per year and the year starts with the October issue. The address is: *UPDATE MAGAZINE*; P.O. Box 1095; Peru IN 46970. Frank Davis and his wife Carol publish the magazine and it was taken over from Bill Jones in Florida that wanted to do something else in his last years. He is older than I am.

ZXir Qlive Alive! is available for $10. a year from ABED KAHALE: 335 W. Newport Road; Hoffman Estates Ill 60195-3106. TELE: (708) 885-4337.

Now that I have replied to you I will now write to the leads that *The Computer Journal* has opened up.

TIMEX/SINCLAIR STILL LIVES.

Sinclairly yours, Donald S. Lambert, Auburn, IN.

Well Donald, you certainly have fun with your many computers!. Do you know anything about the emulators for ZX machines. I tried to use the BIN file of the FORTH ROM and was unsuccessful. The piggy back chip holder sounds like there is more to using the FORTH ROM than just sticking it in the socket. Maybe you can direct one of your readers to do an article on using different ROMs in ZX's. What say you?

I feel since IBM set their size and mark on the disk formats currently used, we are stuck with them. Whether effiecnet or best size really matters little. They are IBM blessed and so be it. Which is another reason to use other systems, like you do.

Well enough for now, thanks again for the letters, and let us know how Dayton turns out. Bill Kibler.

Real Computing

By Rick Rodman

32-Bit Systems

All Readers

TCP/IP & 32 Bit News

Tiny-TCP

With Tiny-TCP working at 9600 baud on a PC-clone, I've started work on the first port to a CP/M system, which, due to a slight change in plans, will be the Xerox 820-II. This isn't working yet, but I can say a few things about this system.

The Xerox 820-II, which is the double-density version of the Xerox 820, has a monitor which performs most of the console and printer activities of the system. The keyboard uses a Z-80 PIO and is interrupt-driven, with a 16-byte FIFO. The printer and communications ports use a Z-80 SIO/0. The monitor's printer driver is not interrupt-driven. The 820-II is a modern design by Z-80 standards: using all Z-80 family chips, and using Mode 2 interrupts. Most CP/M systems, by contrast, use 8080-style interrupts, known in the Z-80 world as mode 0.

The advantage of mode 2 interrupts is that you can have up to 128 different interrupts. The vector supplied by the chip is an index into an array of word pointers in a buffer pointed to by the I register. On the Xerox 820-II, this register points to FF00 hex. Unfortunately, it isn't clear which vectors aren't used or can be used. The monitor uses some parts of the FF00 page for other things. Anyone having a commented source listing or disassembly of this monitor, please contact me! But don't get the wrong impression - this is actually a very well-documented system.

I still plan to do the Rainbow. Since it's a dual-processor system, the work will be a little more complicated. The Rainbow User's Group supplied two communications programs: Modem 9.12, and Kermit-86. Modem 9.12 does not use interrupts; Kermit-86 does. What's the significance of that? At 9600 baud, the significance is that Modem 9.12 is *worthless.*

And this is where I'm leading to: Almost none of the available CP/M communications software uses interrupts, and so the overlay drivers and status-poll techniques used in those programs will not work at higher speeds. If you want to run your network, or your modem, for that matter, at 9600 baud or higher, you're going to have to learn to use interrupts. It's not that hard, and there have been many articles already explaining the topic thoroughly, so get used to it.

I won't say that you should learn interrupts because that's what the Big Boys do, because it isn't: most mainframes and minis use *DMA* for their serial ports. You thought *interrupts* were complicated?

But now let's return to the 32-bit world... Linux news

Linux 1.0 has been released! This version consists mostly of bug fixes from the previous releases. Linus Torvalds says he has some future enhancements in mind, but his main purpose in stabilizing the 1.0 release was to get all of the Linux fans working from a common code base. Both the Slackware diskette releases and the Yggdrasil CD-ROM releases are available with the 1.0 Linux release. While Slackware appears to be the more popular release, presumably because it's on diskette, I have several CD-ROM drives and have ordered the Yggdrasil "Plug and Play Linux" release.

While Yggdrasil themselves list the P&P CD at $39.95, it is available discounted from many suppliers, including Just Computers!.

The Yggdrasil release includes Linux plus various Gnu tools, the Xfree86 X Window software, the Tab (or Tom's) Window Manager, TCP/IP, and all source for everything. Actually, if you bought the entire Slackware package including all of the source files, you would be buying a fairly large stack of floppies, and paying a considerably larger amount of money than the price of Yggdrasil. You would also be investing a lot more time installing the software, and need a larger hard drive. And there's a greater risk of defective media or accidental data damage. When you balance that against the cost of installing a CD-ROM drive - just $50 for a nice LMSI CM-205 from Corporate Systems Center! - you might consider that *not* getting a CD-ROM drive is false economy.

At any rate, in the 0.99.7A release I presently have running, I haven't been able to get the network support to function. The software is supposed to support either 3Com 3C503 or Western Digital Ethernet cards, and I have at least one of each - but there is a serious design problem with both of these cards: they must use interrupt 3. It seems that the PC world has worse interrupt congestion than the CP/M world, doesn't it? Interrupt 3 conflicts with COM2, the second serial device, and it's hard for me to live with only 2 serial ports as it is. The 1.0 release is supposed to support NE2000-clone boards, which I currently use at interrupt 5. The old 3C501 so beloved of our esteemed Editor allows the interrupt to be selected - I

suppose that the 3C503 is supposed to have been an improvement.

The upshot is that I won't be able to say much more about network support under Linux until I get the new release. You folks wondering if you should get into networking should jump in - it's really a lot of fun! Network boards are cheap at hamfests, and you could use them with free software (e.g. KA9Q TCP/IP).

Windows NT

Not all accept the inevitable ubiquity of Linux. A company in Washington State offers, at a substantially higher price and without any source code, an operating system called Windows NT. The first release of this operating system was version 3.1. The next version, about to be released, is called version 3.5. (The company seems to have a problem with version numbers. Their word processor went from version 1.1, to version 2.0, to version 6.0 in less than a year, with no numbers in between.)

Windows NT comes with TCP/IP. To load it, you have to first enable the protocol, then enable the FTP server, then correct the bindings. This is fairly easy to do, and the ftp software works well from an NT system prompt. There is no NFS client or server provided, but that doesn't bother me because I seldom use NFS.

The Unix world is in the midst of a gradual conversion from NFS to the Andrew file system (AFS) which features greatly improved speed, among other things. AFS was first described in 1989, but things move slowly in the Unix world. One thing you can be sure of is that AFS will be much more complicated than NFS.

OS/2

Recent talk about OS/2 impels me to put in a good word for it.

At the present time, OS/2 2.1 is outselling Windows NT by a factor of more than 50 to 1. This is a recognition by the market of the fact, not that OS/2 is more stable or bug-free or anything else, but that *you can run all of your Windows and DOS programs.* One problem with NT is the terrible support for DOS and Windows programs. OS/2 is actually better than DOS for running DOS programs.

"What?" you say. I'll give you an example. Remember the 43-line EGA mode in WordStar? I imagine everyone has tried it - and turned it off immediately. Echh. But I've started using it again - in a DOS window, on my OS/2 machine (which has an 8514-clone board), it doesn't "crunch up"! Instead, it uses 43 regular lines of regular characters in the regular font. Beautiful! Plus I can run other DOS windows at the same time and do compiles, links, etc.

The DOS boxes in OS/2 are completely virtualized. As I mentioned before, you can actually boot DOS in one. However, there's usually no need to do so. A better feature is the way you can control the video modes and emulation. By changing the settings for "Video Mode Restriction" to "CGA Only", you can create a DOS box which has 768K of low memory - perfect for doing compiles and links. And don't worry about the awful CGA fonts - because, in a DOS window, it uses the regular characters in the regular font, so it looks every bit as good. Since you have *multiple* virtualized DOS boxes, each one can be set a different way.

Now all of this makes OS/2 the best platform for running DOS and Windows programs, and the best platform for doing development for DOS and Windows. There really isn't any competition.

I'll admit that as a platform for developing programs for OS/2 *itself,* things are not as good. The debugger doesn't work, so it's difficult to catch segmentation problems - there's no Dr. Watson for OS/2. OS/2 is multithreaded, itself, but PM isn't - so you have to write multithreaded applications for even fairly simple jobs. If a PM message handler does a DosSleep or reads a named pipe or anything else - pow! Reset button time. (I've been told that NT has a multithreaded GUI.) But at least you *can* do things like DosSleep. In Windows, one of the hardest situations is when you have to wait for something that might take some time. There just isn't any way to do it but loop. You can't even *simulate* a semaphore, because you're on the same stack with the routine you're trying to interlock with.

PC-532 news

The latest news in the PC-532 world is NetBSD, a full Unix, which has been ported to the machine by some folks and is finally becoming stable. I don't have it running on mine yet, but hope to obtain it from someone who is fairly soon. NetBSD is an AT&T-free version of BSD Unix which differs, in some way, from the FreeBSD version, which, in turn, differs from the BSD386 version presently being offered by Berkeley. This is another variation on how the Unix world resembles 14th-century Europe with its jumble of principailities and duchies. Still, the primeval Ugaritic omniglot common to all Unices will suffice for most purposes. NetBSD will allow improved networking to the PC-532, including support of X clients, NFS, and other neat stuff, so it is an improvement over Minix.

The Obituaries

The venerable name of NCR, a company almost a hundred years old, has gone into the bit-bucket recently. Built into a giant in cash registers by Thomas J. Watson, who went on to found IBM, they also produced a variety of computers and other hardware before being absorbed by AT&T. Now AT&T has eradicated the last traces of that name.

More sadly missed by *TCJ* readers will be Commodore. Thousands of Commodore 64 machines, the first computer with 64K bytes of RAM for under $300, are still in use today. It's made of chips you see nowhere else, and has a really slow floppy drive, but you can do amazing things with it. Then there's the C-128, which not only runs C-64 software but CP/M Plus, too. After that, Commodore produced the Amiga, another machine with an unusual, but very powerful, design.

These are fun machines, and open machines, with well-publicized addresses, even schematics, published for all to see. (Try to get a schematic for a Macintosh.) Unlike Macs and PCs, too, the Amiga's high-resolution graphics are broadcast-video-compatible by design. This led to the creation of the Video Toaster, a board set which provided some incredible real-time manipulation of video images. The Amiga was really a better attempt at a home computer than its competitors, because with its digital audio and video it could be easily incorporated into a stereo/TV system. Commodore dared to be different, *really* different. It might be that some company will keep the Amiga and Toaster alive, but the company that dreamed the original dream is gone forever.

Next time

Writable CD-ROM is revolutionizing everything. There's no way you *won't* be affected by it. We'll go into what to look for - and what to watch out for. I also hope to be able to review FreeBSD, another Unix for PC-clones.

Where to call or write

Real Computing BBS or Fax: +1 703 330 9049
E-mail: rickr@aib.com
Mail: 8329 Ivy Glen Court, Manassas VA 22110

Yggdrasil Computing, Inc. +1 510 526 7531
P.O. Box 8418, Berkeley CA 94707-8418

Just Computers! (Linux CD-ROM) +1 707 769 1648
P.O. Box 751414, Petaluma CA 94975-1414

Slackware, Steven Balbach, 5437 Enbered Terr., Columbia MD 21045, +1 410 740 1157

Corporate Systems Center (CM-205 CD-ROM drive $49) +1 408 734 3475, 1294 Hammerwood Ave., Sunnyvale CA 94089

Regular Feature

ZCPR Support

Beginning Z-System

The Z-System Corner II

By Ron Mitchell

The Great White "ZED"

Canada is known for many things; water, snow, arctic air, the Toronto Blue Jays.

We also mis-pronounce the letter Z. Up here it's ZED. It probably has something to do with the fact that winter is 6 months long. It tends to freeze the brain and distort our speech.

Nonetheless, there's a healthy contingent of Canadian CP/M-ers and some of us are into Z-System, the subject of this article, and hopefully more to follow. Being a newcomer to ZED is quite an adventure. I've only just started. Sometimes I can learn more efficiently by trying to explain things to other people. Read on.

If you're already a whizbang genius in Z-System, you might find this series a little slow. Still I would like to hear from you. We need to interest new people in old things. The population of my CP/M group here in Canada's capital is down to 4 or 5 stalwarts. We need more. They are out there and they are listening, but somehow we're missing them. Since the responsibility for communication rests with the party having something to communicate, perhaps we need to try a different tack. I will take suggestions from anywhere and everywhere as to what that tack should be. If you're an old hand, I'll take your advice. If you're a novice I'll take your questions.

This is the beginning of what will hopefully be a series of articles on Z3PLUS and NZ-COM, the world of Z-System. It will be written from a newcomer's point of view because that's the only one I've got. We'll deal with the pluses, payoffs, the prices and pains that come when you decide to learn something new. As I experience them, you'll read about them. Perhaps I'll be able to carry on for a page or two.

I have here four books, which will need to be mastered if this is not to be a severely short series.

1) A loose leaf bound collection of computer printouts exploring ZCPR2. No dates, no names - except for the source code listing at the back of the book. "Z-80 Based Command Processor Replacement Version 2.0" Copyright 1982 1983 by Richard Conn.

2) "ZCPR3 The Manual" copyright 1985, New York Zoetrope, Inc ISBN 0-918432-59-6.

3) "NZ-COM The Automatic Z-System" Copyright 1988 by Jay Sage and Bridger (no relation) Mitchell.

4) "Z3PLUS The Z-System for CP/M Plus" Copyright 1988 by Jay Sage and Bridger Mitchell.

It's a Start

The last two are yours when you purchase Z3PLUS or NZ-COM from Jay Sage whose business address appears elsewhere in this issue of TCJ. More follows about that later on and also about the difference between the two.

First we need to exchange a few pieces of philosophy, and to fix firmly in mind some concepts. That will probably take the balance of this article and perhaps some of the next. It is important to understand why Z-System is worth our attention at all. It is also important to understand what an operating system, any operating system, does for a computer. First the philosophy.

What goes around comes around. In my ten years as an unpaid professional computer hobbyist of novice standing I have met many wonderful people. These people have been of immense help to me as I learn how to compute and why computers work at all. Some of these people border on gifted levels of intelligence, if not outright genius. They can quite literally make a small computer sing. I feel that I want to give something back to these people, without whose efforts my experience of hobbyist computing would have been far less satisfying than it has been.

I have met other people who regularly have more questions than answers. They come to user group meetings armed with long written accounts of their experiences. Their notes form an agenda for club meetings everywhere and they provide their contribution by being curious, attentive and persistent. These people give the willing guru something useful to do while at the same time serving as a necessary thorn in his side.

There is a third category. These are users who bought a home computer for a specific purpose such as word processing and who's ambitions might never drive them to produce a program or ask a question. If they show up at user group meetings at all, they will stare at the presenter of a topic glassy eyed for a full 40 minute session without saying a word. Chances are they joined your user group because it seemed like a good thing to do, but they don't believe they have very much to offer. Worse yet, some members of this category will turn out with the

single hope of getting something for nothing, something that they didn't have when they walked through the door. They don't think too much about contributing.

Each of us probably has the potential to fit into all three categories at one time or another. We give, we take, we teach and learn. What goes around comes around, and as it turns out the quality of our experience usually has something to do with what we contribute.

There is always something new to be learned when it comes to computers. The old machines can teach the hobbyist much about the new ones. At very modest cost, they can show us whether or not we're even interested. Despite the claims of some that eight-bit computing is dead, there is a very real role to be played by a machine whose every processor cycle can be readily understood by anyone who has just a little time and inclination to be curious and to think for himself. Contrary to the popular hype, the inner workings are not as complicated as they are made out to be. The math deals with quantities that are understandable and yet the principles apply whether you're dealing in K or in MEG. A microprocessor may be a Pentium or a Z-80, but it's job remains to fetch a computer instruction and execute it, nothing more, nothing less.

So if you've just picked up a venerable old beast from the local garage sale and you've blown away the dust and cobwebs, welcome. Your local computer supplier may not even know what CP/M is, and most likely will not be of much help. Some salesmen, recognizing that they're not about to make an instant 'G' or two won't even give you the time of day. In your local neighborhood, you are likely to be all by yourself. But there is help, and perhaps someday you'll be in a position to provide it to someone else. Perhaps not a dollar will change hands.

ZED?

What does all of this have to do with the Great White ZED?

Plenty. It's my aim to dispel some of the ignorance about operating systems in general and CP/M 2.2 in particular. That's just a side objective. The chief aim is to describe in detail as understandable as I can make it just how Z-System capitalizes on CP/M's strengths while providing for some pretty impressive improvements at the same time. I want to do that, and in doing so to leave you with more knowledge about the operating system that your dusty old Z-80 uses to do its thing. My reward will be hearing fewer stories like the one that follows.

Having just joined a rather large computer club here in town a year or so ago, I met a few of the members and introduced myself. There was the usual session of 'show and tell' as I described my own background as a computer hobbyist. In the course of unloading all of that I mentioned that I was interested in CP/M. The silence that followed almost flattened me. And then someone said, (roughly paraphrased)

"Well I would have been interested in CP/M too, except that I've never seen it do anything useful."

And that, believe it or not is a fairly common perception.

In this day and age of impressive visuals and Graphic User Interfaces, (GUI's) the poor old operating system doesn't get much attention any more. Getting down to the 'DOS prompt' as they call it is something seldom done much by the average Windows enthusiast. Straying that far into uncharted territory is seen as being too technical. Most users prefer to stay closely linked with the application that is doing the job they want done, and to leave the underpinnings to the experts.

There's nothing wrong with any of this given that the category of users last described a few paragraphs ago exists and it's members are probably doing us all a favor by staying out of the operating system. What is annoying however is the fictitious hype that subsequently develops when these people try to give advice to others. The statement about CP/M not doing anything useful is typical.

So let us quickly see what CP/M does, without belaboring the topic too heavily. Then let's see what it does not do. Later we will focus our attention firmly on 'ZED' in order to see how some very talented hobbyists have, over the past few years given us quite a kit of tools indeed.

To begin at the beginning....

Consider the situation that would prevail if there were no physical connection available between your brain and your left (or right) big toe. Let's assume for the sake of argument that your brain had been informed somehow about the existence of toes and their general purpose in kicking things around. Perhaps there was a ROM chip somewhere in your psyche that informed your brain about these things each morning when it powered up, providing just enough detail to get the brain curious. This ROM chip, if it was that smart, would no doubt provide other information too about eyes, fingers, arms and all sorts of other strange things required to make the brain's life more interesting. Without this sort of information, all your brain could do would be to sit around encased in your skull and think pure thoughts. There would be no possibility of getting into action of any sort.

Fortunately for most of us there is a connection between the brain and it's peripherals. If you kick the wall with your big toe you'll soon find that the connection is alive and well. Your brain is able to initiate the action of kicking by sending a rather complex set of messages to the leg and the toe doing the kicking. The complex set of messages received from the big toe following the action performed is known as pain.

Now it could well be that there is memory somewhere in the scull. Not just ROM, but also RAM. If the brain had had the presence of mind to search this memory for prior occurrences of pain related to kicking things, it might have dispensed with the whole idea. The bet is fairly certain that if the action/pain was not

recorded before it certainly will be now. And perhaps next time the brain will think twice.

The analogy could be extended much further, but the point is this. While a microprocessor can fetch and execute millions of instructions per second, there won't be much point in such furious activity unless something happens in the outside world. The operating system allows the microprocessor to connect to the outside world giving it access to what are commonly referred to as 'resources'.

memory (ROM and RAM)
peripherals (monitors, disk drives, modems, whatever you have hooked up).

The script might look something like this. There are a few actors in our story; the processor or 'BIG P', a keyboard whom we'll call 'K' for short, and a disk drive named 'Screwup'. PC stands for Program Counter which is actually within the processor. Finally, our operating system is called CP/M. The conversation goes something like this.

Big P: Who wrote this mess? PC where are we? I've been sitting here waiting for Screwup to load the boot block. You would think we were running the complete works of Bill Gates!

PC: Somewhere in the Console Command Processor, sir. CCP has control and is not reporting a specific Hex address at this time.

Big P: Number One will you get it together and tell me what we're doing. And what's a CCP?

CP/M: One moment sir, we're awaiting a character from the keyboard.

Big P: You're awaiting a what from where? Number One, what the hell is a keyboard?

CP/M: We have one out there sir, don't concern yourself with the port number or the details, I'll keep track of that. One moment, I'll see if it has a character ready.

Big P: Make it so Number One, haven't got all day.

CP/M: Sir..... keyboard says it has no character ready. The user hasn't typed a thing in several minutes. He may have fallen asleep.

Big P: Humans! Can't live with 'em, can't live without 'em. Ok, next instruction please. What! Who does he think I am...some kind of trapeze artist? Jump where?

PC: We're way up in high memory, sir. Number One is using the CCP to re-try the prompt. No response so far.

Big P: Well at least we're not waiting for Screwup to load another 128 bytes. When that starts we'll be here all day.

CP/M: Sir, K reports the user has typed a character. And another.... and another.....and here it comes..... Yes! A return.

Big P: Amazing, What's all this? What did you do Number One, prompt him or something? DIR what's DIR?

CP/M: Let me check my resident command table, sir. If it's there I can I can tell you how to execute it right away.

Big P: And what if it's not?

CP/M: Then we'll go to the directory that Screwup has provided from the disk and check to see if there's a program by that name.

Big P: Make it so Number One!

And on we go. This little scenario may not perhaps be completely accurate, but it does convey in general terms the relationship that exists between the processor and the operating system. To make the point once again in human terms, if you had no operating system, you might be able to think about moving your arm, but you certainly wouldn't be able to do it.

We'll come back to system architecture as the need arises. At this point we've said enough about it to allow you to realize that your whizbang Z80-A functions something like a jellyfish without legs. The real work of managing system resources is done by CP/M, and part of CP/M is the Console Command Processor. As the name would indicate, it is the CCP's job to process commands. You the user are going to sit there and issue commands, one after another, all night, sometimes several on one line. Those commands are going to result, hopefully, in something useful happening.

Now in our little scenario there was reference to a table of resident commands. Under CP/M 2.2 you can do 6 things without reference to any other application or utility or computer program of any sort. These 6 resident commands are located within and are known to the CP/M operating system, and with them you can:

- Obtain a directory of the currently logged storage media,
- Type out a text or ASCII file,
- Rename a file,
- Erase a file from the directory,
- Change user areas on your currently logged storage media,
- Save a portion of RAM to a file on the storage media.

The commands, listed in order are: DIR TYPE REN ERA USER SAVE.

And that's all. It's not much really, when you think about it in a certain sense, but you will be surprised as you use any operating system how may thousands of times you come back to these basic operations.

Starting New

If you're a newcomer to CP/M , at this point you have every right to feel taken and somewhat disappointed. You've no doubt learned about some of the other things that your computer is capable of, and perhaps you've even dabbled in your machine's version of BASIC to some extent, where you can at least add two plus two without even loading a program. Now suddenly you have been re-

duced to much less than that, and you can't even add two plus two any more. I don't know about you, but it took me a long time to understand that the price of shedding layers of software right down to the operating system was a loss of capability that others had given to me through their applications, including a BASIC interpreter. However, I found I able to run larger and more efficient applications specifically designed to do the job at hand without carrying program baggage not needed for that job. With CP/M the payoff was being able to use applications designed for other machines, and feeling at home with whatever CP/M machine came to hand. A bigger payoff was an improved understanding of how my computer functions when equipped with only a minimum amount of program code.

These are things worth knowing. Consider this. A 64K computer has 65,536 bytes of memory available. You will most definitely use a chunk of this memory for various housekeeping functions such as putting a picture on your monitor or opening a file on your disk drive. Your operating system, whatever brand it be, will take up around 8000 to 10000 bytes of memory, some less, some more. If you use BASIC as a 'platform' for all of your work, you're going to use between 24000 and 30000 additional bytes of memory that your BASIC interpreter will require. That means that at least half of your computer's memory is taken up with things over which you have no control. That leaves the other half for your program or your word processor file.

