PERQ Workstations

R. D. Davis

Last Updated: November 6, 2003 from the Sept. 7, 1991 edition.

Contents

Chapter 1

Preface and Dedication

This document needs to be updated to reflect the information provided for PERQ Fanatics from Dr. A.R. Duell; Tony has been an amazing source of ingenious information about maintaining, repairing, preserving and understanding the PERQ workstations—and other computers, and I can't call this document anywhere close to being somewhat complete without including some of that information, which I'd planned to add while trying to complete this incomplete document over a decade ago. There's also some other information about the PERQs that I'd like to eventually include, which may be of interest, or which may be yawn inducing... you decide. Hopefully I'll get 'round to completing another version of this, with all of the aforementioned information added to it.

This document is dedicated to Tony Duell and the creators of the PERQs, to whom I say "Thanks for your work and ingenious hacking!"

Chapter 2

History

Note: Unless otherwise noted with a specific month or year, all items below fit somewhere within either the year or decade specified, but possibly not in the exact order in which they are mentioned within that generalized time period.

2.1 PERQ History as Told by Those Who Were There

Note about the PERQ's history from Brian Rosen, as it appeared in the PERQ-Fanatics mailing list on July 30, 1993:

> I might be able to shed a little light on chronology, but I only really remember the order of things rather than the actual dates:
>
> It started at CMU, where there was a very active engineering laboratory headed by Bill Broadly that built hardware for CS research. Several somewhat interesting projects came out of that, including the first 16-bit A-D and D-A systems for speech and music research, a writable control store for a PDP-11 model 40, and a cute vector (calligraphic) display that did 50,000 vectors at a 60Hz refresh rate. Three Rivers Computer Corporation was started by Broadly, Stan Kriz, Jim Teter, Paul Newbury and myself, with a little help and encouragement from Raj Reddy. Its first products were versions of the above projects. We all worked full time for CMU and moonlighted at 3RCC.
>
> In 1976, I left to go to work at Xerox, at the Palo Alto Research Center, where I worked on a successor to the Alto (the first worksta-

> tion, against which all others sprang) called the Dolphin. When the Dolphin was completed, I returned to 3RCC and started work on the Perq. That was 1978. At that time, CMU was starting a project to change the way it provided computing to the CS people from timeshared mainframes to "personal" computers or workstations. They put out a request for proposals, to which 3RCC responded. The first showing of the Perq was as Siggraph, 1980, and the first machine was delivered to CMU in late 80 or early 81. It had 256K of memory, 27MB disk (14"), 4K of writable control store, an 8" floppy disk, and a 768 x 1024 monochrome raster display. At the time, Sun and Apollo were dreams. The Perq had a lot in common with the Alto, but was significantly different in many ways.
>
> One of the early customers for Perq was Bob Hopgood, at Rutherford Appleton Labs in the U.K. His enthusiasm for the Perq caused the Science and Engineering Research Council (SERC) to try to adopt it as a computing platform for SERC sponsored research. However, SERC could not go around pushing a U.S. system. So, it influenced ICL to enter into a collaborative effort with 3RCC to make PERQs in the U.K. Eventually, ICL provided some funding for 3RCC and engaged in some co-operative R+D efforts.
>
> 3RCC, which changed its name to PERQ Systems in around 1983, never had stable financing, made several management mistakes, never had effective marketing, waited too long to switch to microprocessors, and foolishly followed CMU's efforts on operating system development (Accent, precursor to Mach). I was Vice President of Engineering through much of the above, so I'll take most of the blame, if you wish.
>
> Perq went belly up in 1985 when its investors refused to supply any further funding.

A note about PERQ history from Robert Rae, of July 30, 1994, from the PERQ-Fanatics mailing list:

> I was involved at the beginning of the UK Perq saga in a rather weird way. We (Bill Clocksin and myself, as I remember, in the AI department at Edinburgh) were contacted around December 1979 by Bob Hopgood of the Rutherford Appleton Lab and Roger Vinnicombe of ICL (through Tommy Thomas of the ERCC), with strict instructions not to tell anyone about anything we knew about the

Perq. As neither of us had heard of it before, this was not too difficult! They then told us all about it. We never did find out why they thought we were already involved ...

Remember that, at that time, we lived in a relatively under-priviliged computing world. The Department's own computing resource was one PDP-11/60 (relatively recent: it had been one 11/10 before that!), and research work had the use of a locally managed PDP-10 which served a national SERC AI community. VAXen were new and wonderful (was the 750 just becoming available?) and a VT100 on everyone's desk was a pipe dream. Xerox Altos, MIT Lisp Machines and Nu bus plans, and the CMU SPICE project made us very jealous. So the prospect of a commercially available VAX style architecture with incredible graphics capabilities at an affordable price was very attractive.

The result was that Rob Witty (then at RAL) and I packed our passports and visited Three Rivers in Pittsburgh on (it says in the report in my hot little hand) the 14th and 15th of January 1980 to have a look at the Perq and meet some of the people involved: mainly Brian Rosen, Stan Kriz, Bob Spuntak and Miles Barel. I think we also spoke that time to Robert Sproull at CMU. It was really exciting to be involved with such a new technology. At that time, they had assembled only five machines, and I don't remember any of them actually being capable of working (but real soon now)!.

Our summary (I hope confidentiality isn't a big issue now) included the statement which I would continue to stand by: "The PERQ software will be little and late".

We actually wanted a Unix box and the Unix port[1] was not done until a lot later by ICL, at Dalkeith just outside Edinburgh. Unfortunately there seemed to be collective brain failure there and they targeted the PDP11 rather than the VAX as their competition, so they went for good performance for small processes. They also went for separate input and output windows (I think that was Nick Felisiack's decision: I name the guilty party). It was also handicapped, as far as we were concerned at least, by a UK religious intention to use token ring rather then ethernet technology, so we couldn't use any existing US networking software and communications were not good. So it never did fullfill its original promise, but it was an

[1] PNX

> excellent graphics platform and, for a short time, it was the nearest we had to the future.
>
> We did eventually use it most (a 3RCC mk2 running Accent) as a benchmark for Common Lisp.
>
> It was all a very long time ago and, I fear, a lot hasn't stuck in my memory.
>
> Robert Rae

2.2 PERQ History as Otherwise Researched

2.3 Late 1960's

Note: I recall reading in some computer magazing about someone else who developed a workstation, with a mouse, prior to the one at XEROX PARC; it was probably in an issue of Byte, Computer Shopper, Radio-Electronics, etc. There was a black & white picture of some man sitting in front of the system that he had designed. I know that I did see this article and am not dreaming this! This was possibly not a true workstation, but it was possibly the predecessor of all workstations since it was the first successful graphics system for CAD type stuff or somesuch.

At Xerox's Palo Alto Research Center (PARC), in the late 1960's, work on the supposedly first computer workstation began.

This was an undertaking which would take several years of research and a great deal of money being invested in the project. The name of this system being designed was the Alto.

In those days, a computer workstation was unheard of, and Xerox was designing the Alto. The Alto had a 10 Megabyte disk drive, 64K of memory, and a lot of interfaces. The use of a 'mouse' and complex audio (speaker system and synthesizer) with computers began here. The basic idea for this workstation was for a user to be able to drag around a two-dimensional picture on a computer's display screen. The Alto had a specialized data path and it had to compute furiously for the raster-op graphics algorythm that it used. The Alto, and later Xerox D-machines, used a peculiar type of microprogramming that no one has since copied. It had multitasking multiprogramming which could run up to eight different microcode tasks at different priorities. It was also the first machine to use an Ethernet interface. BitBlt/RasterOp also originated with the Alto.

It was around this time that Digital Equipment Corp. (DEC) had started producing minicomputers and time-shared systems.

2.4 1972/1973

Stan Kriz and Brian Rosen were undergrads in 1972/1973, attending Carnegie Mellon University (CMU). While at CMU, they had access to this information about Xerox and the Alto system.

2.5 1973

Xerox's experimental workstation was completed; all high- performance graphic workstations are descendents of this machine. The cost for one was $80,000. (the production version of the Xerox Alto was the Xerox Star? It was very proprietary and didn't do well [BF]). Several hundred Altos were produced and some were given to universities such as Stanford, MIT, Rochester and CMU.

Note that a possible connection between the similarities of PERQ Pascal and Modula-2 may have something to do with the fact that Niklaus Wirth spent some time at PARC, and that the PERQ was modeled after the Alto. Something to check out: is there some possible connection/similarity between Wirth's "Lilith" computer system and the Alto or PERQ?

2.6 1974

When Brian Rosen and Stan Kriz graduated, they worked at the university for several years and started the Three Rivers Computer corporation in 1974. When Three Rivers started, it specialized in producing high-quality graphics equipment.

Three Rivers Computer started out in the basement of four of the founders who lived on Craig Street in Oakland.

Brian Rosen went off to work at XEROX's PARC.

The Systems Development Division of Xerox was formed to exploit the features of the Alto for office automation purposes. Two new systems were the result of this effort: the Dolphin (1976) and the Dandelion. Of the two, the Dolphin bore the most similarities to the PERQ. Brian Rosen and (first name?) Charney, who were from Three Rivers, were part of the five-man design team for the Dolphin. The majority of the design work for the Dolphin was done by (?) Charney. The Dolphin was intended to become

the Star processor, however, after it was completed, Xerox decided that it was too expensive. The Dandelion was then chosen to replace the Dolphin as the 8010 Star workstation, since it was less expensive.

There were a lot of internal problems, "politics", at Xerox regarding the decision not to market the Dolphin. As a result of this, Brian Rosen returned to Three Rivers Computer with ideas for the PERQ.

2.7 1975

Implementation of the RIG OS, precedessor of Accent, began on an early version of the Data General Eclipse computer at the University of Rochester.

2.8 1976

Fall of 1976: First usable implementation of RIG came on-line; a network of DG Eclipse systems and Xerox Alto's was used.

2.9 Late 1970's

Carnegie Mellon put out a proposal for SPICE (Scientific Personal Interactive Computing Environment) environment. They wanted a system which would be similar to a Xerox Star or Apollo, only less expensive. The original PERQ was developed as a response to this proposal.

he SPICE project was sponsored by the Defense Advanced Research Projects Agency (DOD), ARPA Order No. 3597, monitored by the Air Force Avionics Laboratory Under Contract F33615-78-C-1551. (1978 ???)

2.10 1978

In 1978, Brian Rosen and Stan Kriz decided to produce a commercial version of the Alto, since Xerox had decided not to produce the workstation. Three Rivers Computer took the concepts of the Alto, and, using all new design, designed the PERQ graphics workstation. The PERQ was the first commercial product to use a bitmapped display.

Brian Rosen was responsible for most of the Perq's design [NF].

2.11 1979

May 1979: The PERQ product was announced. Brian Rosen headed the PERQ development.

June 1979: The SERC's Rutherford Appleton Laboratory was the first customer to place an order for a PERQ.[2]

August 1979: The PERQ prototype was demonstrated at the ACM's Siggraph in Chicago.

2.12 1980

In The first PERQ was delivered.

Sept. 1980 or October 1980: Note that there was a conflict between two newspaper articles with regards to the date. Three Rivers moves into the building at 720 Gross St. in Bloomfield with 35 full-time employees and 10-15 part time employees.

Sept. 1980: Note that the following information appeared in the ACM's publication "Accent: A communication oriented network operating system kernel" [Rashid, Robertson]:

> An implementation of the Accent IPC Facility as a communication facility for VAX/UNIX has been in use since **March, 1980** [7]. Network servers written in PASCAL have also been implemented and are now in use. The PERQ implementation of the full Accent Kernel is now (**September 1980**) being tested and refined.

Note: Another possible (???) reason for the Perq and its use with the Spice project (taking a wild guess here), and the selection of this project done at CMU by the DOD, may have something to do with (again, something related to Xerox) the fact that the XEROX Pilot OS did not meet the first (and the most controversial) requirement in the design of Accent, which was that a logically distinct and independent address space was to be used for each process supported by the kernel. Also note that while this was an advantage in the area of reliability executing programs written in languages that are not "safe" (the possibility that a properly written program could

[2]Conflict: the above was from the RAL SERC paper, but according to a newspaper article by William H. Wylie, CMU was the first customer. However, a third source (Comp. Graphics & Apps.) cites that the first order came from RAL.

attempt to use memory addresses which don't belong to it), it did cave the drawback of causing Accent to run slower.

MIT Researchers were working on two systems which used the concepts of the Alto using very different technology. This first was known as the Lisp Machine project; the MIT Lisp Machine had all the features associated with a workstation, but was much more expensive. There were two spinoffs of this project which were less costly lisp machines produced by Symbolics and LMI.

The other MIT project which evolved from knowlege of the Alto was the Nu machine; this was to be a workstation using a Motorolla 68000 microprocessor as its CPU. In various attempts to turn the Nu machine into a commercially viable product, MIT teamed up with Exxon Enterprises, then Heath/Schlumberger, and then Western Digital. All of these attempts failed. The only thing that resulted from this was a deal between MIT and Western Digital in which Western Digital was to make a Lisp processor board which would plug into the bus of the Nu machine.

After some officials from Prime Computer took a tour of MIT and saw the Nu machine, Lisp Machine, and the Alto, it is runmored that the idea for the Apollo workstations was formed. The head of engineering at Prime Computer, William Poduska, started Apollo. Apollo had the added benefit of much more financial backing than Three Rivers had. Apollo introduced its Domain processor.

2.13 1981

January, 1981: Edward Fredkin drafts James Gay from Information Int'l to become President of Three Rivers.

April 1981: The first line of code for the Accent OS is written.

Note: there is a conflict here. The above statement was derived from the publication "From RIG to Accent to Mach", [Rashid], page 4. Refer to the time period for **September, 1980** to see the conflict. Should this date have been **April of 1980**?

Note: Accent is a registered trademark of Accent International, Inc. and the product which Accent designates is sold as a spice (which contains only monosodium glutamate).

Soon after the PERQ went into production, Apollo got ahold of the idea of what Three Rivers was doing and got ahead of them.

Apollo was able to deliver the first Apollo Domain in the **fall of 1981**; a great big deal was made over the new Apollo system. It entered into the

fledgling workstation market with a much higher profile than the PERQ had.

Note: there is a possible conflict here; this information comes from the company confidential stuff in the Perq-3 design docs. notebook; the article in Venture states that the Apollo Domain was introduced in 1980 – was it just announced to the public then or was it actually produced and delivered at that time?

In California, some people at Stanford University who were very knowlegable about Altos designed an advanced graphics computer terminal which they intended to use with a VAX running Berkley Unix. They intended for this to be the worlds most advanved front end for a VAX running BSD. In essence, this terminal contained the features of a workstation and they named it the Stanford UNiversity Terminal, or SUN, and licensed its design to three companies. The most well known of which is Sun Microsystems (find out the names of the other two). Sun Microsystems then started producing low-end workstations entry priced at low $8900 for a system with no disk, no software, and no networking capability.

Unlike Three Rivers, SUN used a lot of "off the shelf" parts, and was able to bring their systems to market more quickly than Three Rivers was able to.

Finally (was this in 1981?), Xerox allowed the Dolphin to be marketed; it was sold with software called Interlisp which was developed at PARC. The Dolphin, Dorado, and Dandilion, all running Interlisp or Smalltalk, were now being marketed by Xerox's Electro- Optic Systems division. These systems primarily competed with the MIT Lisp Machines.

Summer 1981: Three Rivers shows its first profit.

August 1981: After a suggestion from SERC, ICL entered into an agreement with Three Rivers in which it would manufacture and market the PERQ in the UK and in Europe. This deal gave ICL the marketing rights to the PERQ in the UK, all of Europe, and in a number of other countries. This arrangement worked well for several years. This resulted in a strong partnership between Three Rivers Computer Company, ICL, SERC, and which later included CMU (which already had strong ties with 3RCC).

Friday, September 1, 1981: CMU received its 100th computer from Three Rivers; this was a ceremonial delivery, since CMU purchased the first Perq computer.[3]

October 1981: Version 1.6 of the Perq Color Display Document writ-

[3]Conflict: SERC paper says it was first.

ten.

1982: Plans for MPOS (multitasking POS) put on hold; this OS was developed by Brian Rosen, Brad Meyers, and a few others. This OS was never released to the public.

January 1982: ICL begins manufacturing PERQs in the UK.

March 1982: Chairman Edgar Fredkin and president Jim Gay resign. New chairman Richard Rifenburgh and new president Aaron Coleman.[4]

June 1982: The targeted completion date by RAL to move Unix Version 7 to the PERQ to run on top of the Accent kernel.

July 1982: SERC Document written. J.M. Loveluck, Rutherford Appleton Lab.

Perq Color Display Document (Colwell, Kriz, Stoner), revision 2.0 was written. It was claimed that retrofitting the Perq for color would be easy.

2.14 1982/1983:

ICL bought the world-wide marketing rights, excluding sales within the United States, to the PERQ and added a FORTRAN compiler to Perqs running under POS (or PQOS). Within about a year or so, ICL added the Landscape monitor as an option.

British Telecom Industries in Felixtowe, England, was looking for a machine to market under their name which was to use a lot of customized software. (Apparently this was a Perq-1?) T.A., who had the job of determining which machines were the easiest to use had this to say about the Perq: "I was supposed to see which were easy to use. The Perq was the funnest and most unlikely (it cost a fortune) of the lot, you could draw pictures and everything! ...I had to demonstrate this machine that couldn't be relied on to boot the same way twice to the deputy-director of BT."

2.15 1983-1984?

PERQ was losing market share to Apollo and Sun workstations; to counter this, ICL decided to develop a version of System V (???) [BF] Unix for it; this provided an improved window manager and networking, provided multiprocessing, and added a C compiler. The ICL office in Bracknell was where the Perqs were supported from. Note: I called there this past fall (Fall, 1990) and was told that PNX was about to be discontinued.

[4] I believe this was March of '82, since the article said "last March" and it was written in Aug. '83.

Due to the slowness of the Accent kernel on the PERQs, "ICL had to do a quick & dirty implementation of PNX in order to sell it to UK universities". [VP] PNX was not originally intended to be used outside of UK universities.

For the PNX (Unix Version 7/System III Unix for the Perq) development, the Perq's microcode was changed from its original Q-Code (similar to P-Code) to a form of microcode which was more appropriate to the C programming language.

2.16 1984:

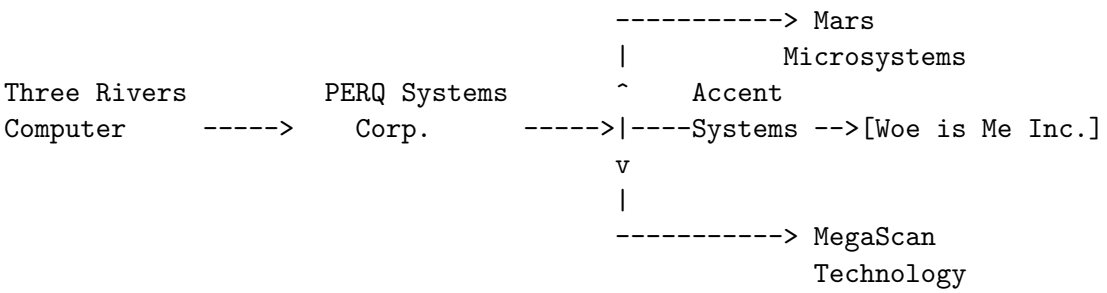
ICL's Dallkeith development center (near Edinburgh, Scotland) was closed down and Spider Systems was formed. One of the co-founders of Spider Systems was involved in the development of Perq hardware.

Spider Systems, working with Perq Systems, PQS, implemented a layer of Unix which ran on top of the Accent OS. (Was this QNIX?).

When the computer slowdown occured in 1984, Three Rivers Computer began to go under. At this time, Accent Systems split up from them, primarily as a maintenence operation for the support of the PERQs. Towards this end, Three Rivers changed its name to PERQ Systems.

In 1984, Perq Systems began working on the Perq3 project; this was a 68020 based machine with seperate I/O and raster processors. The main memory consisted of 8 Megabytes of RAM.

Fall, 1984: It was realized at CMU that a new hardware base would be needed for the Accent OS if Accent was to be spared the same fate that RIG met up with. There was also the problem of Accent, running a Unix compatibility package (QNIX?), not being able to run many pieces of Unix software which were being written at the time. Mach, an OS similar to Accent, was then conceived. It was to be completely compatible with Unix and would run on a wide variety of hardware other than the PERQs.



2.17 1985

In **September of 1985**, Perq Systems went out of business and their work on the Perq-3 project ceased.

CMU stopped buying PERQs; they had previously been purchasing them due to close ties between the CS faculty and Three Rivers; many people were forced to use PERQs for their projects and a users revolt ensued. Within a year, many of CMU's one hundred and fifty PERQs vanished from CMU. A few were sold to customers of Three Rivers Computer/PERQ, but the majority of them were sold for scrap metal for $5 apiece to a local surplus dealer.

Varityper purchased the manufacturing rights to the Perq T-2, but soon discontinued the manufacture of them.

Some of the members of PERQ Systems' software team returned to CMU, such as Dave Golub who became involved in the Mach OS.

When PERQ Systems Corp. went out of business, several new firms arose from the ashes.

As this is being written, **June 23, 1991**,[5] the most direct descendent of PERQ Systems, Accent Systems, is on ts very last legs. It no longer goes by its original name. It is interesting to also note that as of 1990, Three Rivers Computer was still listed in the Thomas Register.

Accent Systems started out as a maintenance organization to repair, and provide parts and supplies for, the PERQs, and was run by former employees of PERQ Systems. The name Accent Systems can be a bit deceiving, since the firms name leads one to believe that they would have more to do with the Accent OS than other apects relating to the PERQs. To the contrary, much of Accent Systems business had to do with suporting PERQs running POS, although they also did sell the Accent OS, as well as PNX (Unix) for the PERQs.[6] Accent Systems went through several changes. First, it merged with a company called Scribe, which produced text-formatting software, and the name became Scribe/Accent. When the Scribe side of the business began losing money, the PERQ part was still making a profit, but

[5] This is when I began writing this manuscript, and when the majority of it was completed. Please refer to the Procrastinator's Anonymous web page for further information: www.rddavis.org/rdd/procrastinators.html

[6] PNX was never supported by Accent Systems, however.

not enough to support both of them. Next, the company name changed to Integra/Scribe, At this point, when one would call for information about the PERQs, it was difficult to get ahold of someone there who even knew what a PERQ was. The name then changed to SAY Technology. By this point, it turns out that when Scribe/Accent was losing money, they went bankrupt, and the landlord had siezed all of the PERQ related assets, all of which remained under lock and key until the lawyers battled things out and it was finally decided that the remaining inventory of what was Accent Systems was to be liquidated. The landlord, Real Estate Enterprises, was not all that interested in the PERQs, and they were in a hurry to reclaim the space taken up by the PERQ stuff, so that they could rent it out to someone else. At this point, only one employee remains from PERQ; Dave Callen. An Accent Systems employee, and former PERQ employee, to help them liquidate the remaning PERQ related equipment.

The sign in the office window of what was Accent Systems now reads "Woe Is Me, Inc.".[7]

Before PERQ Systems went under, a new system was in the design stages; the PERQ-3. This system ended up being designed under contract for Crosfield Electronics, Ltd., which was an English company.

When Perq went under (**late 1985?**), Crosfield Electronics purchased the rights to the engineering designs for a machine referred to as the PERQ-3B or the PERQ-5 in various documents; this system was in the midst of development when Perq Systems closed up shop.

Note that Crosfield Electronics helped to set up two small companies which were to help finish the development of a color graphics workstation, formerly known as the PERQ-3B. These two companies were Brian Rosen's MegaScan Technologies, which was to work on the Perq-3's hardware and Connor Scelza Associates (CAS) which was to work on the software.

According to ADF, the Perq 3B was originally designed to run under the Accent OS, however, some (perhaps later) internal (Perq) company documents show that a SunOS type of Unix OS was to be implemented.

2.18 1986:

Spring of 1986: The demise of the RIG OS occured due to the obsolescence of its base of Data General Eclipse and Xerox Alto hardware. Interesting that this should happen around the same time as the demise of PERQ Systems.

[7] this was back in 1991

May of 1986: The successor to Accent, the Mach OS, now runs, successfully, on a wide variety of machines including many VAX systems, IBM RT/PC, SUN 3, and Encore MultiMax. Also interesting that Mach begins to really take off as the Perqs fade into oblivion; there is no mention of any effort to get Mach to run on the PERQs.

A company, Intran, which produced software for the design of forms and graphics that ran on the Perq, ported the software over to a Sun platform. The software was ported from Perq Pascal to C. In order to do that, they loaded PNX onto a Perq and then used a PERQ utility to "convert the software" [JG]. Intran is now a subsidiary of Xerox.

This is only being mentioned here because the SERC document mentions something about being able to convert the POS .SEG files by some process by which they will be made runnable under Unix; however, it appears that this was done under a Unix system (QNIX?) which ran on top of Accent. Note also that CL found some MPOS stuff on a disk which was used for PNX; any connection here?

2.19 1986/1987

The Perq-3B came into production in **1986/1987**. Crossfield did much of the "iron-work design" [ADF]. As originally designed by Perq Systems, the Perq-3 needed to have most of its designs of worst-case analysis reworked in order to make the manufacture of the system possible. The only exception was John Straits RasterOp, a display accelerator deemed "excellent" by Crosfield; it was the only card in the system which did not need to have any changes made to it. In addition to John Strait's RasterOp, many of the Perq-3's original features were kept. Crosfield wanted to use the Perq-3's for picture processing.

MegaScan Technolog was started by one of the founders of PERQ Systems, Brian Rosen. Stan Kriz who was was the analog engineer for the PERQ also came to MegaScan, which had its offices in the old Perq office in Fairfax, VA. Stan Kriz and Max Salinas remained in Fairfax while the remains of the Perq group stayed in Pittsburgh.

Crosfield took over total control of the Perq-3 and MegaScan Technology was no longer involved with the project. Soon after, "troubled-times at MegaScan Technologies caused the split off of Rosen & Co. to form Mars".[ADF] Stan Kriz stayed on at MegaScan.

When MegaScan completed their work on the PERQ-3 project, they began work on a TMS34010-based, "low-cost" [MS], frame buffer for the

MegaScan monitor; this had front ends for both the VME and PC-AT buses.

In 1990/1991, MegaScan was sold to a company in Massachussetts and will be moving there. MegaScan produces a 300 dpi monitor with 3,000 x 4,000 resolution at a 70 Hz, non-interlaced, scan rate.

Another company (Connor Scelza Associates?) was formed by Don Scelza, and Gerry Connor, "technical and management leaders on software respectively" (from Perq Systems).

Mars Microsystems was founded by Brian Rosen. Note that the name of this company is the same as the code name of the PERQ 3B, a system which PERQ Systems would have produced, had they stayed in business. Mars Microsystems now manufactures an IBM PC-AT based machine that has a SPARC processor manufactured by Tatung. This system uses an "AT" type bus which accepts a board containing an Intel 80386 microprocessor; this board is used for the MS-DOS operating system. A seperate high-speed bus is used for the machines video and memory. MS-DOS output is redirected to the main video memory and it is possible that this machine uses some sort of shared memory configuration. The main operating system is Sun OS. Note that a Sun OS like Unix OS was also planned for the Perq 3B by Perq Systems Corp.

2.20 1997

July 1987: As of July 1987, the Spice Rack at CMU no longer contains any PERQ hardware; the PERQ stuff has been replaced with Lisp machines and SUNs.

2.21 Things whose time period is questionable

Probably sometime in the early to mid 1980's: The U.S. Navy (or USAF?) spent $2,000,000 in R & D in order to use Perq(s) on the U.S.S. Carl Vinson, an aircraft carrier. The Perq(s) was to be used to monitor ship operations, repairs, and the deployment of aircraft. Some related project called ZOG was being worked on at CMU. ZOG is now known as KMS which is being developed by a company called Knowlege Systems Inc. in Murrysville near Pittsburgh and had been ported over to run on Sun and Apollo machines instead of the Perqs; KMS is described as a "hypermedia product".

Siemens Corp. (US) used the Perqs for artificial intelligence research. Note: this was at Siemens at Forrestal Center, 105 College Road East, 3rd floor (Princeton, NJ ?).

On some of the early Perqs, the floating-point microcode was slightly error-prone; for example: 10.0 * 10.0 = 98.9 Which version(s) of the software contained this flaw?

The appearance of the SUN-2 was largely responsible for PERQs demise, [VP] and the SUN-3 was the "last nail in Perq's coffin" [CL]. When did these SUNs appear on the market? 84/85?

A lot of work was done with the PERQs at The University of Kent at Canterbury (UKC) including the development of a new windowing manager, a pattern/grey-ness editor, debugger, higher lavel graphics interface, amongst many other things. Also, Peter J. Brown did a system called GUIDE which was something like a hypertext system. "You have a text system where certain words within it can be expanded by click so that the text substitution is inserted in place (and some more fancy stuff)." [HPS] Information on GUIDE was published rather widely. Also, no microde programming was being done at this time at UKC because ICL considered it to be strictly secret; they did not want the details of it known. This was within a two-year timeframe, most likely between 1983 and 1985, since they continued to use the PERQs until they were taken off the market.

Many PERQ orders got cancelled in favor of SUN1's because they got tired of waiting for PERQs to be delivered. The PERQ was superior in many ways to the SUN-1 which had a 68000 CPU with no user-writable microcode, lacked the PERQs fast BitBlt graphics engine, and it had a "kludgey MMU that imposed some annoying restrictions". [HS]

Countries where the PERQs seemed to be fairly widely used: U.S., British Isles, Germany, Canada and Switzerland (legal to get bootleg copies of software there, and may be a good source for finding some PERQ software to run under PNX or copies of the PNX operating system). The use of the PERQs in Japan is a possibility, but it is questionable as to how widespread their use is or was.

```
Perq-1 production began/ended: ?
(number of units produced: ?)
Had the most reliable disk drive, least reliable I/O boards.

Perq-2 production began/ended: ?
(number of units produced: ?)

Perq T-1 production began/ended: ?
(number of units produced: ?)
```

```
Perq T-2 production began/ended: ?
(number of units produced: approx. 5000)

PERQ 24-bit CPU productiob began/ended: ?
(number of units produced: ?)

Speech card for the PERQ introduced: ?

Multibus board for the PERQ introduced: ?
```

What were the dates for the releases of the different versions of the PERQs operating systems?

A picture of a PERQ aboard a U.S. Navy ship (S.S. Carl Vinson?) appeared in Newsweek magazine.

Work on floating-point math chips began, but not implemented (other than making room for them on the I/O board).

PERQ Systems had redesigned the entire PERQ CPU as that it would require only five chips. Note that this new CPU would have been the functional equivalent of the original microcoded CPU board. Unfortunately, PERQ was unable to obtain the required financing, in the amount of approximately one-million dollars, which was necessary to put the redesigned CPU into production. This new CPU would have increased reliability and speed as well as decreasing the size of the system. A reduction in the size of the PERQs cabinet was to be made; it would have resembled one of todays "tower" type cases for IBM-type PC's.

The next project following the redesigned CPU was to have been a true 32-bit PERQ system.

New company, ETI, started by Jim Gay after he left PERQ Systems (1986?). This company had something to do with artificial intelligence stuff and some of the PERQs Lisp stuff?

Spider Systems possibly performed some early implementation of networking for the Perqs [BF].

"I recall Nick Felisiak of Spider (now there's part of the Perq legacy: all those spin off companies formed by the redundant Perq 'brains') telling me that someone was writing a 'book' about the Perq. Did anything come of that?" [Charles Curran, July 30, 1993, PERQ-Fanatics mailing list.]

"3RCC eventually became [part of?] Accent Systems, along with Unilogic. Unilogic called itself Scribe Systems for a while, and S.A.Y. bought

it all out. While I worked for Accent/Scribe, the Accent folks bought the assets of a company from Taxachusetts called Canaan. Now *there* was a box to hate – it was an IBM 370-compatible box the size of a PERQ that ran VM.CMS." [Anthony A. Datri, July 30, 1993, PERQ-Fanatics mailing list.]

On a seperate note from the timeline - since ICL was bought out by a Japanese company a year or so ago, could this have anything to do with the group (Rikei) who took the majority of the PERQ stuff from Accent Systems to Japan?

Chapter 3

Accent Systems Corp.

Originally formed as a computer marketing and service organization

Expanded its marketing strategy to encompass software and hardware integration by forming its Application Products Division which focused on computer integrated manufacturing requirements within the aerospace, automotive and federal/defense markets.

Sold the MaxPro environment for the PERQ LN-3000 series systems. Maxpro was designed to integrate applications for decision support, computer aided engineering, computer aided drafting,computer aided manufacturing and text processing through a common front end (user interface?). Software available, either separately or bundled together, including an image scanning package, WYSIWYG word processing system, a drawing program for CAE/CAD applications, a free-hand sketching program and spreadsheet package. Text and illustrations could be combined into a single document using an interactive word processor or optional batch document composition system (Mint?) for output on the Canon-CX laser printer. (Was MasPro related to IDOCS?)

Selected MaxPro packages were available for hardware including Apollo and DEC MicroVAX.

```
Address of Accent at the time that MaxPro was being marketed:

  5907 Penn Avenue
  Pittsburgh, PA 15206

Phone number then was: 1-412-361-3200
```

```
President then was: Richard Cox
```

Chapter 4

More PERQ History

Three Rivers computer Corporation produced several types of computer/electronic equipment besides the PERQ computers including:

- High performance calligraphic (vector) display processors
- Color raster display equipment
- Management information presentation systems featuring color graphics
- Special purpose memory systems
- High Fidelity Audio A-D and D-A conversion systems

Brian Rosen's reply to my question about the PERQ's speech circuitry:

> Three Rivers made a 16 bit ADC and a 16 bit DAC before we made Perq's. They were intended for research into computer speech and music. 16 bit converters in those days were unheard of (way before CD was a dream). We did "is it live or is it Memorex" tests with it - microphone and good speaker system with a switch - in one positiom you got "live" music. In the other position you got A to D followed by D to A. You could not hear the difference!. This, in 1974! [BR, mailing list, Dec. 7, 1992]

4.1 Graphic Wonder

In a message to Chris Lamb from Henry Spencer on Aug. 1, 1992, from Chris Lamb's message of Aug. 7, 1992:

"Any interest in other products of Three Rivers? I ask because we own Graphic Wonder (aka GDP-1) serial number 001, 3RCC's very first commercial sale... and it's in storage and slated to be disposed of. The old 11/45 it was on has been spoken for, but the buyer isn't vert interested in the 3RCC stuff."

In a message to Chris Lamb from Henry Spencer on Aug. 5, 1992, from Chris Lamb's message of Aug. 7, 1992:

"They tried to do the Graphic Wonder as a production item, but it wasn't very successful, only a few customers. People wanted smarts rather than speed, mostly, I gather."

"I'm definitely the contact for this, and I probably know more about the thing than anyone hereabouts now. The first thing to know is that this is definitely a pdp11-specific beastie: it wants to talk to a Unibus. It looks like a chunk of dual-ported memory plus some control registers. The hardware is essentially a simple-minded processor; you load a program for it into the dual-ported memory and say "go". If you can find enough memory for them (there is a subroutine mechanism, but the GDP-1 was definitely a bit short on memory), you can refresh something like 100,000 vectors before the flicker gets too bad. It's not too intelligent, e.g. no hardware coordinate transformations, but it is fast."

"Physically, it is a great big HP monitor plus a couple of rackmount boxes full of electronics (one for GDP itself, one for memory) plus a power supply You might have to put together some fresh cabling; some of the bits and pieces got cannibalized for other things, I think. We also have the electronics (less monitor) for a raster color display they built, probably of value mostly for spare parts (in particular, it has two more of the memory boxes)."

Three Rivers was started in **1970** and incorporated in **1974** and entered the "untapped" workstation market in **1980**.

The term PERQ is derived from the word perquisite - an exclusive right to added value. The design philosophy of the PERQ embodies the term perquisite by providing a personal computer system that accomodates people, rather than one that requires people to adapt to it.

Note that heresay says that PERQ was an acronym for Pascal Eval Real Quick.

As recently as **1990**, there was a listing for Three Rivers Computer in the Thomas Register (check this out again for better details). Although a phone number was listed, it was no longer a working number.

"PERQ systems has evolved in its 10-year history from the pioneer developer of high-performance graphics workstations to the manufacturer and

marketer of the world's most advanced high erformancs graphics network systems." (most likely written in 1984)

PERQ Systems Corp was founded in 1974 as Three Rivers Computer Corporation by a group of engineers from Carnegie-Mellon University. Their first efforts concentrated on custom designed graphics systems.

In 1980, the company pioneered the introduction of the industry's first high-performancs graphics workstation. Since that time, in conjunction with Carnegie-Mellon University, PERQ Systems has been developing Accent, a network-wide virtual memory operating system, and the LINQ open systems local area network.

In the past year (1983-1984) PERQ Systems has:

- Raised approximately $10 million in a private placement
- Consolidated operations in a new 70,000 sq. ft. manufacturing and office facility (Liberty Avenue) in Pittsburgh.
- Introduced the PERQ 2
- Changed name from Three Rivers Computer Corp. to more closely associate itself with its popular line of products.
- Installed a new top management team
- Averaged a one hundred percent growth in sales.

4.1.1 Historical notes from Chris Lamb

> The Perq was the first commercially available machine in its class. Though Xerox built the Alto and other experimental machines, they were not offered for sale to the public.
>
> The Perq could be called the first "3 M" machine: 1 MIP computing power, 1 Megapixel display (ok, the portrait was slightly less than a million pixels if you're being pedantic :-), and 1 Meg of RAM in a single-user system. That was a very important milestone in computing, and the Perq really did define the whole industry. Apollo and Sun and all the others that followed took *years* to build a system around commercial microprocessor that could compete with the Perq for graphics performance.
>
> The Perq really did "push the envelope": for $30,000 USD (in 1980) you had the most powerful single-user computer available,

> with excellent graphics performance on a 100 dpi display, a local 24MB disk, a megabyte of memory, sound output , RS232, GPIB, and Ethernet ports, and that in and of itself was an astonishing feat.
>
> Outside of the US, ICL was pushing PNX - which just happens to be the first Unix with an in-kernel window manager, and distributed file service, etc.
>
> Inside the US, Accent was paving the way in distributed OS research - with an elegant microkernel (they squeezed it into 16K of WCS - with two language instruction sets - and room to spare!) and a new and powerful window manager (when X was still rev 10? and still a bogus hack floating around MIT).
>
> Perq "Spice Lisp" was the first workstation implementation of Common Lisp.

[CL, Nov 2, 1992]

4.2 Alt.sys.perq

On **March 2, 1993**, I posted the proposal for alt.sys.perq to alt.config. Malcolm J. Shute was very helpful with the drafting of this proposal, and was extremely encouraging about the newsgroups creation. In fact, he suggested that such a group be created. I sent the control message for the creation of this newsgroup and Uunet added it to the groups they carry on April 5, 1993.

4.3 PERQ-Fanatics Mailing Lists

The original list was started by Chris Lamb around **November (exact date?), 1992**. A while after he stopped working at TSSI, he could no longer make changes to the subscribers of the list and soon afterwards, that list ended; the last message was **May 8, 1993**. Chris attempted to start another list from the computer at a company he co-founded, but due to a network connection problem, that list never became a reality.

Jamie L. Jones, in Wales, started a new PERQ-Fanatics mailing list on **June 10, 1993**. Jamie appointed Malcolm J. Shute and I to be co-administrators of the list.

In an attempt to get as many subscribers to this mailing list as possible, and in an attempt to dig up as much information about PERQ as possible,

I sent Jamie a list of all of the people that I had ever contacted about the PERQs and everyone that I knew of to have ever used, or knew anything about, PERQs. The results of this were quite interesting, to say the least! While some very useful and interesting information was posted to the mailing list as a result of this, there were also a lot of angry people who demended to be removed from the mailing list. They were quite angry that they were "automatically" subscribed to it without their permission. This "automatic subscription" occured on July 29, 1993, a few weeks after Jamie, Malcolm and myself debated the ethics of subscribing people in this manner.

4.4 Original uCode

"They told me they developed a lot of it [PERQ microcode] on a PDP-11 running RSX (or something like that). The floppy format was the same fot that reason." [Tom Brusehaver, mailing list, Nov. 30, 1992]

Chapter 5

The Accent OS

Accent was developed jointly by PERQ Systems Corp and the SPICE Project in the Computer Science Department at Carnegie-Mellon University. "SPICE" is an acronym for Scientific Personal Integrated Computing Environment.

- fundamentally designed to enhance network services
- processes that are remote communicate with each other, by the use of messages, in the same way as processes on a local machine
- is a product of over 5 years of research dedicated to achieving a very attractive alternative to time shared computing systems
- provides interprocess communication
- transparent access to network resources
- multiple language support
- demand paging
- advanced virtual memory management
- a multi-process, message based, operating system
- provides a 32 bit paged virtual address space for each user process
- workstation memory and disk space are managed by a demand paging feature which offers a 512 byte page size

- another facet of Accent substantially reduces network traffic by permitting users to access remote files, without the need to transfer all the information to their own disk. Only those file pages that are actually needed are moved.
- priority scheduling with preemption and aging
- compute bound processes (those that don't require user interaction) are automatically lowered in priority, giving faster response to current user commands
- the Accent user interface includes a multiple window display controlled by the Accent window manager. Users can simultaneously access as many as 64 different windows on the screen. These windows can be overlapped and manipulated on the screen as if they were papers on a desk. A series of icons displays the status of each active process.
- Languages supported include: Pascal, FORTRAN 77, LISP, ADA (Adaplus from Siemens/CMU?), Modula-2 and C.
- A messge-based, multiprocess operating system providing a demand-paged 32-bit virtual address space for each user process.
- designed to be a truly distributed, transparent operating environment
- network-wide interprocess communication, concurrent active windows, multiple instruction sets, multiple processes with a flexible process scheduling facility

Accent provides a transparent remote file and server prcess access; for example, a LISP program on one machine may invoke a process on a second machine, access data on a third and and send screen output to a fourth, all managed by the operating system without the program having to know where these resouces are or even understand how they are accessed.

Simultaneously supports co-equal environments which include the native Accent environment, LISP and Qnix, a UNIX System V environment licensed from A.T. & T. The fact that these environments are co-equal means that a program in LISP, for instance, has access to all the facilities of either native Accent or Qnix.

With LINQ, provides multiprocessing capabilities, a distributed file system and transparent access to network graphics, computation and data resources.

Original PR release about Accent, "PERQ Systems Announces New Virtual Memory Operating System"? Article bearing Pittsburgh phone numbers appears to originate in Anaheim, Calif. on May 15, 1984.

Developed jointly with the Computer Science Department of Carnegie-Mellon University over the last five years (five years back from 1984; 1980-1984?)

Provides high performance graphics and transparent access to network resources.

Features multiple, co-equal instruction set architectures.

A multi-processing operating system with network-wide 32-bit memory addressability. It incorporates a hierarchical file structure and supports demand paging, diskless nodes and multiple languages such as C, FORTRAN and Pascal.

Conceived from the beginning to be a portable operating system, the first release of Accent supports the PERQ Systems line of high-performance graphics workstations.

Window manager supports the covered window paradigm, supporting as many as 64 different windows on the screen simultaneously. Windows can also be overlapped.

Status reporting icons provide current performance information to the user. According to Aaron Coleman, PERQ Systems president, these icons differe from others currently available from other manufacturers; "...they show the user exactly what process is occuring and provide information or prompts. It's one of the ways in which we allow users to get maximum performance from out systems.".

"Transparency means that when a user calls on a process residing on another machine in the network, Accent will respond as if the function resided on the user's machine. The user won't need to know where functions reside in the network" [Coleman]

System functions on the LINQ network are activated by sending messages to the appropriate server process through the Accent kernel. The kernel provides low-level support for the operating system by managing interprocess communication.

Allows optimal software development and operational flexibility because it supports multiple, co-equal, environments. These environments include the native Accent, a LISP environment for artificial intelligence applications, as well as an advanced UNIX System V implementation, QNIX. These environments can reside simultaneously under Accent. "For example," says Coleman, "a window on the screen can be running LISP while at the same time a second window is running QNIX, and a third is operating under the

native Accent environment. Programs written in LISP or under the UNIX operating system can be easily ported to Accent using this feature."

The Accent kernel provides facilities which offer unique microcode-implemented language support that yield multiple co-equal instruction set architectures.

Each process utilizes an instruction set that is optimized for the language on which the process is written; Pascal, C, FORTRAN or LISP.

Says Coleman, "Accent changes instruction sets dynamically, depending upon the language of the process it is executing. This allows processes to run more efficiently."

Allows processes in different languages to communicate with each other through Accent Interprocess Communication (using MatchMaker?)

- Priced at less than $1,000
- Available for shipment in July, 1984
- integrated on-line help facility
- icons for window control and process status
- windows can be on local machine or on another machine on the network
- progress bars in both the icon and window title
- over 100 utilities (oh really???)
- virtual terminal emulation (what does this mean?)
- Accent architecture can be divided into three basic levels:
  - Accent kernel
  - co-equal environments
  - Accent window manager

5.1 The Accent Kernel

Accent employs a small kernel. An Accent process consists of:

- a 32-bit virtual address space, process state
- process state (including the state of any macrocode and microcode registers

- access to one or more port capabilities

An Accent port is a kernel-managed and protected simplex communications channel.

Message passing is the sole means of communication in Accent, both process to process and process to kernel.