So...you say, that's life in the fast lane, so what! Well, not necessarily. For example, by running your application from BASIC, you are carrying with you a complete set of mathematical routines including floating point representation and all the math formulae that Pythagoras wished he had access to. You may not be able to access it with your word processing program. All of these routines are occupying memory space underneath your word processor that could be used to give you more space for your own file. You pay a price not only in space but in speed of execution. Depending on the application, this could all be unnecessary overhead.

What CP/M allows us in addition to the six resident commands noted previously is a standardized toolkit of elementary abilities without which not much of anything useful could be done at all. We will get to these tools in the next article. In the meantime, I want to finish off this one by setting the stage for Z-System. The capabilities provided by CP/M 2.2 and CP/M 3.0 are important, there's no dispute about that. But there are certain things that we would like the operating system to be capable of which were somehow left hanging when CP/M hit the streets. Let's deal with two of those, because as a new user their absence almost caused me to discard CP/M entirely until I found out what I was doing. A few fellow users have fallen by the wayside over these things.

One friend of mine who ventured into CP/M concluded flatly that most of the public domain programs he had tried on his computer did not work. What he meant to say was that they did not work 'right out of the box'. They had to be installed or amended to provide for proper operation on his particular terminal. His particular terminal was somewhat different from my particular terminal. We had not been to the same garage sale.

But, claimed my friend, CP/M is supposed to allow programs to run on a wide variety of computers.

And that is certainly true to be sure, but one thing hasn't been said. Under stock CP/M (2.2 or 3.0) the program you want to run must be 'overlaid' to recognize a specific set of what are known as 'escape codes'. We'll talk more about these codes later. For the moment, suffice it to say that without them, CP/M does not know how to clear your screen, position the cursor on your screen, insert or delete a line of text, insert or delete a character, turn on the highlighting or bright video that your terminal can do, and a number of other functions which are said to be 'terminal specific'. Taken together, these functions are referred to as the TERMINAL CAPABILITY or TCAP.

Z-System allows you to specify your TCAP once when you first install it. From then on that becomes part of your operating system and is conveyed to any application program that you happen to be running. If the application is Z-system compatible, it does not need to be installed, it simply says to the operating system, "Ok where am I and which terminal has he got hooked up this time?" The information is there.

Another particularly frustrating item of business that Z-System takes care of is the dreaded error message:

BDOS Error on A: R/O

If you have not yet encountered this error message on your stock CP/M system then it's probably safe to say that you haven't yet loaded CP/M or used it for very much. My manual explains it this way:"Drive has been assigned Read-Only status with a STAT command, or the media in the drive has been changed without being initialized with a CONTROL-C. CP/M terminates the current program as soon as you press any key."

Right! It does just that, leaving you to invent whatever swearwords your frame of mind suggests to you. They will be colorful. Experienced CP/M-ers have told me that this is a holdover from the days when it was considered important to protect certain classes of user from themselves. It was a safety mechanism to prevent the possibility of something being written to the wrong disk. Z-System eliminates the requirement and you are on your own to decide what disk you should and should not be writing to. Of course with that freedom comes the responsibility of keeping track. This is by and large not a big deal.

These are only two advantages of Z-System. Next time we'll talk about many more.

Ron Michell at CompuServe 70323,2267, or GEnie as R.Mitchell31.

Special Feature

Intermediate Users

CMOS and RC's

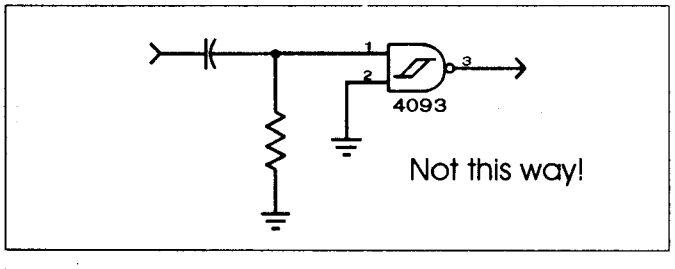
Little Circuits

by Dave Baldwin

Most logic circuits need an oscillator or some kind of timing circuit in order to operate. CMOS logic IC's can be used with resistors and capacitors in a number of ways to make oscillators and one-shots. But there is one circuit that causes a number of problems even though it's shown in a number of data books. I wonder if the people who draw some of the sample circuits intentionally put errors in or leave important things out so that us 'amateurs' have trouble making things that work.

THE WRONG WAY

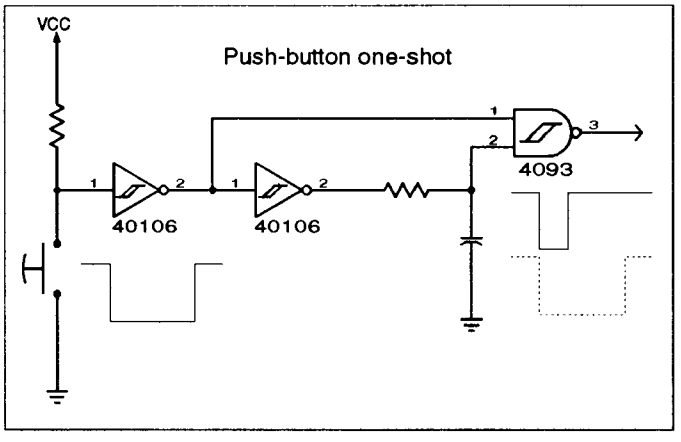
Data books for CMOS IC's show capacitive coupling as one way to generate a simple one-shot. In the circuit shown below, when you put a high on the left side of the cap, the cap charges up to 5 volts through the resistor. If you bring the input low, the cap tries to take the input of the gate to -5 volts. The input of the gate has diodes that clamp the input voltage when it goes below ground or above the supply. If that was all, that would be fine. But the current through the diodes usually goes into the substrate of the IC. Even with the other input shorted to ground as shown below, this can cause false outputs or even damage the IC. If you just have to use this circuit, add at least a 1K resistor in series with the input pin. This will limit the diode current to 5mA peak which is within the specs for the protection diodes.



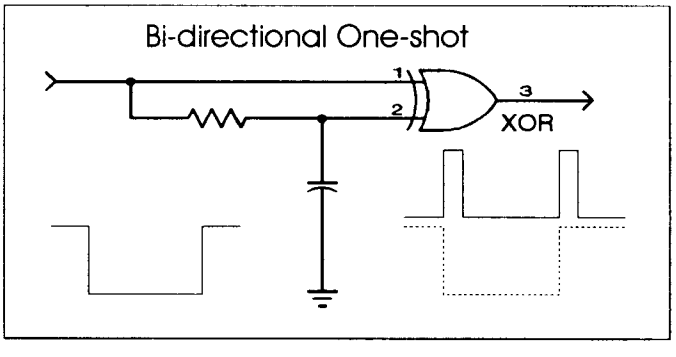
THE RIGHT WAY

The right way to use RC's to generate timing with CMOS gates is to put the resistor in series and the cap in parallel with the input. This way, the input voltage never exceeds the supply rails. The one-shot IC's available like the CD4538 and the 74HC4538 operate this way. You can look them up in the data books. I'll show you a couple of circuits that aren't in the books.

The push-button one-shot is circuit I've used a number of times. You can drive the input with another IC instead of a push-button to get a short pulse from a longer one. You can also add a cap to ground at the input to the first IC to help debounce the switch. Be sure to put a small resistor in series with the switch to limit the discharge current from the cap.



The next circuit is a bidirectional one-shot that gives you a pulse each time the input changes states. The output pulses at twice the input frequency. If you substitute an AND gate for the XOR, you get a delay/pulse filter circuit that lets positive pulses through after a delay and doesn't allow very short pulses through at all. Substitute an OR gate and you get the same thing for negative going pulses.

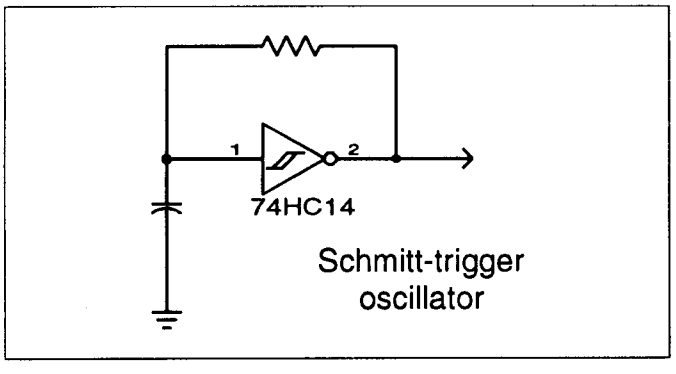


RESTRICTIONS

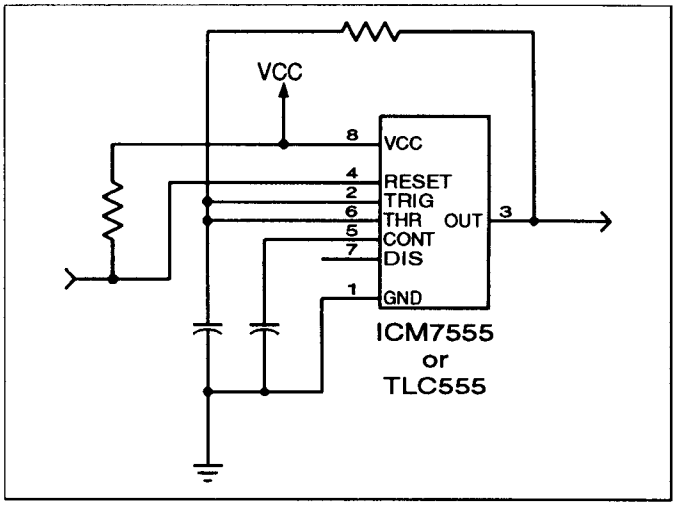
Both standard and high-speed CMOS have limits on how slow the input signal can change. Both technologies use input buffers on the inputs that are inverters consisting of a single N-channel and a single P-channel device. The input buffers draw very little current when they are all the way high or low. In between, however, both the N-channel and the P-channel input transistors are on and can draw enough current to burn up the IC if left that way. Standard CD-type CMOS draws less current than HC and input transitions can take as long as 1 millisecond at supply voltages up to 15 volts. HC high-speed CMOS draws a lot more current and transition times should be 500 nanoseconds or less with a 5 volt supply. The exception to these rules are gates with schmitt trigger inputs. These gates are designed to accept slow signals at the inputs. The gates shown in the push-button one-shot circuit have schmitt trigger inputs.

OSCILLATORS

The simplest CMOS oscillator circuit is a schmitt trigger gate and a capacitor and a resistor. I used to use the CD4093 to make gated oscillators, but the manufacturers have changed the specs in the last few years. They still have schmitt trigger inputs, but the upper and lower thresholds are so close together that they are unusable as oscillators. The 74HC14 and the 40106 still have reasonable (if somewhat sloppy) thresholds according to the databooks. The duty cycle is dependent on the input thresholds which are temperature and process dependent. You can use an open drain output like the 74HC05 or a transistor to gate the oscillator on and off by shorting out the cap.



A better oscillator uses the CMOS version of the 555 timer chip. The upper and lower thresholds are set by resistors inside the IC and are pretty stable. The CMOS version of the 555 will oscillate up to 1 MHz. It only requires a resistor and a cap to oscillate at about a 50% duty cycle. With more parts and different connections, it can be set to other duty cycles. The Reset input can be used to gate the oscillator on and off. And, of course, the 555 can always be used as a one-shot.



If you really need a 50% duty cycle, you should design the oscillator for twice the desired frequency and follow it with a flip-flop to divide the frequency by two. Because the flip-flop triggers from the same clock edge each time, the output will be a 'perfect' square wave at 50% duty cycle.

CONTACT

Well I signed up for Internet access only to find out that I didn't know what I was doing. Got to get books. When I figure out how to use it, I'll publish my address. Right now, I wouldn't know how to find a message if you sent it to me. You can always reach me through DIBs BBS at (916) 722-5799, 1200 to 14.4 kb, 24 hours. There is a TCJ conference where you can leave messages. There's a special logon that gets you directly to the TCJ conference and file area and skips the new user questionaire. Call (916) 722-5799 and use the following logon:

First name? <COMPUTER>
Last name? <JOURNAL>
Password? <SUBSCRIBER>

The TCJ file area has a ProComm script for logging on and a couple of files that Bill uploaded. All of the Little Circuits articles are available in the TCJ file area in PM4 format. If you also want access to other areas, log on with your own name.

DIBs Electronic Design

Dave Baldwin

6619 Westbrook Dr.
Citrus Heights, CA 95621

Voice/Fax (916) 722-3877
DIBs BBS (916) 722-5799

Regular Feature

68xx/68xxx Support

Languages and More

Small System Support

By Ronald W. Anderson

Thoughts on Languages

Continuing from last time, I thought I might divert for a few paragraphs and talk about the relative merits of a high level language vs. assembler. The example that I have handy is (sorry) from a PC, but the principles apply in any system and to any processor. I have a Citizen GSX-140 printer. When I get a new printer I generally write a few little command utilities to change the print mode. In the case of program listings I like the 12 CPI draft mode. I found that I could write a program in C to open PRN as an output file and write the command code to it, then close the printer. Having done so, let me report the program file sizes. The C source file is 503 characters, pretty short. The C .exe file is 9,339 bytes! The Assembler source file is 883 bytes. The .com file that it generates is 38 bytes.

That's right, the C compiler generates 9,339 bytes of code and the assembler generates 38. Assembler is nice for little jobs like this. I'll present the C code here:

```
// set citizen to draft 12 pitch hi density
#include <stdio.h>
#define ESC 0x1b

FILE *printer;
void main()
{
        printer = fopen("lpt1","w");
// turn off condensed mode
        putc(0x12,printer);
// set 12 pitch
        putc(ESC,printer);
        putc(0x4d,printer);
// draft mode
        putc(ESC,printer);
        putc(0x78,printer);
        putc(0,printer);        // this is an ASCII code 0 or a null character
// hi density
        putc(ESC,printer);
        putc(0x7e,printer);
        putc(0x42,printer);
        putc(1,printer);
        fclose(printer);
        exit(0);
}
```

The following is a slight modification of a demonstration program that comes with the A86 assembler package, see below. Here is the assembler code:

```
; Following is a minimal program for sending a sequence of bytes to the
; PRN printer-device. The codes that are sent appear in the DB statement
; following the label CODES.

PRINT_CODES:
        MOV     DX,PRN_NAME     ; point to the printer's device name
        MOV     AX,03D01        ; MSDOS codes for "open for writing"
        INT 33                  ; call MSDOS to open the printer device
        XCHG    BX,AX           ; swap the printer's open-file handle into BX
        MOV     DX,CODES        ; point to the codes we are outputting
        MOV     CX,LENGTH       ; load the number of code bytes
        MOV     AH,040          ; MSDOS code for "write"
        INT 33                  ; write the codes to the printer
        MOV     AX,04C00  ; MSDOS codes for "successful process termination"
        INT 33                  ; go back to the operating system

PRN_NAME:  DB   'PRN',0
CODES:  DB      012,01b,04d,01b,078,0,01b,07e,042,001
LENGTH  EQU     $-CODES
```

Having done these, I thought I had an idea to make the C program smaller. Basically, I thought, I would define a string of characters like the DB above in the assembler program and then have C print it to the printer. I hit a snag. One of the characters is a null, an ascii 0. C detects the end of a string by looking for a zero, so if I were to use the C library function to output a string to a file fputs(), it would stop at the null. A little thinking got me the following that does work:

```
// set citizen to draft 12 pitch hi density
#include <stdio.h>
#define ESC 0x1b
FILE *printer;
char codes[] = { 0x12, ESC, 0x4d, ESC, 0x78, 0, ESC, 0x7e, 0x42,0x01 };
void main()
{
        int k;
        printer = fopen("lpt1","w");
        for (k=0;k<10;k++) fputc(codes[k],printer);
        fclose(printer);
        exit(0);
}
```

That is, I could force the output of the code 0 and on to the end of the string by doing my own "for loop" to output the ten characters in the array. It worked fine. Original program 9,339 bytes of .exe code. Short version 9,005 bytes. Though the source is much more compact, the effort saved only 3% in the .exe code.

The assembler code obviously requires you to know a lot more about the hardware and the DOS service functions than the C program does. This code was adapted from a demo program that came with a nice shareware package called A86. It is a shareware '86 style assembler that doesn't require all the formalities of Microsoft's MASM. At least for a COM file it is pretty straightforward. At any rate, you get the idea that the assembler version might require more understanding, but the resulting code is more "pleasing" in it's size. Perhaps I retain too much of the old "save memory" attitude from the days when 32K was a lot of memory, but somehow 38 bytes seem a lot better than 9,339 bytes (or 9,005) to me.

A86 is available if anyone is interested see note at end of this article. The manual (on disk for you to print) is large and

detailed. Registration fee is $50 and if you intend to do any assembler programming on your XT on upward, this might be easier to use than MASM, (and I'd rather support a good shareware producer than MicroSoft). This package generates pre-286 code and so is runnable on ANY PC including the now "antique" XT.

Well, perhaps enough of that. There is one point that must be made. The high level languages carry a lot of overhead along with any of the functions that you use. In this case the C program is about 250 times larger than the assembler one. Don't expect this ratio to hold, however. High level languages do much better as the program size grows. I've seen ratios in other systems (the 68000 in particular) of 2 or 3 to 1 (hll to assembler). Lastly, if you want to avoid learning how to program in assembler, you can do it in C and not worry about the code size. Learn C and you can program just about anybody's computer.

Forth

I hope I don't offend anyone here. These next few paragraphs are STRICTLY my own opinion, and I am speaking for myself, not anyone else. I spent a lot of time with Forth quite a few years ago in the 6809 context. I installed a Fig. (Forth Interest Group) Forth and made an editor work with much frustration. At the time I thought it was much harder than coding in assembler. The 6809 assembler has just over 55 very regular instructions. There are some instructions that don't work in all the addressing modes and you have to learn a few exceptions. Forth (the version I was using) had about 250 words (instructions) to learn. I hasten to add that this was quite a few years ago. I note that "standard" Forth has grown to 1000 words and then to 2500 words.

At the time there was not much in the way of floating point arithmetic available, and that, perhaps was the biggest problem I had with it. In my applications there are many calculations that require floating point and scientific functions. In assembler I could write my own floating point arithmetic and scientific functions (and I did, though the sci functions were crude by my current standards).

In my tests I found NONE of the claims of Forth to be true. It was supposed to be fast. My favorite Pascal like language PL/9 could beat it easily in, say, a program to find the prime numbers from 1 to 10000, using the same algorithm. Assembler could beat it by a wide margin, and the total code was far larger than the equivalent assembler version.

I repeat that this is strictly my own opinion. I'm not suggesting that someone else ought not to like Forth, just saying that I gave it a pretty good try, and yes, I do have Leo Brodie's book. Some of the other reasons I didn't like Forth at the time were that it made me use it's operating system, while I had a better one available to me. It used a special format for disk files so a disk with Forth programs had to be separate and distinct from all the other disks I had. Some of these objections were overcome later when I bought a version of Forth for the CoCo. It wrote standard FLEX operating system files, and yet used standard Forth screens. By the way, I didn't like UCSD Pascal for exactly the same reason. It forced me to use a PRIMITIVE operating system compared to the better one I used every day for everything else. I had to run a utility to "pack" the disk after deleting files. FLEX took care of such housekeeping chores in a much better way, and automatically.

In spite of my negative feelings about Forth, I have to say that I like Reverse Polish Notation (RPN). My first good calculator was an HP, and my current one is their RPN version. I have a hard time using anything else. I don't like RP as applied to simple decision statements as in "if else then". Sorry but I don't think backwards in words. I'd rather program the way I write.

Just in case someone out there doesn't know what Reverse Polish Notation is, let me illustrate. The key sequence to add two and two on a "normal calculator is: **2 + 2 =** . On a Reverse Polish calculator it is: **2 (enter) 2 +** . On the surface that seems more complicated, but let's look at something more complex: such as **(2+5) * (12-3)**. On a normal calculator you press those keys in that sequence and add **=**.

A RP calculator has no parentheses, but uses a stack instead. You key: **2 (enter) 5 + 12 (enter) 3 - *.** The **2 enter 5 +** displays 7 on the display. The **12 enter** pushes the 12 onto the stack. **3-** subtracts 3 from 12 and displays 9. ***** multiplies the 9 by the 7 already on the stack and displays 63. The rules are always the same. Calculate the parenthesized expressions first, starting at the deepest nesting level and leave the results on the stack. With a little practice (though it sounds hard) you will never want any other calculator.

Quite a few years ago, Consumer's Report did an article on calculators. They severely downgraded the HP because of the "peculiar notation". A few months later they retracted their criticism. They said that all the people in the office wanted the HP because they could get the right answer sooner with it, and they found it simpler to use.

If you ever code an expression evaluator for a compiler, this technique works very well. Start at the deepest level and work out.

Other High Level Languages

My language of choice is Pascal. When I try to write "pseudo code" to document an algorithm it always comes out Pascal. Tilmann Reh pointed out that Pascal is messy when you start using pointers. In my dumb opinion, pointers in Pascal are good for only one thing, linked lists. Pointers in C are vastly more versatile (read that useful). C is a pain to learn. You can ignore some of the more esoteric things like the question mark operator which serves only to make the source code very cryptic in my opinion, but pointers are very handy devices giving the language the advantage of indirect addressing. C is sort of a "universal assembler" in one sense, and a full fledged high

level language in another. Tilmann made one comment in his letter in issue 64, that while it was true that I could translate a Pascal program to C rather easily, the reverse is not true. I agree 100%. You could consider Pascal a subset of C in one sense. It is a much smaller language, one designed to teach students how to write programs.

Speaking of Pascal, I heard a cute story about Niklaus Wirth, the author of Pascal. He was speaking at a conference once and someone asked him the correct pronunciation of his name. (He is Swiss). "Well, you can call me 'Veert', or you can call me by value, 'Worth'".

Speed and Power

A number of years ago I wrote a book called "From BASIC to Pascal". As the title implies, it was intended to be a simple book describing how to program in Pascal for those who had learned to program in BASIC. One of the subjects of the book (one chapter) was sorting. I wrote some programs to sort random integers to compare various sorting algorithms. At the time I was using Lucidata Pascal on a 6809 running at 2 MHz. I sorted 1000 random integer values and timed the sort. The two fast sorts for a large number of items were the Shell Metzner sort and the Quicksort. The simplest approach, the bubble sort took 22 seconds short of 1 hour to sort 1000 items. The Shellsort took 100 seconds, and the quicksort took 47.

Lucidata Pascal is a Pcode compiler that does a partial compile and then runs in interpreter mode. It is sort of between BASIC and a native code Pascal. I'll give it a factor of ten in speed for that, so let's say that the shell and quick sorts took 10 and 4.7 seconds respectively (or would have with a native code compiler).

Today I decided to code these sorts in C and see how fast they run on a 386-40 PC. (About now I can hear you saying "Pascal is his favorite language but he writes programs in C"). Remember the Listerine commerical about the mouthwash you hate twice a day? C is like that to me. I grumble about it but it is eminently useful and quick and it doesn't get in my way when I want to do something.

Anyway, when I coded the sort programs in C and ran them on my 386-40 system they ran far too fast to time. I extended them to sort 25,000 items. By my theory, that ought to take about 125 times as long to sort, so I would expect 1250 seconds for the shell sort and about 550 for the quicksort. After a little tweaking I ran the two with the computer doing the timing via a nice utility in the C library that reads the system time.

The shellsort ran 25000 items in 2.86 seconds, and the quicksort ran them in 0.71. These times are 500 to 700 times faster than the old 2 MHz 6809, which couldn't even handle such a large number of items without running out of memory. Of course these times don't include reading the input data file of random numbers or writing the output file of sorted ones. (It is always a good idea to check to see if a sort program is actually sorting the items). Obviously the quicksort does better compared to the shellsort as the number of items gets larger. It is just about four times as fast for the list of 25,000 items. If there is any interest in the subject of sorting (if I get a few letters of encouragement) I'll devote a column to it one of these times.