One of the Kernel's main functions is to handle the interprocess communication (IPC). Most other operating system functions are provided by sending messages through the IPC facility to server processes.

Server processes may reside on the local machine or may be accessed through the network. Neither the client process nor the server process ever needs to know that the partner in the transaction is not on the local machine.

Provides priority scheduling with preemption and aging; an aging algorythm automatically lowers the priority of a compute-bound process, thus allowing fast response for highly interactive processes. This is a typical situation found in makeup situations where compute- bound background tasks must not interfere with cursor tracking or placement routines.

Memory and disk space are managed by a demand paging feature in the kernel; this paging feature is responsible for handling references to virtual addresses which are not currently resident in physical memory. This job is done completely by passing messages between the kernel, the pager, and other server processes. The Accent message system and virtual memory systems interact so that it is possible to send a very large (megabytes) message between two local or remote processes.

Provides facilities which offer unique microcode implemented language support, yielding multiple virtual machine environments. Under Accent, each process utilizes an instruction set which is optimized for the language in which the process is written. Accent changes instruction sets dynamically depending upon the language of the process which is executing.

The file system incorporated into the Accent kernel features a tree-structured directory. Reliability has been enhanced through the use of special hardware (what special hardware?). The file system is transparent across the network, so access to remote files is the same as access to local files. Access control lists protect file access for system security. Files are mapped into the process virtual address space and are treated as virtual memory.

5.2 Co-Equal Environments

One of the most unique features of Accent is its ability to support multiple co-equal environments.

5.3 Accent Window Manager: Sapphire

Windows act as multiple virtual displays; they are rectangular regions of the screen that contain one or more processes.

Windows can be fully visible, partially visible, partially covered or totally covered while the processes are operating as well as when they are idle.

Window size and location are dynamically alterable by the user or by program. All window manager functions can be invoked by the mouse, the keyboard or by a user program.

Up to 64 windows can concurrently exist, either on or off the screen.

Offers protected access to rectangular bitmaps, including graphics raster-op, line drawing, special string display functions and special typesetting functions.

Characters are written on the screen by passing character strings to the window manager, together with font and positioning data. Font data is stored as bitmaps.

Icons are provided to assist the user in wondow control and actively monitor process status. The Accent window manager features a unique icon system which is superior to previous work. Unlike older systems, where icons represent "closed" or inactive windows, Accent icons are always visible, whether or not a window is on-screen. Icons are used to assist in window control and in monitoring process status. Each icon contains the name of the process, indicators for Attention and Waiting-for-input, and percent done progress bars.

The window manager tracks movement of the mouse and can dynamically alter the cursor pattern when the user moves the mouse from one area to another. The cursor, roughly the size of a postage stamp, can contain a fair amount of detail. The logic behind a large (56 x 64 pixel) available cursor is that it is the user's focal point, and therefore can be used to pass important information to him. A standard set of cursor shapes is provided, plus utility software for creating new cursors (it is???).

Progress bars appear on screen in both the icon and the window title.

Reflects Accent's dedication to transparent communications by allowing users to create windows on a local machine that control processes running

on any other machine residing on the LINQ network.

5.4 Matchmaker

A program available with Accent which creates a remote procedure call to the message system, automating the linkage between client and server processes.

May be used to generate procedures for automatically sending and receiving messages between processes which may be written in any of the supported languages.

Matchmaker does all the work of appropriately packing the procedure arguments into messages (on the client side), extracting incoming procedure arguments from message fields (on the server side), formatting the response (on the server side), and returning that data to the client process. It is language sensitive, i.e. understands how to translate data representations between languages.

With a Matchmaker generated interface, clients access server functions by making simple procedure calls.

The Matchmaker generated code handles the entire message send and reply sequence.

This is network transparent, so that a client on one machine can effectively call a procedure on another machine across the net.

5.5 Microprogramming

Accent makes use of microprogramming for a number of low-level system operations.

The machine contains a 16K (48-bit word) block of Writable Control Store (WCS). Of this, a 4K block is reserved for user microprograms.

Because microprogramming is critical to PERQs own efficient use of the machine, much care has been taken to provide microprogramming development tools which are easy to use.

Microprogram development is supported by a microcode assembler and placer.

Accent supports execution of user microprograms by allowing dynamic microcode loading/execution, including calls to microcode from any of the languages supported under Accent.

In compute-bound applications such as font outline-to-raster conversion, picture compression, raster scale and rotate, etc., microcoded subroutines

can increase speeds by factors greated than 10:1.

Multiple instruction sets corresponding to the supported languages co-reside in Writable Control Store. Each language's instruction set is optimized for that language.

Accent switches instruction sets dynamically, based on the language of the running process.

Because of optimiztion and co-residency, programs written in any of the languages inherently run more efficiently, as do multi-language applications systems.

5.6 Other Info.

According to PERQ Systems President Aaron Coleman, Accent can be ported to other vendors computers, making this a highly versatile system. (prior to it becoming Mach, was this ever done?)

Accent was designed to maximize the overall efficiency of a heterogenious network of workstations with varying capabilities, as well as independent storage and I/O subsystems

Accent was a mature, fully documented and operational on several hundred workstations in production environments.

Research continued at Carnegie-Mellon nnnniversity while PERQ Systems held responsibility for further Accent development and support.

Accent is an open, non-proprietary system.

Source code was available and PERQ Systems would support a vendor wishing to port Accent to any appropriate hardware. (did this occur? on what systems? how was appropriate hardware defined?)

5.7 Accent and Printing/Publishing

Accent is ideally suited for Computer-Aided printing and Publishing because it supports:

- highly interactive graphics on raster displays under the control of an innovative window manager.
- very large programs using demand-paged virtual memory techniques.
- I/O systems which can support sustained high-speed data transfers for graphics and text data bases

- networks ranging from one workstation to hundreds of workstations, servers and other CPUs from multiple vendors.

Automated prepress systems must by definition be easily expandable at low cost, and must provide ease of interfacing specialized devices. The network philosophy and transparency provided by Accent, when coupled with Ethernet, provides the backbone for acheiving both objectives.

5.8 Porting POS Code to Accent

"I presume they just didn't have the manpower and time and money to make Accent have a POS-compatible shell or be able to detect POS binaries and run them in a special mode...There's nothing in Accent that would prevent that, and in fact, it would be a neat project! Obviously old POS programs would have to be forced to run under Sapphire and use Accent-style device handling, but by writing new versions of the old standard POS libraries (screen, memory, io_*, etc.) that simply mapped the POS calls to the proper Accent server (or the kernel) you can basically fool the old programs into behaving themselves. They'd run a bit slower of course, if they were highly-rasterop intensive, since most POS programs always assume they can blast bits wherever they want, without any overhead of a windowing system. But I don't think the technical issues would be that overwhelming if there was time and energy to do it." [CL, Oct. 10, 1992]

"I believe that we thought that all applications would be migrated to Accent, and POS would die. POS always seemed to be a temporary solution. It would have been possible to emulate POS, but it may not have been fast in some cases. If it were done right, it would have been possible to have multiple POS windows."

5.9 Accent S5

According to Dave Callen, Accent S5 is a bit buggy... I haven't verified this, since I've been using Accent S6.

5.10 Naming of Accent

> "CMU planned to port Accent to every computer they owned. Accent of course is the prototype of Mach, which is ported to every computer CMU owns. There was, as I recall, a debate about what

> to call it, and Accent won. I don't recall CMU saying we could not use the name [Solar], I remember it as being some of us didn't want to call it Accent. By the way, do you know why it is called Accent? It is so named because it is a message based operating system. Message -¿ msg -¿ MSG -¿ Monosodium Glutimate -¿ Accent!"
>
> [Brian Rosen, mailing list, Dec. 9, 1992]

Chapter 6

The Action List

"Action" items that were discussed at a meeting on Jan. 10, 1983 at Alpha. What, or where, was Alpha?

Present at meeting:

```
Brian Rosen
Jim Gay
Ed Fredkin
Geoff Potter
Jeff Howell
Ken Young
Joe Novak
```

Decide if a backplane stiffener is necesary to prevent bowing when inserting cards. To be designed by ICL is needed. Assigned to Geoff Potter. It was determined that this was needed; prototypes were to be procured. Assigned to Ed Maples.

Determine if any changes to backplane are necessary to support landscape. Assigned to Tim Taylor.

Check wiring of connectors for EIO to backplane. Jeff Howell.

Design and modify Z80 test code and PROM. Assigned to Dirk Kalp and Bill Hulley.

Document kludge display cable assembly for production. Ken Young.

Build I/O kludge cables for Jan. K1 machines. Andy Verostic. (Note: Andy Verostic was later at Megascan)

Document production cable from vendor (MOLDCON). Ken Young.

Design PAL on backplane using a VAX program. Brian (Rosen?).

Design fixture for setting time of day clock and write program to set clock with fixture. Bill Hulley.

(note: this must be the device and software mentioned in the ICL T2 Service Guide)

Design appropriate insulation and handling procedures for TOD clock. Dale (?) at ICL.

Determine most appropriate fix for the filter frame that is visible through front grill. Ken Young.

Prevent puncture of Ige connectors - determine most apropriate fix (shorter tails & insulator) and document it. Joe Schmidt (ICL)

Mounting on +55 supply to be improved. Design better mounting arrangement. Ken Young.

Paint plenium of Jan. and Feb. frames in-house. Bob Tysarczyk.

Run thermal tests on entire unit, esp. CPU. Geoff Potter.

Lay out labels. Jim Taylor.

Japan floppy voltage: will floppy withstand 100V, +/- 10V? Yes. Geoff Potter.

Test new tablet PCB, validate artwork. Jeff Howell.

Design and document changes to landscape housing. Ken Young.

Prepare burn-in software. Lee Harris.

RFI test for FCC regulations. Geoff Potter.

Modify boot switch to allow easy operation at any angle of tilt. Geoff Potter and Ken Young.

Design new layout of backplane. Geoff Potter.

Determine if power supply mounting can be changed to make floppy removal/installation easier. Ken Young.

Respecify gasketing on back of cabinet/backplane. Ken Young.

Improve shock hazzard protection: design covers for all AC wiring to minimize chance of accidental contact. Dale (?) (ICL)

Design labels warning of shock hazzard. Ken Young.

Determine if barrier strip harness can be redesigned to minimize possibility of incorrect assembly. Ken Young.

Circuit breaker service difficult and hazardous; redesign cabinet to mount circuit breaker to frame. Dale (?) (ICL)

Paint adhesion has been a problem; determine if problem has been fixed by vendor. Ed Maples and Ken Young.

Rear panel connector mounting is difficult; determine if it can be improved. Geoff Potter.

Determine if DDS visibility can be improved; no change. Geoff Potter.

Check speaker mounting to see if accoustics are acceptible; redesign mounting if necessary. Andy Verostic.

Go around and get more feedback on keyboard key legending, respecify if necessary. Art Lim.

Chapter 7

Adverts and Etc.

Miscellaneous excerpts from PERQ advertisements & Historical Info.:

7.1 PERQ-1

"Human engineering: designing systems for the people that use them is evident through the design of PERQ."

"PERQ is designed to live in your office, plug into the wall, utilize existing air-conditioning, fit in with your furniture, and not take up too much of your valuable desk space. All of these considerations are part of Three Rivers committment to develop systems for people."

"Maintainability is another important facet of PERQ. From its innovative construction techniques that allow a reparman to replace any module in 15 minutes using only a #2 Phillips screwdrver, to its diagnostic display system that tells you what's wrong with your system. PERQ shows that maintainability is an integral part of its design concept."

Note: the following example from an ad seems to demonstrate that PERQ systems was "clutching at straws," unable to clearly target its intended buyers. Much literature tends to aim towards scientists and engineers as users of PERQS, however the following excerpt from an advertisement seems to clearly be aiming at the wrong people and trying to make office work appear overly complicated:

> Managers and office workers use PERQs to solve the complex problems of the office of the future on one system. The PERQ is a word processor, a mail system, a filing system, and a large computer all in one at your desk; and it can communicate with other Ethernet compatible office equipment.

The PERQ system is a highly integrated set of components designed to provide a complete computing environment for a single user. PERQ is a personal computer with sufficient power and facilities to perform a wide variety of tasks found in offices, research facilities, business and government. By concentrating on the needs of one person, Three Rivers Computer Corporation has been able to offer computer users a very powerful tool for computing needs, at an attractive price. The key to the small size and low cost of PERQ is that all components of the system were designed together, with the goal of minimizing system complexity.

PERQ combines the work capacity and processing speed of a minicomputer with the price of a traditional microcomputer and adds a graphics capability that is superior to both.

The foundations of PERQ's high performance are its large memory capacity and processing power, giving it exceptionally large computation ability and speed of operation, and enabling the flicker-free image on the display to be manipulated in the most complex ways, with such speed that even realistic animation is possible. PERQ's unrivalled graphics capability has put it in a league of its own.

The graphics tablet works with a variety of pointing devices for point-and-act efficiency in using pop-up menus and and in the manipulation of screen images.

PERQ Systems believes that product support and corporate response to emerging market requirements are fundamental. In developing PERQ AI, many experts from major universities (which ones aside from CMU?) and research institutions were consulted. In supporting PERQ AI, PERQ systems is always responsive to customer needs - from installation and training through on-going customer service and product enhancement.

According to PERQ Systems Corp. President Aaron Coleman, "PERQ AI represents a breakthrough in artificial intelligence workstations as well as a symbol of our transformation to a company providing high-performance network systems."

[from May 15, 1984 press release]

"Back in 1980, we pioneered a new concept in computing"

Markets PERQs were aimed at:

- electronic technical publishing
- university and corporate research
- artificial intelligence
- federal and special systems (Navy and CIA)
- computer aided design (CAD)
- computer aided engineering (CAE)

International sales: nearly 45% of sales were international. Overseas, PERQs were sold through these major distributers:

- In the UK, Europe, Asia and Australia: International Computers, Ltd. (ICL), Europe's largest computer company.
- In Japan by Rikei.
- In India by OMC Corp.

7.1.1 PERQ Systems and cooperative agreements:

PERQ Systems has sought cooperative relationships with other companies such as ICL in the United Kingdom and OMC Corp. in India, involving joint product development, marketing and manufacturing activities. These relationships have contributed to the acceleration of PERQ Systems product planning and development efforts.

PERQs were shipped with a bootable copy of the most recent POS operating system (were later models shipped with both POS and Accent?)

Where marketed and sold by ICL, before the 3RCC name change, the PERQ was called the ICL/Three Rivers PERQ.

Blurbs from the ICL PERQ T2 Service Guide:

"The system satisfies the needs of scientists and engineers whose work requires a combination of dedicated processing power and interactive graphics facilities."

"The product is subject to continuous development and additional facilities will be introduced from time to time."

Chapter 8

Advent Systems

Note: Prolog Software's Picasso program was copied from Advent System's Paint software.

Chapter 9

Autologic Development

The Autologic Design Document references the following DARPA Internet documents:

- Assigned numbers, RFC 790, J. Postel, Sept. 1981 [ASSIGN]
- Internet Protocol Specifications, RFC 791, Sept. 1981 [IP]

3RCC: developed a reliable, flow controlled protocol, for Autologic called MESSAGESYSTEM.

BOX: the hardware configuration to be supplied by Auscom tp support MESSAGE.

SERVER: the total software package in the AUSCOM box

Chapter 10

PERQ Circuit Boards

Due to the "weird" form factor of the PERQ boards, it would be very expensive to custom fabricate them. [Dave Callen]

"Dave (Callen) said that all the scrap dealers refuse to take Perqs because apparently the circuit boards are all "cadmium plated" or something like that, and they can't reclaim the "good stuff" from them. There IS justice in the world! They can't just dump all the remaining Perqs for scrap 'cause nobody will take it! Hooray!" [Chris Lamb]

Chapter 11

PERQ Bus System

Communication with the input-output subsystem is by means of the 16-bit bi-directional I/O data bus (IOD). Data read from the bus is available to the A side of the ALU. The ALU result is buffered for transfer to I/O and memory boards through the bus.

Both CPU and I/O boards can access the memory through the memory data and address highways. The CPU uses the I/O data bus for display generation control.

The memory highway and I/O data bus are used for information transfer with the CPU and memory boards.

I/O devices are prevented from accessing memory if the hold bit of a microinstruction is set.

11.1 Memory Data In (MDI)

The 16-bit words on the memory data in highway (MDI) are loaded into a buffer and a parity bit for that word generated. Dependent on store function, 1,2 or 4 words are collected and written to the array.

The fixed disc and other channel controllers on the input-output boards have direct memory access and use four word data transfers.

A bad parity facility allows the parity checking logic to be tested.

11.2 Memory Data Out (MDO)

The four words each with a parity bit are loaded from the array to the memory data out buffer. Words defined by the fetch function are enabled

on the memory data out highway (MDO). Parity is checked for each word enabled.

11.3 IO Bus (IOB)

The I/O bus (IOB) connects the CPU to I/O devices. It consists of an 8-bit address (IOA) which is driven from a microword field and a 16-bit bidirectional data bus (IOD) which is read via AMux and written from R (Result).

Chapter 12

Computer Conservation Society

```
Article 586 of alt.sys.perq:
Path: mystica!uunet!spool.mu.edu!howland.reston.ans.net!pipex!demon!envex.demon.co.uk!c
From: chris@envex.demon.co.uk ("Chris P. Burton")
Newsgroups: alt.sys.perq
Subject: Computer Conservation Society (LONG ARTICLE)
Message-ID: <760740223snz@envex.demon.co.uk>
Date: 8 Feb 94 20:43:43 GMT
Sender: usenet@demon.co.uk
Reply-To: chris@envex.demon.co.uk
Organization: Envex Services
Lines: 170
X-Newsreader: Demon Internet Simple News v1.27

Since the PERQ is what we might call a vintage machine, I assume readers
are interested in such machines and might not have heard of, or would
like to know more about, the Computer Conservation Society (the CCS).
The CCS is a Specialist Group of the British Computer Society and was
started about four years ago after an approach from the Curator of
Computing at the National Science Museum in London. He is responsible
for the care, for ever, of machines in the collection, but with little in-
house expertise available as to how to care for them; and he was often
approached by old-timers offering their skills. The original idea of the
CCS was to try to bring these together in an organised way. For the
```

first three years, the Science Museum took on a BCS officer to manage the ‘‘Information Age’’ project, and he was allowed time and facilities to also act as Secretary to the CCS, thus ensuring it got off to a stable and solid start. Although he no longer works for the Museum, he is still the Secretary, working from home.

Broadly the objectives are:

Promote the conservation of historic computers
Develop awareness of the importance of historic computers
Encourage research on seminal historic computers.

Hardware, software and user applications are all included in the remit.

So far as I know, there is no equivalent in the UK to the Charles Babbage Institute in the USA, though there are academic departments, e.g. at the University of Warwick, with an interest in the history. The CCS thus provides a focus for these activities.

What counts as historic? As a rule of thumb we would say it should be at least twenty years old, but that shouldn’t rule out important younger things which might need rescuing. So early IBM 360s and ICL 1900s count. Commodore Pets count (yes, there are dozens about). PERQs count, just. I saw two in the Science Museum archive store about a couple of weeks ago. Visicalc counts (pun not intended). I think it will be some time before 486 PC clones count. BESM counts (we are very interested in Soviet cold-war-vintage machinery).

There are roughly 300 members of whom about 10\% are actively involved in some way. Membership is open to anyone interested in computer conservation and the history of computing, and is currently free. Members give their time voluntarilly. There are half a dozen Corporate Members including ICL, Unisys, Digital, Bull HN Information Systems and Vaughan Systems, who donate a large amount of money annually to pay for running the CCS.

In order to meet the objectives, the CCS does the following things.

WORKING PARTIES

There are a number of working parties, each led by a chairman, and

containing typically a dozen members who are prepared to give some time and effort, typically a half day every six weeks. Working parties are formed as required and the current ones are:

Elliott 401 WP. Very active, currently conserving and restoring to working order this very important, unique, valve machine built in 1952 and in the care of the Museum.

Pegasus WP. Not quite so active, but has maintained in working order the also very important valve machine Pegasus, made by Ferranti Ltd from 1956 on. Not yet on display in the Museum

Manchester Group Pegasus WP. Newly formed working party operating at the Manchester Museum of Science and Industry to conserve, but perhaps not restore to working order, their early Pegasus, recently ‘‘discovered’’.

Elliott 803 WP. Small but active, have restored and maintain the Museum’s Elliott 803 which was an early transistor machine.

DEC WP. Active, maintains the Museum’s several sorts of PDP8 and a PDP12.

S100 Bus WP. Active, masses of Altairs, North Star Horizons etc.

Totalisator WP. Suspended, awaiting space and expertise. Yes, it is considered that the electro-mechanical dog track totalisators from the late 1920’s are large-scale, multi-terminal, real-time computers. The Museum has three, I believe, in storage at the moment.

Software and Emulation WP. Not active, but a lot of work on emulation, see below.

Working parties work on Museum objects under very strict curatorial rules and procedures. Because they can only meet infrequently, progress is at a much slower pace than in say industry or academia.

ARCHIVING

The CCS has a good relationship with the National Archive for the History of Computing based in Manchester, with loosely separate spheres of interest.

There is a large and growing collection of manuals, manuscripts, sales literature and so on from the early days. It is catalogued but not easily accessible yet.

SIMULATION AND EMULATION

Hard to draw the line between them. Quite a lot of interest has been shown in the museum world by some of the simulators written by members to model the behaviour of early, and in some cases extinct, machines. The techniques are applicable to any large, functionally opaque object. At the moment simulators for EDSAC, Pegasus, Elliott 803 and Ferranti Mercury exist, and the Elliott 401 is under development. An emulator for the original Manchester Prototype has been obtained by a devious route from the net, but I have yet to work out it's provenance.

SEMINARS

At least one all-day seminar is held at the Science Museum annually, where usually pioneers of some system give talks and discussions. This year's will be on the IBM 360, coinciding with the 30th anniversary of the launch.

LECTURE PROGRAMME

A programme of evening lectures is given during the year on relevant topics.

PUBLICATIONS

All members receive the quarterly bulletin called ‘‘Resurrection’’, currently free of charge. This has short articles, transcripts of lectures and reports on work in progress.

INFORMATION CAPTURE

The CCS recognises that not only are early machines dying off, but so are many of the pioneers. An urgent programme of interviewing and video- and audio-recording pioneers has been started, but is not progressing well due to lack of suitable interviewers.

The above paragraphs describe most of the CCS activities. However,

ambitions are wider. Some members are also active in getting Bletchley Park, where cryptanalysis was done in the last World War, together with some of the remaining Huts, saved for the nation. It looks as though they will succeed. The very large site and buildings will, among other things, house a Museum of Cryptography, and possibly a Museum of Computing. It will provide a permanent and spacious base for the CCS, which will be able to work on old systems of its own as well as of the Science Museum. It is expected that there will be adequate storage space, so that early artefacts and documents can be properly stored and worked-on. In the meantime, the CCS is grateful to accept objects, particularly if the owner can hold on to them for a few more months.

Finally, although this posting has been intended to expose the activities of the CCS and not as a recruiting drive, if you feel you would like to join then send an application to the Secretary of the CCS, 15 Northampton Road, Bromham, Bedfordshire MK43 8QB, UK. You will be put on a ‘‘steam’’ mail list on a database. He would like to know:

  Your name and address, post code and phone number
  Your BCS membership number if any. (membership of BCS is NOT a requirement)
  Your personal experience of which early computers as what (designer,
    builder, maintainer, user or whatever). Younger people not experienced
    on early machines are particularly welcome if they are willing to
    learn from us old-timers while we are still lucid.
  Your experience with hardware technologies (delay lines, drums, tapes,
    valves, early transistors, card and tape readers and punches etc)
  Do you have early programs, either tape, cards or coding sheets which
    you are willing to share by copying.
  Your willingness to help with recording the history of computing.
  Your willingness to help with restoration of computers.

Ask him if he has a recent issue of ‘‘Resurrection’’ for you to sample.

Sorry, very few of us have email access, so we have to work via postal service.

{I have not seen any other newsgroups to cross-post this to - .folklore is not quite the right one. If anyone has any other ideas I’d be glad to cross-post.}
--

Chris P. Burton, not far from Oswestry, Shropshire, UK.

Chapter 13

Real Time Clock

A crystal controlled clock-calendar chip, MSM5832, with battery backup is used. The lithium battery's shelf life is approximately four years. The clock is set during board test and can be set again on site using a special tool.

Note: a 3.6 volt lithium battery is used.

The circuit allows the PERQ CPU to request the date and time from the Z80 system thus removing the need for operator entry when logging in.

13.1 Setting up the real time clock

Setting up the real-time clock on the PERQ2 EIO board can be done as follows using the real time clock setting kit. This kit consists of:

- a physical tool (Dil clip) 86013116. Possibly this is nothing more than two wires terminating in clips and connected to a switch. Eg:

```
 >--------------|/|-----------------<
clip   wire    switch      wire    clip
```

- a floppy disk 80043716 (a special boot floppy?)

Refer to T2 service guide for more info. on this.

Chapter 14

PERQ and Its Competition

PERQ Systems entered the untapped workstation market in 1980 with the PERQ.

Newer vendors such as Apollo and Sun Microsystems with low-cost workstations amd networking capabilities entered the market and proved to be formidable competitors.

Sun used "off the shelf" hardware components in order to achieve low prices. (get more proof of this)

Dataquest, in May of 1984 believed that the PERQ had a solid workstation product in the LN3000 and strong networking capabilities in LINQ. The introduction of these products signified a strong leap forward for PERQ which thrusted the company full-force into the workstation market with vendors such as Sun Microsystems and Apollo. (note: author of Dataquest article: Gail Levy)

In a message I wrote to Chris Lamb: "BTW, did you see the report on CNN about workstations? It was one of those 3-minute technology updates (or whatever they call them). First off, they defined workstation: "A souped up PC". Next, they proceeded to make an apparently carefully researched statement which said that Sun was the first workstation! I wonder how much Sun paid them to say that. I called CNN about it, but they wouldn't return my call." [RDD, November 1992] Note- the above, if put in the book, should be worded to say something along the lines of "from what I recall..." for legal reasons.

Chapter 15

The alt.sys.perq Control Message

```
Newsgroups: alt.sys.perq
Control: newgroup alt.sys.perq
Subject: newgroup alt.sys.perq
Approved: rdd@mystica.UUCP

Description: a newsgroup for the discussion of PERQ graphics workstations.

This newsgroup was discussed in alt.config, and there were only positive
responses for its creation.  Please carry this group.  Attached is a
copy of the charter/proposal that was posted to alt.config.  Thank you.


                Charter/Proposal for alt.sys.perq
                ---------------------------------

The purpose for the proposed newsgroup, alt.sys.perq, is for
discussions related to PERQ graphic workstations, manufactured by PERQ
Systems Corp. (PQS) in the United States and, under license from PQS,
by ICL in the United Kingdom.  PQS was formerly the Three Rivers
Computer Corp.  (Pittsburgh, PA).

PERQ graphic workststions were the first {\em commercial} graphic
workstations in their class ever produced, and have a great deal of
historical merit.

This newsgroup would help the owners of PERQs, often known as ``PERQ
```

Fanatics'', to maintain these systems, which is becoming a difficult task, to say the least, now that PQS in no longer in business to support these systems. Additionally, the company that was set up to support these systems by PQS, Accent Systems, is also no longer in business. Needless to say, the survival of the remaining PERQs may very well depend upon the creation of such a newsgroup.

Many of the known present owners of PERQs have the goal of preserving these machines, in working order, for as long as possible. PERQ owners would benefit greatly by having a newsgroup devoted to the PERQs, which would provide an informal forum for the discussion of PERQs. This would be a place where they can exchange help and advice, technical information, operating hints, etc. about the PERQs, and exchange stories and anecdotes on the use, history and future of PERQ graphic workstations. etc.

We feel that we need a newsgroup for reasons which include the following: The PERQ owners (``PERQ Fanatics'') have been organized as a mailing list for some time, and regularly exchange ideas and information regarding PERQs. It is estimated that only about one-percent of the current PERQs in use today are targeted by this mailing list. Not only would such a newsgroup provide current PERQ users ready access to help and advice from other current PERQ users, but it would also provide help and information from the vast pool of past PERQ users.

We have discovered that many former users of PERQs have retained an interest in these fascinating, historic, machines and have fond memories of using them in days gone by. The group of present PERQ users would greatly benefit from the ability to draw on this larger pool of advice, and past PERQ users would have the benefit of having a forum in which they could exchange reminiscences (this would also be of benefit to the present PERQ users).

No existing groups cover the use and maintainance of PERQ workstations. Therefore, the discussion of matters pertaining to PERQs falls primarily under two umbrella groups, comp.sys.misc and alt.folklore.computers. Neither group really applies specifically to the users of PERQs. Additionally, other newsgroups which pertain primarily to graphics workstations are, for the most part, used by the users of systems which are presently manufactured and supported by the manufacturers. Many of the readers of such workstation-specific newsgroups may not have an interest in the PERQs, and we'd be taking up unnecessary net resources by posting out PERQ related information in such groups.

The reason we feel that this newsgroup should be in the alt.
hierarchy, rather than the comp. hierarchy, is to attempt to determine
if a comp.sys.perq newsgroup is justified, which we hope that it will
be. However, the creation of a comp.sys.perq newsgroup would only be
justified if we can locate on the net, an order of magnitude greater
than we have at present, contributors to the group consisting of
present PERQ users, experts, past users, past experts, past PERQ/ICL
employees, preservationists, etc.

Please vote YES to alt.sys.perq! Thank you. :-)

--
Robert D. Davis   ...uunet!mystica!rdd | rdavis@jhunix.hcf.jhu.edu

Chapter 16

The PERQ CPU

- Executes approximately one million Q-codes (high-level machine codes) per second, which corresponds to a processor power of two thirds of a DEC VAX 11/780 in certain cases.
- Yields considerable speed increases when critical functions are microcoded.
- 16-bit ALU (Arithmetic Logic Unit)
- 20-bit, 17 function, ALU
- up to 1 million Pascal P-codes per second
- 4K to 16K x 48-bits of 170ns writable control store (much faster than main memory)
- The central processing unit single printed circuit board is a 16-bit wide, 170 nanosecond instruction time microprogrammed device.
- The instructions are single byte instructions followed by up to 4 bytes of parameters.
- Processor has a 170 nanosecond microcycle time.
- Uses 48-bit wide microinstructions.
- Most data paths in the microengine are 20 bits wide
- Data entering/leaving the procesor are 16 bits wide; the extra four bits allow the processor to calculate real addresses in a 1 megaword (2MB) addressing space.

- Virtual addresses are kept in a doubleword in memory, but calculations on addresses can be single precision within the processor.

16.1 PERQ-1

- The microprogrammed CPU performs many of the housekeeping tasks for the I/O devices. Microcode for the disk, for example, implements device control block (DCB) dequeing, buffer manipulation, track seek operations, etc. Using microcode in this manner reduces controller complexity significantly. Ths disk controller itself is comprised of less than 40 integrated circuits.

- The CPU is a horizontally microprogrammed processor with a 48 bit wide microword.

- The processor includes a 256 word dual ported general purpose file, a 16 level hardware implemented expression evaluation stack, and barrel shift and mask logic.

- Special hardware in the processor provides an extremely efficient mechanism for emulating byte coded instruction sets. Simple 1 byte instructions can execute in a single microcycle.

- Hardware support for the RasterOp bit manipulation primitave allows bit addressed raster data to be streamed at 64 bits per memory cycle.

- The microprogram sequencer implements two-way and multiway conditional branches, loops, and micro subroutines with a 4 level call stack.

- In order to service the needs of the I/O controllers, the microprogrammed processor can be interrupted (microinterrupt) which is implemented with a multiple priority, zero overhead change of microstate.

- The CPU is designed to be user microprogrammed. This means that the whole system architecture can be changed for efficient performance of particular operating systems or languages– to become for example a C engine, a Pascal engine, or a LISP engine. Individual time-critical routines in software can also be microprogrammed to improve speed of execution.

- The data control structures are very straightforward, with a few baroque quirks, so that user microprogramming is easy. (NOTE: use this as a direct quote from the brochure "PERQ System Architecture", TRC-12)

- The 4K writable control store is fully supported with assembler, loader and debugger, and is expected to be a popular option.

- the processor has 20 bit internal data paths so that it can manipulate real addresses without resorting to double precision arithmetic. The programmer does not see this, it merely improves efficiency.

- High speed microprogrammed 16-bit CPU

- high level language directed architecture

- integrated I/O controllers

- Native instruction set is the P-code byte sequences that a compiler generates for an "ideal" PASCAL (or other structured language) machine.

- microcode supports a P-code byte sequence that a compiler... (See above)

- executes in excess of 1 million P-codes per second

- 10-20 times faster than conventional interpreted P-code

- instruction set is modifiable so that additional languages can be supported without compromising execution speed

- a writable control store option is available for uses to do their own language development, or to optimize application programs

- the microcycle time is 170 ns

16.2 PERQ2, T2, LN3000

- proprietary, bit-slice bipolar CPU with 170 ns. cycle time

- microprogrammable to optimize speed and memory usage for programs written in specific higher order languages including Pascal, C, FORTRAN 77, and LISP to provide multiple instruction set architectures within a single workstation.
- 16K Writable Control Store
- Hardware RasterOp bitmap manipulation, enabling bit-addressed raster data to be moved at up to 32 Mbits/sec
- 20 bit real address registers
- hardware diagnostics implemented via microcode with results displayed automatically
- proprietary processor supports multiple instruction sets simultaneously resident in the Writable Control Store. This enables workstations to support entirely different instruction sets for LISP and other languages such as Pascal and C.
- 32-bit, segmented virtual address space mapped onto a 20-bit physical address

16.3 Differences between the PERQ1 and the PERQ1A CPU

Only the PERQ1A has:

- 16K writable control store
- 14-bit computable goto with the address coming from the processor shift output.
- Single precision multiply step and divide step hardware.
- A base register for addressing the X and Y registers.
- A readable victim latch.
- Ability to use a long constant in a microinstruction which pushes the expression stack

The assembler reflects features of the hardware specific to the PERQ1 or PERQ1A CPU by restricting usage of the following features:

- The percent sign which signals the use of the base register may only be used on the PERQ1A.
- MQ, RBase, Victim, LeapPop, Goto (shift), MultiplyStep and DivideStep may be only used on the PERQ1A.
- LatchMA may only be used on the PERQ1.

16.4 Writable Control Store

The 48-bit micro-instructions are held in the control store. This store consists of:

- Read only memory (ROM) holding instructions used during the first phase of system establishment and then disable.
- Random access memory (RAM) loaded during system establishment.

The PERQ T2 version of the CPU board has a control store of 16K 48-bit words (microinstructions).

Microprograms reside in a writeable control store (WCS).

16.5 Micro-Sequencer

- Addresses the control store.
- Is a complex single-chip micro-sequencer
- Has a 5-level call stack
- Inputs are jump parameters, interrupts and operands
- Allows the execution of one microinstruction while fetching the next micro-instruction from the control store.
- Uses the first byts of each Q or C code instruction to form the start address of the microcode that executes it.

16.6 Micro-instructions

Addressed micro-instructions are buffered in a 48-bit micro-instruction register from which the various fields control operand selection, mill function, jump conditions and memory access.

Single instructions combine arithmetic or logical operations with memory access and conditional jumps.

16.7 Extension Bits

The CPU basically handles 16-bit words but four extension bits are employed within the engine for simplification of multi-length operations and the generation of 20-bit memory addresses.

Four extension bits are employed within the microengine for simplification of multi-length operations and in addition to the 16-bit words handled by the CPU, generate 20-bit memory addresses.

Most of the data paths in the microengine are 20-bits wide. The data coming in and out of the processor (e.g. I/O and memory) are 16 bits wide. The extra four bits allow the microprogrammed processor to calculate real addresses in a one megaword address space. The assumption is that virtual addresses are kept in a double word in memory but calculations on addresses can be single precision within the processor. The programmer never sees the 20-bit paths.

16.8 ALU

The 17-function 20-bit arithmetic logic unit produces micro-instruction results which are buffered for use as:

- Memory address and data in
- Input-output bus address and data
- Source operand register input

Outputs (R - Result) are fed back to the XY registers as well as being fed to the memory data output and memory address registers.

If the hold bit is set, the contents of R are written into the register specified in the X field of the microinstruction word. When reset, no XY registers are modified.

16.9 A and B multiplexors

Distributed multiplexors select operands for the ALU from:

- The memory
- The I/O bus
- The CPU internal hardware

Memory data coming from the memory is sent to the ALU via the AMux.

16.10 XY Registers

The XY registers provide general working registers and local storage for certain Q or C code registers.

There are 256, 20-bit registers, any two of which may be addressed simultaneously from the X and Y fields of a micro-instruction.

Registers addressed from the X field are available to the A input of the ALU.

Registers addressed from the Y field are available to the B input of the ALU.

The XY registers are loaded with selected ALU results.

The XY registers form a dual ported file of general purpose registers.

The X port outputs are multiplexed with several other sources (the AMux) to form the A input to the ALU.

The Y port outputs, multiplexed with an 8 or 16-bit constant via the BMUX, form the B input to the ALU.

The XY registers receive input from the ALU output (R).

16.11 Expression Stack

Is a push-down pop-up E (expression) stack

Is a 16-level stack

Holds up to 16 20-bit items, providing efficient local storage for partial results.

The expression stack is loaded with selected ALU results which are then available to the A side of the ALU.

Is written from R

Is read on AMux

Is used by the intermediate code (Q-Code or C-Code) interpreter to evaluate expressions.

16.12 OP Registers

Also known as the OP file

The operation registers are loaded from memory with eight Q or C code bytes. The eight bytes are addressed sequentially by a 4-bit byte position counter (BPC), and made available to the A side of the ALU.

The most significant bit of BPC indicates that the OP register needs refilling and is employed by the microcode as a jump condition.

The first byte of each Q or C code instruction is used by the micro-sequencer to form the start address of the micricode that executes it.

Is a special 8 x 8 RAM into which opcodes and operands that are part of the instruction byte stream are buffered.

Is loaded 16 bits at a time from the memory data inputs

Its output is 8 bits wide and can be read via AMux or can be sent to the micro-addressing section for opcode dispatch.

Its read port is addressed by the 3-bit (or 4-bit?) BPC. Note that some documentation says the BPC is 3 bits wide and other documentation says that it is 4 bits wide.

16.13 Raster Op

The Raster operation (RasterOp) function is supported by special hardware.

The RasterOp function manipulates, on bit boundaries, areas of the memory holding the display screen areas (windows)

Part of this hardware also allows the manipulation of ALU results. Items of 16-bits may be shifted left or right 0 to 15 positions and masked within one micro-instruction time of 170 nS.

(CPU) Has special processor hardware and microcode, known as RasterOp, which performs logical operations on memory. This is especially useful for rapid manipulations of the display image.

The RasterOp Q-Code can be invoked by the PERQ Pascal compiler.

The RasterOp hardware performs shift, mask and merge operations on 64 bits at a time which are pipelined into the hardware. The speed is limited by the memory cycle time of 680ns for 64 bits.

16.14 Micro state register

The micro state register allows the transfer of CPU status and extension bit's value to the memory.

16.15 Shift Matrix

Is part of the special hardware provided for the RasterOp operator.

16.16 Micro-Instruction Register

Buffers 48-bit addressed microinstructions.

16.17 Micro State Register

Allows the transfer of CPU status and extension bits value to the memory.

16.18 Byte Position Counter (BPC)

Also known as the byte program counter

Is a 4-bit BPC according to some documentation and a 3-bit BPC according to other documentation.

Sequentially addresses eight Q or C code bytes from memory and loads them into the operation registers.

The most significant bit of the BPC indicates that the OP register needs refilling and it imployed by the microcode as a jump condition.

Addresses the read port of the OP file

Can be read as the Micro State Register (UState) via AMux

Chapter 17

Design

Chris Lamb was given (by Dave Callen) some of the original, handwritten PERQ design notes and diagrams - "lots of pages of test vectors all written out by hand".

> Just as a point of reference, I did, about five (ten?) years ago sketch out some ideas for a new 32-bit Perq CPU and I/O system, built around the original 16-bit microcoded design. I think their basic CPU architecture was solid and potentially scalable - obviously they took one step to 24-bits, but I think the 32-bit design was a bigger jump and would have been a real hummer. Basically, I think they were onto a RISC-like path before any nothion of "RISC" was really being discussed. My ideas (not being a hardware guy at all, mind you) were for a chip or chipset with multiple functional units running in parallel, with a very large bank of registers (like the 256 in the original PERQ) and large secondary cache, and a very intelligent memory/dma controller for better I/O throughput. I think the one thing I was really intrigued about was a RasterOp chip that worked with the memory controller, so that it could asynchronously do RasterOps in memory (or to and from or within a dedicated frame buffer) and never have to worry about memory access collisions or refresh delays. And of couse, for color machines, I advocated a "planar" approach, where you'd have one rasterop chip for each bit plane in memory, and would for simple moves, run them all simultaneously (some *really* neat tricks could come of that! Like selectable window depths, overlay planes, Z-buffering, etc). [CL, Oct. 10, 1992]

Chapter 18

PERQ Mass Storage (Disk & Tape)

The seek complete signals from the drive are not used, the microprogram provides more than adequate delays. Note: this is for 5-1/4" and 8" drives. Does this slow the system down? Does it cause incompatibility problems with some hard disks?

18.1 Fixed Disc Drive Controller, 5-1/4" and 8" discs

Connection to the drive is by a 50-way stripline from the EIO board off card connector JB. Controller commands and parameters are loaded into the controller registers from the I/O data bus and the controller returns status to the bus. The dedicated controller logic generates all the signals required for head movements, head selection and data transfer. The DMA controller handles the transfer of data between the disc controller data buffer and the PERQ memory.

Note that there are at least two types of 5-1/4" hard drive controllers for the PERQ T2 systems: one for the ST-506 drives and one, specially made, controller for the Seagate ST-412 10 MB hard disk. The two cannot be interchanged [according to Dave Callen].

According to the PERQ schematics, there are also two types of ST-506 controllers: one made for Micropolis drives and the other for Maxtor drives.

18.2 Shugart SA4000 series hard disk (PERQ1)

- PERQ has a built in 12 megabyte (formatted) rigid disk
- the disk uses Winchester technology
- with 97 ms average access time (12 Megabyte?)
- with 87 ms average access time (24 Megabyte?, SA4004?)
- 7 megabit/sec transfer rate.
- as an option, 24 megabyte capacity is available

The SA4000 is almost entirely dependent upon the host computer for the intelligent control of the disk's mechanical operations.

If a LOUD, terrible, squeal is heard: the heads may not be seeking properly due to a problem with the drive electronics. Try swapping the 4 boards on the disk containg the hard disk controller ciruitry. [Dave Callen told Chris Lamb this]

Why it takes the PERQ-1 systems so long to boot, in response to Malcolm's question about this and about why his disk sounds like it spins up to a faster speed and then stabilizes at a slower speed:

> The disk drive spin up is like most disk drives, when you turn it on, there is an accelleration phase, where the motor tries to get the disks spinning as fast as possible. When the drive nears the desired speed, the servo takes over and tries to keep the drive at the rated speed. When the drive is near the rated speed, the heads are unloaded and moved around a bit to settle them on the right place. All of this makes noise. As far as I know, it is very rare to have the disk spin too fast, and then slow down.
>
> Boot time is a compromise between diagnostic coverage and impatience. If you run a good diagnostic, you can alert the operator to a failure before things get flakey. Good diagnostics take a long time to run. This is especially true of memory diagnostics.
>
> The way to tell what it is doing is to look at the DDS. Since it tells you what failed, as it does it, it tells you what passed, so by looking at the DDS list, you can figure out what it is doing (next potential failure). It does spend a while on memory diagnostics.
>
> I don't recall all of this but, as I remember, there are two sets of diagnostics; one is a check of the micromachine and enough

> of the memory to load the boot microcode. The boot microcode loads a more extensive diagnostic, then loads memory with an image containing the operating system and the real microcode. The microcode is loaded into the WCS and the OS is started.
>
> I distantly recall some problem with disk reading taking a long time on the 1A, but I thought we fixed that one. Memory tests take the longest single time slice as I recall.
>
> [Brian Rosen, mailing list, Jan. 27, 1993]

18.3 Floppy - Shugart SA851 or SA858

Also optionally(!) available is a 1 megabyte double sided, double density, IBM compatible floppy disk drive.

Note: does the above refer to the 8" drive with a format which may be compatible with larger IBM minis/mainframes, such as the RT-11 format used by the PERQs 'floppy' program), or do they mean an optional 5-1/4" floppy drive which is compatible with IBM PCs? Also note that RT-11 is from DEC, not IBM. Why not say DEC compatible instead??? Addendum: yes, this refers to compatibility with IBM 8" disks.

Answer: yes. This refers to the 8" floppy drive.

Interface connections to the single (floppy) drive within the processor cabinet are carried by a 50-way stripline and the EIO board off-card connector JA.

The PD765 floppy disc controller chip on the Z80 bus takes commands and parameters from the Z80 CPU. This chip has direct access to the Z80 memory for data byte transfers.

The track to track seek time is 8 milliseconds.

Serial data is transferred to and from the diskette at 62.5 KBytes per second.

Other specs:

- 8" double-sided, double density, soft sectored
- 1 Mbyte formatted capacity
- rotational speed: 360 rpm
- average access time: 96 ms.
- transfer rate: 62.5 Kbytes/sec.