680X Assembler

I promised some continued words for the beginners on Assembler and the differences between the 6800 and the 6809 in that respect. Last time I couldn't resist a quick example of how the 6809 was easier to program than the 6800. I ought to mention that the 6809 assembler by Motorola and that sold by SWTPc for their computers and the FLEX operating system all were capable of assembling 6800 assembler source code. You could simply reassemble your old code and the result would run on the 6809. Many of the software suppliers did just that when the 6809 boards became available. It wasn't until some time later that 6809 specific code was written and the advantage of the 6809 became more apparent.

Of course, I knew almost nothing about programming when I started writing 6800 code, but when the 6809 came along, I had programmed in BASIC, and Pascal, and so had a better idea of what a structured program was all about. Naturally I wrote at least somewhat better code for the 6809. I had intended at this point to get into the instruction sets and a comparison of them, but the past couple of months have been hectic at work and I have gotten little done in the area of column writing. By now, Bill should be into working with the last thing I sent him, and I have to get this off. I've filled it with random thoughts about programming and programming languages, but haven't had the several hours it would take to wade through a reasonably lucid explanation of the instruction sets of the 6809 and the 6800. I'm truly sorry, but better I get something going than skip an issue. Time permitting I WILL get the assembler code discussion going next time.

I do have time for a few more generalizations since I can simply sit here and put thoughts on my screen without having to do a lot of digging for information. Realize that it has been years since I've done any "serious" coding in assembler for the 6809 or 6800, though I have now and then done a simple utility like the one above in 86 assembler for similar purposes.

Maybe I could "say a few paragraphs" about assembler programming in general. First, it is a much more detailed pursuit than programming in any higher level language. Secondly it is very processor specific, though once you have learned how to program for one processor in assembler, the next one is not as hard. Third, because an assembler program generally has more machine instructions by a large margin, it is more difficult to debug. It takes longer to write and longer to make it work once done. You will get tired of writing or importing large chunks of code for such simple things as outputting text to the screen, particularly if you want to do any sort of formatting of the output on the screen. You will have subroutines to put the

cursor at a spot on the screen, subroutines for clearing the screen, etc.

If you want to do much more than quick integer math, you will have to write or purchase (or find on a bulletin board or in someone's old code collection) a floating point math package. I've done a number of these over the years, with trig, log, exp, and square root functions included. If enough of you would be interested in going that deeply into assembler programming, I'd be happy to pull the code out and start publishing a bit of it each time along with explanations of how it works.

Not only is assembler code processor specific, it is also very operating system specific. You note in the above 86 example that there are system calls in the form of the INT 33 instruction, each call preceded by stuffing the appropriate register with the appropriate code. Such programming requires an extensive knowledge of the operating system. Fortunately, you who are interested in 6809 systems will find FLEX was much simpler. System calls were via jumps to subroutines rather than software interrupts. Each system call in FLEX is described concisely in the programmer's manual which was shipped with FLEX at one time, and available separately at another. Contrary to this, I spent at least $300 on books supposedly on the subject of programming a PC in assembler before I found one that really had some information in it. Most of them would say that you could do something and then not tell you how!

Just in case someone is interested, the only GOOD book (that I've found) on PC System Programming goes by that title ("Pc System Programming"). I would hope there is a later printing available by now. It is published by Abacus, 5370 52nd St. S.E., Grand Rapids, MI 49512. The price a couple of years ago when I bought it was a whopping $59.95 complete with a megabyte of programs on two disks. It is worth half a dozen of any of the other books I've ever found on the subject. My copy has a 1990 date, so it is just into the 286 era. It devotes most of it's pages to the XT, and so is very appropriate to mention here.

Printer

I have a last minute addition, though it is not about programming or old computers. I've been using a Citizen GSX-140 24 pin printer for some time. I found a package called Emulaser that works with WordStar. WordStar can output Post Script printing instructions for a Laser printer. Emulaser can interpret Post Script instructions to set up a bit graphics image of the page to output to a dot matrix printer. My Citizen takes several minutes to print a page in that mode. I've been looking at the ads for the Canon BJ-200e "bubble jet" printer. Last week I got one at work to use in my office. It is FAR better than I would have anticipated in my wildest dreams. In it's native mode it prints nice black text in a limited number of fonts, and it has a draft mode in which it prints in dark grey. According to the manual, the draft mode uses half as much ink as the quality mode. When used with Emulaser, it is fantastic. I have a large selection of scalable fonts that I can use, and to my eye, it is so close to the quality of a laser printer that I am overjoyed. I can't tell the difference. I've looked at it's output and that of a good laser printer with a magnifier, and the outlines of characters are equally smooth (or equally ragged depending on how you look at it). The printer is essentially silent. It does a page in graphics mode via Emulaser just as fast as it does a page in native mode. With my 386-40, Emulaser takes about 15 seconds to translate a page of Post Script instruction file into the bit graphics. The printer than takes about 45 seconds to print a page full of text, proportionally less for pages with white space on them (like program listings).

The Canon was purchased locally at an electronics supermarket called "Best Buy" new to this area about a year ago. The price was $271. Computer Shopper had several sources for these, all at about $285 to $295. This is at the top of my want list for home since I stay up later than anyone else in the house, and with one of these I could actually print letters at 2 AM without bothering anyone. I've printed a manual for my PAT editor (last week during lunch hour one day) and it really looks like it is offset printed. I'll take it to the local copy place and get some copies to put in notebooks for reference.

You could use one of these on an old 6809 system running FLEX in the Epson emulation mode in which it has about six fonts in numerous sizes. My only disappointment was that the manual was missing the list of control codes that could be used to set the printer's modes, type fonts, etc. I found a note in the manual that I could call a help number and they would fax the instruction codes to me at no charge. I haven't done it yet, but I intend to do it as soon as I get tired of playing with the new printer. I've been printing memos and class notes for a C programming class with it.

I guess the reason for the lack is that most people now simply buy a word processor package and most all of them have their own drivers for nearly any printer, so we computer users don't have to write our own anymore. WordStar has complete drivers for the BJ in native mode and in Epson LQ emulation mode. In native mode there are only a couple of fonts, but there are widths all the way from five characters per inch to 18, and there is a double height mode and a combination double height and double width mode. In LQ emulation mode there are half a dozen fonts with Italic and Bold as well.

A86 Assembler

Available from: Eric Isaacson, 416 E. University Ave.
Bloomington, IN 47401-4739.
Phone (812) 339-1811, for Visa and MasterCard orders.

Eric sells the registered version for $50. If you want to try the package out first, it can be found in some of the shareware distributors' catalogs. Try TSL, (The Software Labs) or Public Brand Software. Both have nearly everything in shareware. You might find this on some of the shareware CD Rom packages that are available also. There is a companion D86 debugger package that I haven't yet tried (and have not registered).

Regular Feature

Kaypro Support

3 Support Articles!

Mr. Kaypro

By Charles B. Stafford

In this edition of Mr. Kaypro, I have added two other support articles for the Kaypro. JW Weaver explains his hardware problems and their solutions. JW shows why using older systems makes for inexpensive repairs. The last article is one on creating keyboard macros for the unused keys on the Kaypro keyboard. Jack Wyatt a local Kaypro user sent me this addition some time back, and it has trickled to the surface for your enjoyment. But first those ever great words of wisdom from Charles himself. BDK.

Infromation for Everyone

Wherein we take a break from the construction project and attempt to catch up on correspondence.

Keep those cards and letters coming, folks, and I'll get to them eventually. In fact when it appears from the content that the answer is urgent, I answer by telephone when I can track you down. If your physical location is obscure, however, I need some clues to give the "Directory Assistance" operator. If you were to include your telephone number, it would be greatly appreciated.

A reader in Northern California (a state unto itself, or is it a state of mind) writes that he has just acquired a Kaypro 2 complete with software and documentation for a mere $50.00. (That, by the way, is a pretty typical price) The machine appears new, I'm told, despite the fact that the original owner purchased it in 1982, and had been using it diligently and regularly ever since. He was surprised to find only 191k drives, so he opened the hood and discovered no serial number or assembly number, but the ROMs were (are) 81-146a and 81-149. The previous owner indicated that the K-2 had quit, and been taken to a local repair shop which had replaced the mother-board, and then discovered that the real problem was the power supply, which was then replaced. His questions are:

1. Is this an "ok" board and is it upgradeable ?
2. What products do I need to get (ROM/boards/etc) ?
3. The CP/M system disk was original, but no version number was shown. How do I find out what version I have and does it really matter ?
4. How do I tell if this is the 65 watt or 85 watt power supply ?
5. Since I have several old hard drives kicking around how do I put them to use, if possible ?

First, machine identification.

This appears to be an example of the second run of single sided KayPros. The first was designated KayPro II and had an 81-110 monitor ROM at u-47 and the original character ROM did not have true "descenders" on the g, p, and q. That was changed during the first production run. The second production run was designated KayPro 2 and had an updated monitor Rom, 81-149. The designations were only silk screened on the side of the "hood", however, and toward the end of the second run, the K-2 hoods ran out before the single-sided drives did, as did the mother boards, so the folks on the assembly line used what they had. As a result there are some "late-model" K-IIs with K-4 mother boards. This does not appear to be one of those however.

Q 1. It depends on your definition of "OK". Yes , it is upgradeable, first to a K-4, and then perhaps as far or farther than my original K-4, which is now a K-28 (20 megabyte hard drive, and 800k floppies, hence K-28). The conversion to K-4 has been covered in a previous issue, and see Q 5 for hard drive info.

Q 2. You don't need any ROMs or boards to be productive, BUT if you're going to do the K-4 modification you'll need a couple of double sided, double density floppy drives (standard IBM 360k issue) and a new monitor ROM, either an 81-292, or a TurboRom, or one of the MicroCornupia K-4 or K-8 ROMs. What this upgrade gets you is doubling of your diskette capacity, 391K vs 191k. If you choose either the TurboRom or the MicroCornucopia Rom you'll also get an improved and easier-to-use Operating System.

Q 3. If you just HAVE TO KNOW, there is a public domain program which will write the contents of your system tracks to a file, which you can then inspect using DDT, or you can use SYSGEN.COM to pull the system into memory at 100h, followed by "save 36 cpm.sys" and then use DDT, or you can take my word for it that everything prior to the 1983 K-10 used CP/M 2.2D in its stock form. The bottom line is that it doesn't really matter anyway, unless you "lunch" your only copy of the operating system.

Continued after Centerfold, page 29.

TCJ Center Fold

Special Feature

All Users

Pertec/Mits 88 - 4 PIO

For this issue of TCJ, it seemed appropriate to provide a schematic of a parallel port circuit. Frank Sergeant's XT project will use a centronics port and we will see a typical PC/XT early design presented there (issue 69). Prior to the early PC/XT machines there were many S-100 and SS-50 designs. I chose the Pertec/Mits version for two reasons. The design shown here is very simple. It contains the necessary address decoding, in and out buffers, and minimum control lines.

Another feature of the design is the use of non-Intel devices, mainly the Motorola 6820. Many SS-50 users will recognize this device from their parallel SS-30 cards. The 6820 is not very much different than the Intel 8255 or Zilog Z80-PIO in functionality and concept. The 6820 has 20 bits of possible data lines. The Intel 8255 provides 24 bits of data, although it has several modes which limit the use of all 24 bits and can result in the same functionality of the 6820 or the Z80-PIO (2 data ports of 8 bits each with two control lines per data port). In pricing the Z80-PIO is less than $2, 6820 and 8255 less than $3.

The simplicity of the address decoding presented is one reason many of TCJ writers do not use PALs. We see that addresses A0 and A1 determine which of the four internal registers are selected (RS0 and RS1 provide 4 by 1 decoding). Address lines A2 and A3, through a combination of using the CS (chip select) lines determine which of the four devices is selected for use. This scheme of using both the normal and negative version of the signals is also used for decoding the higher address bits.

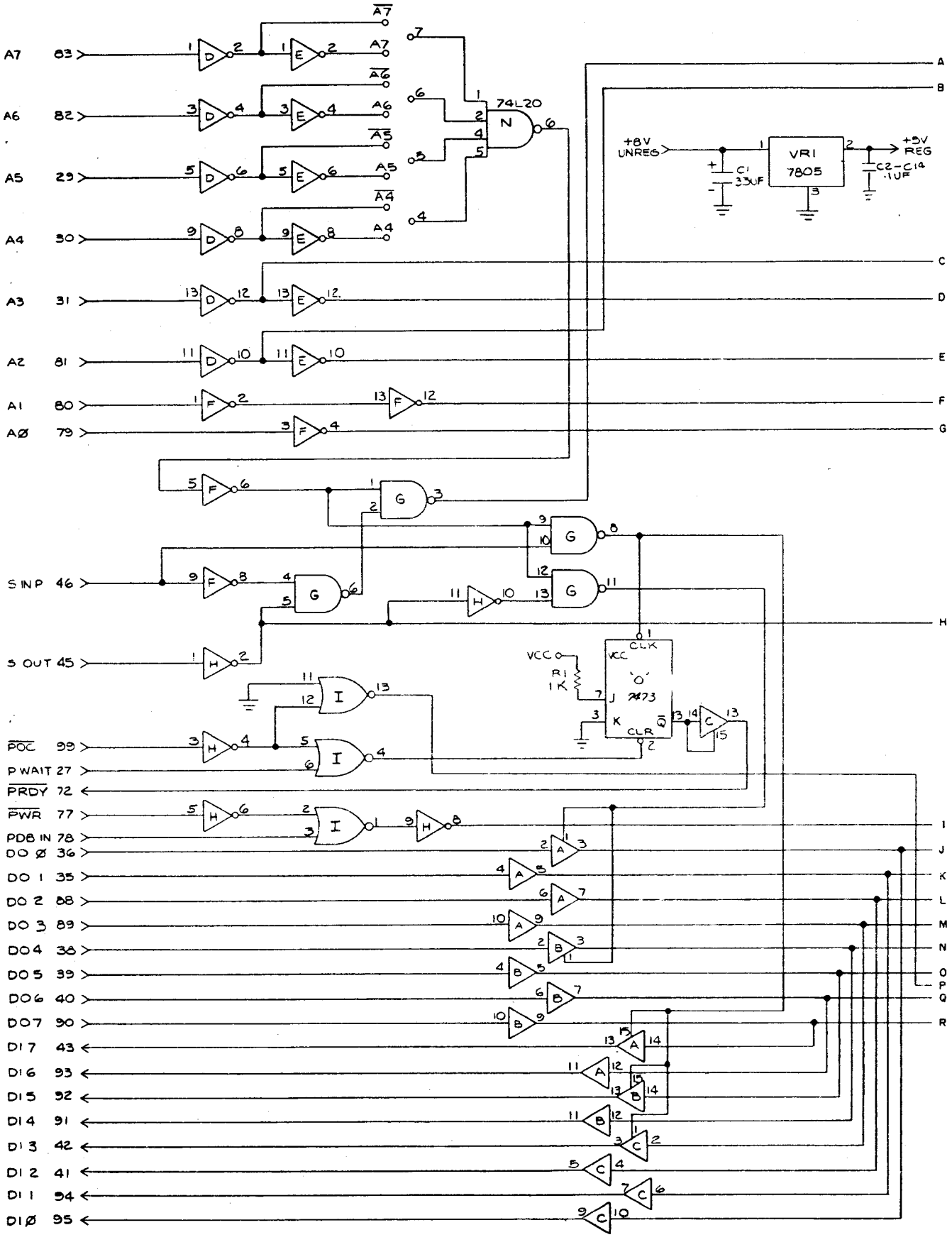
In the 8080 or S-100 world, all I/O (Input and Output) uses an 8 bit addressing range. Thus only the lower 8 bits of address are decoded for chip selection. For systems with a full 16 bit address decoding using another 74L20 and a few AND gates could be added to allow for selection only if all the upper 8 bits were all zeros (low/0 V). However you need to be careful, as some CPU chips "echo" the lower 8 bits on the upper 8 bits when doing I/O requests!

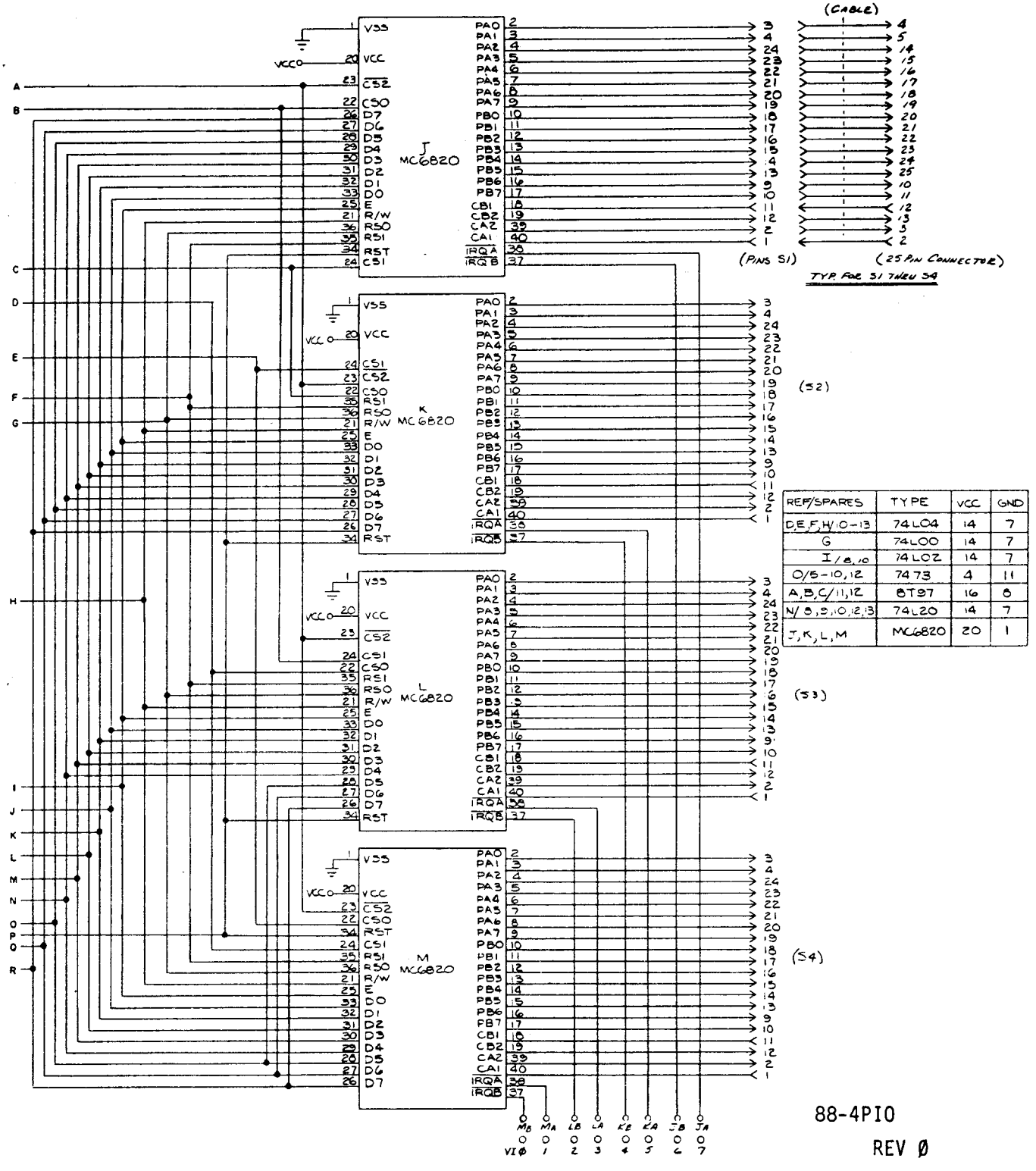
Not shown in the schematic are any form of buffering or voltage conversion of the IN/OUT signals from the 6820s. The schematic assumes that all uses of the data are to or from 5 volt TTL type logic devices (such as other 7400 or 6800 devices). Should you desire to drive stepper motors or other higher current requirements, buffers and drivers will be needed. Often used these days are opto isolated signals which prevent shorts or improper voltages from reaching the main devices (opto's cost $.47 each, while PIO's are $2 each). Typical current limits for outputs from 6820 is 5 to 10 milliamps, about enough to drives one or two TTL devices or a single transistor driver device (such as 2N2222).

To build a parallel port interface for almost any other bus system would vary little from this design. Whether the CPU be an 8031, 80386, 68HC11, or 68040, the design will need the same items. An address decoder will be needed with enough lines decoded to guarantee that only the parallel device desired is selected. Input and Output buffer and gates are needed to make sure that the devices are sending or receiving data on the CPU bus correctly (put data on the bus when reading from the device). The gates and buffers must be "gated" at the correct time for the type of CPU and thus some combination of control signals associated with the address selection will be needed. Shown in our schematic these are the SINP/SOUT (whether the data request is a IN from Port or and Out to Port). /POC, /PRDY, /PWR, PDBIN, and PWAIT (direct "handshake" signals from the CPU).

For further understanding of this circuit, "The S-100 Bus Handbook" by Dave Bursky, from Hayden Book Company in 1980, contains detailed discussion on pages 64 through 70. For discussions on interfacing to S-100 (complete with copy of S-100 standard) see "Interfacing to S-100/IEEE696 Microcomputers", by Sol Libes and Mark Garetz, last publisher was Dr. Dobbs. An excellent book for understanding the PC/XT interfacing scheme is the reference sited by Brad Rodriguez, "Interfacing to the IBM PC" by Lewis C. Eggebrecht. This book contains sections that add to the material presented by Frank Sergeant, including using the parallel port to drive a stepper motor (chapters 18 and 19).

For PC compatible systems, caution is needed as not all parallel ports are programmable or usable. Many new single board compatibles have the parallel port generated from within a single ASIC chip. These devices support the Centronic standard and may be difficult to impossible to use other than as the standard allowed. It is common practice not to support all the centronic signals in many of these compatible systems.


A7 83
A6 82
A5 29
A4 30
A3 31
A2 81
A1 80
A0 79
SINP 46
S OUT 45
POC 99
P WAIT 27
PRDY 72
PWR 77
PDB IN 78
DO Ø 36
DO 1 35
DO 2 88
DO 3 89
DO 4 38
DO 5 39
DO 6 40
DO 7 90
DI 7 43
DI 6 93
DI 5 92
DI 4 91
DI 3 42
DI 2 41
DI 1 94
DI Ø 95
74L20
+8V UNREG
VR1 7805
+5V REG
C1 33UF
C2-C14 .1UF
R1 1K
VCC
CLK
CLR
A
B
C
D
E
F
G
H
I
J
K
L
M
N
O
P
Q
R



Pertec/MITS 88-4PIO, quad parallel 8-bit port card

PIA PERIPHERAL INTERFACE LINES

The PIA provides two 8-bit bi-directional data buses and four interrupt/control lines for interfacing to peripheral devices.

Section A Peripheral Data (PA0-PA7) – Each of the peripheral data lines can be programmed to act as an input or output. This is accomplished by setting a "1" in the corresponding Data Direction Register bit for those lines which are to be outputs. A "0" in a bit of the Data Direction Register causes the corresponding peripheral data line to act as an input. During an MPU Read Peripheral Data Operation, the data on peripheral lines programmed to act as inputs appears directly on the corresponding MPU Data Bus lines. In the input mode the internal pullup resistor on these lines represents a maximum of one standard TTL load.

The data in Output Register A will appear on the data lines that are programmed to be outputs. A logical "1" written into the register will cause a "high" on the corresponding data line while a "0" results in a "low". Data in Output Register A may be read by an MPU "Read Peripheral Data A" operation when the corresponding lines are programmed as outputs. This data will be read properly if the voltage on the peripheral data lines is greater than 2.0 volts for a logic "1" output and less than 0.8 volt for a logic "0" output. Loading the output lines such that the voltage on these lines does not reach full voltage causes the data transferred into the MPU on a Read operation to differ from that contained in the respective bit of Output Register A.