- average latency: 83.3 ms.
- head load time: 50 ms.
- settling time: 15 ms.
- track to track seek: 3 ms.

Floppy drive was tested to see if it could withstand 100V, +/- 10V, for use in Japan. It can withstand this.

The front of floppy drives is slightly wider for the T2 systems than for the PERQ1 systems; the cover in the PERQ1 systems will not fit back on if a floppy from a T2 is installed. Aside from this, however, PERQ1 and PERQ-T2 floppy drives may be substituted.

Deceptive floppy directory listing:

I noticed that the same floppy can appear to be both a file-system floppy as well as an RT-11 floppy. For example: when I inserted a duplicate "Run" floppy and, forgetting that it was a file-system floppy, issued a "fl dir" command. An empty floppy directory listing appeared and showed 900 blocks to be free. When I remembered that this was a file-system floppy, I mounted the floppy and used the directory command; all of the files were still there. Note that this was a double-density floppy (was it previously formatted as an RT-11 floppy?)

Brian Rosen responded to my discovery:

> When the Perq was first invented, the computer we had to do software development on was a PDP-11. So, the first floppy disks POS could read were RT-11 floppy disks, written by the PDP-11. As I recall, we made the first few blocks of a file system formatted floppy have a header that looked like an RT-11 floppy so we could write POS floppies on the PDP-11 or PDP-11 floppies on POS. I believe that you cannot have files in POS format and in RT-11 format on the same disk; only the first few disk blocks have some commonality.
>
> [BR, mailing list, Feb. 12, 1993]

Sometimes a PERQ's floppy drive doesn't seek the disk heads to home before attempting to mount a floppy (PNX will report "failed to open /dev/flop" or "/fd not found"). Here's a solution to this problem from Jamie L. Jones (August 13, 1993):

> My disk drive wasn't seeking the disk heads to home before attempting to mount the disk - it would try to mount it / format

> it etc. from whatever position the heads were in. To solve this, I used to take "sparkie" apart, and carefully push the heads back manually. [...] As it was impractical tp take it apart all the time, here comes the kludge:
>
> I would open the door of the drive, and carefully stick a ruler in, forcing the heads back.. yuck. but it worked.
>
> Incidently, after the heads were pushed back, and the disk mounted, it would always seek back and forth correctly. Howevr, as soon as I unmounted and tried to remount, no joy. - I would have to manually push the heads back again... weird eh?
>
> Anyway, it sort of 'fixed itself' after a while (as have most of the perq problems I've had [I'm sure it's alive :-)])
>
> So, I would suggest you look at the drive as you try to mount the disk. Do the heads correctly "home" before attempting to mount?
>
> Just a thought..
>
> Jamie.

18.4 PERQ2 (T1) 8", 35 Meg., Micropolis 1200 Hard Disk

"The (Micropolis) M1200 Series 8" disk drives have a built-in intelligent controller for handling mechanical operations."

Serial data is transferred to and from the disc drive at 720 kilobytes per second.

The track to track seek time is 12 milliseconds and the average data access time is 50 milliseconds.

A further 8 inch basic or slave drive may be daisy-chained from the master drive, but cannot be fitted in a PERQ2 T1 cabinet.

It takes approximately 30 seconds for the drive to come up to speed.

There is a delay in the extablishment process to allow the fixed disk to come up to speed. This is 30 seconds for the Micropolis 8" on a PERQ T1.

18.5 PERQ1 and PERQ2 with 14" and 8" hard disks

The PERQ1 (14" disk) and PERQ2 (8" disk) systems will boot from the hard disks when all of the following conditions have been met:

- Disk drive is up to speed
- PLO clock has locked to servo track
- Two minutes have elapsed since power up
- The drive has been selected

The following paragraph is from the PERQ Disk Control document:

> The control bus structures of the SA4000 and the M1200 are sufficiently different that they cannot be directly substituted for one another. Yet, they are enough alike that a simple adapter circuit will allow, with the proper software support, the M1200 type of drive to be used on an SA4000 controller. Such an adapter has been built. The adapter converts the eight-bit control bus of the M1200 to a bus that is physically compatible with the SA4000 control bus.

18.6 PERQ T-2 5-1/4" hard disk drives

(brands/models?)

- 5-1/4" Winchester disks
- 1 disk standard. 2nd integral disk optional
- 43 - 288 Mbytes capacity (Note: Accent only: POS limit approx 32 MB)
- rotational speed 3600 rpm
- average access time: 30 ms.
- transfer rate: 5 Mbits/sec.
- average latency: 8.33 ms.

Micropolis 300 series 5-1/4" disk drives:

- Takes about 30 seconds for the drive to come up to speed.
- track to track seek time is 6 milliseconds

- average access time is 30 milliseconds
- Serial data is transferred to and from the drive at 6 Mbits per second
- A further basic or slave drive may be daisy-chained from the master drive, in the same cabinet.

There is a delay in the establishment process to allow the fixed disk to come up to speed. This is 20 seconds for a Micropolis 5.25 inch drive on a T2.

18.7 PERQ T-2 W/Multibus

* Note: Multibus only works under Accent OS (???)

3459 Multibus Storage Module disk controller/formatter is a single Multibus board which provides:

- 24 bit DMA addressing
- 8/16 bit data - 8/16 bit address
- multiple memory addressing modes
- buffered (full sector), direct and cached modes
- 1:1 interleave in direct and cache modes
- intelligent caching
- bad track and sector replacement
- bit rates of 20 Mbits/sec
- control for up to 4 SMD drives
- automatic error correction
- overlapped seeks
- programmable sector and gap sizes, interleave factor, number of
- sectors/track, heads/unit
- a full track of information can be transferred to memory in one disk revolution in direct mode
- special formatting allows multisector/multitrack operations without missing a disk revolution

18.8 SMD Disk Drive

- model 3456 Storage Module Disk Drive
- storage capacity (unformatted) = 474 Mbytes
- number of disks = 6
- number of heads = 20
- average seek time = 18 milliseconds
- data transfer rate = 1.859 Mbytes per second
- MTBF = more than 10,000 power on hours
- can be mounted in the model 3411 expansion cabinet's 19 inch rack

According to Brian Rosen [mailing list, Dec. 9, 1992], there were not many PERQ systems made that used these.

18.9 Tape controller (9-track)

- the model 3458 tape controller is capable of handling 1/2" magnetic start/stop or streaming drives
- capable of controlling up to 8 start/stop or streaming, PE or NZRI formatted drives
- full 24-bit addressing
- DMA operation
- buffered, direct or streaming data transfer modes
- bus lock option during DMA transfers
- programmable interrupt option
- on-board 4K buffer standard, up to 16K optional
- automatic retry for all recoverable errors

18.10 Streaming tape controller

- part of the M/LO board
- allows for the direct connection of the QIC-02 streaming (model 3521) tape drive

Note: by building a special board (not normally available to Perq customers?), it is possible to use the streaming tape drive with either POS or Accent. No multibus board is required for this. The special tape controller board requires very few parts and may be plugged into either the OIO or the CPU option slot.

Some notes about building the board are in my message to Chris Lamb of Aug. 19, 1991.

18.11 Tape drive (9-track)

- model 3457 1/2" magnetic tape drive (mfr.: Pertec)
- fully automatic loading
- 25 IPS start/stop mode for transactional applications
- 100 IPS for high-speed streaming disk dumps
- large storage capacity
- fully ANSI and IBM compatible
- built in diagnostics
- compact size (8.75" height)
- microprocessor based
- high data reliability

Some 9-track 1/2" tape drives may be attached via the GPIB

"The Perq did support a GPIB tape drive with a 488 interface. I don't recall the name of the vendor, but Idea does nota sound rignt; that may be the product name. An old HP GPIB drive might work. If not, it wouldn't be hard to fix the code for any specific drive as long as the hardware worked." [Brian Rosen, mailing list, Dec. 8, 1992]

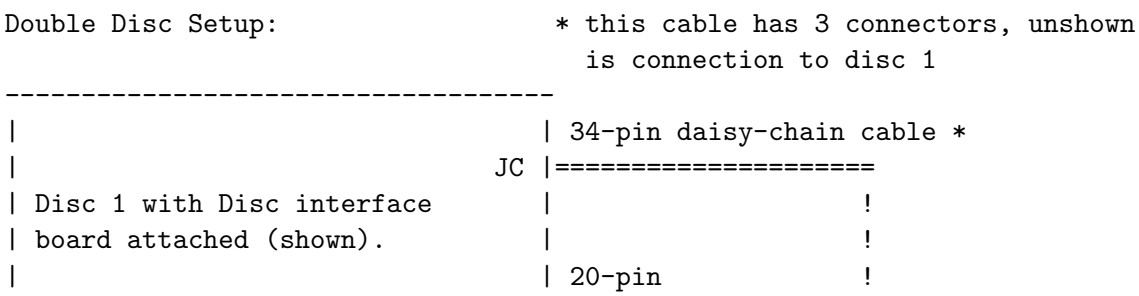
18.12 Streaming cartridge tape drive

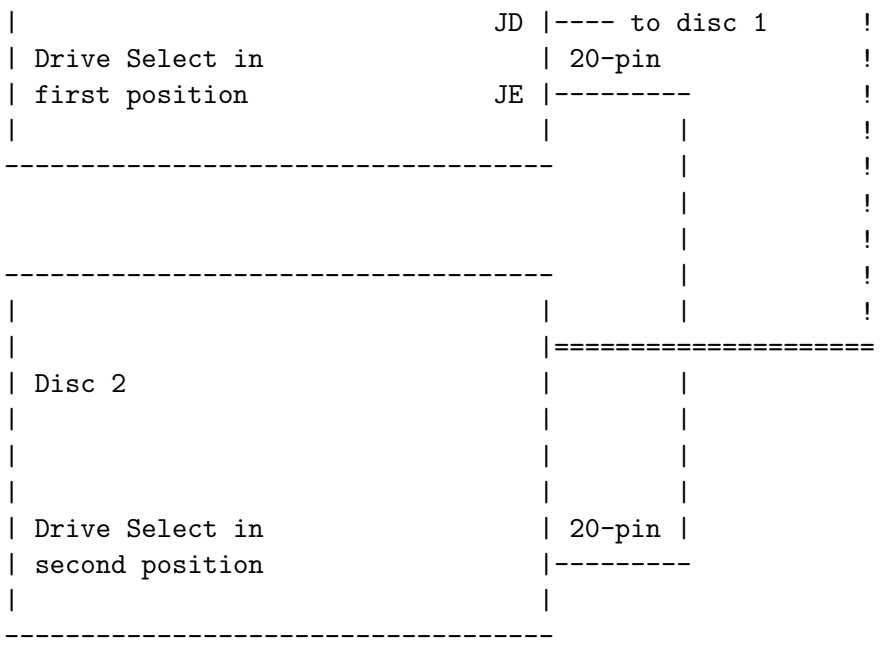
- model 3521 (tape drive mfr: Archive)
- transfer rate: 30 IPS (20 Megabyte), 30 Kbytes/Sec
- capacity: 21.6 MB (unformatted), 20 MB (formatted)
- recording tracks/form: 4 track serpentine
- recording code: run length limited (RLL)
- head format: read while write with seperate erase
- recording density: 8,000 bits/inch
- data reliability - recoverable error rate: no more than 1 in $10\hat{8}$ bits
- device reliability - MTBF: ¿= 3,500 hours, MTTR: ¡= 30 min.
- holds approx. the same amount of data as 16 8" floppies
- uses DC-300A type cartridges
- QIC-02

According to Chris Lamb, it may be possible to daisy-chain up to three or four of these tape drives.

QIC-02 tape drives use the same 50-pin connector as SCSI drives, but are not SCSI. QIC-02 is a newer, smarter interface and is used by many "PC" tape drives.

18.13 Adding a Second Hard Disc





Daisy chain cable: This is a flat straight-through conected cable; doesn't get "twisted" like the daisy chain cable for PC's. For example, if ribbon cable is laid flat out, and one socket has the connectors showing, the others will also; If one socket is face down, they will all be face down with no pin connectors showing.

Notes:

1. Only one controller (disc interface board) is needed for two drives.

2. POS will only recognise the first drive, except for the POS utilities which are used for disc formatting, etc. The first drive (or only drive) in a PERQ should be jumpered as the first drive (either a 1 or a 0, depending upon manufacturer's notation) and the second drive should be jumpered as the next one.

3. Remember that if using a drive that is not in the disk.params file, and you add it to this file, do not place it in this file as the last line! The EOF marker in this file must not be disturbed!

18.14 Adding a SCSI interface for Disc or Tape

Chris Lamb has microcode examples of how to write the microcode to set up a DMA channel.

> And if you think about it, the protocol between the PERQ and the SCSI chip would be MUCH simpler than the Shugart stuff, and I bet even less of a hassle than the EIO disk interface.Since SCSI is a very high-level from the host's point of view, it would really just be an issue of learning how to talk on the bus, and how to fit in with the other boards and talk DMA. If you think about it, that's probably all the streamer board is - a generic Perq interface that passes commands between the microcode and the intelligent conroller on the tape drive. So if I went with the second idea, of building a sort of "secondary EIO board", it would be just like how the Perq currently deals with the Z80! [CL, March 24, 1992]

Chapter 19

PERQ Display

"Recall that the Perq had its frame buffers reside in main memory, and the CPU got every other memory access (poor man's dual porting)." [Robert Colwell]

Makes use of 64-bit parallel RasterOp hardware

Data for the display screen is written by the CPU to an area of memory allocated for that purpose.

The black and white monitor is driven from the display control on the memory board.

The display assembly box holds (this is for the PERQ2 series):

- The CRT, scan PCB and transformers
- DC power regulator or mains power unit
- display unit interface panel
- loudspeaker, volume control and brightness controls.

The four display unit types are (just for the PERQ2 series?):

- The T1 portrait display F2362/22; this contains a KME assembly 80019760 powered by a +55 volts DC from the T1 processor power supply.
- The T2 portrait display F2362/24; this contains a KRIZ assembly 7734942 powered by +24 volts from the T2 processor power supply.
- The Moniterm landscape display F2362/23, assembly 7734546; with AC mains power supply input.

- The T2 KRIZ landscape displays (not available at the time the T2 Service Guide was printed, edition 1, 1984). Contains a KRIZ assembly powered by +24 volts from the T2 processor supply.

19.1 Display Control

The CPU controls the display content by writing the required bit patterns to the display data memory area.

The CPU also loads parameters to the memory display logic circuits which define the memory area employed, and the active screen size. Parameters are updated by the CPU in response to line counter interrupts from the memory display logic circuits.

The screen is refreshed 60 times a second by the memory display logic circuits. The refresh rate is defined by timing circuits on the board interleaving menory accesses with external requests. Display and cursor address counters mean that only infrequent CPU parameter updates are needed.

19.2 Video Signals

Data read from the memory array in resonse to display and cursor addresses is serialised and transmitted to the display unit. Horizontal and vertical synchronisation signals are also provided by the display logic timing circuits.

19.3 PERQ1 Display

- high resolution 768 point by 1024 line display
- bit-mapped in memory
- displays a raster scanned image
- 15” portrait display (approx. 8-1/2 by 11)
- size of A4 paper
- non-interlaced display; all 1024 lines are refreshed 60 times per second to provide a flicker-free high resolution display
- the display bit map occupies a part of main memory, and special hardware and microcode in the processor facilitate rapid manipulation of the image

- for text, charactes are "painted" into the bit map from a software defined font which can be any size, shape or complexity
- multiple fonts are supported as well as proportionally spaced characters to give the screen typeset quality.
- uses P104 phosphor
- makes use of 64-bit parallel RasterOp hardware.

The (all or some of?) PERQ-1(A) displays for PERQs in the U.K. have the following information on a tag inside them:

Video Monitors, Inc. model no.: M1000A Customer: International Computers, Ltd.

19.4 PERQ2 Display

- free-standing, raster-scanned, bit-mapped CRT
- 60 Hz non-interlaced refresh rate
- 100 dots/inch resolution
- P4 white tube phosphor
- hardware cursor user-definable up to 56 x 64 pixels
- direct refresh from main memory
- portrait screen 768 x 1024 pixels, 15" diagonal
- landscape screen 1280 x 1024 pixels, 20" diagonal
- display 1.3 million pixels
- a different document indicated that the landscape display is 19", not 20". Were there two different sized landscape displays? The other specifications for the display are the same.

19.5 Color display

Information about the display for the ColorPerq:

- 8-bits/pixel, 256 color-map entry format
- screen: 1280 pixels x 1024 lines
- Hitachi 19" color monitor, model 3619
- Just after booting, the display contained a "wild-looking purple-and-yellow picture" (Robert Colwell - also, a slide of this)
- Another document mantions a different(?) color display: 1024 x 768 pixels, 30 HZ interlaced, landscape display (Dataquest document)
- Was to be a "low-cost" system and for this this reason, an "expensive" monitor was rejected.
- When the color option was to be used, all memory would reside in the cabinet for the color hardware; this is because this much memory would not fit into the current PERQ cabinet. It was decided to provide the full compliment of memory that a PERQ is equipped to address: 1 MWord (2 Megabytes) using 256 16K RAMs plus 16 64K RAMs for parity. With 256 RAMs, 32 bytes during a single memory cycle could be looked up; an aggregate memory bandwidth of 753 MBytes/second. This bandwidth is enough to support both screens, the color and B&W monitors, simultaneously.
- 1024 x 1280 pixels x 8 bits/pixel color mapped
- Color cabinet houses four boards:
  - Color system interface board
  - Two identical 1 MByte memory cards
  - Color video card
- Each of the 1 MByte memory cards contain:
  - 1 MByte memory: 136 64K RAMs
  - Address drivers: 12 chips
  - Octal registers: 42 chips

  - Control and random logic: 30 chips
  - Testing circuitry: 15 chips
  - A total of 235 chips

- Color map: 256 x 32 (10 red, 10 green, 10 blue, 10 extra bright)
- programmable 64 x 64 x 2 cursor
- B&W and Color supported simultaneously, each with its own cursor.
- Physical packaging was to be "Beautiful in every way".

19.6 Kriz Type Monitors

Were these the ones used with the PERQ1's?

The Kriz[1] type monitors required 24V from the power supply unit, as opposed to the 55V required by other monitors.

19.7 Random Memory Displays

> Recall that the display is driven from main memory; when the OS boots up, a segment of main store is allocated and "locked down", and the display hardware is given a pointer to the start of that block. When the system crashes, usually that information is corrupted, so what gets put on the screen is just tandom sections of main memory, which may include the area that had the display image. I'd say the PERQ crashes in a more spectacular fashion than any other computer I've ever seen! :-) [CL, Nov. 6, 1992]

[1] Named after the PERQ graphics/display guru, Stan Kriz

Chapter 20

PERQ Documents Cataloged Thus Far

"1K Line Portrait Monitor Yoke Mods to Discom 127320-1" (PERQ-3?), file: 1kyokemods

"1K Monitor Theory of Operation", April 11, 1983 (PERQ-2 or 3?)

"20 Fascinating Facts You Didn't Know About ICL" (mentions PERQ)

"256K Memory Function" (Mars memory board scheme, PERQ-3), file: memtxt

"Accent: A Communication Oriented Network Operating System Kernel"

"The Accent LISP Users Guide" (The Red Pages), Aug. 1, 1984 (LISP M2)

"Accent Version S6, Amendment No. 1, April 29, 1985 (Release Notes)", files: s6amd1.pro [3Prose],

"Accent Version S6, Amendment No. 2, May 2, 1985 (Release Notes)", files: s6amd2.pro.doc, s6amd2.pro [3Prose]

"Accent Version S6, Amendment No. 3, June 14, 1985 (Release Notes)", files: s6amd3.pro.doc, s6amd3.pro [3Prose]

"Accent Version S6, Amendment No. 4, August 2, 1985 (Release Notes)",

files: relusr.pro.doc, relusr.pro [3Prose]

"Accent Version S6, Amendment No. 4, August 2, 1985 (Changes to the System Administration Manual)", files: relsys.pro.doc, relsys.pro

"Accent Version S6, Amendment No. 4, August 2, 1985 (Changes to the Programming Manual)", files: relprg.pro.doc, relprg.pro

"Accent Version S6, Amendment No. 4, August 2, 1985 (Changes to the Languages Manual)", files: rellng.pro.doc, rellng.pro

addendum to Pepper documentation, file: addendum.doc

"Autologic Development Design Document: PERQ Interface Design" by Sandeep Johar, May 7, 1982, forst draft. (IBM to PERQ via Ethernet software), 15 pages.

"BadMap V1.0x Documentation (customer released version of BadSector)", file: badmap.doc

burn-in documentation (very brief), file: burnin.doc

"Business PERQs Up for City Computer Builder"

"A Butler Process for Resource Sharing on SPICE Machines"

"C" interactions with Accent network server, file: cinter

"C Version 2.0, May 31, 1985 (Release Notes)", for Accent, files: c2rel.pro.doc, c2rel.pro [3Prose]

"C Version 2.0, Amendment No. 1, July 16, 1985 (Release Notes)", for Accent, files: c2amd1.pro.doc, c2amd1.pro [3Prose]

"Cache Control Logic", file: cache.v1.doc

Canon printer docs for Accent (including print server info.)

Canon-CX software info. (some of it is not true), file: read.me

"PERQ Color Display Manual, February 28, 1985", for use with Color Display Software Version C500.1 and the PERQ Color 500 display system, files: color5.pro.doc, color5.pro [3Prose]

"Computer Firm Leaves Basement Origins Behind" (3RCC)

"A Controlled Window Management Environment for the PERQ", David Barnes, Software Tools Centre, U.K.C.

"Cursor Design - A PERQ Cursor Builder", file: cursdesign.doc

"Daemons and Dragons: Cool Daemons" (PERQs, CMU Coke Machine)

"Data General Veteran Joins Local High-Tech Firm" (MegaScan), Pgh. Business Times

DDS description file (PERQ-3?), file: IopDDS.doc

"A Debugger for POS PERQ Pascal" by Brad A. Myers, May 20, 1983. 3RCC, 13+ pages.

"A Debugger for POS PERQ Pascal" (from user library)

"A Debugger for a Graphical Workstation" - summary and intro. only J.D. Bovery at U.K.C.

deflection yoke - technical information (PERQ-3?), file: ytext

"Design Note on Logic Section ADI4" (PERQ-3?), file: adi4.doc

"Design Note on Logic Section ASI4" (PERQ-3?), file asi4.doc

"Design Note on Logic Section CACHE4" (PERQ-3?), file: cache4.doc

"Design Note on Section PRO4" (PERQ-3?), file:

DiskTest docs (for PERQ-3?), file: DiskTest.doc

"A Display Architecture for Driving Two Different Bit Mapped Displays from One Frame Buffer" (PERQ)

"Distributed Cooperative Processes and Transactions" (Accent/SPICE)

"Five Years Ago", Sept. 19, 1990, ICL & PERQ

"Five Years Ago", Oct. 5, 1990, PERQ Systems

"Five Years Ago", Aug. 5, 1991, ICL dumped PERQ

FixBoot floppy "Tech Bulletin", file: fixboot.doc

"FixDisk V1.0x Documentation (customer released version of DiskTest)", file: FixDisk.doc

flames (a few comments about PERQs), file: flames

FontEd documentation, file: fonted.doc

"From RIG to Accent to Mach: The Evolution of a Network Operating System"

"Generalized Path Expressions: A High-Level Debugging Mechanism" (PERQs, Accent)

Kelly Hickel - his listing from "Who's Who in High Technology"

Horizontal deflection board documentation (PERQ-3), file: htext

"The ICL/Three Rivers PERQ and Distributed Interactive Computing" by J.M. Loveluck and F.R.A. Hopgood, Rutherford Appleton Labs, from Comput. & Graphics magazine, 1983.

"Integrated Project Support Environments", John McDonald

"The Integration of Virtual Memory Management and Interprocess Communication in Accent", full article + abstract

"IP/TCP for POS - Project Description" by Chuck Beckett, April 5, 1982 3 pages

"An Interactive Graphics Editor for Document Preparation" (PERQs)

"KMS: A Distributed Hypermedia System for Managing Knowlege in Organizations", (PERQs, ZOG)

"List of Fundamental Building Blocks Neded for the S/A/Z, Mixer and C/R Chip Designs" (PERQ-3?)

"List of Questions About the Vipor Video System" (PERQ-3?)

"Low Resolution Landscape Monitor - Timing for 90 Hz Frame Rate", file: ls86hz.t

mail messages to files on SPICE VAX at CMU, file: getmail.txt

Mars Closes UNIX/DOS Gap

"Mars Wins Race to Announce SPARC and Intel-based Machine"

"Matsushita Pays £8M for 62% of Office Workstations Ltd." (PERQs)

"MegaScan Allows Viewing of Images as they will be Output", P.C. Week

"MegaScan Leaving Town", Pgh. Business Times

"MegaScan Monitor, V1.0" (theory of operation and specs for custom components), file: monitr.doc

"MegaScan Receives $3.3 Million in Venture Infusion", Pgh. Business Times, 1987

misc. video pattern programs documentation (PERQ-3?), file: vid-misc.doc

"Monitors Give X-Ray Film Challenge" (MegaScan)

"MPOS Users Guide" by Bob Amber, 1982, 3RCC (document possibly incomplete), 10 pages.

"MPOS Users Guide", Incomplete. March 31, 1982.

"MProm - Program PROMs from a Bin File"

"Multibus User Guide", file: multi.mss [Scribe]

NewPart documentation, March 85

"Pascal Version 12.8f Release Notes, Feb. 1, 1985", file: ps128f.pro

"Pascal Version 12.8g Release Notes, April 10, 1985", file: ps128g.pro

Pepper help file, file: edhelp.ehelp

"PERQ and Advanced Raster Graphics Workstations"

"PERQ Color Display Concept Document" by Robert P. Colwell, J. Stanley Kriz and David Stonet, July 2, 1982. revision 2.0, 13 pages.

PERQ (PERQ-3?) focus transformer data

"PERQ Reports Revenues Double"

PERQ Systems Corp. (info. from Pgh. High-Technology Council Membership Directory)

PERQ TCP/IP, file: tcp.mss

"The PERQ Workstation and the Distributed Computing Environment" - R.A.L.

"PERQ ¡-¿ Z-80 Messages" by Bill Glass and Donald A. Scelza, Aug 14, 1981, 8 pages, intro, and table of contents.

PERQ-3 8-Megabyte memory, file: time.text

PERQ-3 SIB Electrical Definition, file: elec.doc

"PERQ-3B Architectural Model Processor Board Interface", file: psd.doc

"PERQ-5 Serializer, V1.0", file: serial.doc

"Pittsburgh's Smaller Companies Striving to Plug the Money Gap" (PERQ)

"PLP-CX Laser Printer Ammendment to User's Guide", POS, Aug. 5, 1985

"Printing Software for the PLP-10 and PLP-CX Laser Printers, March 15, 1985", files: laser.pro.doc, laser.pro [3Prose]

"Printing Software for the PLP-10 and PLP-CX Laser Printers, V1.0, May 15, 1985" (Release Notes), for Accent S6, files: lasrel.pro.doc, lasrel.pro [3Prose]

"Print Three: Desktop Publishing on the PC Comes to the Copy Shop" (they originally used PERQs)

"Program Development on a Graphical Workstation" (PERQs)

PNX man pages for: treewalk, curses (no, a cursor editor!), curses2, shades, logo, thumb, splot and minit.

"The PNX Window Manager Interface" - describes undocumented PNX system calls

"POS F.2 Test Plan" by Donald A. Scelza, Nov. 18, 1981, 3RCC, 3 pages & intro.

"POS F.2 IO Test Plan" by Jerry L. Conner, Feb. 3, 1983, 3RCC, 3 pg. & intro.

Python IOP diagnostic test descriptions

"Quick Guide to Disk Control" by John R. Rose, 3RCC, Feb. 7, 1983. Appears to be a preliminary document. 16 pages.

"Quick Guide to Disk Control", Oct. 22, 1982, 3RCC, 7 pages

"RISC Workstation puts DOS on Module" (Mars)

"Sequencer PROM Encoding" (PERQ-3?), file: seqcoding.doc

"Shamos Completes Long-awaited Deal" (Accent Systems)

"SIM - Synchronous, Register Transfer Simulator V2.5" by John Strait, Sept. 25, 1984. 6 pages.

SMD disk module documentation, file: smd.doc

"Sparc-like Stations Hobble Into Sunlight" (Mars)

"Specification Document: POS Pascal IO Specification", project name: POSF2CIO, Nov. 12, 1982, third draft, by August G. Reinig, 36 pages.

"Specification for High Voltage Power Supply" (PERQ-3), file: hvpsspec

"Specification for High Voltage Power Supply" (PERQ-3?), file: hvpsint

"State of the Newspaper Market from Both Sides of the Atlantic" (QED, Agfa, Crosfield)

"Survivor of Wilmot's Carnage Eyes Top ICL Job" (mentions PERQs)

"The Symbolic Debugger for POS G.4 Pascal" (1 page)

"Tablet Test Procedure" (PERQ-3?), file: tablettestproc

"Tablet Theory of Operation" (PERQ-3?), file: tablettheory

tape documentation (9-track?), file: tape.doc

tape drive controller initialization documentation (9-track?) file: mt-drv.doc

TAR documentation, file: tar.doc

TAR program help, file: tar.hlp

TCP bugs, file: tcp.bugs

"Theory of Operation of the P3MCNTL Wirewrap Board", also known as the WWSIB. (PERQ-3?) file: theoryopp3mctl

"Theory of Operation for Memory Board", file: memtheory.operation

"Theory of Operation of the P3MCNTL Wirewrap Board", also known as the WWSIB. (PERQ-3?) file: topp3

"Theory of Operation of the P3MEZ Printed Circuit Board" (PERQ-3?) file: theoryopp3mez

"Theory of Operation of the WWSIB Wirewrap Board" (PERQ-3?), file: theoryopwwsib

"A Three-processor LISP Machine Architecture Based on Statistical Analysis of Common LISP" (SPICE LISP)

"Three Rivers Computer" ("We're going to be very, very big")

"Three Years Ago", Aug 5, 1986, ICL dumped PERQ

timings for PERQ-3 monitors, file: stan

"A Tool for Providing Programs with Menus", J.D. Bovery, Computing

Laboratory U.K.C.

"Two Startups, One Winner" (PERQ, Apollo)

"User Facilities", a description of the userfacilities for Accent; the comments in this file describe programs that were to be part of future releases. Since this was a very long Scribe file, only the first to pages have been printed so far, and this is where the relevant data appears to be. The rest is still on disk to be printed at a later time. file: userfac.mss

"Video Controller" - 1/3 page technical description

video self-test docs (PERQ-3?), file: VidTest.Doc

virtual memory test docs (PERQ-3?), file: VMemTst.Doc

voltage conversion procedures (110 to 220 VAC), file: convert.txt

"What the KPROC Sequencer Does" (PERQ-3?), file: memtheory.operation

"Xerox Network Systems (XNS) Protocol Package for the PERQ Workstation", files: xns.pro.doc, xns.pro [3Prose]

"Xerox Network Systems (XNS) Protocol Package, V1.0 (Release Notes)", files: xnsrel.pro.doc, xnsrel.pro [3Prose]

"Xerox Network Systems (XNS) Protocol Package, V1.0 (Release Notes), Ammendment 1", files: xnsamd.pro.doc, xnsamd.pro [3Prose]

Z80 cross assembler docs, file: z80.prose

Chapter 21

PERQ File System

Each disk partition must be fewer than 32,768 blocks

Each block is 256 words (512 bytes)

Recommended partition size is 10,080, or fewer, blocks so that the Scavenger utility can handle the partition in one pass.

Chapter 22

PERQ Finance

This chapter refers to the PERQ financial documents which haven't been included in this document yet.

Chapter 23

Floating Point Unit

This unit is not used by the software and the components are only fitted on early boards.

The EIO board can hold a 8087 numeric data procesing chip with control and interfacing circuits. This sub-system communicates only with the PERQ CPU through the I/O data bus.

The PERQ CPU can load instructions to a queue in the FPU and examines it's status register for indication that results are ready for retreival.

Note: the floating point, clock, speech and timer were not supported at first release. (what first release?)

Chapter 24

Glossary of PERQ CPU Terminology

- A-side of ALU
- address space
- ALU - See arithmetic logic unit
- AMux
- arithmetic logic unit (ALU):

  The PERQ's ALU is 20-bits wide and performs 17 functions. The ALU combines its A and B inputs according to the data contained in the ALU field of the microinstruction.

  Functions of the ALU include:

```
A
B
NOT A
NOT B
A AND B
B AND NOT A
A NAND B
A OR B
A OR NOT B
A NOR B
```

```
A XOR B
A XNOR B
A+B
A+B+OldCarry
A-B
A-B-OldCarry
```

Note that OldCarry is the carry from the microinstruction which immediately preceeds the present microinstruction. It is used for multiple precision arithmetic.

The micro-instruction results that the ALU produces are buffered for use as:

1. input for source operand register
2. data in and memory address
3. I/O bus data and address

The result (R) output is fed back into the XY registers and to the memory address registers and the memory data output. If the hold bit is set, the contents of R are written to the register specified in the microinstruction's X field. If the hold bit is reset, no XY registers will be changed.

- arithmetic operations
- assembler (see microcode assembler)
- B-side of ALU
- base register
- bit boundaries
- BMux
- BPC - byte position counter
- BPC - byte program counter
- byte position counter (BPC)
- byte program counter (BPC)

- buffer
- bus
- C-code
- call stack
- central processor unit (CPU)
- computable goto
- CPU - central processor unit
- data in
- distributed multiplexors
- divide step hardware
- dual ported file
- EStack - expression stack
- ESTK - expression stack
- expression stack (EStack or ESTK)
- extension bits
- intermediate code (see also Q-Code and C-Code)
- interrupts
- IOA - I/O address bus:
- Is a part of the I/O
- IOB - I/O bus:

  The I/O bus (IOB) connects the CPU to the I/O devices. It consists of an 8-bit address (IOA)

  IOD - I/O data bus:

  A 16-bit bi-directional I/O data bus (IOD) which is a part of the I/O bus (IOB). Data read from this bus is available to the A side of the ALU. The ALU result is buffered for transfer to I/O and memory boards through this bus.

- I/O address bus (IOA) - A part of the I/O bus.
- I/O bus (IOB)
- I/O data bus (IOD) - A part of the I/O bus.
- jump conditions
- jump parameters
- logical operations
- MADR - memory address registers
- MAR - memory address registers
- mask
- MDI - memory data input
- MDO - memory data output
- MDX - memory data input, extended.
- memory address
- memory address data highway (MADR) - A part of the memory highway.
- memory address highway:

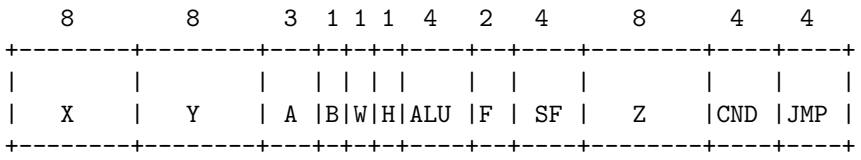
  The memory address highway is 20 bitsy wide. When a word address is placed on this highway by the CPU or I/O boards, memory is addressed and the address gets buffered on the memory board.
- memory address registers (MAR)
- memory cycle
- memory data highway
- memory data input (MDI) - A part of the memory highway.
- memory data input, extended (MDX)
- memory data output (MDO) - A part of the memory highway.
- micro-addressing section

- microcode assembler
- microcycle:

  The amount of time that it takes the microengine to execute one micro-instruction. One microcycle on a PERQ takes 170 ns.

- microengine
- micro-instruction:

  The format of a PERQ microinstruction is:



  The twelve fields if the horizontal microinstruction have a total width of 48 bits. Each of these microinstructions is a sequence of phrases produced by the microassembler from one or more instructions. The PERQ microengine executes each microinstruction in one microcycle (170 ns).

  H - The hold field. When the hold bit is set, I/O devices are prevented from accessing memory.

  ALU - ALU field: encodes the function that the ALU uses to combine its A and B inputs.

  F - Function field: controls interpretation of the contents of the SF and Z fields.

- micro-instruction beats
- micro-instruction register
- micro-instruction results
- microprogram
- micro sequencer

- micro state register (UState)
- microstate condition codes
- microword field
- mill function
- multi-length operations
- multiplexor
- multiply step hardware
- NextOp (Opfile[BPC])
- NMI - non-maskable interrupt
- non-conditional interrupt
- non-maskable interrupt (NMI)
- opcode
- operand
- operand file (OP file)
- operand register (OP register)
- operand selection
- OP file - operand file
- OP register - operand register
- processor shift output
- push down stack
- Q-code
- R - result
- R bus - result bus
- RAM - random access memory

- random access memory (RAM)
- random access store (see random access memory)
- RasterOp
- RasterOp operator
- readable victim latch
- real address
- refresh (dynamic array memory)
- refresh (display scren)
- result
- result bus (R bus)
- shift matrix
- source operand register
- start address (of microcode that executes an intermediate code)
- system establishment - the boot process
- UState - microstate register
- victim latch
- virtual address
- WCS - writable control store
- writable control store (WCS)
- X field (of a microinstruction)
- X port outputs (of ALU)
- XY registers
- Y field (of a microinstruction)
- Y port outputs (of ALU)

Chapter 25

PERQ GPIB

(PERQ1)

PERQ provides a full IEEE 488-1975 standard implementation of the General Purpose Instrumentation Bus.

The GPIB provides a simple, compatible way of interfacing a wide range of medium speed peripherals, as well as laboratory facilities, production test equipment and instrumentation.

A 9914 intelligent GPIB controller chip and transceivers handle all the interface control and data transfers. The controller is programmed by the Z80 CPU and has direct access to the Z80 memory for data byte transfers.

- The PERQ-1 GPIB is rated at 170 K Baud
- The PERQ-2 GPIB is rated at 30 K Baud

Chapter 26

PERQ Graphics

- Graphics primitives are extremely fast
- "RasterOp" moves arbitrary bit rectangles
- Operates at about 31,000,000 pixels per second
- Line drawing operates at 475,000 pixels per second
- Text display at about 16,030 characters per second, which is equivalent to 160 K Baud

RasterOp hardware provides eight ways to combine images. All run at full speed.

1. **Replace** for moving
2. **Compliment** for inverting
3. **AND** for masking
4. **AND-not** (NAND) for erasing
5. **OR** for combining
6. **OR-not** (NOR) for coloring
7. **XOR** for temporary updates or setting to white
8. **XNOR** for inverted temporary updates or setting to black

RasterOp will operate on buffers anywhere in main memory. This allows many fonts to be used at the same time.

Pictures can be generated off screen to avoid flicker (the user doesn't see the immediate image).

Real-time animation and smooth movement are easy to accomplish.

Pop-up menus and pop-up windows can restore the original screen picture when removed.

26.1 PERQ Related Graphics Files

In the Poskanzer Bitmap Collection (FTP'able from lth.se and other FTP sites), there is a bitmap file named "perq". This file is in Sun raster format and the picture is mirrored spheres displaying the word PERQ.

I suspect that this picture file was originally, in a different (.pic?) format, a part of the original PERQ picture file collection at C.M.U., as I suspect a few other files in that collection to be... not a lot of them, buit at least apparenty a few may have survived.

What ever happened to the C.M.U. PERQ picture files collection?

How can the files from the Poskanzer Bitmap Collection be converted to PERQ .pic files?

> Actually, the bitmap RObert mentions is a couple of ray-traced spheres floating above the PERQ Systems Corp. logo, which receds into the distance. One ball is reflective, the other opaque; the light source is from above your left shoulder as you view the bitmap. It is on the ftp site as a Sun rasterfile, black and white only (not colow :-(and not greyscale). The dithering algorythm used was very good, though! At a viewing distance of about 3-5 feet it looks very, very smooth! [CL, Nov. 28, 1992]

26.2 Strawberry Fairchild

The "Strawberry Fairchild" program, found on a tape with the Audre software, is sort of weird and has a bit of surrealism thrown in. It starts out with four playing cards on the screem, then this surrealistic scene and poem, about dolphins and the moon, appears as an angry looking moon appears at the top of the screen (the background of the screen in black), a poem in large, eerily shaped, letters appears below it, images of dolphins are under the poem and a the image of a dolphin glides across the screen. The next

thing to appear is a large strawberry and some closing messages. I think the name of the company that created it is something like Painted Lady Software and the author was apparently heavily involved in the AUDRE project.

26.3 Perq Cursor Files

"Once I tore apart the busy-bee image. I think there were only a couple bytes of header (3 maybe 7), and the image (no compression or run length encoding). Maybe the file format is:"

```
byte   data
0      magic number (maybe not?)
1      width
2      height
```

[T. Brusehaver, mailing list, Nov. 30, 1992]

26.4 SPic

SPic is a picture file viewer that displays large, scrollable, picture files under POS. These files are not standard PERQ .pic files, and they have no extension. Come to think of it, Sun raster files have no extension either... and are scrollable. I wonder... could this be the viewer needed for viewing such files on a PERQ??? I'll have to test this!!! The files that I've found so far are titled: Florida, Florida1, Florida2, Heather, Sandcastle and Vanessa (all of the girls have their clothes on).

There was also an XSPic.run file, but no accompanying .seg (or .pas) file. What was the difference between SPic and XSPic?

Chapter 27

PERQ I/O - RS-232, GPIB, speech, etc.

27.1 PERQ1 I/O

The IO channel multiplexes 4 high bandwidth (network, disk and two optional controllers) devices which can deliver 10 M bits per second to each device with all 4 devices running simultaneously.

For lower speed devices, an 8 bit microprocessor is included in the system to provide a low-cost method of interfacing simple peripherals. Incorporating a standard microprocessor permits the use of off the shelf LSI components to reduce the cost and complexity of, for example, the RS-232 channel and the IEEE-488 bus interface.

The high speed controllers and the microprocessor communicate with the CPU via a 16 bit IO bus.

27.2 I/O Z80 Subsystem

The system consists of a 4 MHz Z80 single chip microprocessor interconnected with supporting circuits by an address bus and a data bus.

The PERQ CPU controls peripheral operations by transferring commands and, or, data to the Z80 subsystem through either the DMA controller channel or the I/O data bus.

The supporting circuits provide the following features:

- The I/O bus interface, two 16-byte registers for Z80 - PERQ CPU communications. Interrupt lines and status bits provide hand-shaking

for transfers through the I/O data bus.

- A 32 byte buffer for transfer to and from the PERQ memory. Transfers are handled by the DMA controller.
- The Z80 memory consisting of 4 KBytes of boot PROM and 16 KBytes of dynamic RAM.
- The Z80 direct memory access chip, 8237-2, allowing the Z80 peripheral controllers to access Z80 memory through the Z80 bus. This circuit also refreshes the dynamic RAM (DRAM).
- Interrupt steering for direct entry to Z80 service routines for devices interfaced through the Z80 system.
- The Z80 system peripherals interface circuits:
  - the floppy drive interface
  - the GPIB
  - Serial input output
  - the floating point unit
  - Real time clock

27.3 LN3000

- I/O controller, for high speed devices, multiplexes up to 6 simultaneous DMA channels at up to 10 Mbits/sec. each; two DMA channels for user options.
- dedicated microprocessor to control low speed devices

27.4 EIO Board (Ethernet I/O)

The same EIO board is used for both the T1 and T2 systems. But two links, JP2 and JP3, must be fitted to the board when used with T2 5.25 inch disks. Links JP2 and JP3 must not be present when the board is used with T1 8 inch disks.

The EIO board provides control for:

- fixed disc drive

- floppy disc drive
- keyboard
- standard tablet
- OSLAN connection
- two RS232 interfaces
- a GPIB
- speech output

This board also holds a real time clock-calendar.

The EIO board circuits for control and data servicing of peripherals fall into two groups:

- state machines for the fixed disc and OSLAN connection
- the I/O Z80 microprocessor system controlling the other devices

27.5 OIO Board (Other I/O)

The optional input-output board can hold logic circuits for three interfaces. These circuits are for the:

- PERQ to PERQ strip-line link
- OSLAN connection
- Canon laser printer

The OSLAN connection circuits on the OIO board:

- drive the physical interface to the OSLAN transciever hardware through the PERQ2 OSLAN B connector.
- provide data encapsulation and link management functions implementing part of the data link layer in conjunction with the CPU and I/O boards.
- provide a channel controller interface at the memory and I/O data highways.

These circuits are functionally the same as those on the EIO board which provide the PERQ2 OSLAN A connection.

27.6 Perq RS-232

27.6.1 PERQ1

- full duplex
- high-speed serial data port
- supports asynchronous, bisynchronous and SDLC/HDLC/ADCCP protocols at up to 56K bits per second
- all line and protocol parameters are programmable
- modem control is standard
- PERQ includes a single full bit stream at up to 9600 baud (Note: this ad mentions nothing about the 56K bits per second like the other did)

27.6.2 LN3000

- two RS232C interfaces; full duplex, synchronous/asynchronous serial data ports; speeds up to 9600 bps
- Speed up to 19,200 bps using serial port A
- Speed up to 9,600 bps using serial port B (note that it doesn't work very well at 9,600 bps, howver. For example, if using PERQ as a terminal at 9600, some characters will be lost occasionally, but if using a communications protocol such as Kermit to transfer files at 9,600 bps, there appers to be no problem except for a few occasional retries at sending data).