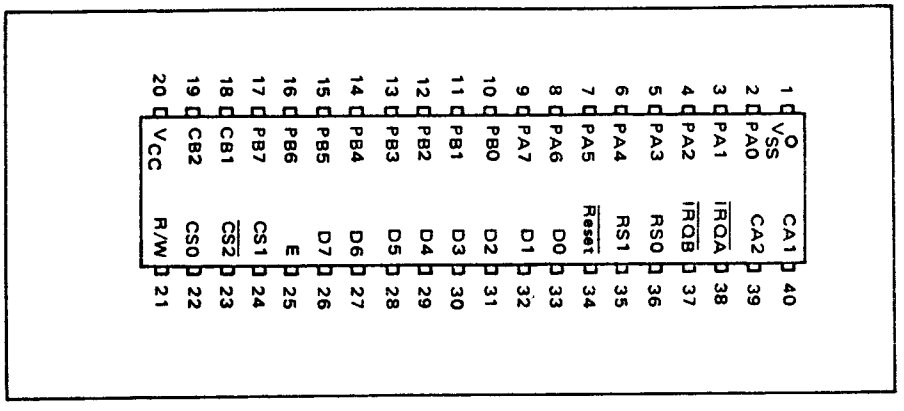
Section B Peripheral Data (PB0-PB7) – The peripheral data lines in the B Section of the PIA can be programmed to act as either inputs or outputs in a similar manner to PA0-PA7. However, the output buffers driving these lines differ from those driving lines PA0-PA7. They have three-state capability, allowing them to enter a high impedance state when the peripheral data line is used as an input. In addition, data on the peripheral data lines PB0-PB7 will be read properly from those lines programmed as outputs even if the voltages are below 2.0 volts for a "high". As outputs, these lines are compatible with standard TTL and may also be used as a source of up to 1 milliampere at 1.5 volts to directly drive the base of a transistor switch.

Interrupt Input (CA1 and CB1) – Peripheral Input lines CA1 and CB1 are input only lines that set the interrupt flags of the control registers. The active transition for these signals is also programmed by the two control registers.

Peripheral Control (CA2) – The peripheral control line CA2 can be programmed to act as an interrupt input or as a peripheral control output. As an output, this line is compatible with standard TTL; as an input the internal pullup resistor on this line represents one standard TTL load. The function of this signal line is programmed with Control Register A.

Peripheral Control (CB2) – Peripheral Control line CB2 may also be programmed to act as an interrupt input or peripheral control output. As an input, this line has high input impedance and is compatible with standard TTL. As an output it is compatible with standard TTL and may also be used as a source of up to 1 milliampere at 1.5 volts to directly drive the base of a transistor switch. This line is programmed by Control Register B.

NOTE: It is recommended that the control lines (CA1, CA2, CB1, CB2) should be held in a logic 1 state when $\overline{\text{Reset}}$ is active to prevent setting of corresponding interrupt flags in the control register when $\overline{\text{Reset}}$ goes to an inactive state. Subsequent to $\overline{\text{Reset}}$ going inactive, a read of the data registers may be used to clear any undesired interrupt flags.



Mr. Kaypro, continued from page 24.

Q 4. The original power supply had a poorly soldered area around the pins for the connector. The pins had acted like a heat sink and the result was not enough heat and/or not enough solder to make a permanent connection. With repeated heating and cooling the joint deteriorated and gave the symptoms of intermittent power supply failure. (this was not limited to K-II/2s, it also happened to my K-2000 laptop's docking station) The "cure" is to reheat and resolder the joints at the pin bases. A visual inspection of the bottom of the power supply would identify this condition. There were three manufacturers of power supplies, Boschert, Astec, and California DC. The Astec seemed to be the most reliable. Only the 85 watt supply had any markings as such, usually on a sticker on a transformer. It is really only significant if you are going to add a hard drive. Replacement of the full-height single-sided drives with full height double-sided drives does not result in any added load, if you use half-height drives, you actually reduce the load, and the modern half-height or 3.5 inch hard drives use less power than one of the full-height single-sided floppy drives.

Q 5. Installation of a hard drive requires not only the drive, but a controller and a way to connect it to the mother board. The most commonly used controller is the Western Digital 1002-05 or -HDO. Both are variations of the same basic MFM controller and are like hens teeth, hard to find. There are two interfaces that I know of, one made by Advent Products, and the Winchester Connection made by Microsphere in Bend Or. Both are plugged into the processor socket on the mother board and the processor plugs in on top of the adapter. A 40 conductor cable connects the adapter and the controller. If the TurboRom is used, two hard drives of up to 55 megabytes could be installed. The TurboRom will boot from the hard drive using either adapter. I have heard of other monitor ROMS but have not been able to get my hands on any of them for testing.

All of these "upgrades" are within the capabilities of a reasonably well coordinated novice and result in an electronic secretary that can type faster than I can, and remember more data, more accurately than I can.

A digression

At a local swap meet a couple of weeks ago I came across a Syquest 306RD removable media hard drive. I couldn't resist the $5.00 price tag, and at a subsequent trip to Silicon Valley I ran across some media at a bargain price. I had never heard of Syquest before but inquiry revealed that a standard MFM or RLL controller would work, so I hooked it up to my test PC-XT with an MFM controller as an experiment. Wonder of wonders it checked out flawlessly, so I formatted 5 cartridges and then the light bulb went ON. This drive has 304 cylinders and 2 heads, and formats to 5 megabytes more or less. That is the same as one logical drive on the K-10. Maybe I could install the Syquest as the second hard drive in my K-28 addressable as E: since the 20 Mb drive is partitioned as A:, B:, C:, and D:, and move my public domain software collection to 5 Mb cartridges. Stay tuned for further developments.

At an electronics surplus shop, I discovered an uncased, 9 inch, amber monitor, with the mounting "ears" for the CRT on the back side of the metal mounting band, just like the "ears" on the CRT in my K-28. The price was right,(I'm not real big on risking more than $5-$10) and the result is that the K-28 now has an amber screen, much easier to see. The change was straight forward uncomplicated, and required no special tools. A couple of "caveats", however, the easiest way to disconnect the High-Voltage lead is with a common screwdriver. Its a U-shaped connector whose legs have barbs which fit into the recess in the side of the CRT.

Lift the side of the plastic skirt and use the screwdriver to depress one leg at a time and "walk" the connector out of its socket. BEFORE you do this, however, leave the machine OFF for 15 minutes or longer so the High-Voltage can bleed off, and then hook a ground wire (any piece of wire with stripped ends will work) from the case to the screwdriver shaft just to make sure.

Coming Attractions:
Next issue we'll finish the personality decoder project. In the near future we'll look at the video speed-up for the '84 machines, and devise a replacement for each of the LSICs in the '84 machines.

Short Commercial:
I now have the MicroCornucopia large Schematics and Theory of Operations for K-II/2/IV, K-10, and 84 K-xx. They are $15.00 each, or any two for $25.00 or all three for $30.00, tax & shipping included. I also have Teac quad density drives and of course TurboRoms as well as lots of advice and comment.

Support for a Kaypro Classic

by JW Weaver

During the process of developing some testing hardware for use by the Kaypro, my Kaypro bit the dusk. Both floppy drives would not work. One or the other, but NOT both. This being a floppy based system, this action is disastrous to development projects. I tried swapping the drives, (actual the cable connections), but no change. Next came replacing the FDC chip, I just happened to have a spare, (pulled it from my Kaypro II), still no change. This is getting serious now. Well the weekend is arriving, and the monthly "1st Sunday of the month swap meet" is 2 days away, so I'll put this aside and take in the swap, after all you never know what bargains are lurking about.

At the swap meet, I found a true Kaypro 10, and the price was under $25.00. Ho-ho, I may not have to solve the Kaypro 4

problem after all. Included with the buy, are two sets of manuals, no software, but that is the easy part. KRASH II BBS contains almost all the Kaypro software that the former Sacramento Kaypro Users Group had in their library. I got home, and started the exploration of this new and wonderful find. Oh-oh, just a few problems to compensate for the low price. Keyboard has two key tops missing and upon further examination, the two key mechanisms are damaged beyond repair. But, I have the keyboard from the Kaypro 4 that can be substituted for this keyboard, minor little thing like case is blue, while the 10's case is gray.

Now to try my new acquisition, plug in the keyboard, also power cord, snap the power switch.... nothing... meekly turned the power switch off, disconnect the power cord from the machine (now its just a machine, not even working). Logic indicates the fuse must be bad. Open up the case, (see side bar), checked that the power connector was secured on the computer board, yes! Well there went an easy solution, so I removed the computer board, and ran a continuity check on the power supply fuse, fuse checks out ok. Now it is going to require some work, removed the power supply and upon close examination saw that a solder joint to a capacitor, had broken. On further examination I found 2 more bad solder joints, at least I can handle these. Powered up the solder station, applied heat to the joints, solder flowed good, must mean the joints are good. Test now imminent, reconnected the power supply inside the Kaypro chassis, also tied volt meter to power connector, turned power on, "wa la" power is good. Replaced computer board and reconnected all cables, secured case cover. Time for the magic to come forth, turned the power switch on, lights come on, but alas, the system does not boot from the hard drive. But it does boot from floppy.

Now is the time to seek help, desperately I called on Chuck Stafford. Chuck spotted the problem right off, hard drive at fault, talk about being in a daze. I could not think straight. But thanks to Chuck, we isolated the problem. However, this still left me with out a fully functioning Kaypro. Went to my favorite surplus store, Tech-Line, and found another Kaypro 10, again at less then $25.00. By this time I'm not so enthusiastic about a new acquisition. At home, connected up, and powered up the system, ok, so it isn't working perfectly, but it is working, and it is booting from the hard drive. Fan was floating loose inside the case, two bolts and it is tied down. Serial port is not working correctly, swap

Opening and Closing a Kaypro for Repairs

REMOVE CASE COVER

1.Remove power cord from connector on back of computer.
2.There are 4 pan-head bolts located on each side of case, and 2 flat heads located on top of the case.

REMOVE COMPUTER BOARD

1.Be sure that the power cord is removed from connector on backside of system.
2.Viewing from the front, there is a 4 pin video connector located on the left side at the rear of board, lift up to disconnect.
3.On the right side, at rear, is the power connector, pull to the right on connector.
4.Also on right, at front of board, is the floppy connector, 26 conductor ribbon cable, lift up to disconnect.
5.At the front, on the right side, is a 4 pin connector, this is the reset lines, lift up to remove.
6.Now you may have an additional cable connected to the expansion connector J9 (in most systems), this goes to the hard drive interface board, lift up to disconnect.
7.On the front edge of the computer board, at each corner, are bolts to remove.
8.On the back side of the case, the 3 I/O connectors, each contain 2 hex shaft thread-ons, these need to be removed.
9.Also, on the back side of the case, left corner, is the last bolt holding the board in.
10.Gently remove board.

REMOVE POWER SUPPLY

1.On the outside of the back of case, there are 4 pan head bolts, located centrally around the carry handle, remove these to loosen the power supply board from the case.
2.Disconnect the cable from the right side of supply.

RESTORE POWER SUPPLY

1.Reconnect the cable to the right side of supply.
2.Hold the supply up to the holes, on the backside of case, screw in the 4 pan-head bolts into the supply mounts.

RESTORE COMPUTER BOARD

1.Place board inside of case.
2.Apply the bolts in order, starting at the left rear corner of case.
3.Restore the 6 hex thread-in's into the I/O connectors.
4.Replace the 2 bolts, holding the board onto two vertical plastic shafts.
5.At the left rear of board, re-connect the video cable (4pin).
6.At the right rear of board, re-connect the power cable to board.
7.At front right side of board re-connect 4 pin reset connector.
8.Behind the reset connector, on right side of board, is the 34 pin floppy drive connector, re-connect, on the board is the number 1 at one end or the other of connector, this must match the stripe on the ribbon cable. (stripe is either red or blue).
9.If you have the extra cable, (50 pin ribbon), re-connect this to the expansion connector, again the connector is marked as which end is the number 1 pin, and cable is striped.

RESTORE CASE COVER

1.Place cover on case, side snap to the front.
2.Screw the 2 flat heads, on top of the cover, in first.
3.Screw the 4 pan heads, on each side, left and right.

ALL DONE.

the two SIO chips, it works now. Raster scan is kind of weak, but I CAN LIVE WITH THAT. So now I have working systems and I had better get busy writing some articles.

Happy Hacking... JW

Definitions
FDC - Floppy Disk Controller chip i.e., 1793 in the Kaypro.
SIO - Serial Input/Output chip i.e., Zilogs Z80-SIO/0

Krash II BBS @ (916) 427-9039 300-9600 8 N 1
Mail to Drawer 180, Volcano Calif. 95689

Keyboard Macros on A Kaypro

by Jack Wyatt

The Kaypro keyboard contains, in addition to the standard ASCII keys, 18 keys and 15 unfilled key positions which produce codes with the high bit set. The Kaypro BIOS uses a table lookup in ROM to convert the codes for the existing keys to 80h - 91h; you can use the Z80 I/O commands to display the primary codes of all the keys.

It is very simple to add keys to the unfilled positions and cut additional holes in the cover to accommodate them, and there is space available in the K10-83 BIOS to utilize them for keyboard macros. Earlier models have no extra space, so to provide macros it would be necessary to enlarge the BIOS and be happy with a slightly smaller TPA. I believe that the codes could also be used with one of the popular key translation programs, such as QK21 or Smartkey, but I have not tried this.

Seven vacant positions are in the top row, one next to the backspace, one next to shift key, and six in the bottom row. I found that keys at the bottom caused me to make too many errors, so use only the top 8. The software, however, provides for all 15 and three extras for codes EA, EB, EC transmitted by the E0 key with shift and control. The high nibble of the special key codes is A to F from bottom row to top. The low nibble begins with 0 for the leftmost, increasing to the right, except row F, where the leftmost key is FE.

KPKBD1.LBR contains system revisions to K10-83 which permit use of existing and added keys as function keys to automatically enter strings of ASCII and control characters of any length within the total memory available, 140 bytes. The revision is entirely in the BIOS; it does not reduce the transient program area as most key translation programs do, and it does not need to be loaded separately.

Files included:
KPKBD.DOC Documentation for the system.
KBIOSXX.AZM The revised BIOS program.
KBIOSXX.HEX Assembled program for overlaying the BIOS or PUTSYS.
ZPUTKEY.COM ZCPR overlaid by KBIOSXX.HEX for Kaypro 10.
KPATCH.AZM Program to patch the desired codes into your system.
KPATCH.COM Assembled and loaded patch program.
P.COM Transmits input directly to printer.

Installation: For Kaypro 10, it is necessary only to run ZPUTKEY and push the reset button. For other models I recommend getting the TINKER disc for your machine.

Revise KBIOSXX.AZM to allow for the different BIOS of these machines. It may be necessary to increase the size of BIOS slightly and adjust the jumps at 0000 and 0005 accordingly, if the space savings I made can not be accomplished in these systems. In this case it will be necessary also to reassemble BDOS and CCP to relocate them. This can be done using DASM and Z80MR. Or perhaps more easily you can use MOVCPM to create a slightly smaller version of CPM.

A>AZM KBIOSXX.xAx (gives KBIOSXX.HEX)

A>DDT (PUT filename)

-IKBIOSXX.HEX

-Rbias (for Kaypro 10, boot is EA00, bias is 3580; for other models, boot will be higher than this; it is 3 less than the address jumped to at 0000. Use EDFILE to look at the jump table at this address, then to find the same table in your PUT program; this is boot. Bias = this address - the wboot address at 0001 + 3).

-G0 (note this is zero, not O)

A>SAVE 40 (your putkey filename)

A>(your putkey filename)

push reset button

The special keys will give whatever has been placed in the VTAB and SNDTAB tables.

Patching:

If you are creating a new ZPUTKEY by overlaying KBIOSXX.HEX, put the desired single-byte codes in VTAB. String codes go into SNDTAB; first the input code from the key, then the string length, then the string. There is no separation between strings, and the last string is ended by ~, code 7E. The location of the input code in VTAB is filled with a null code 00 to activate the string.

In my Kaypro 10 version of KBIOSXX the order of codes in VTAB is F1 F2 F3 F4 B1 C0 C1 C2 D0 D1 D2 E1 E2 E3 E4 D3 C3 B2 FE F0 F5 F6 F7 F8 F9 E0 B0 A0 A1 A2 A3 A4 A5 EA EB EC. This is shown in the KPATCH table mapin; it is dictated in part by the original conversion table, which is in ROM.

After the put program has been overlaid, use KPATCH to change the key translations. It is not necessary to know where the codes go in memory; just follow the program instructions. The program copies two sectors which contain tables

VTAB and SECTAB into memory at 0A00, and after the changes have been completed it writes the sectors back into the file.

You may get the records to modify either from file A:KTEMP.COM or from BIOS in memory. After modification the records may be returned either to A:KTEMP.COM or to memory BIOS. If you would like to try out the patches before saving, return them to memory and exercise them. If satisfactory, load KPATCH again, get the records from memory, then return them to KTEMP without further change.

KPATCH can display the translated codes either in hex form or as ASCII and control characters, as you choose.

PIP KTEMP.COM=(your PUT filename)[v]

KPATCH

"record from file KTEMP, enter f; from memory, enter m"

"store in file KTEMP, enter f; in memory, enter m"

"enter a for ASCII, h for hex display"

Make these choices as you choose.

"enter special key" if you enter a standard ASCII key the request will be repeated, otherwise the current translated value will be displayed in the form you have chosen.

"a = accept, c = change" if you wish to change it, enter c

"enter the new byte (^@ = 0 = skip to string table)" enter the key for the ASCII or control code you wish this key to produce. If the key is to produce a string, enter ^@ to give a 00 code in the table. (^ means hold the control key down. Codes are entered as ordinary keyboard symbols, but as you enter they are displayed in hex form.)

"{hex code for key} not found; add string? y/n"

or

"{string displayed} a=accept,d=delete" d deletes it "replace? y/n"

If you answered y to either add string or replace:

"space {amount of space left for strings (decimal)}. You need 2 more bytes than the string length.

"enter string of ascii and control chars, ending with ~"

If you exceeded the space available:

"table full, you MUST enter ~"

"enter special key" proceed with the next key to look at or ~ to end.

You may check your changes by reentering the same key before ending the run; the program will display the new values. At end the revised tables are written into KTEMP.COM or into memory BIOS.
KTEMP

push reset button

check operation

rename KTEMP to any filename you choose. You may wish to have more than one version.

KBIOSXX and ZPUTKEY included in this library contain codes in VTAB and SNDTAB which give the standard translations for the cursor keys and number pad except for demonstration purpose the comma key has been coded to produce the entry "function ,". Try it out; zputkey, push reset button, push comma in number pad. Some other strings are included for the unfilled key positions.

My main objective in this was macros for printer control strings using ^P, but some of the controls are changed if entered at the CPM prompt, so to use them at this point it is necessary to first hit p to call P.COM, which transmits the string directly to the printer without change.

Since making these changes I have installed the Advent Turborom in one machine. The Advent BIOS does not have the keyboard string routine which is in the Kaypro BIOS, so it can not handle the macros. However it is an 8K rom with only about 5.5K used, so there is plenty of room in rom to contain this routine and much more whenever I get time to work on it. Of course I could use MOVTURBO to provide more room, but I would prefer to use all that wasted space and at the same time eliminate the original translation table from rom.

I have not tried ZCPR3, so do not know what modifications might be required for it.

Jack Wyatt, SMUG, 916 925 3309, 578 Garden St. Sacramento 95815

PC/XT Corner

by Frank Sergeant

Regular Feature

Intermediate Users

A Bit Everything Part I

Forth, Stepper Motors, Parallel Port, PC Memory Map, Blues

When you hear your cat purring, you probably don't consider he might be trying to communicate important details. You take it merely as a sign of contentment, scratch him behind the ears, and continue with what you were doing. Lately I've come to believe my girl friend treats me the same way when I'm discussing computers and electronics with her. Oh well, there is still **you** right?

Table of Contents

This was really one article, now in two parts: my general complaining and recent experiences, a discussion of the PC memory structure, and part II an example of how to use Forth and very little hardware to control a stepper motor from the PC's parallel port.

Several people have written with questions about the layout of the PC's memory and about the PC's parallel port. Coincidentally, I have built a simple stepper motor controller circuit, using mainly the PC's parallel port and Forth. I'll tend to those items next issue, discussing my other PC adventures and opinions, now. Keep them cards and letters coming. Otherwise, I would be afraid to discuss the PC's memory layout, etc, assuming you know all about it already.

A disclaimer that should go with all my articles is that I don't know what I'm talking about. Feel free to write me directly or to write letters to the editor to correct me, so eventually we can all get the details straight. Especially if I say something like, "The designers at Intel thought ..." I mean, how the hell would I know what they thought; it's just a literary (literary?) device. It is too inconvenient to sprinkle "it **might** work this way" and "I have a vague feeling it was sort of like ..." throughout the article, so I'll talk as if I know what I'm talking about, but we both know better.

I'm in a rush. I am madly racing to finish a mixed C and assembly language conversion project for a client before Summer II starts at SWT (Southwest Texas State University).

Email, the Internet, and SWT

I love email. It so much more pleasant than dealing with traditional letters and the post office. If you don't have email access, that's ok; I'm still glad to get your letters. I may stay a student forever, just to keep the cheap Internet access. Cheap? — everything has its price. Often when I'm discussing school, my girl friend **knows** I'm not purring. She says I should just buy commercial access to the Internet and say to hell with SWT. I'm often close to doing that. Today, for instance, I am **not** looking forward to going back in Summer II. But, I am committed. I promised to teach the intro to computer architecture labs and one introduction to campus computing facilities seminar (as if I knew anything about the damn campus computing facilities!).

One problem is continuity of email address. When a student leaves SWT, bang! that email address is gone. To keep a year round SWT computer account, a student must be enrolled during both long semesters and both summer terms. Although I'm not enrolled during Summer I, I get to keep my account because I am a TA (teaching assistant). The ACM (Association for Computing Machinery, 1-817-776-6876 or Account-Info@ACM.ORG), of which I am a member, offers a permanent email address for $89 per year. Well, they only charge $10 per year for the email address, but you have to be a member, which costs about $79 per year ($24/year if you are a full-time student. This is a forwarding service only. You still need a real email account somewhere. You give out your permanent ACM email address to all your friends and tell the ACM your current email address. I think this is a great idea, except for the cost. For about $100/year a GEnie account also offers Internet email access. So do AOL (America On-line) and Compuserv, but I don't know their rates. GEnie is the only one of them with a local San Marcos telephone number.

For an additional fee — I think it runs about $20/month plus long distance — ACM will give you an account on their own computer. Let's see, $79 + $10 + (12 times $20 = $240) + (1 hour/week times 52 weeks times $7.20/hour = $374) = $703 per year. Pretty soon you're talking real money. And, what if you wanted to spend an hour a night instead of an hour a week? In the mean time, SWT tuition and fees (forget books for the moment and the wear and tear on me) cost over a $1000 per year to enroll in just one course each term. For this, I get unlimited Internet access, homework assignments, and a chance of an eventual Master's degree. Economically speaking, though, I have not factored in the opportunity cost: theoretically, I could do something more productive with my time that would

swamp any apparent savings from staying in school. I sort of want a Master's degree, but I don't think it is very important at my age and with my previous job experience. The people willing to pay for my special programming skills don't particularly care about any degrees and I am not likely to apply for the sort of "straight" job that would care.

Dem Mean Ol' Upgrade Blues

I'm behinder than ever on the conversion project. In connection with it, I've had to (had to?) upgrade my computer system, and it always takes me a while to get things connected and working right. I had (had?) to add a cdrom drive, but there were no spare bays in my AT-style case, so I also had to upgrade to a tower case.