27.7 PERQ GPIB

27.7.1 PERQ1

- PERQ provides a full IEEE 488-1975 standard implementation of the General Purpose Instrumentation Bus
- the GPIB provides a simple, compatible way of interfacing a wide range of medium speed peripherals, as well as laboratory facilities, production test equipment and instrumentation

27.8 Speech and Sound

27.8.1 LN3000

- Speech and sound generation hardware
- Can be used for a speech synthesizer to go along with voice-input circuitry in order to provide a user interface less conventional than a screen and keyboard.

27.9 Temperature Sensing

The Z80 circuitry performs some sort of voltage/temperature checking.

Code in (POS) pascal library IO modules reports voltage and temperature settings.

27.10 Misc.

Noted in POS F.2 Test Plan: "(I/O related) Bugs, restrictions, etc. encountered during testing will be reported to Chelmsford (ICL, England?) software engineering."

Chapter 28

PERQ Keyboard

28.1 PERQ1

- 60 keys
- Most likely component to fail is the PROM.

28.2 LN3000

- Detached, solid state keyboard stretches up to 8' from display (er, from the back of the display)
- 84 keys (including numeric keypad and for user definable function keys)
- ASCII character code

Chapter 29

Link Board

The link board allows the direct connection of one processor unit to another.

This feature is not generally available but it is used in manufacturing test and other special applications.

The OIO board has the link logic built in.

The link board provides for an external connection to the I/O data bus. Control and data lines from the link are isolated from the bus by receiver amplifiers on the link board. Control and data lines to the link come from buffer registers on the link board.

The link board also holds a test pattern PROM and echo facility logic circuits for checking the link.

The link, if fitted, is the forst choice as a system extablishment load device. A sequence of 16-bit word transfers is used to load 256 microinstructions into the control store RAM. There is no sum check on the data which is transferred directly to the control store.

About connecting link boards: "From what I recall of a conversation with Dave Callen, you connect them in a big "X" – the top connector on one board to the bottom connector on the other and vice versa..." [CL, April 8, 1993]

About adding link circuitry: "The link is easy to add, but you would have to have the schematics to do it. The CX support, as I recall, was more complicated." [Brian Rosen, mailing list, Dec. 9, 1993]

Chapter 30

PERQ-LISP/AI

"...with the "PERQ-AI" workstation..." (sounds like theres a lot of marketing hype going on; lots of different names for the same basic workstation hardware)

Uses PERQ-LISP (the same as Spice LISP?)

a low-cost, high-performance Lisp workstation

delivers Lisp performance comparable to a VAX 11/780 at a quarter of the price

PERQ-Lisp is a superset of Common Lisp, featuring:

- tagged architecture (data items are "tagged" with a data type field)
- lexically-scoped variables (in both the interpreter and compiler)
- keyword optional parameters to functions
- multiple return values (fpr example, an integer division function can return both an an integer quotient and a remainder)
- user-controllable error handling and dynamic, non-local exits
- powerful macro facilities
- stream-based I/O facilities
- formatting and pretty-printing facilities
- a rich set of numerical data types (e.g., IEEE floating point, rational numbers, and arbitrary precision integers), with generic numerical primitives

- string, array and vector data types with "sequence" operations
- bit and field manipulation
- hash table facilities
- user-defined data types (records or structures)
- graphics primitives
- access to window manager
- interprocess communication facilities
- alien data structure handling facilities for dealing with message data from other languages such as Pascal, C or FORTRAN.

Integrted into the PERQ Lisp environ,ment is a screen-oriented EMACS-like editor (the Hemlock editor) which provides facilities for editing, evaluating and compiling Lisp expressions and saves results in either the Lisp environment or a file. It is written entirely in PERQ Lisp and is readily extensible.

extensive on-line help and documentation

powerful debugging facilities

The PERQ's architecture can be microcoded for optimum Lisp performance

A press release for the PERQ AI workstation appeared at the same time as one for the Accent operating system, May 15, 1984. This press release also orginated from Anaheim, CA.

Running Common Lisp PERQ AI has the same performance as the VAX 11/780 at one fifth the price (other sources mention one fourth the price; did the cost of a VAX 111/780 increase, or did the price of a PERQ decrease?)

Says Coleman, "We chose to implement Common Lisp because it is the de facto standard used by a significant group of companies and institutions that include Carnegie-Mellon University, Massachussetts Institute of Technology, and Lawrence Livermore Laboratories. PERQ Lisp incorporates many features which make it a valuable tool for symbolic processing."

"PERQ AI was developed as a response to the market need for a versatile, low-cost, artificial intelligence workstation," says Coleman. (is he referring to hardware or software? Was the T2 new in 1984???) He added that PERQ Systems Corp. is an experienced, high-volume (?) OEM workstation supplier.

PERQ AI incorporates a powerful screen-oriented editor into the PERQ Lisp environment that is modelled after the well-known TOPS-20 EMACS editor.

PERQ AI is part of PERQ Systems Corp. LINQ line of advanced high performance workstations (what were the others in this line? weren't they also basically the same T2s?)

Chapter 31

PERQ 3410 series Multibus/Laser Option board

- provides the PERQ workstion with the ability to connect to Multibus-compatible devices ... , as well as to streaming cartridge tapes and to a laser printer.
- Access to these devices through the PERQ workstation is fully supported under the Accent operating system.
- Any properly equipped device on the network can have access to M/LO supported devices connected to the PERQ M/LO PCB (printed circuit board)
- The M/LO PCB can be ordered in three configurations:
  - to support Multibus and streaming tape
  - to support a laser printer
  - to support Multibus, streamer and laser printer
- One multibus controller can be integrated into the PERQ card cage
- Expansion cabinets are available, each capable of housing up to 14 Multibus compatible controllers
- SMD disk controllers and disk drives and "Pertec" 9 track tape controllers and tape drives, are also available

- Full function Multibus adapter conforming to the IEEE-796 standard
- features a 24-bit address bus, with either 8 or 16-bit data paths
- One device on the bus is the current "master" which controls the bus. Other devices contend to become masters if they wish to control a transfer directly, rather than relinquishing control to the workstation housing the Multibus Adapter.
- Multibus allows multile CPUs on one bus. The PERQ CPU may serve as Multibus master and initiate I/O or memory transfers.
- DMA transfers to/from memory: for DMA transfers, PERQ microcode initializes a PERQ DMA channel to accept transfers from a mapped portion of Multibus address space.
- Multibus-initiated DMA requests to the mapped portion of address space will be accepted by the M/LO PCB as a slave and and be routed through the standard PERQ DMA channel to PERQ memory. Interrupt requests on the Multibus will be fielded by the M/LO PCB and cause a PERQ interrupt. The Multibus interrupt vector will be passed to the interrupt service microcode to notify the appropriate PERQ device driver.
- Multibus interrupts mapped to PERQ interrupt system (see above)
- The Multibus adapter supports both byte and word transfers at any Multibus address with the PERQ as master. Full 24 bit memory and 16 bit I/O addresses are supported. Both byte and word transfers may be made to or from PERQ memory as a slave. Eight maskable, nonvectored interrupts are encoded by priority and asserted on a single PERQ hardware line.

Note from CL: because the PERQ CPU acts as the Multibus master, it placed to heavy a load on the CPU and would really slow it down.

Chapter 32

PERQ Memory System

Very large virtual address space of 4 GBytes

Contains the display's bit-map

The dynamic RAM array is organized for the parallel access of 1, 2 or 4 16-bit words in each memory cycle.

One (680 nS) memory cycle is equal to four 170 ns CPU micro-instruction beats.

Access to sequential row addresses, to refresh the display screen, means that no specific refresh mechanism is required for the dynamic array.

Memory is addresses when the CPU or I/O boards present a word address on the 20-bit memory address highway. This address is buffered on the memory board.

Memory may not be accesses by I/O devices when the hold bit of a microinstruction is set.

Address inputs may be modified when the hold bit is set in conjunction with the jump field.

The memory printed circuit boards provide the basic random access store, drivers, buffers and control and also holds the logic circuits generating the visual display signals.

The random access memory (RAM) array is organised for the parallel access of 1,2 or 4 16-bit words in one 680 mS memory cycle, which is four CPU micro-instruction beats.

The CPU or I/O boards present a word address on the 20-bit memory address highway. This is buffered on the memory board and used to generate bank, column and row selection and the two phase 8-bit chip addresses.

Different board types are used for systems with portrait and landscape displays.

The memory board contains:

- basic random access store
- drivers
- buffers
- control
- logic circuits generating the visual display signals

Maximum memory capacity: 2 MBytes (1 Megaword)

Can be accessed by other boards through the memory data and address highways.

Display generation circuitry on the memory board is controlled by the CPU via the I/O data bus.

The memory board signals the CPU when any requests for memory access are made; the CPU has logic circuits to deal with any contention.

32.1 Memory Access

Micro-instructions control CPU access to the memory allowing:

- The storage of 1,2, or 4 16-bit words
- The fetching of 1,2 or 4 16-bit words

These words may be accessed in forward or reverse order (MADR:MDI:MDO).

Timing for the transfer, and CPU use of multiple 16-bit words, requires the provision of only single word buffers within the CPU for memory data in and out.

Requests for memory access are signalled to the CPU board where logic circuits deal with any contention.

32.2 Direct Memory Access (DMA)

The direct memory access controller has 6 channels for access to the PERQ memory board. The EIO board circuits use four of these channels. One for the fixed disc, two for OSLAN, and one for the Z80 system. The other two channels are for I/O options.

The DMA controller has header and address registers for each channel which it updates after data transfers. The registers are loaded initially by the CPU.

Data transfers initiated by the peripheral result in quad-word transfers of data with the memory board.

32.2.1 PERQ1

- a minimum of 256 kilobytes of main memory
- 680 ns (average) cycle time
- 1 megabyte RAM option will be available
- features virtual addressing scheme with segmentation, swapping and large address space.
- 32-bit address space
- a parity option is available for the RAM
- To support the wide bandwidth requirements of the CPU display, disk and network, the PERQ memory system is designed for a 200 megabit per second aggregate memory bandwidth.
- The memory plane is organized as 64K of 64 bit wide words.
- Memory cycles are split between the display and the rest of the system with 50% of the bandwidth available to the display.
- The display controller is completey integrated into the memory controller, resulting in a compact, low-cost design.
- The memory connects to the CPU and the IO channel controller with a 3 bus design (Address, Data In, Data Out) which can transfer 16 bits every 680 ns.

32.2.2 PERQ2, T2, LN3000

- 1 to 2 Mbytes of MOS semiconductor RAM with parity checking
- memory cycles 64 bits every 680 ns.
- approximately 200 Mbits/sec. aggregate memory bandwidth, shared equally between display and processor

- integrated memory/display controller
- screen bit-mapped directly from any area of memory

One to two Mbytes of MOS RAM with parity checking (no longer an option?)

Chapter 33

PERQ Microcode

Kwok W. Sheh wrote the following comment in the POS I/O uCode source code:

"In memory: This code is dedicated to all of those who have lay the foundation for the perfection of the current microcode technology."

Brian Rosen had this to say about the above quote:

"Most of the original microcode was written much earlier than Sheh's effort, and was fairly amazing." [BR, mailing list, Dec. 7, 1992]

In reply to Malcolm's questions about horiz./vert. uCode:

"No, vertical [microcode] does not follow from Writable Control Store. In a "horizontal" microcoded machine, the (micro)instruction set is wide, and consists of several fields that directly control the hardware. On a vertical microcoded machine, the instructions are typically narrower, and have several levels of decoding to control the hardware." [BR, mailing list, Dec. 7, 1992]

When asked what the microinstruction set is like:

"It is a 48 bit horizontal format, with register fields, ALU ops, special ops and branch fields." [BR, mailing list, Dec. 7, 1992]

"As I recall, there were two fields that selected which register from a dual ported register file. Two fields that selected what the ALU sources were (one choice was the register file, but there were others). An ALU field that selected an ALU operation, a "Write" bit that caused the dual port register file to be written, two control fields that did one of several functions, a branch control field and a branch target field." [BR, mailing list, Dec. 7, 1992]

Questions by Malcolm Shute, answered by Brian Rosen:

MS: Suppose, to pick an absurd example, I had a weird application,

which would go N times faster, if only I had an XXX instruction in the instruction set. The definition of XXX is that it takes two arguments (I'll leave it up to you to suggest where these should be, in registers, in memory, etc), and takes one of those arguments, multiplies it by 6, performs a bitwise-and using the other argument, and stores the result somewhere (back at the origin of the former argument).

Could I add such an instruction?

BR: Yes

MS: If so, how would I do it? Wouldn't it look like a block of microinstructions to be executed one after another? Wouldn't this be vertical microcode?

BR: You do indeed write a series of microinstructions to accomplish what you want. The "way you do it", has nothing to do with whether you have a vertical or horizontal microinstruction set.

Basically, a horizontal micro-architecture has a wide micro-word, which is broken down into fields, each field controlling some part of the hardware. As examples, the Perq has the X and Y fields which directly address the dual port registers, the A dna B fields which directly control the ALU input multiplexor, the ALU field that directly controls the ALU, etc. In a horizontal microword, there can be, and often are, field combinations that don't make sense, or duplicate other instructions. An interesting game we often played was to figure out how many ways to program a microinstruction that was effectively a NOP. A varient is to figure out a NOP that does the most amount of work for nothing (like A multiplied by B without saving the result is "better" than A plus B without saving the result, because it takes more work to multiply than add).

With a vertical microarchitecture, you encode the instruction fields tightly so that it takes fewer bits to represent the possible operations that you can do; sort of like what a regular instruction set for a classic machine language would do. The trade-off is size of the microword vs hardware complexity and speed. H machines tend to be fast, but need lots of microcode bits. V machines tend to be slower, but have more instructions for a given number of microcode storage bits. Typical H style microwords are 32-128 bits wide. Typical V style is 16 or 24 bits. Typically, you can get more done with an H style word. For example, in a single Perq microinstruction you can

fetch two words from a register file, ALU them, execute a special function like do an IO operation, branch to a microsubroutine while pushing the current PC on a stack; all in one instruction. Typically, a V style machine could only get one of those things done in one instruction. As a result H style tends to be fewer microinstructions per opcode.

As a point of reference, the majority of microcoded computers are horizontal machines. Vertical microcoded machines are rare.

Perq programs often did exactly what you described above; they added a few instructions to the instruction set. There is a call in the OS to load microcode, and a way to invoke this code. Generally, you add a new opcode to the main opcode dispatch (case) loop, which has an entry point in your new microcode. The inline assembler directive in Perq Pascal could insert your new opcode when you wanted to call it.

We worked hard to make the opcode dispatch loop as small as possible so that instruction set emulation was as fast as possible. The way it worked was there is an 8 byte opcode cache (OP?), that had an autoincrementing counter (BPC) that addressed it. One of the codings of the branch control field is JMP@BPC(OP) (I've probably got the field names wrong). This clause said:

Look to see if BPC is "overflowed" if so, branch to the opcode refill code

If not, get the byte out of the OP cache that is pointed to by BPC

Branch to one of 256 locations depending on the value of the byte - this is the instruction dispatch itself

Increment BPC

At the end of every microcode fragment the implements a QCode, this clause appears on the Branch field. The net result is that if there was a byte remaining in the OP file, there is a zero instruction overhead for dispatch; the first microinstruction of the next QCode is executed in they cycle immediately following the last microinstruction of the previous QCode. If the OP file was empty, the branch was to the opcode refill sequence that started a 4 word (8 byte) memory read, where the data was saved in the Op field. The BPC was then cleared, and the opcode dispatch clause re-executed.

I just answered the question of how big BPC is. It is 3 bits plus an

> overflow bit. It is implemented as a 4 bit counter, but the msb is a branch condition.
>
> This was different from most machines, where opcode dispatch takes 2-6 instructions per opcode; at least that was the case in 1979!" [MS, BR, mailing list, Dec. 9, 1992]

"Horizontal microwords are wide (e.g. 40 to 100 or more bits). Each subfield controls a data function drectly, and all the operations specified by the various subfields can be executed in parallel..." [book: "computer structures....", mailing list, Dec. 14, 1992]

"Whereas vertical microprograms are characterized by long sequences of narrow microwords, horizontal microprograms are characterized by short, intertwined sequences of wide microwords." [book: "computer structures....", mailing list, Dec. 14, 1992]

Chapter 34

Perq models

PERQ 1 - Has either a PERQ1 or a PERQ-1A CPU

PERQ 2 (same as a T1 or K1?) has PERQ-1A CPU

PERQ T2 - has PERQ-1A CPU

34.1 High performance LN3000 series:

PERQ LN3000 (PERQ T2 hardware) - Note that T2 systems made in the UK by ICL had a slightly different cabinet. The design of the front cover was slightly different and the ICL PERQ T2 systems did not have a recessed power switch and would have been easier to accidentally turn the power off.

34.1.1 PERQ LN3500 (PERQ AI) (PERQ T2 hardware)

- implements a version of Common Lisp that is specially suited to the needs of the Artificial Intelligence community

34.1.2 PERQ Color Workstation

- a system that illuminates layouts or designs in up to 256 colors from a palette of over 16.7 million

34.1.3 PERQ Audre (Still basically a T2)

- Audre stands for ADvanced Digitizing and REcognition. The software for the Audre system is by a company named Audre, Inc.

- a system which offers advanced optical digitizing capabilities across a network

34.1.4 Notes

Note: the PERQ Color Workstation, LN3500 (PERQ AI), and the PERQ Audre are all configuration options of the PERQ LN3000 line. (was this nomenclature confusing to customers/prospective customers?)

Note: Circuit breaker service was hazardous and difficult; cabinet was redesigned to mount circuit breaker to frame - was this new cabinet the T2 cabinet?

Note: PERQ K1 (T1?) cabinet had a problem with paint adhesion.

Differences between PERQ-1 and PERQ-2 systems:

PERQ-2 is:

- more than 15dB quieter
- dissipates less heat
- produces less E/M radiation
- has a low-profile, ergonimically designed keyboard
- a landscape monitor is available
- 16K WCS is standard
- 2MB of memory is standard
- has a battery backed-up real-time clock
- has a flat-surface pointing device, active area extends to edge
- faster gpib: up to 170 K Baud instead of 30 K Baud
- has a second rs232 channel that can operate at up to 19,200 baud

34.2 PERQ T4

Varityper 1280. 24-bits.

34.3 PERQ-3 and PERQ-5

Codenames Mars and Jaguar.

34.4 Notes on PERQ models from Chris Lamb:

"Now, is this the "3" that I'm familiar with - the 32-bit custom machine they were developing? Or is it the Motorola 68K-based system they tried to put out right as everything fell apart? (I swear I saw an announcement for the "Perq IV" touting 4MB of memory, a 32-bit processor, etc. but I don't have the right issue of IEEE something-or-other from five years ago)" [CL, June 9, 1992]

Model designations, according to Chris Lamb (Oct. 22, 1992), were:

Perq-1 - original machine with 4K CPU, up to 1 MB RAM, portrait (sometimes refered to as the Perq-0, when 256K RAM and a 12 MB disk was standard?)

Perq-1A - Original model with 16K CPU, up to 2 MB RAM, portrait. Still used "IOB" for the I/O and the "OIO" board for ethernet.

Perq-2 - A horrendous rush-job to get the "new" machine out the door. 8" disk interface and new cabinet design, with up to 2 MB memory and 16K CPU and either portrait or landscape monitors. Few of these were built and the 8" drive controllers are infamous for their flaky performance.

Perq-T2 - The "real" Perq-2, with the 16K CPU, up to 2 MB RAM, the EIO board with Ethernet built in [wasn't this available with the original PERQ-2??? ..and also an option on the T2?], 5.25" disk support. Possibly more of these built than any other model.

Perq-3 - A completely different architecture. Not the "real" Perq by any means. I'm very unclear on the specs for these machines. As I was privy to some inside company info., the Perq-3 was going to be a full 32-bit machine, and was actively in development when the company went down.

Perq-IV - A "t2" with a 24-bit board set and backplane mods. This allowed up to 4MB of memory (though with 24 address lines, 16 MB is the theoretical maximum...though to my knowlege 4MB boards were the highest capacity offered.

[CL]

34.5 Notes on K1/K2 from Brian Rosen

When asked about the Krismas system (the K1) and the K2:

"As I recall, and I'm not too sure of my memory here, the K1 was an early prototype of what became the T2. It was called Krismas because the eternal story about when any major project would be finished was "around Christmas time", so that the constant exhortation for getting finished was "Christmas is coming!". I don't remember the K2, it may have been the board set of a T2 in the old chasis." [BR, mailing list, Dec. 7, 1992]

Chapter 35

MPOS - PERQ Multiple Process OS

- Multiple process capability gives the user the capability to have more than one context established at a time. This allows rapid switching from editor to compiler to debugger for instance, without normal "start-up" delays.
- Multiple processes permit background I/O spooling, network accesses by other systems, etc. without disturbing the user.
- The operating system also supports the PERQs virtual memory system which manages very large programs with ease.
- Provides a multiprocessing environment for PERQs
- Permits concurrent execution of up to 32 processes.
- To run multiple processes, multiple shells are created with the New-Shell command
- Each shell has its own window, but each process can have multiple windows.
- Windows may overlap one another
- The puck (mouse) is used to control window size and placement
- A row of boxes at the top of the screen is used to indicate the status of each process.

- Does not support Ethernet
- Statistics command not implemented
- Bye command does not support the /OFF switch
- Supports inter-process communication which uses a basic message system consisting of mailboxes and message operations that send messages to, and receive messages from, the mailboxes.
- Utilities not converted to MPOS as of March 31, 1982 were: FixPart, MakeBoot, Partition and Scavenger.

35.1 MPOS File system

- distributed file system which supports:
  - multiple, tree structured directories
  - file versions
  - linked and contiguous files
  - security protection
- access to files on other PERQ systems as easy as if they were stored on the local disk
- all critical information is redundantly stored, and verification checks on disk operations insure the integrity of files

35.2 MPOS Display Window Manager

- the window manager partitions the screen into separate areas or windows
- windows may be moved around the screen enlarged or contracted in two dimensions, scrolled and clipped under direct user control
- windows can overlap each other and can be as large as the entire screen or as small as a postage stamp
- menus and “light buttons” are also supported by the window manager
- the process mechanism uses the window manager to allow direct user control of multiple concurrent processes

35.3 Weekend Wonder Crew

"It took more that one weekend to get the whole job done, but most of it was done in the 2 day extravaganza. I don't remember everyone who participated, but John Strait, Don Scelza, Brad Myers, and I were there. There must have been 3 or 4 others, but I don't recall who."

Chapter 36

PERQ Mysteries to be solved

- What/where was Alpha? (this was where "action items" were discussed on Jan. 10, 1983)
- Why wasn't the floating point hardware used?
- According to the T2 Service Guide, "the OIO board has the link logic built in". Is this all OIO boards? If not, what changes would need to be made to add this to an OIO board.
- Why would they design a system with a clock/calendar that the users could not set without special hardware and software?
- Why was no software written that would allow POS programs to run as processes under Accent?