Well, I couldn't just move the innards from my old case to the new one. My bedroom office was in such a mess I had to pull **everything** out of the room and the closet to vacuum and straighten up. I couldn't vacuum with the lousy old vacuum we had, although I tried until I was sick of the hose falling apart in my hand. So I had (had?) to buy a Shop Vac at Wallmart. That may have been my best purchase. It really **sucks**. The room hasn't been this clean in years.

Finally I began to install all my old parts into the new case. This took a while. The cdrom drive was part of a "multi-media kit" that included a Soundblaster board, which also serves as the controller card for the cdrom drive. The 2nd of the 3 Soundblaster disks could not be read on my machine (or any of the other 3 machines I tried it on), so I couldn't install the software or test the sound card.

The (unlabeled) installation disk for the cdrom drive seemed ok, but none of the menu choices seemed to be for the drive I had, but I tried it any way. Nothing worked. I didn't know if the cdrom drive or the controller board (the Soundblaster board) was defective or whether I had installed it incorrectly. Boo! I didn't want to send the drive back and wait for a replacement, nor did I want to screw things up further by tinkering too much. I'd never seen a (computer) cdrom drive before, so I didn't know what to expect. Gradually, over several days of fooling with it, I traced down various shareware this and thats and found a program that would access the cdrom drive. I could tell it to open the tray (eject) and close the tray. This worked fine, which told me the drivers and hardware connections were generally correct. But read the drive? No! "Drive not ready" no matter what I did.

Finally, I took the cover off and looked and looked and unscrewed this and that. Now I know what one looks like inside. I have a Matsushita (Panasonic) drive. It's cute. A metal bridge above the spinning (spinning?) cd disc holds a plastic disk with a magnet that clamps to a mate below the disc. My disc was **not** spinning. Should it be? All the time? Surely some of the time! When I removed the bridge altogether and tried to read the drive the disc did spin. I replaced the bridge loosely, again it spun. I replaced it completely. No spin. The bridge was bent, but so neatly that I thought it was **supposed** to be that way! I removed the plastic parts from the bridge, unbent it by hand, replaced the plastic parts, and reassembled everything. The drive has been working perfectly since then. It's a good thing, too, as I would rather not explain my tampering to Pacific Coast Micros (800-581-6040, 619-581-6040, FAX 619-581-0125).

Vendors

I have mixed feelings about Pacific Coast Micros. They caught my attention when a friend at SWT spent all day checking RAM prices and found them to have the lowest: $39/megabyte as of May, 1994. So, as part of my upgrading for the conversion project, a month ago I bought two 1-Mbyte SIMMs (single in-line memory modules) and a 14.4Kbps FAX/Modem and a larger harddrive (Maxtor 345 Mbyte) from them. I thought the prices were reasonable ($39/meg, $129, $275 respectively) and service fairly prompt. I was disappointed with the FAX/Modem software. I haven't been able to figure out how to send a FAX yet, but it isn't my highest priority. Then in early June I bought the cdrom ("multi-media kit") with Soundblaster 16 soundcard for $299 and tower case for $99, which I do not feel are such great prices. I was slightly annoyed that the cdrom drive didn't work at first and that the documentation accompanying all this was <crude adjective deleted> poor. I think they've hit upon a very successfuly strategy: offer reasonable RAM prices (reasonable? RAM was $29/Mbyte several years ago just before our government started hassling the oriental RAM suppliers) and collect the gravy from the one-stop-shopping syndrome.

However, I don't know if I will buy more from them in the future. I phoned on a Monday for a replacement for disk #2 of the 3 Soundblaster 16 installation disks. A disk arrived by air express the next day! Mainly, I think, I want to feel they care about me and my problems. Unfortunately, it was the wrong disk. Yes, it was the #2 disk for Soundblaster installation software, but it was for a newer version of the software and wouldn't work with the other two disks I had. So, I phoned them Tuesday for a whole matching set of installation disks (oh, I wish I'd thought of that originally!). Over a week later the new disk(s) still had not arrived, so I phoned again. Where is their prompt service? How come the replacements weren't here sooner? They wouldn't take any of these follow-up support phone calls on Saturday, even though their sales department is open Saturday, nor on their 800 number. Plus they have one of those damn voice mail phone deals where it answers to a recorded voice saying to press certain keys to reach customer service. Then it rings and rings, at my expense, and is never answered. Then I call back and eventually reach someone. Well, I finally spoke to someone who said he'd send the other two disks of the set from which he sent the #2 disk — they were right on his disk. Sure enough, he didn't do it! He sent me a full set of 3, the same version I had first received, and again the #2 disk was unreadable. I think this is **very shortsighted**! Their very best advertising comes from keeping customers happy after the sale. How well they do this

affects what the customers say to friends who are wondering where to buy computer equipment, never mind the customers who might also happen to write the occasional magazine article! Can I recommend them? Not really. After the last screw up, I telephoned Creative Labs, the manufacturer of the Soundblaster 16, with credit card in hand, hoping to **buy** a set of installation disks or find an ftp site for them. Apparently Creative Labs is sending me a set at no charge, but they haven't arrived yet. Maybe next time I need to buy something I'll get a Computer Shopper magazine and try some other vendor.

CDROMs

There ought to be a way to take a $79 mass-marketed CDROM player from Radio Shack or K-Mart and convert it to computer use. But, the computer versions are cheap enough that there is not much incentive to do this. I probably would have waited another year to buy a drive, except the software upgrade for the conversion project I'm working on is best bought on CD. I've also been looking longingly at ads for the DDJ (Dr. Dobbs Journal magazine) CD containing all their issues from 1988 through 1992, or some such. I think this is overpriced at $79 (unless, of course, you **need** it, then it's cheap). I've clipped most of the articles from those years and saved them, but a manual search is a tedious thing indeed. Further, consider the savings in space and clutter! I haven't bought it yet, but I probably will eventually. Year by year, more such info will be available on CD. This is the wave of the future. You'll not only have a CDROM drive, you'll have a juke box of 'em with multiple CDs on-line at the same time. A business can consider putting the accounting info on CD instead of on microfiche, etc. etc.

Simtel

I'm vague on the history, but Simtel is the big collection of MS-DOS software. I think it was on a military site but was shadowed (is that the word?) on various ftp sites, especially OAK.oakland.edu. I think the military site is dead, but the archive lives on at OAK and through Coast to Coast Telecommunications (810-623-6700, FAX 810-623-0040), run by Keith Peterson. Various people sell CDs containing some or all of this massive DOS collection of shareware and public domain software, but Coast to Coast, as I understand it, maintains the archive, claims about 300 additions to it per month, and puts out all of it on a pair of CDs quarterly. I ordered a pair when I ordered the drive, and the pair arrived first. By reading them on a friend's computer with a CDROM drive, I was able to extract some of the software that helped me determine my drive was connected correctly. (If only I had a working CDROM drive, I could read these CDs to tell me how to get my CDROM drive working.) The big danger with these shareware collections on CD is the amount of time you spend browsing through them. Each CD is equivalent to a massive harddrive.

61 Floppy Disks versus 1 CD

I mentioned the reason I had (had?) to get a CDROM drive was a software upgrade. $164 to upgrade to version 10.0 of Watcom C/C++ on CD or pay that, plus an extra $99, to get the software on 61 floppy disks instead. Can you believe it? 61 floppy disks? Those prices do not include the manuals. The manuals are on disk. Yes, for another $169 or whatever, you could get a set of printed manuals. I have been somewhat annoyed at Watcom (is it just that I can't be happy?), but it is what we are committed to at the moment on this conversion project. I find their documentation (printed or on-line) INADEQUATE and unsatisfying! I find their technical support likewise. I cannot do this project, at this time, entirely in Forth, or I would. I am converting a large system of 16-bit Borland C and MASM/TASM (Microsoft Macro Assembler/Borland Turbo Assembler) assembly programs to 32-bit Watcom C and TASM32. The theory is that so much is already done that, in spite of the disadvantages of C and ordinary assembler, converting the existing system is "better" than redoing it in Forth. I have heard (and used) this reasoning for many, many years. At each point in a project you can say, well, we were wrong. It really would have been cheaper to redesign and do it correctly from scratch. However, NOW, it is cheaper to continue the way we have been going than to start over. A little later you say, well, we were wrong again. It would have been cheaper...

I had so much trouble getting all this to work, both with their old version 9.5 and with this new CD version 10.0, that I seriously considered recommending returning the upgrade for a refund. I finally got it going, after many, many unpleasant hours of effort, and agree that the debugger in v10.0 is better than the one in v9.5. For that alone, I want to keep it for this project. Doing it in Forth we would not even need a debugger, as the whole Forth system is a debugger. Watcom claims the new version requires 8 Mbytes of RAM, and a full installation takes around 175 Mbytes of harddisk space. I've installed much less, as I don't need the OS2/Windows/etc stuff — just the DOS & DOS Extender stuff, so I am ok for now on my antiquated '386SX-25Mhz 4Mbyte RAM machine, although I have to WAIT for the long compiles. I think software written **right** ought to run fast on **even** a 25 Mhz '386 machine. (Yeah, yeah, get a CPM machine).

PC Memory Layout: History

Look, the standard setup was an 8-bit data bus and 16-bit address bus. Who knows why, except powers of 2 are convenient. Sure, there are 4-bit busses and 12-bit busses, etc. We think a 3-bit or 7-bit bus might be weird, but that's just because of what we grew up with. When memory was so expensive you settled for 4K of RAM and dreamed of upgrading to 16K, you never thought you'd ever fully populate the 64K address space that 16 address lines provided.

Address Bus and Data Bus

What's a "bus," you say? It is a set of signal lines that various components of a computer system can tap into in parallel, rather like a large, diesel, street vehicle takes several people at a time to various destinations — just get on the bus — just put

the signal on the bus — stay in your lane (time slot), though, or you might have a collision (called bus contention in the computer world). To be more concrete, the 16 address lines, taken as a set are called the address bus. The set goes from the CPU to all the different memory elements. The PC, however, also has something called DMA (Direct Memory Access). This is really a separate controller unit that can put addresses onto the address bus when the CPU is not using the address bus, to control the transfer of information between memory and an I/O device (such as your floppy disk) without requiring the CPU's constant attention. The 8 data lines, taken as a set are called the data bus. Whatever device is sending data writes this to the data bus while whatever device(s) want that data, read it from the data bus. In a small computer system, the CPU and the memory might be the only devices on the data bus, with the transfer going in either direction, depending on whether the CPU is writing to the memory or reading from the memory. I'm getting worried. The letters with the questions encouraged me for a moment, but I'm getting cold feet. Is this level of information really of any use to any of you? I feel most of you will just have to skip right over it, but if you **don't** know it, you **need** to know it. Should those get it here or from some stranger on the street?

Memory Map

So, how do you know that 16 address lines provide a memory space of 64K elements (bytes, in our case)? The math behind it is so simple you can do it on your fingers. You do have 16 fingers, don't you? Oh well, you can get started with only 10 fingers. Hold up one finger and say "my finger can be on or off. Thus, it can choose between two individual things." Then hold up a 2nd finger and say "my second finger can also either be on or off. Thus, it can choose between 2 sets of things. Thus, with 2 fingers, I can pick one of two sets of things and then one of two items inside the chosen set — that is, I can select 1 of 4 items." This gets tedious to say as you move to 3, 4, 5 fingers, but the idea is the same. A shorter way of saying it is "2 times 2 = 4, times 2 = 8, times 2 = 16, times 2 = 32, times 2 = 64, times 2 = 128, ... , times 2 = 65,536" while you put up one finger after another until all 16 are up.

Prices Fell

Unfortunately, memory prices fell. What do you do with 16 address lines if you want to address more than 64K. (Since a "K" is 1024, 64K = 64 times 1024 = 65,536.) You **could** add more address lines to the CPU. But, if your CPU is already built and designed and paid for, you might want to add those lines outside the CPU. This was and is done on 8-bit systems (ie systems with 8-bit data/16-bit address busses) by adding a latch. You write the higher order addresses, say 4 bits, to a 4-bit latch and let the CPU take care of the other 16 address lines. You could use a DIP switch and set the upper 4 bits by hand. Then you could have 16 banks of 64K bytes. How did I come up with the number 16? Well, count on your fingers: 2 x 2 x 2 x 2 = 16. You'd have to have some of your operating system duplicated in each of the 16 banks, though. Or, you could dedicate 32K to the operating system — it would never be switched out, and let a 4-bit latch switch among 16 banks of 32K for the other half of the CPU's address space. Naturally, you could extend this idea to an 8-bit latch, etc. Assuming we all buy the idea that a computer can turn on an external device, for example it can write to a printer, then it is but a small step for the computer to write the data to its **own** address latch. Now you don't have to fool with the DIP switch settings, and the computer program can determine which bank it wants to access.

Forethought

The next step is to build the latches inside the CPU and give it 20 address lines. Now you have gone from an 8-bit CPU (e.g. the Intel 8080a) with a 64K address space to an 8-bit CPU (e.g. the Intel 8088) which can address 16 times 64K = 1M. You could think of the 8088 as having a 4-bit internal latch. The 8088 has internal 16-bit registers, so you could consider it to be a 16-bit CPU, although it's external data bus still has only 8 lines. When it wants to read 16 bits, it does so with 2 separate fetches from the memory. If Intel's designers had only known this damn, little, inconvenient, control processor would one day grow up to rule the world, they might have designed it differently. Essentially the 8088 and 8086 are the same CPU except the 8086 has an external 16-bit data bus. To the programmer they are the same CPU. From then until now, the 80286, 80386, 80486, and "Pentium" are all backward compatible with the miserable 8088/8086 chip.

Segments

While you can think of the 8088/8086 as having a 4-bit upper address latch, things are more complicated than that. It actually has a set of 16-bit segment registers whose value is shifted left 4 bits and then added to the 16-bit address in whatever other register contains the main address, to give the final address that is placed on the 20-bit address bus. Other than politicians, theiving governments, women, and cars, this segmented architecture has caused more problems throughout the world than anything else. If you already understand segments, there is no point in beating the subject to death. If you don't, then I'm not sure we can cover it adequately here, but I have to try:

There are 4 segment registers: DS (Data Segment), ES (Extra Segment), CS (Code Segment), SS (Stack Segment). Each can hold a different value. This is the 16-bit value that is shifted left by 4 (i.e. multiplied by 16) and then added to some other address. What other address? It depends. For example, to fetch the next instruction to be executed, the address — in Intel terminology this is now called an "offset" because it can be thought of as the offset from the base address established by the segment register — of the next instruction is contained in the IP (instruction pointer) register. In most other CPUs you would call this the PC (program counter) register. When accessing an instruction, the CS:IP pair are used. Suppose CS = $0001 and IP = $0100. The actual address put on the address

bus for that instruction fetch would be (16 times $0001)+$0100 = $0010+$0100 = $0110. I'm using the dollar sign to indicate hexadecimal and no dollar sign to indicate decimal. So, the CS register is used with IP for fetching instructions. The DS register is used with most other registers for addressing data. The ES register is occasionally used for data access, as when copying a string, where the DS:SI (source index register) pair address the source and the ES:DI (destination index register) pair address the destination. The only one left is SS. Pushes and pops to and from the stack use the SS:SP pair.

Memory Models

If all the segment registers contain the same value, then you have what is commonly called the tiny model. With DOS C, etc. compilers you often have to tell them whether you want code generated for the tiny, small, large, huge, whatever model. If code is in aa separate segment from the data, but all the data is in a single segment (i.e. DS=ES=SS, but CS may be different) then you have the small model, etc. etc. By have your data and code and stack spread out in different segments, you can have an address space of 4 times 64K without changing values in the segment registers. By changing the values, you can address the full 1Mbyte which the 8088/8086's 20 address lines allow.

The Famous 640K Barrier

Oooh, it gets worse. If I'm supposed to have a meg, how come I only have 640K? The PC's design dedicates the upper 384K to video display RAM and various ROMs, such as your system BIOS ROMs and any ROMs on the various adaptor cards. For example, your hard disk controller card or video card might have its own ROM.

Compatibility at the Hardware Level

To various people's regret, and as pointed out by Tilmann Reh and others, lack of adherence to standard methods of accessing facilities in PC systems causes lots of problems. If the original PC had provided adequate performance with standard access methods, things might have been different. Performance is so poor when using the standard operating system for video output that **everyone** bypasses the operating system and writes directly to the video hardware. This means such software will not run correctly on machines that are not essentially identical to the original PC **at the hardware level!** The original PC had a Motorola 6845 CRTC (Cathode Ray Tube Controller) chip. The latest '486 clone has an ASIC (Application Specific Integrated Circuit) that, among other things, mimics the 6845 chip down to its internal register level. Old software that thinks (thinks?) it is writing to a real 6845 works fine with the ASIC. This backward compatibility comes at a terrible expense in hardware complexity. We can't start over and do it right, we have to maintain compatibility forever with the past — or so many people think. Now comes the PowerPC fast enough to emulate a '286 running Windows (ok, a '386 or '486 which is emulating a '286 running Windows).

So, the 80286 emulates an 8088/8086. An 80386 emulates both an 80286 and an 8088/8086. An 80486, ... Most of the "fast" Intel CPUs spend most of their life pretending to be an 8088, running real mode 8088 code, including the segments and all the trouble they cause.

Real Mode versus Protected Mode

When the '386 wakes up, it wakes up in real mode. This means it behaves just like an 8088/8086 and runs in 16-bit mode. A very complicated protocol can set up various data structures needed by protected mode, and transfer the '386 CPU into protected mode. At this point, the '386 is not limited to segments 64K bytes long. It can now access segments up to 4 Gigabytes long! Also, the 16-bit registers ax, bx, cx, dx, si, di, sp, bp become 32-bit registers eax, ebx, ecx, edx, esi, edi, esp, ebp. Essentially, after going through the pain of putting the CPU into protected mode, you can set all the segments to a single base address and thereafter ignore segments altogether, as each of the 32-bit registers is large enough to hold an offset that can address anywhere in the entire 4 Gigabyte address space. (Let's see, at $39/Mbyte ...). Finally, we have the pleasant flat address space that the 68000 family has offered from the very beginning. Beware, though, that some people have good reasons for recommending the continued use of segmentation in protected mode. I am not one of these people, but they have a point. Put each data structure into its own segment and let an accidental access past the end of the data structure cause an exception, that is, let your program blow up instead of happily continuing with bad data and program bugs. There is more, such as paging and virtual memory, but let's leave that alone for now.

Extended versus Expanded Memory

I'm sure you are sick of these terms, so I'll just say that extended memory (XMS) is the rest of your PC's RAM above 1 Megabyte. It is generally only accessible to you directly if you are running in protected mode. Expanded memory (EMS) is more of an idea than a reality. At one time, extra memory cards could be plugged into a PC. The PC couldn't address the memory directly, but circuitry could map parts of the card's memory into 16Kbyte expanded memory frame buffers addressable in the 384K region just above the main 640K, i.e. in the lower 1Meg region that even an 8088 could address. Software that needed access to huge amounts of RAM could then run. Nowadays, software that expects to find EMS RAM usually gets from what is really extended memory, via a utility program that accepts the EMS requests and provides it from XMS memory. Another special term is "high memory," which means the first 64K above the 1Meg limit. It just goes on and on, making the Motorola 68000 series of processors look awfully pretty.

Remember my GEnie email address is F.SERGEANT or, through the internet, f.sergeant@GEnie.geis.com in case I say to hell with school and lose my fs07675@academia.swt.edu account.

Special Feature

Intermediate Users

Part 4: IBM PC Bus

MULTIPROCESSING FOR THE IMPOVERISHED

by Brad Rodriguez

MEMORY

I'll start this installment off with something easy: the on-board memory, shown in Figure 1. U20 is a 27256 (32K byte) EPROM. Recall from the previous installment that the EPROM is the only device which does *not* use the address generated by the memory mapping logic; instead, it uses the CPU address lines A0-A14 directly. OEPROM\ comes from the decoding logic, and will be asserted (low) only for processor Read cycles when A15 is high — i.e., any read to CPU addresses 8000-FFFF hex.

You might wonder, then, why A15\ (the logical inverse of A15) is connected to the EPROM's chip enable input. It's true that this signal is logically redundant, since the EPROM's output enable can only be asserted when A15 is high (A15\ low). You could tie the EPROM's CE\ input low and the circuit would still work. But many EPROMs will consume less power when CE\ is high, so it's worthwhile disabling the EPROM whenever you can. Connecting A15\ to CE\ costs no extra logic — I needed A15\ anyway — and reduces the average power consumption of the board.

U21 is a byte-wide static RAM. Since the RAM appears in the 1 MB "mapped" address space, this chip uses address lines MA12-MA16. (A0-A11 are the same for all devices.) RAMRD\ and RAMWR\ are asserted (low) for read and write cycles, respectively, to addresses in the 127 KB "on-board RAM" range, E0000-FFBFF. Again, tying ONBOARD\ to the RAM chip enable input is logically redundant, but costs nothing and saves some power.

Thanks to the consistent "JEDEC" pinout of byte-wide memory devices, you can install a 128Kx8, 32Kx8, or 8Kx8 RAM chip in the U21 socket. (If you install a 28-pin device in this 32-pin socket, be sure to install it in the "bottom" of the socket, i.e., with pin 1 of the IC inserted into pin 3 of the socket.) The low address, data, and control lines are all the same. Only the high four address lines (MA13-MA16) are different, with the following results:

128Kx8 RAM (e.g. 628128): all four address lines are significant, so the chip occupies the full 127 Kbyte space. The last 1K of the RAM is inaccessible because this space is reserved for I/O.

32Kx8 RAM (e.g. 62256): the high two address lines are don't-cares (not even connected). This means that the mapped addresses E0000, E8000, F0000, and F8000 will all access the same location in the RAM chip. The net effect is that the 32K RAM appears *four times* in the 127K address space. The first three "appearances" allow access to the full 32K of RAM, but the fourth "appearance" (F8000-FFBFF) is shortened by 1K for I/O.

8Kx8 RAM (e.g. 6264): the high three address lines are don't-cares; but MA13 is tied to the *active-high* chip enable input CE2 of the 8K RAM. This means that MA13 must be high to access this chip, and so the 8K RAM will appear *eight times* in the on-board RAM space, starting at addresses E2000, E6000, EA000, EE000, F2000, F6000, FA000, and FE000. Again, the last "appearance" of the RAM is shortened by 1K for I/O.

You can do the same trick with U20, and install a 16Kx8 or 8Kx8 EPROM (making sure that the unused address lines are *high*). But there's little advantage to this: 27256s are as cheap as 2764s these days, and — unlike the RAM — you can't re-map the unused EPROM address space to other devices.

With a 2 MHz 68B09 (8 MHz oscillator), you should use 200 nsec or faster memory chips. The 1 MHz 6809 can use slower parts...but it's getting hard to find parts slower than 200 nsec!

PARALLEL I/O AND TIMERS

Figure 1 also shows U22, a Zilog Z8536 Counter/Timer/Parallel I/O chip. This chip provides two 8-bit parallel ports, a 4-bit control port, plus three 16-bit counter/timers. Several similar "multifunction" devices exist, including the Rockwell 6522 and Intel 8256. After much deliberation, I selected the Zilog part based on ease of interfacing, low cost, availability, and flexibility (in roughly that order).

Recall that eight I/O selects, IO0\ to IO7\, are generated for the eight 128-byte regions in the 1K "on-board I/O" address space FFC00-FFFFF. Select line IO2\ is used here. The Z8536 uses only two address lines, A0 and A1, and thus needs only four bytes of the 128-byte region FFD00-FFD7F. (Since address lines A2-A6 are "don't-cares", the chip appears 32 times in the 128-byte region.) LCLIORD\ and LCLIOWR\ are the read and write signals for devices in the "on-board I/O" space.

The interrupt output from this chip, INT\, can be jumpered to either the NMI\ or the IRQ\ input of the 6809. The remaining interrupt signals on the Z8536 pertain to Zilog's interrupt daisy-chaining scheme, and can't be used with the 6809 (at least, not without a *lot* of extra work). IEI is Interrupt Enable In, and must be high for this chip to generate an interrupt. IEO is Interrupt Enable Out and can be ignored. INTACK\, when held low, causes the Z8536 to place an "interrupt vector" on the data bus. Since the 6809 can't use this interrupt vector, INTACK\ is simply tied high.

The Z8536 requires a clock signal for its operation. This clock need not be synchronized with the CPU, but the BUFOSC signal is convenient and just the right frequency. If you are using faster 6809s, with a 6 MHz or 8 MHz oscillator, make sure you buy a Z8536 of the right speed (6 or 8 MHz).

The parallel I/O pins are brought out to J7. The strange wiring of this connector is for ease of PCB layout.

EXPANSION BUS

It might be desirable to expand the I/O of a single processor, without going through the rigamarole of the multiprocessor bus. Perhaps you want one of the 6809s on the bus to be a SCSI server, or perhaps you're using this as a single-board computer (SBC). The expansion bus allows you to connect additional I/O devices to the "private bus" of the CPU.

Since this is intended exclusively for I/O chips, only the low 7 address lines are brought to the connector. Five of the I/O selects are brought out, so you can add five I/O chips with no extra decoding logic. You can use either the "Intel-style" RD\ and WR\, or the "Motorola-style" R/W\ and E, to control the I/O read and write operation. (Warning: many 65xx family peripheral chips will *not* work with this board, due to address timing problems.) The "add-on" I/O also has access to the CPU interrupt lines NMI\ IRQ\ and FIRQ\, RESET\, and the oscillator signal BUFOSC.

The best way to use this bus is to build a "daughterboard" that plugs directly onto the CPU card. Don't use a long piece of ribbon cable here; the added capacitance on the bus lines will foul everything up, even the memory and I/O on the CPU board. Also, you are limited to *one* LSTTL load on the E signal. In general, if the daughterboard is going to be at all complex, you should also buffer the address bus, data bus, and R/W\.

THE IBM PC BUS INTERFACE

Most multiprocessor busses — such as VME bus, Multibus, or Q-bus — are much too complicated for a simple educational project like this one. Originally I had intended to invent my own multiprocessor bus. However, after reading about a 6809 board using the IBM PC bus (TCJ #64 p.47), and discussions with various Interested Parties, I realized the advantages of being able to use IBM PC peripheral cards.

A bit of history: the original PCs all had 8-bit CPUs, and so allowed 8-bit peripherals to plug into 8-bit expansion slots through a 62-pin edge connector. The PC/AT, with its 16-bit CPU, needed a 16-bit bus; but IBM wanted to keep using the cornucopia of 8-bit plug-in cards. So they kept the 8-bit bus, and put the additional 8 data bits (plus some more address bits and extra control lines) on a second, 36-pin edge connector. This is why you see AT (286), 386, and 486 motherboards having a certain number of "8-bit" and "16-bit" expansion slots. The "8-bit" slots have one edge connector, and the "16-bit" slots have two. All of the old 8-bit peripherals work fine in a 16-bit system. (To confuse the issue, a *third* edge connector has now been invented for 32-bit peripherals.)

Since the 6809 is an 8-bit processor, I only desire to support the basic, 8-bit, PC bus. (There's no shortage of 8-bit perhipheral cards!) But this bus does not include control signals for multiprocessing, and leaves no pins unused. A separate processor arbitration bus is required. Rather than mess around with ribbon cables or custom edge connectors, I decided to use the second edge connector of the PC/AT-style cards for the arbitration bus. This allows me to use commonly available 16-bit "passive backplanes" for the multiprocessor system. (A "passive backplane" is a board which has only the edge connectors for the PC bus. The CPU and all of the rest of the motherboard electronics sit on a plug-in card. In theory, this lets you upgrade CPUs more easily.)

Important: the second edge connector is NOT used in accordance with the PC specification. You can NOT use 16-bit IBM PC cards with the ScroungeMaster II, and you must NOT plug the SM II into a 16-bit slot of a PC motherboard. Use ONLY 8-bit peripheral cards, and ONLY a passive backplane! Passive backplanes are available at swapmeets — I found several at the Trenton Computer Festival — and Alltech Electronics Co. (602 Garrison Street, Oceanside, CA 92054, phone 619-721-7733, fax 619-721-2823) has advertised 16-bit backplane boards for $10 in Nuts & Volts.

Figure 2 shows the bus logic. (For a complete explanation of the IBM PC bus signals, see the sidebar.) U16 through U19 are the tri-state bus drivers. These drivers can only be enabled when the signal DRIVENBL\ is low, which you will recall occurs only when a) this CPU is attempting to access an "external" address and b) the bus has been granted to this CPU.

U19 is a bidirectional driver for the data bus. During CPU Write cycles this outputs data from D0-D7 to the PC bus lines XD0-XD7. During CPU Read cycles, data from the peripheral card on XD0-XD7 is output to the 6809 bus D0-D7. The direction of transfer is controlled by the 6809's R/W\ signal, via the DIR input of the 74LS245.

U16, U17, and U18 buffer the address bus from the 6809. This is always output by the CPU, so these are unidirectional buffers. U18 also buffers the control lines IORD\, IOWR\, MEMRD\, and MEMWR\. Like the address lines, these are always output by the CPU and input by the peripheral cards. They must pass

through a 74LS244 buffer so that they can be disabled when another CPU "owns" the bus.

The *active-high* RESET line is input by all CPUs and peripherals. It is normally generated by the motherboard. Since we are using a passive backplane, it must be generated by a plug-in card. To simplify matters, any 6809 board can output this signal by installing JP1. Only *one* 6809 board (the "bus master") should have this jumper installed. U9A is a simple RC reset circuit; U9B inverts the active-high RESET to the active-low signal used on the 6809 board.

The CLK and OSC lines are also normally generated by the motherboard. CLK must be synchronous with the processor, but is otherwise rather vaguely defined (see sidebar); to meet this requirement, all of the 6809s use CLK as their (4, 6, or 8 MHz) oscillator signal. OSC is a 14.31818 MHz signal — a hangover from the days of 4.77 MHz XTs, useful for video

THE IBM PC BUS

The IBM PC bus is undoubtedly the most widely-used and poorly-documented computer bus in history. Much of my information comes from reading PC peripheral card schematics, the technical reference manual for my old XT clone, and the excellent book by Lewis Eggebrecht [EGG90] which I only discovered this year. This is a brief summary, starting on the component side of the connector:

IOCHK\ (I/O Check) can be pulled low by a peripheral card to signal an error condition. The ScroungeMaster II ignores it.

D0 through D7 are the data bus.

IORDY, when pulled low, stretches the memory or I/O access cycle. This is called WAIT\ on Z80s and MRDY on the 6809.

AEN (Address Enable) is output high by the PC motherboard to indicate that a DMA cycle is in progress. Since the SM II has no DMA, it ties this line low.

A0 through A19 are the 20-bit address bus.

RESET is the active-high reset line.

GND, +5, -5, +12, and -12 are, I hope, obvious.

IRQ2,3,4,5,6, and 7 are six active-high interrupt lines. Each of these can be pulled high by a peripheral card to generate a CPU interrupt. Normally each line can be used by only one IBM PC peripheral card. The PC has an eight-input interrupt controller chip; the SM II limits each CPU to handling one bus interrupt.

DRQ1,2, and 3 are three active-high DMA Request lines. Each of these can be pulled high by a peripheral card to request a Direct-Memory-Access data transfer. Normally each line can be used by only one PC peripheral card. The PC has a four-input DMA controller chip; the SM II does not support DMA.

DACK1\, DACK2\, and DACK3\ are the active-low acknowledge lines for DRQ1, 2, and 3. These are not supported by the SM II.

DACK0\ on IBM PCs and PC/XTs indicates that a dynamic RAM refresh is taking place. (The 8-bit PCs used one DMA channel as a cheap & simple refresh controller.) This is not supported by the SM II.

SMEMR\ and SMEMW\ are the memory read and write strobes. When active (low), they indicate that a valid 20-bit address is on the bus, and a data transfer can take place.

IOR\ and IOW\ are the I/O read and write strobes. When active (low), they indicate that a valid *10-bit* address (A0-A9) is on the bus, and a data transfer can take place. Intel and Zilog CPUs distinguish between I/O and memory references; and in the IBM PC, only 10 address bits are used for I/O. For the 6809 a 1K address space is designated "I/O" and the IOR\ and IOW\ signals are generated accordingly. Note that a bus cycle can be a memory *or* an I/O cycle, but not both — thus, if IOR\ is low (active), SMEMR\ had better be high.

CLK is a clock signal derived from the 8088 support chips, and really not defined any better than that. It is synchronous with the CPU; its exact timing specs are prohibitively complex to emulate with the 6809. Fortunately, it's rarely important.

T/C (Terminal Count) when high, indicates that a DMA operation is complete. This is not supported by the SM II.

ALE (Address Latch Enable) when high, indicates when the address is being output by the 8088 CPU. Like CLK, it is difficult to emulate on the 6809, and rarely used.

OSC is a 14.31818 MHz signal used for old (CGA) video cards and not much else. It is not synchronized to the CPU clock.

For more details, I highly recommend Eggebrecht's book.

display circuits but not much else. CLK and OSC are produced by one designated "master" 6809 board, by the simple expedient of installing Y1 and Y2 *only* on that board. U7F buffers CLK for each 6809 board, since there are many loads placed on this line.

The PC bus has six *active-high* interrupt lines, IRQ2-IRQ7. For simplicity, I have decreed that each 6809 board can handle *at most one PC bus interrupt line.* Thus, to handle all PC interrupts, you would need six 6809 CPUs. Jumper JP16 selects which of the interrupt lines this CPU will respond to as the "external interrupt" XIRQ. This is then inverted (see the CPU schematic) and may be routed to the CPU's IRQ input, or a "pseudo interrupt" on an I/O chip.

DMA is not supported by the ScroungeMaster II. The complications of DMA and multiprocessor arbitration are more than I wanted to attempt for this project, and the CPU board was big enough already! So, the DMA control lines DRQ2, DACK0\, DACK1\, DACK2\, and DACK3\ are all ignored. For experimentation I allow the DMA request lines DRQ1 and DRQ3 to be jumpered as interrupt inputs to the 6809. This will not cause an electrical conflict with PC peripheral cards, but you *cannot* use the DMA function of those cards, since the required DACKn\ acknowledge signal is not generated. (Maybe some TCJ reader will design a plug-in DMA controller board.)

You also cannot use plug-in dynamic RAM boards. Most of these depend upon the DACK0\ signal for refresh timing.

The IORDY (a.k.a. WAIT\) signal is routed directly to the wait-control logic, as described in the previous installment.

Strictly speaking, pullup resistors should be installed on the motherboard for the signals IORD\, IOWR\, MEMRD\, and MEMWR\. This is so they won't be inadvertently asserted (pulled low) when all of the CPUs are "off the bus." (I should have put an optional pullup on the 6809 board; sorry, I forgot.) Likewise, the interrupt lines should have weak pulldowns, so that "missing" peripheral cards don't produce an interrupt. This can be handled by removing JP4 when the selected external interrupt line is not in use.

BUS ARBITER

U31 (on Figure 1), and U28 (on Figure 2), constitute a "round-robin" bus arbiter. U31 is a free-running three-bit counter which is present *on the "bus master" CPU board only.* It runs at the BUFOSC rate (synchronous with the 6809s) and counts 0,1,2,3,4,5,6,7,0,1,2,3... in binary.

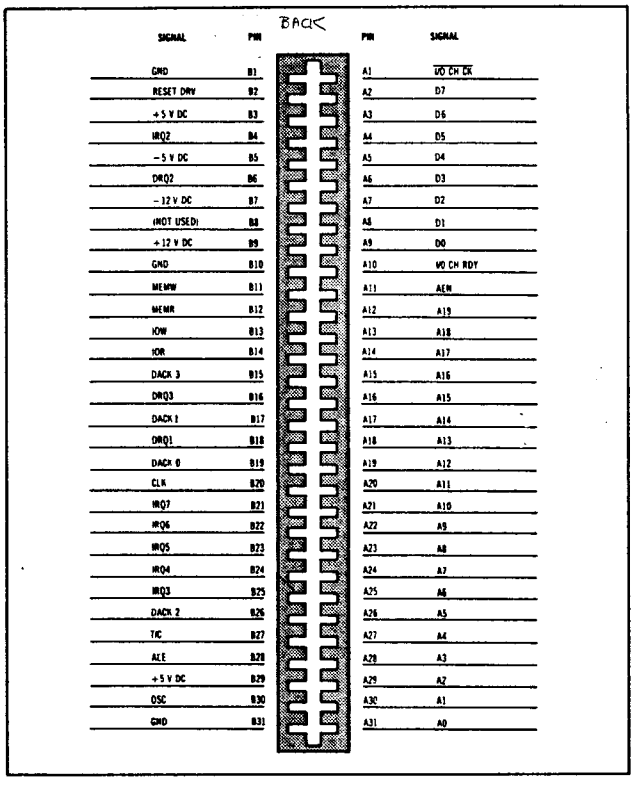
Each of the eight CPUs which can be installed in a system is assigned a unique number from 0 to 7, by installing the corresponding jumper in JP17. When the counter reaches that number, *and* that CPU is requesting the bus (REQ is high), the corresponding output of U28 will go low, and thus GRANT\ will go low.

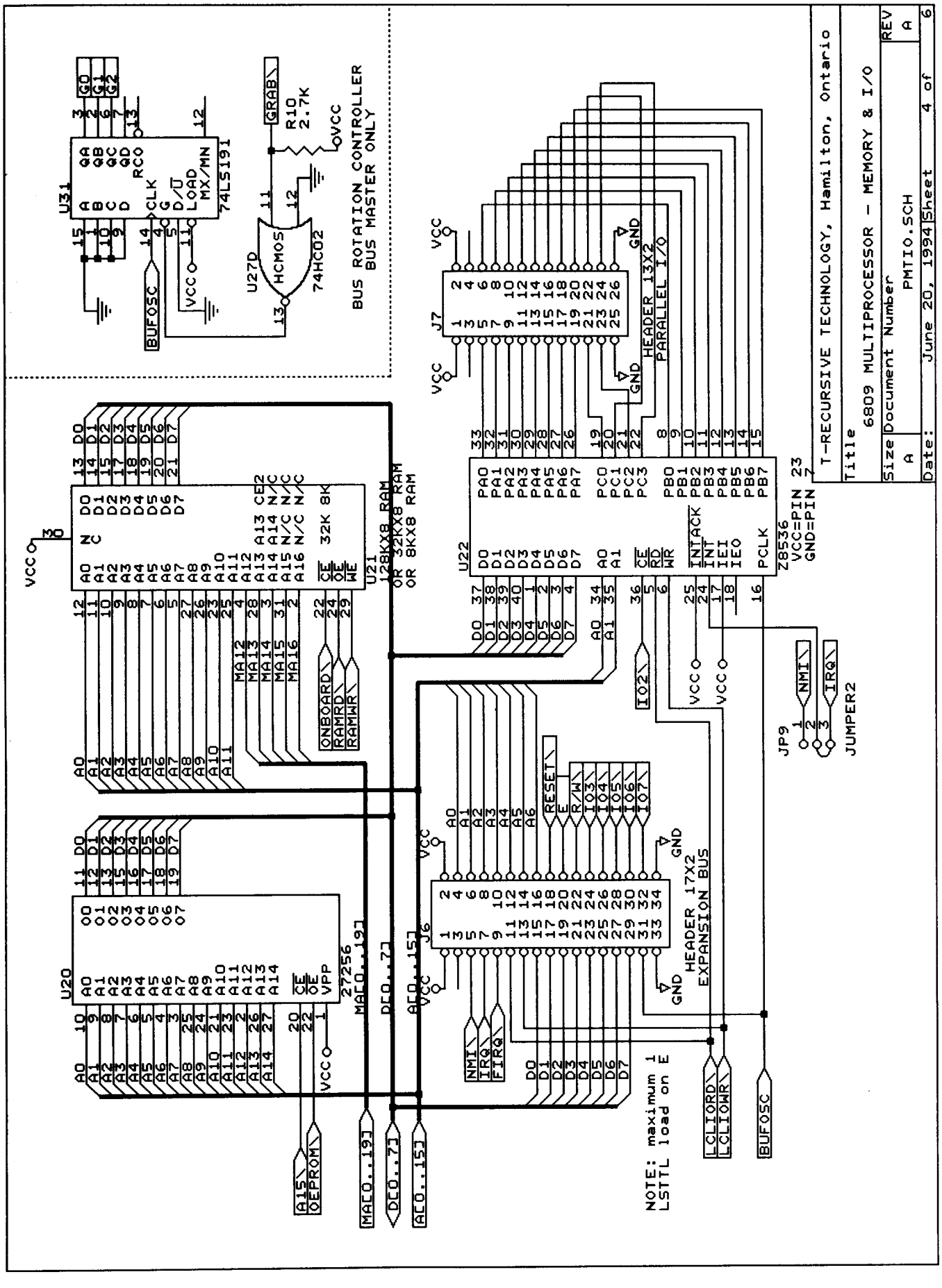
This low signal is also "diode-OR'ed" onto the GRAB\ line on the arbitration bus. (The diode, and pullup resistor R10, are necessary because all the other CPUs are trying to pull this line high.) Any CPU can pull GRAB\ low, to indicate that it has "grabbed" the bus. This halts the counter U31 (via inverter U27D), which "freezes" the count at the CPU number which now "owns" the bus. The counter will remain frozen until that CPU de-asserts (pulls low) its REQ line, signalling that it is done with the bus. Then all the outputs of U28 go high, and the count continues.

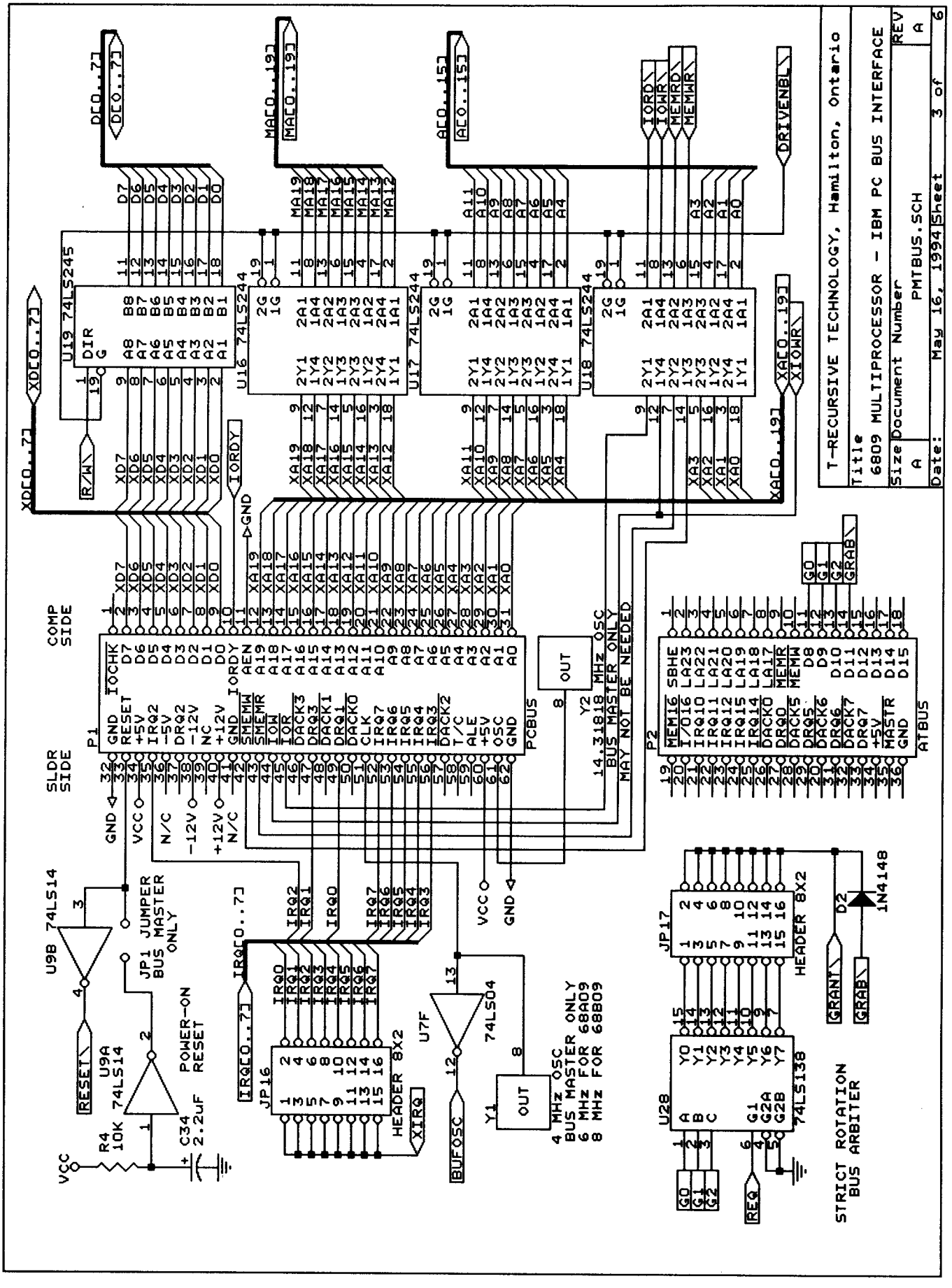
No matter what order the CPUs make their requests, they will always be granted the bus in strict rotation. This satisfies the requirement (mentioned in the previous installment) that no one CPU can wait more than 15 usec for the bus. *Remember that U31 is installed in only one CPU board; U28 is installed in every CPU board.*

REFERENCES

[EGG90] Eggebrecht, Lewis C., *Interfacing to the IBM Personal Computer*, Second Edition, SAMS, 17111 North College, Carmel, Indiana, 46032 (1990), ISBN 0-672-22722-3. The best book I've found for descriptions of the IBM PC and PC/AT busses. Available from JDR Microdevices, 2233 Samaritan Drive, San Jose, CA 95124, phone 408-559-1200, fax 408-559-0250, if you can't find it locally.




T-RECURSIVE TECHNOLOGY, Hamilton, Ontario
6809 MULTIPROCESSOR - MEMORY & I/O
PMTIO.SCH
June 20, 1994
4 of 6
BUS ROTATION CONTROLLER
BUS MASTER ONLY
74LS191
HCMOS
74HC02
HEADER 13X2
PARALLEL I/O
Z8536
128KX8 RAM
OR 32KX8 RAM
OR 8KX8 RAM
27256
HEADER 17X2
EXPANSION BUS
JUMPER2
NOTE: maximum 1
LSTTL load on E


T-RECURSIVE TECHNOLOGY, Hamilton, Ontario
Title
6809 MULTIPROCESSOR - IBM PC BUS INTERFACE
Size A
Document Number
PMTBUS.SCH
REV A
Date: May 16, 1994 Sheet 3 of 6
U19 74LS245
U16 74LS244
U17 74LS244
U18 74LS244
P1
P2
PCBUS
ATBUS
U9B 74LS14
U9A 74LS14
JP1 JUMPER BUS MASTER ONLY
POWER-ON RESET
JP16 HEADER 8X2
U7F 74LS04
4 MHz OSC BUS MASTER ONLY
6 MHz FOR 68A09
8 MHz FOR 68B09
JP17 HEADER 8X2
U28 74LS138
STRICT ROTATION BUS ARBITER
1N4148

Regular Feature

Group Reviews

SUPPORT GROUPS FOR THE CLASSICS

GiMiX lives!

Some time back I gave the impression that GIMIX systems had ceased to be. Several others had confirmed that thought. Yet, several weeks ago I got one of those packs of post cards advertising computer products. Sure enough there was a card from GMX saying they had the trademark for GIMIX and GHOST products.

Could this be the same GIMIX of past, surely not. A simple phone call puts me right and to shame. I talked with Robert Philips, the original owner and starter of GIMIX. We tried to determine how so many people could be wrong about them not being in business. Robert indicated that his partner of some time (Richard Don, I think that is the correct spelling, but not sure) retired about three years ago, and may have been the source to the rumor. Richard did mostly the business and marketing side and thus most people would have dealt with him, as I did once.

Robert has great records of all his deals and was able to pull my name and product serial number up in quick order (wish my mailing program worked as good as that!) When asked if they support older systems, the answer was YES! In fact the company has been able to retain many of the original technicians that have been there from 1975 (not many companies in the computer industry can boast of that record.)

On parts, he has a complete list and can tell you if the one you want is available, the finding may take longer as some have been boxed and stored safely away. They have many of the older 6800 and 6809 systems still around and running, and for a fee could probably fix any of the 6800/6809 products ever produced. Robert indicated that he has thrown very little away over the years, and in fact has complete sets of magazines neatly stored away. So if you really need to find a reprint of some article, give them a call.

One problem for the misinformation about GMX is that they did move a few years back to a suburb of Chicago, but Robert still has the old phone number forwarded to his new location. I tried information last year, and was unable to get any help, but possibly I used GIMIX or some such, and it must be GMX? Anyway you look at it, they have been supporting the old systems and their new line of 68K products non stop from 1975.

Their new line is mostly 68030/040 based EISA bus systems, with a new line of 68340/349 systems for embedded usage in the works. Their main operating system is still OS-9, and in fact they supply Microwave with all their development hardware, even for the Multi-Media CDI project (and to Philips as well!) Their regular base of users has changed lately and thus, the post card is an attempt by them to get the word out that they still are producing good systems. Their new line will be in the $500 price range with too many features to list.

So if you have a GIMIX, or SwTPC, or any 6800/6809/68K system needing support and information, call Robert Philips at GMX, 3223 Arnold Lane, Northbrook, IL 60062, (800) 559-0909.

Zed-FEST EUROPE/GEnie

One treat GEnie users had was to talk on-line with the Zed-FEST participants. That happened on Saturday, June 18. The heavily edited transcript that I captured is below. I am sure a more accurate and complete version is on GEnie (#8417).

LIVE FROM GERMANY - 3rd EUROPEAN Zed-Fest

Room 1, The General Club room.
** <B.KIBLER> is here.
<[C=64/128 Out]> What is the Club-80 terminal? Is it something we can use on the 128? If so what features are there?
<[Helmut Jungk]> The CLUB-80 terminal is for use on ECB-bus machines like CPU280, Prof80 also
<BW.MILLER> Does Europe have easy access to Internet?
<[Helmut Jungk] JAY.SAGE> Europe is not a person
<BW.MILLER> (for FTP?)
<[Helmut Jungk]> {ftp is available to all the people with an Internet agreement for this action. Wim Nelis here is our man to ftp stuff. What do you have? Hello to Bill Kibler!
<BW.MILLER> Jeff-CPM was just online a few minutes ago, he runs one of the CP/M repositories, plus is our software librarian here on GEnie. Helmut, I wasn't sure how many europeans using CP/M have access to internet.
<B.KIBLER> Hello Jay/Helmut and all, am a Bit late but hope all is going well...
<BW.MILLER> It's getting easier and easier to access internet here in the states, wasn't sure about people using it in europe.
<[Helmut Jungk]> In Germany, it is easier to get mail access to Internet. Well, Bill you were cutting my picture as well, ey?
<B.KIBLER> oh yes your pictures... seems that PageMaker 4

doesn't like some of the pictures, they just come out black..solid black.
<[Ludo] VANHEMELRYCK> Jan, Can you update me on UZI?
<[Jan Taalman] JAY.SAGE> I have not seen it yet for myself, but plan to do so very soon
<[Jan Taalman]> It is indeed an upgrade of the stuff doug braun wrote.
<[C=64/128 Out]> What is 'UZI' is it something usable on the 128? What does it do?
<[Ken] K.OWEN2> Would this be pheasible on a Z180 with a Meg of RAM?
<[Jan Taalman]> Because of hardware restrictions, chances are small it will run on an 128. Z180 stands a lot better chance though.
<[C=64/128 Out]> ok just what is it then? I have no clue. like I said I'm VERY new to CP/M
<[Michael] M.CRAFTON> Is it possible to up grade my Televideo systems to run 'UZI'?
<[Jan Taalman]> UZI is a complete unixlike user interface, file system and kernel for the z80 and up environments
<B.KIBLER> Jan .. when has this new work on UZI, unix like replacement for cp/m based machines, been started and by whom and what is the ultimate objective (just bug free or what)...
<BW.MILLER> Is UZI ready for release, fairware, shareware, commercial?
<[Jan Taalman]> Televideo will have a CPU problem, according to the people "in the know" here.
<[Jan Taalman]> The original UZI has been around for several years, but rarely seen in a running state. The Z280 version is shareware
<[Ludo] VANHEMELRYCK> I think Doug Braun (UZI) is in Israel right now?...
<[Jan Taalman]> Yep. to the best of my info he is. He has internet access there but i would have to look it up
<[Ludo] VANHEMELRYCK> I do have Braun's address..
<[Jan Taalman]> We need to get regular file update info, the zsus service has not been very active lately, any suggestions ?
<[Ludo] VANHEMELRYCK> Waiting for the CP/M-CDRom
<B.KIBLER> ok just found UZI review of sorts in issue #58 of TCJ...
<B.KIBLER> Last time I talked to Walnut Creek CD ROM folks the CPM CDROM was in the works, now almost a year since he got my stuff...
<[Michael] M.HURST> apprently they are selling it, I have seen messages about this
<[Michael] M.CRAFTON> Yes, I've been waiting for that CP/M CDROM!
<[Jan Taialman]> I believe the cdrom has arrived when i hold it in my hands, and not one minute earlier.<[Beery] BW.MILLER> Each time I talk to Walnut creek, they say one more month.
<[Michael] M.HURST> wow I thought from what i read it was out!!
<B.KIBLER> CPM CDROM STILL not available, I just called them..
<[Jay] JAY.SAGE> Bill, did they give another estimate
<B.KIBLER> NO estimate in fact people on phone had no idea what I was talking about..will have to call Brian later in the week and talk to him as he is the one sitting on the source material.
<[Jay] JAY.SAGE> I was told several weeks ago that the CPM CDROM was now fully organized and should be ready for production soon.
<B.KIBLER> some time ago they told me that it was finished and it would only take about a week to start making them..but I wonder...
<[Jay] JAY.SAGE> {Actually, we are going to have to sign off soon. There is
another kind of festival we are planning to attend — one with BEER! It is now almost 10 pm here, so we have to be going soon.
<[Beery] BW.MILLER> OK Jay, what's the drink of the city where you are at?
<[Jay] JAY.SAGE> Actually, I guess there is no local beer. This is a wine-growing area. But I think we will be drinking beer. Since there are about 14 of us here, we can cover all of you by drinking one extra. We will be glad to comply!
<[Ken] K.OWEN2> Thanks

And thank you Jay and all the GEnie participants. Bill Kibler.

TCJ Staff Contacts

TCJ Editor:
Bill D. Kibler, PO Box 535, Lincoln, CA 95648, (916)645-1670, GEnie: B.Kibler, CompuServe: 71563,2243, E-mail: B.Kibler@Genie.geis.com.

Z-System Support:
Jay Sage,1435 Centre St. Newton Centre, MA 02159-2469, (617)965-3552, BBS: (617)965-7259; E-mail: Sage@ll.mit.edu. Also sells Z-System software.

32Bit Support:
Rick Rodman, BBS:(703)330-9049 (eves), E-mail: rickr@virtech.vti.com.

Kaypro Support:
Charles Stafford, 4000 Norris Ave., Sacramento, CA 95821, (916)483-0312 (eves). Also sells Kaypro upgrades, see ad inside back cover.

S-100 Support:
Herb Johnson, CN 5256 #105, Princeton, NJ 08543, (609)771-1503. Also sells used S-100 boards and systems, see inside back cover.

6809 Support:
Ronald Anderson, 3540 Sturbridge Ct., Ann Arbor, MI 48105.

Users Groups and Project Reports:
JW Weaver, Drawer 180, Volcano, CA 95689, BBS: (916)427-9038.

Regular Contributors:
Dave Baldwin, Voice/FAX (916)722-3877, or DIBs BBS (916) 722-5799 (use "computer", "journal", pswd "subscriber" as log on), Internet dibald@netcom.com, CompuServe 70403,2444.

Brad Rodriguez,Box 77, McMaster Univ., 1280 Main St. West, Hamilton, ONT, L8S 1C0, Canada, Genie: B.Rodriguez2, E-mail: b.rodriguez2@genie.geis.com.

Frank Sergeant, 809 W. San Antonio St., San Marcos, TX 78666, E-mail: fs07675@academia.swt.edu.

Tilmann Reh, Germany, E-mail: tilmann.reh@hrz.uni-siegen.d400.de. Has many programs for CP/M+ and is active with Z180/280 ECB bus/Modular/Embedded computers. USA contact Jay Sage.

Helmut Jungkunz, Munich, Germany, ZNODE #51, 8N1, 300-14.4, +49.89.9614574, or CompuServe 100024,1545.

Ron Mitchell, GEnie as R.Mitchell31, or CompuServe 70323,2267.

USER GROUPS

Connecticut CP/M Users Group, contact Stephen Griswold, PO Box 74, Canton CT 06019-0074, BBS: (203)665-1100. Sponsors East Coast Z-fests.

Sacramento Microcomputer Users Group, PO Box 161513, Sacramento, CA 95816-1513, BBS: (916)372-3646. Publishes newsletter, $15.00 membership, normal meeting is first Thursday at SMUD 6201 S st., Sacramento CA.

CAPDUG: The Capital Area Public Domain Users Group, Newsletter $20, Al Siegel Associates, Inc., PO Box 34667, Betherda MD 20827. BBS (301) 292-7955.

NOVAOUG: The Northern Virginia Osborne Users Group, Newsletter $12, Robert L. Crities, 7512 Fairwood Lane, Falls Church, VA 22046. Info (703) 534-1186, BBS use CAPDUG's.

The Windsor Bulletin Board Users' Group: England, Contact Rodney Hannis, 34 Falmouth Road, Reading, RG2 8QR, or Mark Minting, 94 Undley Common, Lakenheath, Brandon, Suffolk, IP27 9BZ, Phone 0842-860469 (also sells NZCOM/Z3PLUS).

L.I.S.T.: Long Island Sinclair and Timex support group, contact Harvey Rait, 5 Peri Lane, Valley Stream, NY 11581.

Coleco ADAM:
ADAM-Link User's Group, Salt Lake City, Utah, BBS: (801)484-5114. Supporting Coleco ADAM machines, with Newsletter and BBS.

Adam International Media, Adam's House, Route 2, Box 2756, 1829-1 County Rd. 130, Pearland TX 77581-9503, (713)482-5040. Contact Terry R. Fowler for information.

AUGER, Emerald Coast ADAM Users Group, PO Box 4934, Fort Walton Beach FL 32549-4934, (904)244-1516. Contact Norman J. Deere, treasurer and editor for pricing and newsletter information.

MOAUG, Metro Orlando Adam Users Group, Contact James Poulin, 1146 Manatee Dr. Rockledge FL 32955, (407)631-0958.

Metro Toronto Adam Group, Box 165, 260 Adelaide St. E., Toronto, ONT M5A 1N0, Canada, (416)424-1352.

Omaha ADAM Users Club, Contact Norman R. Castro, 809 W. 33rd Ave. Bellevue NE 68005, (402)291-4405. Suppose to be oldest ADAM group.

Vancouver Island Senior ADAMphiles, ADVISA newsletter by David Cobley, 17885 Berwick Rd. Qualicum Beach, B.C., Canada V9K 1N7, (604)752-1984.

Northern Illiana ADAMS User's Group, 9389 Bay Colony Dr. #3E, Des Plaines IL 60016, (708)296-0675.

OS-9 Support:
San Diego OS-9 Users Group, Contact Warren Hrach (619)221-8246, BBS: (619)224-4878.

Atari Support:
ACCESS, PO Box 1354, Sacramento, CA 95812, Contact Bob Drews (916)423-1573. Meets first Thurdays at SMUD 59Th St. (ed. bldg.).

Forth Support:
Forth Interest Group, PO Box 2154, Oakland CA 94621 510-89-FORTH. International support of the Forth language. Contact for list of local chapters.

OTHER PUBLICATIONS

The Z-Letter, supporting Z-System and CP/M users. David A.J. McGlone, Lambda Software Publishing, 149 West Hillard Lane, Eugene, OR 97404-3057, (503)688-3563. Bi-Monthly user oriented newsletter (20 pages+). Also sells CP/M Boot disks, software.

The Analytical Engine, by the Computer History Association of California, 1001 Elm Ct. El Cerrito, CA 94530-2602. A ASCII text file distributed by Internet, issue #1 was July 1993. E-mail: kcrosby@crayola.win.net.

Z-100 LifeLine, Steven W. Vagts, 2409 Riddick Rd. Elizabeth City, NC 27909, (919)338-8302. Publication for Z-100 (a S-100 machine).

The Staunch 8/89'er, Kirk L. Thompson editor, PO Box 548, West Branch IA 52358, (319)643-7136. $15/yr(US) publication for H-8/89s.

The SEBHC Journal, Leonard Geisler, 895 Starwick Dr., Ann Arbor MI 48105, (313)662-0750. Magazine of the Society of Eight-Bit Heath computerists, H-8 and H-89 support.

Sanyo PC Hackers Newsletter, Victor R. Frank editor, 12450 Skyline Blvd. Woodside, CA 94062-4541, (415)851-7031. Support for orphaned Sanyo computers and software.

the world of 68' micros, by FARNA Systems, PO Box 321, Warner Robins, GA 31099-0321. E-mail: dsrtfox@delphi.com. New magazine for support of old CoCo's and other 68xx(x) systems.

Amstrad PCW SIG, newsletter by Al Warsh, 2751 Reche Cyn Rd. #93, Colton, CA 92324. $9 for 6 bi-monthly newsletters on Amstrad CP/M machines.

Historically Brewed, A publication of the Historical Computer Society. Bimonthly at $18 a year. HCS, 10928 Ted Williams PL., El Paso, TX 79934. Editor David Greelish. Computer History and more.

Other Support Businesses

Hal Bower writes, sells, and supports B/PBios for Ampro, SB180, and YASBEC. $69.95. Hal Bower, 7914 Redglobe Ct., Severn MD 21144-1048, (410)551-5922.

Sydex, PO Box 5700, Eugene OR 97405, (503)683-6033. Sells several CP/M programs for use with PC Clones ('22Disk' format/ copies CP/M disks using PC files system).

Elliam Associates, PO Box 2664, Atascadero CA 93423, (805)466-8440. Sells CP/M user group disks and Amstrad PCW products. See ad inside back cover.

Discus Distribution Services, Inc. sells CP/M for $150, CBASIC $600, Fortran-77 $350, Pascal/MT+ $600. 8020 San Miguel Canyon Rd., Salinas CA 93907, (408)663-6966.

Microcomputer Mail-Order Library of books, manuals, and periodicals in general and H/Zenith in particular. Borrow items for small fees. Contact Lee Hart, 4209 France Ave. North, Robbinsdale MN 55422, (612)533-3226.

Star Technology, 900 Road 170, Carbondale CO, 81623. Epson QX-10 support and repairs. New units also avialble.

Star-K Software Systems Corp. PO Box 209, Mt. Kisco, NY 10549, (914)241-0287, BBS: (914)241-3307. 6809/68000 operating system and software. Some educational products, call for catalog.

Peripheral Technology, 1250 E. Piedmont Rd., Marietta, GA 30067, (404)973-2156. 6809/68000 single board system. 68K ISA bus compatible system. See inside front cover.

Hazelwood Computers, RR#1, Box 36, Hwy 94@Bluffton, Rhineland, MO 65069, (314)236-4372. Some SS-50 6809 boards and new 68000 systems.

AAA Chicago Computers, Jerry Koppel, (708)202-0150. SS-50 6809 boards and systems. Very limited quanity, call for information.

MicroSolutions Computer Products, 132 W. Lincoln Hwy, DeKalb, IL 60115, (815)756-3411. Make disk copying program for CP/M systems, that runs on CP/M sytems, UNIFROM Format-translation. Also PC/Z80 CompatiCard and UniDos products.

GIMIX/OS-9, GMX, 3223 Arnold Lane, Northbrook, IL 60062, (800)559-0909, (708)559-0909, FAX (708)559-0942. Repair and support of new and old 6800/6809/68K/SS-50 systems.

n/SYSTEMS, Terry Hazen, 21460 Bear Creek Rd, Los Gatos CA 95030-9429, (408)354-7188, sells and supports the MDISK add-on RAM disk for the Ampro LB. PCB $29, assembled PCB $129, includes driver software, manual.

Corvatek, 561 N.W. Van Buren St. Corvallis OR 97330, (503)752-4833. PC style to serial keyboard adapter for Xerox, Kaypros, Franklin, Apples, $129. Other models supported.

Morgan, Thielmann & Associates services NON-PC compatible computers including CP/M as well as clones. Call Jerry Davis for more information (408) 972-1965.

Jim S. Thale Jr., 1150 Somerset Ave., Deerfield IL 60015-2944, (708)948-5731 Sells I/O board for YASBEC. Adds HD drives, 2 serial, 2 parallel ports. Partial kit $150, complete kit $210.

Trio Comapny of Cheektowaga, Ltd., PO Box 594, Cheektowaga NY 14225, (716)892-9630. Sells CP/M (& PC) packages: InfoStar 1.5 ($160); SuperSort 1.6 ($130), and WordStar 4.0 ($130).

Parts is Parts, Mike Zinkow, 137 Barkley Ave., Clifton NJ 07011-3244, (201)340-7333. Supports Zenith Z-100 with parts and service.

When calling for support, say you saw their name in

The Computer Journal

The Computer Journal

Back Issues

Sales limited to supplies in stock.

Volume Number 1:

- **Issues 1 to 9**
- **Serial interfacing and Modem transfers**
- **Floppy disk formats, Print spooler.**
- **Adding 8087 Math Chip, Fiber optics**
- **S-100 HI-RES graphics.**
- **Controlling DC motors, Multi-user column.**
- **VIC-20 EPROM Programmer, CP/M 3.0.**
- **CP/M user functions and integration.**

Volume Number 2:

- **Issues 10 to 19**
- **Forth tutorial and Write Your Own.**
- **68000 CPU for S-100.**
- **RPM vs CP/M, BIOS Enhancements.**
- **Poor Man's Distributed Processing.**
- **Controlling Apple Stepper Motors.**
- **Facsimile Pictures on a Micro.**
- **Memory Mapped I/O on a ZX81.**

Volume Number 3:

- **Issues 20 to 25**
- Designing an 8035 SBC
- Using Apple Graphics from CP/M
- Soldering & Other Strange Tales
- Build an S-100 Floppy Disk Controller: WD2797 Controller for CP/M 68K
- Extending Turbo Pascal: series
- Unsoldering: The Arcane Art
- Analog Data Acquisition & Control: Connecting Your Computer to the Real World
- Programming the 8035 SBC
- NEW-DOS: series
- Variability in the BDS C Standard Library
- The SCSI Interface: series
- Using Turbo Pascal ISAM Files
- The Ampro Little Board Column: series
- C Column: series
- The Z Column: series
- The SCSI Interface: Introduction to SCSI
- Editing the CP/M Operating System
- INDEXER: Turbo Pascal Program to Create an Index
- Selecting & Building a System
- Introduction to Assemble Code for CP/M
- Ampro 186 Column
- ZTime-1: A Real Time Clock for the Ampro Z-80 Little Board

Volume Number 4:

- **Issues 26 to 31**
- Bus Systems: Selecting a System Bus
- Using the SB180 Real Time Clock
- The SCSI Interface: Software for the SCSI Adapter
- Inside Ampro Computers
- NEW-DOS: The CCP Commands (continued)
- ZSIG Corner
- Affordable C Compilers
- Concurrent Multitasking: A Review of DoubleDOS
- 68000 TinyGiant: Hawthorne's Low Cost 16-bit SBC and Operating System
- The Art of Source Code Generation: Disassembling Z-80 Software
- Feedback Control System Analysis: Using Root Locus Analysis & Feedback Loop Compensation
- The C Column: A Graphics Primitive Package
- The Hitachi HD64180: New Life for 8-bit Systems
- ZSIG Corner: Command Line Generators and Aliases
- A Tutor Program in Forth: Writing a Forth Tutor in Forth
- Disk Parameters: Modifying the CP/M Disk Parameter Block for Foreign Disk Formats
- Starting Your Own BBS
- Build an A/D Converter for the Ampro Little Board
- HD64180: Setting the Wait States & RAM Refresh using PRT & DMA
- Using SCSI for Real Time Control
- Open Letter to STD Bus Manufacturers
- Patching Turbo Pascal
- Choosing a Language for Machine Control
- Better Software Filter Design
- MDISK: Adding a 1 Meg RAM Disk to Ampro Little Board, Part 1
- Using the Hitachi hd64180: Embedded Processor Design
- 68000: Why use a new OS and the 68000?
- Detecting the 8087 Math Chip
- Floppy Disk Track Structure
- Double Density Floppy Controller
- ZCPR3 IOP for the Ampro Little Board
- 3200 Hackers' Language
- MDISK: Adding a 1 Meg RAM Disk to Ampro Little Board, Part 2
- Non-Preemptive Multitasking
- Software Timers for the 68000
- Lilliput Z-Node
- Using SCSI for Generalized I/O
- Communicating with Floppy Disks: Disk Parameters & their variations
- XBIOS: A Replacement BIOS for the SB180
- K-OS ONE and the SAGE: Demystifying Operating Systems
- Remote: Designing a Remote System Program
- The ZCPR3 Corner: ARUNZ Documentation

Issue Number 32:

- TEMPORARILY OUT OF STOCK -

Issue Number 33:

- Data File Conversion: Writing a Filter to Convert Foreign File Formats
- Advanced CP/M: ZCPR3PLUS & How to Write Self Relocating Code
- DataBase: The First in a Series on Data Bases and Information Processing
- SCSI for the S-100 Bus: Another Example of SCSI's Versatility
- A Mouse on any Hardware: Implementing the Mouse on a Z80 System
- Systematic Elimination of MS-DOS Files: Part 2, Subdirectories & Extended DOS Services
- ZCPR3 Corner: ARUNZ Shells & Patching WordStar 4.0

Issue Number 34:

- Developing a File Encryption System.
- Database: A continuation of the data base primer series.
- A Simple Multitasking Executive: Designing an embedded controller multitasking executive.
- ZCPR3: Relocatable code, PRL files, ZCPR34, and Type 4 programs.
- New Microcontrollers Have Smarts: Chips with BASIC or Forth in ROM are easy to program.
- Advanced CP/M: Operating system extensions to BDOS and BIOS, RSXs for CP/M 2.2.
- Macintosh Data File Conversion in Turbo Pascal.

Issue Number 35:

- All This & Modula-2: A Pascal-like alternative with scope and parameter passing.
- A Short Course in Source Code Generation: Disassembling 8088 software to produce modifiable assem. source code.
- Real Computing: The NS32032.
- S-100: EPROM Burner project for S-100 hardware hackers.
- Advanced CP/M: An up-to-date DOS, plus details on file structure and formats.
- REL-Style Assembly Language for CP/M and Z-System. Part 1: Selecting your assembler, linker and debugger.

Issue Number 36:

- Information Engineering: Introduction.
- Modula-2: A list of reference books.
- Temperature Measurement & Control: Agricultural computer application.
- ZCPR3 Corner: Z-Nodes, Z-Plan, Amstrand computer, and ZFILE.
- Real Computing: NS32032 hardware for experimenter, CPUs in series, software options.
- SPRINT: A review.
- REL-Style Assembly Language for CP/M & ZSystems, part 2.
- Advanced CP/M: Environmental programming.

Issue Number 37:

- C Pointers, Arrays & Structures Made Easier: Part 1, Pointers.
- ZCPR3 Corner: Z-Nodes, patching for NZCOM, ZFILER.
- Information Engineering: Basic Concepts: fields, field definition, client worksheets.
- Shells: Using ZCPR3 named shell variables to store date variables.
- Resident Programs: A detailed look at TSRs & how they can lead to chaos.
- Advanced CP/M: Raw and cooked console I/O.
- Real Computing: The NS 32000.
- ZSDOS: Anatomy of an Operating System: Part 1.

Issue Number 38:

- C Math: Handling Dollars and Cents With C.
- Advanced CP/M: Batch Processing and a New ZEX.
- C Pointers, Arrays & Structures Made Easier: Part 2, Arrays.
- The Z-System Corner: Shells and ZEX, new Z-Node Central, system security under Z-Systems.
- Information Engineering: The portable Information Age.
- Computer Aided Publishing: Introduction to publishing and Desk Top Publishing.
- Shells: ZEX and hard disk backups.
- Real Computing: The National Semiconductor NS320XX.
- ZSDOS: Anatomy of an Operating System, Part 2.

Issue Number 39:

- Programming for Performance: Assembly Language techniques.
- Computer Aided Publishing: The Hewlett Packard LaserJet.
- The Z-System Corner: System enhancements with NZCOM.
- Generating LaserJet Fonts: A review of Digi-Fonts.
- Advanced CP/M: Making old programs Z-System aware.
- C Pointers, Arrays & Structures Made Easier: Part 3: Structures.
- Shells: Using ARUNZ alias with ZCAL.
- Real Computing: The National Semiconductor NS320XX.

Issue Number 40:

- Programming the LaserJet: Using the escape codes.
- Beginning Forth Column: Introduction.
- Advanced Forth Column: Variant Records and Modules.
- LINKPRL: Generating the bit maps for PRL files from a REL file.
- WordTech's dBXL: Writing your own custom designed business program.
- Advanced CP/M: ZEX 5.0×The machine and the language.
- Programming for Performance: Assembly language techniques.
- Programming Input/Output With C: Keyboard and screen functions.
- The Z-System Corner: Remote access systems and BDS C.
- Real Computing: The NS320XX

Issue Number 41:

- Forth Column: ADTs, Object Oriented Concepts.
- Improving the Ampro LB: Overcoming the 88Mb hard drive limit.
- How to add Data Structures in Forth
- Advanced CP/M: CP/M is hacker's haven, and Z-System Command Scheduler.
- The Z-System Corner: Extended Multiple Command Line, and aliases.
- Programming disk and printer functions with C.
- LINKPRL: Making RSXes easy.
- SCOPY: Copying a series of unrelated files.

Issue Number 42:

- Dynamic Memory Allocation: Allocating memory at runtime with examples in Forth.
- Using BYE with NZCOM.
- C and the MS-DOS Screen Character Attributes.
- Forth Column: Lists and object oriented Forth.
- The Z-System Corner: Genie, BDS Z and Z-System Fundamentals.
- 68705 Embedded Controller Application: An example of a single-chip microcontroller application.
- Advanced CP/M: PluPerfect Writer and using BDS C with REL files.
- Real Computing: The NS 32000.

Issue Number 43:

- Standardize Your Floppy Disk Drives.
- A New History Shell for ZSystem.
- Heath's HDOS, Then and Now.
- The ZSystem Corner: Software update service, and customizing NZCOM.
- Graphics Programming With C: Graphics routines for the IBM PC, and the Turbo C graphics library.
- Lazy Evaluation: End the evaluation as soon as the result is known.
- S-100: There's still life in the old bus.
- Advanced CP/M: Passing parameters, and complex error recovery.
- Real Computing: The NS32000.

Issue Number 44:

- Animation with Turbo C Part 1: The Basic Tools.
- Multitasking in Forth: New Micros F68FC11 and Max Forth.
- Mysteries of PC Floppy Disks Revealed: FM, MFM, and the twisted cable.
- DosDisk: MS-DOS disk format emulator for CP/M.
- Advanced CP/M: ZMATE and using lookup and dispatch for passing parameters.
- Real Computing: The NS32000.
- Forth Column: Handling Strings.
- Z-System Corner: MEX and telecommunications.

Issue Number 45:

- Embedded Systems for the Tenderfoot: Getting started with the 8031.
- The Z-System Corner: Using scripts with MEX.
- The Z-System and Turbo Pascal: Patching TURBO.COM to access the Z-System.
- Embedded Applications: Designing a Z80 RS-232 communications gateway, part 1.
- Advanced CP/M: String searches and tuning Jetfind.
- Animation with Turbo C: Part 2, screen interactions.
- Real Computing: The NS32000.
- The Computer Corner.

Issue Number 46:

- Build a Long Distance Printer Driver.
- Using the 8031's built-in UART for serial communications.
- Foundational Modules in Modula 2.
- The Z-System Corner: Patching The Word Plus spell checker, and the ZMATE macro text editor.
- Animation with Turbo C: Text in the

graphics mode.
• Z80 Communications Gateway: Prototyping, Counter/Timers, and using the Z80 CTC.

Issue Number 47:

• Controlling Stepper Motors with the 68HC11F
• Z-System Corner: ZMATE Macro Language
• Using 8031 Interrupts
• T-1: What it is & Why You Need to Know
• ZCPR3 & Modula, Too
• Tips on Using LCDs: Interfacing to the 68HC705
• Real Computing: Debugging, NS32 Multi-tasking & Distributed Systems
• Long Distance Printer Driver: correction
• ROBO-SOG 90
• The Computer Corner

Issue Number 48:

• Fast Math Using Logarithms
• Forth and Forth Assembler
• Modula-2 and the TCAP
• Adding a Bernoulli Drive to a CP/M Computer (Building a SCSI Interface)
• Review of BDS "Z"
• PMATE/ZMATE Macros, Pt. 1
• Real Computing
• Z-System Corner: Patching MEX-Plus and TheWord, Using ZEX
• Z-Best Software
• The Computer Corner

Issue Number 49:

• Computer Network Power Protection
• Floppy Disk Alignment w/RTXEB, Pt. 1
• Motor Control with the F68HC11
• Controlling Home Heating & Lighting, Pt. 1
• Getting Started in Assembly Language
• LAN Basics
• PMATE/ZMATE Macros, Pt. 2
• Real Computing
• Z-System Corner
• Z-Best Software
• The Computer Corner

Issue Number 50:

• Offload a System CPU with the Z181
• Floppy Disk Alignment w/RTXEB, Pt. 2
• Motor Control with the F68HC11
• Modula-2 and the Command Line
• Controlling Home Heating & Lighting, Pt. 2
• Getting Started in Assembly Language Pt 2
• Local Area Networks
• Using the ZCPR3 IOP
• PMATE/ZMATE Macros, Pt. 3
• Z-System Corner, PCED
• Z-Best Software
• Real Computing, 32FX16, Caches

Issue Number 51:

• Introducing the YASBEC
• Floppy Disk Alignment w/RTXEB, Pt 3
• High Speed Modems on Eight Bit Systems
• A Z8 Talker and Host
• Local Area Networks—Ethernet
• UNIX Connectivity on the Cheap
• PC Hard Disk Partition Table
• A Short Introduction to Forth

The Computer Journal Back Issues

• Stepped Inference as a Technique for Intelligent Real-Time Embedded Control
• Real Computing, the 32CG160, Swordfish, DOS Command Processor
• PMATE/ZMATE Macros
• Z-System Corner, The Trenton Festival
• Z-Best Software, the Z3HELP System

Issue Number 52:

• YASBEC, The Hardware
• An Arbitrary Waveform Generator, Pt. 1
• B.Y.O. Assembler...in Forth
• Getting Started in Assembly Language, Pt. 3
• The NZCOM IOP
• Servos and the F68HC11
• Z-System Corner, Programming for Compatibility
• Z-Best Software
• Real Computing, X10 Revisited
• PMATE/ZMATE Macros
• Controlling Home Heating & Lighting, Pt. 3
• The CPU280, A High Performance Single-Board Computer

Issue Number 53:

· The CPU280
· Local Area Networks
· Am Arbitrary Waveform Generator
· Real Computing
· Zed Fest '91
· Z-System Corner
· Getting Started in Assembly Language
· The NZCOM IOP
· Z-BEST Software

Issue Number 54:

· Z-System Corner
· B.Y.O. Assembler
· Local Area Networks
· Advanced CP/M
· ZCPR on a 16-Bit Intel Platform
· Real Computing
· Interrupts and the Z80
· 8 MHZ on a Ampro
· Hardware Heavenn
· What Zilog never told you about the Super8
· An Arbitary Waveform Generator
· The Development of TDOS

Issue Number 55:

· Fuzzilogy 101
· The Cyclic Redundancy Check in Forth
· The Internetwork Protocol (IP)
· Z-System Corner
· Hardware Heaven
· Real Computing
· Remapping Disk Drives through the Virtual BIOS
· The Bumbling Mathmatician
· YASMEM
· Z-BEST Software

Issue Number 56:

· TCJ - The Next Ten Years
· Input Expansion for 8031
· Connecting IDE Drives to 8-Bit Systems
· Real Computing
· 8 Queens in Forth
· Z-System Corner
· Kaypro-84 Direct File Transfers
· Analog Signal Generation

Issue Number 57:

· Home Automation with X10
· File Transfer Protocols
· MDISK at 8 MHZ.
· Real Computing
· Shell Sort in Forth
· Z-System Corner
· Introduction to Forth
· DR. S-100
· Z AT Last!

Issue Number 58:

· Multitasking Forth
· Computing Timer Values
· Affordable Development Tools
· Real Computing
· Z-System Corner
· Mr. Kaypro
· DR. S-100

Issue Number 59:

· Moving Forth
· Center Fold IMSAI MPU-A
· Developing Forth Applications
· Real Computing
· Z-System Corner
· Mr. Kaypro Review
· DR. S-100

Issue Number 60:

· Moving Forth Part II
· Center Fold IMSAI CPA
· Four for Forth
· Real Computing
· Debugging Forth
· Support Groups for Classics
· Z-System Corner
· Mr. Kaypro Review
· DR. S-100

Issue Number 61:

· Multiprocessing 6809 part I
· Center Fold XEROX 820
· Quality Control
· Real Computing
· Support Groups for Classics
· Z-System Corner
· Operating Systems - CP/M
· Mr. Kaypro 5MHZ

Issue Number 62:

· SCSI EPROM Programmer
· Center Fold XEROX 820
· DR S-100
· Real Computing
· Moving Forth part III
· Z-System Corner
· Programming the 6526 CIA
· Reminiscing and Musings
· Modem Scripts

Issue Number 63:

· SCSI EPROM Programmer part II
· Center Fold XEROX 820
· DR S-100
· Real Computing
· Multiprocessing Part II
· Z-System Corner
· 6809 Operating Systems
· Reminiscing and Musings
· IDE Drives Part II

Issue Number 64:

· Small-C?
· Center Fold last XEROX 820
· DR S-100
· Real Computing
· Moving Forth Part IV
· Z-System Corner
· Small Systems
· Mr. Kaypro
· IDE Drives Part III

Issue Number 65:

· Small System Support
· Center Fold ZX80/81
· DR S-100
· Real Computing
· European Beat
· PC/XT Corner
· Little Circuits
· Levels of Forth
· Sinclair ZX81

Issue Number 66:

· Small System Support
· Center Fold: Advent Decoder
· DR S-100
· Real Computing
· Connecting IDE Drives
· PC/XT Corner
· Little Circuits
· Multiprocessing Part III
· Z-System Corner

Issue Number 67:

· Small System Support
· Center Fold: SS-50/SS-30
· DR S-100
· Real Computing
· Serial Kaypro Interrupts
· Little Circuits
· Moving Forth Part 5
· European Beat

SPECIAL DISCOUNT

15% on cost of Back Issues when buying from 1 to Current Issue or all four volumes.
10% on cost of Back Issues when buying 10 or more issues.

Subscriptions (CA not taxable)	U.S.	Canada/Mexico (Surface)	Canada/Mexico (Air)	Europe/Other (Surface)	Europe/Other (Air)
1year (6 issues)	$24.00	$32.00	$34.00	$34.00	$44.00
2 years (12 issues)	$44.00	$60.00	$64.00	$64.00	$84.00
Back Issues (CA tax)	add these shipping costs for each issue ordered				
Bound Volumes $20.00 ea	+$3.00	+$3.50	+$6.50	+$4.00	+$17.00
#20 thru #43 are $3.00 ea.	+$1.00	+$1.00	+$1.25	+$1.50	+$2.50
#44 and up are $4.00ea.	+$1.25	+$1.25	+$1.75	+$2.00	+$3.50
Software Disks (CA tax)	add these shipping costs for each 3 disks ordered				
MicroC Disks are $6.00ea	+$1.00	+$1.00	+$1.25	+$1.50	+$2.50

Items: _____

_____

Back Issues Total _____
MicroC Disks Total _____
California state Residents add 7.25% Sales TAX _____
Subscription Total _____
Total Enclosed _____

Name: _____
Address: _____
_____
_____
_____
Credit Card # _____-_____-_____-_____ exp ___/___

Payment is accepted by check, money order, or Credit Card (M/C, VISA, CarteBlanche, Diners Club). Checks must be in US funds, drawn on a US bank. Credit Card orders can call 1(800) 424-8825.

TCJ *The Computer Journal*

P.O. Box 535, Lincoln, CA 95648-0535
Phone (916) 645-1670

Regular Feature

Editorial Comment

PLC & Forth

The Computer Corner

By Bill Kibler

Well another issue is going to the printer and here is my chance to talk about things happening to me. This issue is a little late due to work assignments on site.

Forth in Action

While being on site and using a portable PC to download the ladder program to the PLC we needed to check some serial communications on the RS422 circuits. We tried using a standard modem program, but our current version did not have a pure capture mode. The serial link used many values below $20 and since these are normally control characters, the program ignored them. Later versions and many newer modem programs have a diagnostic mode which can be used to capture to disk all data from or to a serial port.

Fortunately I had a copy of polyForth with me and had done some serial testing previously. That night I hacked out a quick serial capture and display program in a little under two screen of code. It is important to note that I spent several hours previously figuring out how polyForth does serial. Most other Forth systems are not fully multiuser as is polyForth.

Being fully mutliuser or multitasking makes bringing another user on-line as simple as adding a few LOAD statements for serial support in the startup load screen. This will provide for the fully interrupt and buffered serial communications needed to handle another user in the background. I tried unsuccessfully to do some serial monitoring in the office without the manual and gave up. The failure I attributed to not understanding the how and why of multitasking and what hooks must be installed before attempting it.

The next day with manual in hand and following the steps outlined, I had a second terminal talking on-line within an hour. The comment screens did explain how the code was put together, but it was the book that explained the multitasking needs and why/how the serial is attached to the task loop. That attachment must also have a user program or in this case a terminal program to be put into the loop. This done, you can have several users on-line at the same time changing and using Forth.

My desire however was to do some diagnostic serial work and thus a slightly different "task" was needed. Several examples were provided in the source code and the previous day I had tried using these not being aware of the needs and problems multitasking presented. I reviewed again what I had done and added the necessary statements to start the task and end it for each transfer and response desired. It worked, allowing me to test how simple or complex talking to this other device would be.

With that knowledge in the back of my mind, and the manual at home, I had to redo some of what I did before from memory. Since this time all I wanted was a printout of data, I chose another sample to modify. This time I had a better feeling about all the supporting code that was needed.

With a little trial and error, I was able to separate the non-essential sample code from the serial terminal communications procedures. Adding my print to screen in HEX statements and the program was running in about two hours of playing around. The next day on site, I ran the code and in 10 minutes saw that our serial link was working as designed without any errors, which was all we wanted to do in the first place.

Since I may want to do this test again later, I plan on doing a turnkey version. Since learning that you must read the manuals first, to take in the more complex intricacies of polyForth, doing a turnkey will have to wait until this issue is at the printer. Since I feel that serial communications testing is very common and an excellent use of Forth, I'll wait on printing my code and how-to instructions.

My overall impressions were very positive of polyForth. I saw rather quickly that much of the price is due to the manuals and also the extensive development that has been done with this version of Forth. Those many years of refinement and adjustments to the code have produced a compact and highly efficient development system. I say system, because it is clearly not just a development language but more.

The source which is provided allows you to do just about anything and in a very large amount of those "things" some sample or basic structure will be present to show you how. Last year when I was looking for a better data base program, I concentrated on packages that had a sample that would be very similar to my desired project. The idea was to only have to make minimal changes to an existing program, and thus little development work.

What I found was several samples that when attempting to make changes, had some hidden structure that prevented the

changes I wanted. In one case the program flat refused to alter the way labels were printed from the data base. It is possible that some rather deeply hidden screen had a simple "type" declaration and changing that would have automatically made the desired changes, but if so that certainly was not obvious.

I intend to use polyForth for my data base work. Why? Two reasons standout: a good database set of words already exists; and true multiuser ability. In what little reading I have done so far, I can see plenty of database tools and good explanations of how to use them (in the manual). Since I have several workstations and extra terminals, I want to put these closer to the phones and shipping location, which is not where I want to have my computers. Setting up a true multiuser data base program now, would save possible changes and problems later.

PT68K Arrives

I finally got my order of a Peripheral Technology 68000 ISA compatible mother board. Since time on site has prevented my putting it in a box that is empty and waiting, a full review will have to wait. I can say that it seems well built and designed. I especially like the four on board serial ports and two floppy controllers. This allows you to bring the unit up without any PC compatible cards.

I am not sure yet of all the options, but so far it appears that it will boot either a serial terminal or Hercules compatible video system. REX a Flex compatible system (supplied with source) uses either of these devices to boot with. OS9/68K (which my copy is still in transit) appears to use VGA cards, mouse and more. Since I am only a cursory user of OS9, I am looking forward to digging more into it when it arrives.

When talking with GMX (GiMiX people), they too have a similar system. Literature with the PT68K indicated other styles of their unit were available. We now have the 6809 board by Brad using the PC interface as well. The C article found one university in New Zealand using the PC system on their 6809 project. I was talking up a Z180 based design and on the Zed Fest round table, talk of a Z180 system using some PC compatible components (I believe) was mentioned. All in all it appears that all users are starting to realize how hard it is to pass up entire working cards for $5.

While at the last swap meet, I saw plenty of used but still working video and disk cards for $5 each. S-100 boards were there, but cost more than similar functioning PC boards. Did I come away feeling that PC compatible boards are better and more desirable, not really. They are cheaper and more available, but seldom do you have any information on these products. Schematics for I/O cards in the PC world are non existent. The only option should it fail is to get another card. Yes, at $5 each that is pretty simple and the way to go. But if your objective varies from what was intended your faced with changing your project to match the hardware or building your own hardware. The PT68K option is very good in that respect, as the needed units are on board and the PC compatible card are optional slots. Very good idea!

The going your own route in the smaller micro arena is busting out big time. I have seen more and more small outfits developing their own products than any other area of computing right now.

When I review these systems, I look at what tools are supplied and as I did with the database search see just how much sample program is already done. My concern for these people is a change for some current big supplier. Right now users like the automotive industry consume some chip makers total production. Should a company decide to change suppliers, we could see that old supplier shift their production facilities over to the consumer embedded market.

PLC Direct

One such change occurred in the PLC market. Koyo made for GE, TI, and Siemens their bottom of the line PLC units. Orders for these units likely dropped below the companies minimum to stay in production. Unlike other companies who often just close down production or shift to other products, Koyo decided on a direct market approach. They set up a southern warehouse, 24 hour 800 number, take credit cards, ship same day, and sell for half the current price the same units are sold under other names.

You can have a complete PLC system from them for about $500. Unlike the small micros which often need power supplies and housings, these are complete bus type PLC units. Their is a power supply back frame, a CPU of several types, and about four sizes of input and output modules available. Their advertising gimmick was a box the exact size of their unit that thanks to a built in rubber band pops into shape when removed from the envelope.

I credit this magic box with selling my boss on trying one of their units for an upcoming show. The size is about 3 x 4 x 9 inches with five I/O slots. They have some fairly decent Windows based software which I just got a chance to load and use. Since I barely loaded and tried it, more will probably follow on PLC Direct systems.

Between PLC Direct and say some big company deciding to enter the embedded market, life could get pretty sticky for the little developers. My only advice is to provide real personal services and develop better interpersonal skills if you hope to stay alive making little systems. It will be your personal relationship with your clients that will keep them from going to a PLC Direct type of organization.

Back to Work

In looking at a few products, keep in mind that just getting a better software program will not solve all your problems. The future still belongs to those who understand in detail what happens inside of computers. Remember too that classic and older systems still provide the best method of learning what really happens inside computer systems. With that, keep hacking, learning, and having fun.

TCJ CLASSIFIED

CLASSIFIED RATES!
$5.00 PER LISTING!

TCJ Classified ads are on a prepaid basis only. The cost is $5.00 per ad entry. Support wanted is a free service to subscribers who need to find old or missing documentation or software. Please limit your requests to one type of system.

Commercial Advertising Rates:

Size	Once	4+
Full	$120	$90
1/2 Page	$75	$60
1/3 Page	$60	$45
1/4 Page	$50	$40
Market Place	$25	$100/yr

Send your items to:
The Computer Journal
P.O. Box 535
Lincoln, CA 95648-0535

NEEDED: Original FIG Forth cartridge for VIC-20. Already have the manual; could also make another cartridge from object code on floppy. FIG never returned my letter. Douglas Beattie, PO Box 47, Oak Harbor, WA 98277-0047

For Sale: complete CP/M 2.2 system used for Z80 development. InterContinental MicroSystems CPZ-48000 single board S-100, 64KRAM, 4Mhz Z80, Dual Disk Controller, Serial/parallel, 8 in. DD drives with 125 disks. Pragmatic Dual Port RAM, Optimal Technology EPROM Programmer with EPROMS, UV Eraser, S-100 Extender, all manuals/docs + Cortesi's CP/M Books, TurboPascal V3, T-Maker III, misc software. Will operate with any RS232 terminal. Best offer takes ALL. Call (303) 651-7360, Robert Zimmerer, Longmont, CO.

Historically Brewed. The magazine of the Historical Computer Society. Read about the people and machines which changed our world. Buy, sell and trade "antique" computers. Subscriptions $18, or try an issue for $3. HCS, 10928 Ted Williams Place, El Paso, TX 79934.

Wanted: SwTPC equipment. Boards, 6800, 6809, 6809/S09, memory, 8K, 16K, 32K, I/O port cards MP-S2, MP-L2, MP-ID, terminals CT-64, CT-82, 2716 EPROM card, any SS-50 items hardware or software. SwTPC 6809/S09 manuals. John Fiorino, 518 - 85th St., Brooklyn, NY 11209.

Wanted: The following Zilog Z80 manuals: CPU Technical Manual, 1977 #03-0029-01; Assembly Language Prgmmr Man. #03-0002-01; SIO Technical Manual; "the Z80 Family Program Interrupt Structure" APP Note # 03-0041-01. Mostek Z80 Assembly Language Programming Manual. Articles by Gary Kildall (the older the better) i.e. "CP/M: A Disk Control Program for Microcomputer System Development", in "Microcomputer Applications", June 1975.
Also any of the following manuals or disks: CP/M 1.3 or 1.4; CP/NET; CP/NOS; MP/NET; MP/NOS; MP/M II; DR GRAPH; DR DRAW; PL/I-80; GSS Plot; GSS Kernel; Data Manager; Screen Manager; CP/M Plus Binder; Microsoft COBOL-80; BIOS of DRI CP/M card (Apple][e). (Note: 8" disks ok, but Epson QX-10 European format better. Send Air Mail please.) ROCHE Emmanuel, 8 rue HERLUISON, 10000 TROYES, FRANCE.

TCJ ***The Computer Journal***
Post Office Box 535
Lincoln, CA 95648-0535
United States

BULK RATE
US POSTAGE
PAID
Lincoln, CA
PERMIT NO. 91

ADDRESS CORRECTION REQUESTED
FORWARDING AND RETURN POSTAGE
GUARANTEED

Telephone: (916) 645-1670