Chapter 37

Various names of people associated with PERQs

Note from Chris Lamb:

"My stepdad worked for 3RCC back in the '80-'82 timeframe; The first "real" computer I ever used and learned on was a PERQ, tutored by him twice a week at "RIDC" - Three Rivers' "Advanced Development Group's" location; I must be the youngest (23) and longest running PERQ fanatic on the planet (10 years!); I have two PERQ-1s and a "t2" that is shipping from Pgh tomorrow; I have TONS of floppies, literature, manuals, and schematics; I've got an official PERQ t-shirt on my wall at home (those are almost as rare as color displays! :-)" [CL, in message to WvH on May 27, 1991]

37.1 Individuals:

- Bob Amber - wrote MPOS Users Guide, 1982 at 3RCC
- Miles Barrel - wrote a lot of the later microcode.
- Chuck Beckett (3RCC) wrote project description for TCPOS (IP/TCP for POS) titled "IP/TCP for POS Descriptopn" on April 5, 1982.
- Steve Clark - an engineer at PERQ, migrated to either MegaScan or Mars (check notes). Wrote the calculator program (calc).
- K. Cochran - ICL Dalkeith, write Rs232 test software
- Ellen Colwell

- Robert P. Colwell - co-authored the "PERQ Color Display Concept Document", July, 1982
- Richard Cox - president of Accent Systems during time when the MaxPro environment was being sold.
- Ed Fredkin
- Jim Gay
- Carol Geyer at PERQ systems (contact for Accent announcement, May 15, 1984), 412-335-0900
- A. Hanzawa - did programming for PERQ Tetris, 1989
- Lee Harris - assigned to prepare burn-in software
- Jeff Howell
- Rich Huber - Used to program the test system used to manufacture PERQs and now works for Brian Rosen at Mars. Had a running PERQ as of December, 1992 (according to Brian Rosen).
- K. Hughes - ICL Delkeith, worked on RS232 test program
- Bill Hulley - designed fixture for setting time of day clock
- I.M. Elliot (testcontrol.pas)
- Kelly F. Hickel (Accent Systems) - modified MBUS.Pas
- Horst (?) - "There was this guy named Horst (I can't remember his last name), who was a CMU guy into ADA. He was hired by Siemens Research, who co-operated with CMU on ADA projects, including a start at ADA for ACCENT. Horst didn't do the work, but another CMU grad student was hired by Siemens to work for Horst did it." [Brian Rosen, mailing list, Jan. 5, 1993]
- Sandeep Johar - Wrote design document for Autologic Development project titled "PERQ Interface Design" (the req. document details the configuration under which the PERQ communicates with the mainframe).
- Dirk Kalp

- Brian Koma at Burston-Marsteller (contact for Accent announcement, May 15, 1984), 412-456-2500. What was Burston-Marsteller? An authorized dealer?
- J. Stanley Kriz - co-authored the "PERQ Color Display Concept Document", designed the Kriz tablet, went on to MegaScan.
- Art Lim
- C. Lindsay (testcontrol.pas)
- J.M. Loveluck & F.R.A. Hopgood, from Rutherford Appleton Lab, SERC, wrote "The ICL/Three Rivers PERQ and Distributed Interactive Computing", April, 1983
- Ed Maples
- Brad Myers - Author of the a lot of the PERQ's graphics software; as of Dec. 1992, he had a PERQ (according to Brian Rosen).
- K."KING of TETRIS".Nakemura - Supervised writing of PERQ Tetris, 1989
- Joe Novak
- Geoff Potter
- George Robertson - Worked on IP/TCP for POS around D.6
- Brian Rosen - Creator of the PERQ.
- Joe Schmidt - ICL
- Kwok W. Sheh - wrote some PERQ microcode and a tribute to that uCode.
- David Stoner - co-authored the "PERQ Color Display Concept Document"
- (?) Stoney
- John Strait - Wrote a LOT of the PERQ software and SIM (Synchronous Register Transfer Simulator). A lot of the software he wrote was low-level coding, such as microcode. It appears that he and Brian Rosen wrote most of the original PERQ microcode. John Strait wrote Prose.

- "John Strait was the original compiler guru, most of the language constructs are his work. He was a Pascal purist, and was always trying to keep extensions and changes to a minimum." [Brian Rosen, mailing list, Jan. 6, 1993]
- Keith Tarvido (Accent Systems) wrote part of MBUS.Pas
- M.J. Tough - ICL Dalkeith, worked on confidence test shell (testcontrol.pas) in 1982
- Tim Taylor
- Bob Tysarczyk
- Terry Vavra - "Terry Vavra did lots of silly artwork and icons and such. His machine at 3RCC's Advanced Development office had a coin slot on it, and some silly stickers and such. Brian may tell us what Terry's job title was, but from what I gather, it was mostly doing silly drawings and stuff! (He did the icons for PerqMan.)" [CL, Dec. 30, 1992]
- "Terry was a technician who was very creative. He worked for engineering and advanced development for several years. He was indeed a wiz at cobbling up icons and silly pictures. We did keep him busy building hardware, but he would come in extra hours to play." [BR, mailing list, Jan. 4, 1993]
- Andy Verostic
- Tony Vezza
- T.J. Watson (testcontrol.pas)
- V. Wilson (testcontrol.pas)
- Ken Young
- Dale (?) - ICL
- John S. (?) in Chelmsford (England?)
- Fred (?) in Chelmsford (England?)
- REH (?) revised calc program

37.2 Companies

- Auscom, Inc. of Austin Texas - supplier of the final hardware configuration to support the Autologic interface project.
- Autologic: the Autologic company of Newbury Park, CA, contractual buyer of the interface system specified by the requirement document [REQ,PO]. (What is this document?) Note: Autologic made an IBM-lookalike mainframe computer.
- Graphic Horizons - this name appears in some .cmd files: "Graphic Horizons ... GetPut program". Apparently, they must have written a program named GetPut that adds a header to the .cmd files that it creates. Contact Vicky Nosbisch and see if she can find anything out about this.
- The Interlan company of Massachussetts, supplier of the QBus-Ethernet interface board for the Autologic project.

```
Accent Systems Corp.
5907 Penn Avenue
Pittsburgh, PA 15206
1-412-361-3200
Richard Cox, Pres.


PERQ Addresses:

Three Rivers Computer Corporation (3RCC)
720 Gross Street
Pittsburgh, PA 15224
1-412-621-6250

Note: above was the former Gulf Oil (?) building.


PERQ Systems Corp.
2600 Liberty Avenue
P.O. Box 2600
Pittsburgh, PA 15230
```

1-412-355-0900
1-800-222-4489

Chapter 38

PERQ Network

The Ethernet and Cambridge Ring networks on a PERQ can operate at speeds of up to 10 MBits per second.

38.1 OSLAN and OSLAN Interface Controller

These logic circuits provide the PERQ interface to an ICL open system local area network Transceiver Unit. The EIO board circuits provide data encapsulation and help in the data link management. A microsecond clock-timer is provided to help in sorting out collisions on the network. The timer is set by software and interrupts the PERQ CPU. The DMA controller handles the transfer of data between the OSLAN controller buffers and the PERQ memory. Separate channels are used for transmitted and received data but transfers are not simultaneous.

Open systems (OS) means that different manufacturers' equipment can be connected to the LAN.

Every packet of information has a 48-bit header which specifies a unique destination device address. All devices listen to every packet of information transmitted.

This is a baseband system. The OSLAN transmission rate is 10 million bits per second (10 MBits).

The ICL OSLAN is like Ethernet. It is a contention network using broadcast techniques. The method of accessing the network is called CSMA/CD. That is carrier sense, multiple access, collision detection. CSMA is sometimes called "listen before transmission". CSMA/CD is sometimes called "listen while transmitting" as well as listening before.

38.1.1 PERQ1

- proprietary wide-band network interconnects PERQ systems on a single coaxial cable using cable TV technology
- up to 64 PERQ workstations can be connected on up to 2000' of cable
- up to 1024 PERQs can be interconnected on a single coaxial cable with a maximum distance of 2.5 km
- broadcasting packets of data at 10 megabits per second, the network allows one PERQ to access files on another system
- the network is also used to provide access to shared resources such as printers and tape drives which cannot be provided to each workstation economically
- by connecting a resource to one PERQ on a network, all other PERQ workstations can share that resource
- Ethernet (trademark of Xerox Corp.) 10 Mbit/sec local network

Under POS D.6, on April 5, 1992, it was planned to get a working version of IP/TCP (TCPOS - IP/TCP for POS). George Robertson was supposed to have been working on this. The scale of effort for this project was to be: "Etc. and blah blah. Blah blah, etc. and so forth." (Actual quote from a short TCPOS document!)

38.2 Ethernet

Contrast Etnernet to the Cambridge Ring.

A number of connected nodes can broadcast messages which include a destination address.

Uses a contention technique, where the different nodes must "listen" to the transmission medium, the ether, and broadcast when the medium is quiet.

Sophisticated hardware is required to check for collisions of broadcast packets.

The propagation delay limits the total Ethernet size.

38.3 XNS

The PERQ 26220 Xerox Network Systems, or XNS, Protocols:

Consists of a set of software modules designed to allow users to develop high-level application programs which involve communication among heterogenious workstations, computers and other devices through one or more local area networks.

The protocols offer the ability to name, locate, and utilize high-level services in a distributed environment without being concerned with network architecture or flow control.

The programmer sees the network as a pool of resources which can be manipulated with simple procedure calls.

Together with the Ethernet local area networks, XNS implements the International Standards Organization (ISO) Open Systems Interconnect (OSI) model. This model describes the transfer of data or control information between applications residing on different machines as a seven layer process. XNS implements seven protocols which together implement levels 3,4, and 5 of the OSI model.

1. physical layer - transmits bit-stream to medium (ethernet specification)

2. data link layer - transfers unit of information to other end of physical channel (ethernet specification)

3. Note: the ethernet specification, which has been accepted as IEEE standard 802.3, defines the physical and electrical characteristics of the network as well as the access method and message deliniation rules. These are denoted as layers 1 and 2.

4. network layer - switches and routes information. At this level, the Internet Datagram Protocol defines the structure and function of the fundamental unit of information, the packet. Each packet is treated as a separate entity, called a datagram, at this level. The function of the Internet Datagram Protocol is to address and route these standard packets from the source application to the destination.

5. transport layer - end-to-end data integrity and quality of service. At this level, the transport protocols provide a variety of facilities which support the reliable exchange of streams of related frames between clients on the network using the Internet Datagram Protocol. These

facilities provide end-to-end error free communication paths, flow control, information for routing messages between networks, and tools for implementing error reporting and diagnostic procedures.

6. session layer - coordinated interaction between end application processes. At this level, the Courier Protocol provides the facility for making the details of network communication transparent to the user. This is done by having access to remote resources structured as procedure calls. For example, the transfer of data from another machine on the network would be seen by the user as a sequence of procedure calls to open a file, move the data and close the file. The Courier forms these call messages and transfers parameters and data using the layer 4 transport protocols.

7. presentation layer - provides code conversion, data formatting

8. application layer - selects appropriate services for the application

XNS is an industry recognized standard which has been implemented on a large number of machines including IBM mainframes, DEC VAX minicomputers, IBM PCs, and dedicated function hardware such as the Xerox 8044 Print Server.

PERQ XNS protocols have been implemented in conjunction with PERQ's Accent distributed operating system.

38.4 LINQ

LINQ is PERQ Systems' local area network. LINQ extends proven ethernet technology with a unique operating system: Accent.

One of the most important aspects of LINQ is that it incorporates a message-based operating system, Accent, that is fundamentally designed to enhance network services. This results in a simpler user interface– both at the programming level and for the user himself, who doesn't need to waste time locating, creatng and monitoring activities at other network nodes.

A user can obtain data or access resources that are available somewhere else on the network, or even on a different network– either locally or remotely– without the need to know the location of that data or resource

LINQ features multiple, co-equal environments for optimal software development and operational flexibility. These environments include the native Accent environment, an advanced UNIX environment licensed from AT&T

(QNIX), and a Lisp environment for Artificial Intelligence applications. A single workstaion can simultaneously run applications in these environments.

The four major elements of the LINQ network are:

- the network operating system (Accent)
- network hardware (LINQ board, ethernet stuff, etc)
- software development environment
- applications software systems

Gateways enable LINQ networks to share data, files, and expensive peripherals with either other LINQ networks, or with foreign networks. LINQ can even be incorporated into other Ethernet compatible networks.

Controlled by the Accent advanced network operating system

Connects workstations, gateways and other manufacturers' computers and devices into a sophisticated open network based on Ethernet technology.

A press release issued on May 15, 1984 stated: "A new local area network designed from the initial concept to provide a non-proprietary, truly distributed, transparent operating environment was introduced at the Computer Graphics '84 Show today by PERQ Systems Corp. (in Anaheim, CA?) Note: this was the LINQ network.

Using LINQ, workstations, various servers, gateways, and mainframes can be networked together.

Gateways allow LINQ networks to share data, files and peripherals with other LINQ networks or foreign networks.

PERQ Systems' older workstations, the PERQ and PERQ-2 can also be used in a LINQ network.

38.5 Cambridge Ring

There are two possibilities for connecting a PERQ to a Cambridge Ring:

- Outbound hardware interface which is relatively inexpensive and can be connected to the PERQ's GPIB.
- Inboard interface which connects directly to the PERQ's I/O bus and is considerably faster than the outboard interface, as well as more costly.

A number of local area networks had been proposed to SERC for the Common Base Policy, among which Ethernet and the Cambridge Ring were the most widely used. (move this to SERC file)

Nodes are connected in a closed loop.

Transmission is based on a system of circulating slots.

A node can load data, which includes a destination address, into an empty slot, and the data is removed by the destination station.

A guaranteed transmission rate is obtained by prohibiting a node from using two consecutive slots.

Chosen as the SERC Common Base Policy standard because of technical advantages and because systems were available from U.K. industry almost two years before an Ethernet product became available.

Over 20 Cambridge Ring LANs were installed in UK universities as of April, 1983, and a large body of experience and software was available.

In order to provide PERQ ring communications within a short period of time, the Rutherford Appleton Laboratory developed an interface which allows the ring to be accessed via the IEEE-488, or GPIB, bus.

The interface to the GPIB is designed to match the "50-way" encoded interface and presents the station as a number of number of status and data registers. The RAL interface translates this node image as a single primary GPIB address with each register having a separate secondary address. The design has been optimized to allow bulk transfers in one direction to take place with the minimum number of GPIB cycles, so that the ring can potentially be driven at full speed.

In terms of the International Standards Organization (ISO) model which divides the communications functions and services of networks into seven levels, the Transport Service Level provided over a Cambridge Ring is the Transport Service Byte Stream Protocol (TSBSP). This allows for a transparent extension from the local Cambridge Ring to the SERC's X25 wide area network, or to other networks. As of April, 1983, a basic block interface for the TSBSP protocol to the Cambridge Ring has been written to run under POS and a version for PNX was being devised.

DICE - Distributed Interactive Computing Environment, a term from the SERC and RAL (Sounds similar to CMU's SPICE).

38.6 Getting TCP/IP Working

Chris Lamb found that the TCP/IP software compiled under Accent S6 after fixing some compiler errors. However, when it is run, it enters the

debugger "way down deep in the Ethernet code". [CL, Aug. 31, 1992]

"Right now, I've successfully compiled the TCP/IP suite under Accent S6, but it crashes and enters the debugger when I try to send out the first packet (deep in the Ethernet driver, so I've got some real fun debugging to do)." [CL, Oct. 22, 1992]

"I'm concentrating on Accent now and may tackle POS later... the code seems to be written such that changing a couple of booleans in a module or two will turn on the proper conditional compilation switches, and the same source files will compile under POS." [CL, Oct. 22, 1992]

"The error is deceptively simple: a range check. But I think it's leading to a memory allocation problem, which is pointing me towards a data structure - the Ethernet packet itself - that isn't getting allocated and intitialized properly, even though the routine that does that *is* getting invoked." [CL, Nov. 30, 1992]

38.7 Eth-Can board

"Another interesting tidbit... when you add an Eth-Can board to a T2 with an EIO board, you should, in theory, have a machine with two working Ethernet interfaces. The OSLAN-A tap on the bulkhead is wired to the EIO slot and the Eth-Can should drive the OSLAN-B tap. That would allow you, with *software support*, to set up gateway machines and partition large networks into more manageable subnets. I've never seen nor heard of a single line of code that would let you do that, but it is a nice thought. :-)" [CL, March 31, 1993]

38.8 Ethernet Connections

> You have to have the proper "pigtail" cable to turn the connector on the back of the machine into a regular AUI cable that can be connected to a thick-wire MAU. On the PERQ-1s, the I/O OPTION connector on the back is approximately 50 pins or so; the normal pigtail turns that into a 15-pin connector that you connect to the transceiver. If you also have a laser printer, there's an even more unique , double-ended cable that splits the I/O OPTION connector into both ethernet and laser printer connectors.
>
> On the T2, you actually have two "OSLAN" connectors. The EIO board drives the "A" port. If you have an "Eth-Can" board (for the

laser printer and the Ethernet, used mostly in PERQ-1s) it supposedly is wired to drive the "B" connector on a T2, though I doubt there is any software that will allow you to gateway between two separate Ethernets. Since I have such a machine, and since I'm working on the TCP/IP stuff (or attempting to, if I had any time available) I may discover otherwise. I'm skeptical, though; I have never read anything in any Perq literature that mentions being able to set up a PERQ as a "bridge" between two seperate subnets. Of course, if the hardware is wired together properly, I don't think there's any reason the I/O microcode and all higher layers couldn't be enhanced to give that capability.

I just "borrowed" some "thick-to-thin" transceivers from work - nifty *inexpensive* little boxes that take a thick AUI type cable in one side (from the Perq) and have a BNC connector for thin-wire cable on the other side. As soon as I lay my hands on a couple of 50-ohm terminators, I'll be attempting to network my Perq-1s at home.

I cut my teeth on CMU's *3MHz* net (with some Xerox Altos!) and 3RCC's 10MHz net out at their Advanced Development Group's office back in Pittsburgh (almost 10 years ago). As a kid, being able to boot a machine (and have it sync its clock to the Vax :-) and print stuff over the net to a 240dpi laser printer was way, way too cool.

Since the Perq is old enough, I suspect it may have trouble talking to 802.3 compliant systems.

[CL, mailing list, Nov. 24, 1992]

38.9 FTP to Sun

"Shoot, there were some disks I *THREW AWAY* that John S. and someone else did for Intran to do FTP to the Sun. It was buggy, but under ideal conditions you could get the files from the Sun to the Perq and vice versa." [CL?]

Chapter 39

PERQ Keeper's Guide

```
    ****************************************
    THE PERQ KEEPER'S GUIDE (to remaining sane)
    ****************************************

Miscellaneous notes regarding the Perqs - Repair, operation, etc.
-----------------------------------------------------------------

Introduction
------------

The information contained within this guide, for the most part, is
necessary knowlege for using and maintaining a PERQ.  However, most of
it is not mentioned in the manuals supplied with the PERQs and was
'inside information'.

This first version of the guide is in ASCII text form.  Future
versions will be in the format of .mss files for the Mint document
processing software and will be distributed in either that form or as
printed documentation.  Copies will be made freely available to
members of the PERQ Preservation Society as long as I am financially
able to do so.

Thanks to (in alphabetical order) the following for the information
that they've provided, which has been included in this users guide.

Dave Callen - formerly of PERQ Systems and Accent Systems.
Had I not been put into contact with him, a lot of PERQ related
knowledge may have been lost.  Without his help, Bill, Chris and
myself may quite possibly not have gotten our PERQs into working
```

condition.

Bill von Hagen
Chris Lamb
Brian Rippon - Advent Systems
Malcolm Schute
Graham Underwood - Advent Systems

Table of Contents

What to do about the ‘‘wraparound’’ problem which occurs on the Perq's landscape monitor: Replace IC-3; this is a phase lock loop (PLL) chip, an MC14046BCP. After replacing this chip, it may be necessary to tweak the second pot from the back of the monitor. This pot is located on the PC board near the speaker.

There is no easy way to exchange monitors between the Perq1 and the Perq2.

In order to exchange landscape and portrait monitors on a PERQ2, it will be necessary to also change the type of memory board used. For example, a landscape memory board is required in order to use a landscape display.

It is not advised to use the PERQ cabinet as the stand for the display; electromagnetic fields from the monitor can damage data on the hard disk. If you insist on using your PERQ in this manner, as I do, you should occasionally use disktest to run a non-destructive reformat. This rewrites all of the data back onto the hard disk as it reads it and doesn't harm anything. This reformat can be safely done on systems containing Advent System's software which has been specially installed using an install disk; it will not require re-installation after a reformat.

Power supplies in the PERQ1 systems are often a problem. This is especially true if a power supply manufactured by Power Components is

used. This type of power supply is best identified by either a label or a large white connector (cable?) on the rear of th power supply.

It is rumoured that the power supplies for the PERQ2 and T-2 systems were not carefully manufactured; rumour has it that they were built in Arizona and assembled by people who couldn't speak English.

CPU, I/O and Memory Boards

* CPU Boards *

Q. How to tell Perq1 CPU boards from those for a Perq2 T-2.

A. The CPU's and memory in the Perq1 and Perq2 (T-1, T-2) are the same if the CPU uses the 16K writable control store CPU. They are both 20-bit internal, 16-bit external, CPUs. There were some 24-bit CPUs made, so that 3 or 4 megabytes of memory could be used in a T-2; however, they were not very common. The PERQs that used these 24-bit CPUs were known as PERQ-4 systems.

* Memory Boards *

The memory board will be different in a T-2 if a landscape display is used; a Perq2 using a landscape display will use a landscape type memory board.

Note that each of the Perq models uses a different type of I/O board, so you may not interchange I/O boards for most different PERQ models.

If possible, avoid the first (early versions) of the landscape memory boards. These are the ones with a small PC board attached to the front of the board; this small PC board contains a few chips, a small trimmer pot and a coil. Refer to the drawing below. The trimmer pot is used to adjust the stability of the video display.

* Ethernet I/O (EIO) Boards *

Some board labels may be misleading. To determine if a IO board is an EIO board, look at the top left-hand corner of the board for the ENET proms.

PERQ1 systems use a seperate Ethernet board, not an EIO board.

In order to use the Accent OS on a PERQ, an Ethernet I/O board or an Ethernet board must be present. Note that when runing Accent, a little icon known the ``Ethernet I/O Eye'' may be present at the top of the display.

[insert Paint drawing of this here]

* All EIO and Non-EIO Boards *

On allmost all I/O boards, EIO and non-EIO, there are five chips missing. These chips were to be used for floating-point math circuitry, employing an Intel 8087 math co-processor.

* Misc *

Q. Are any of the boards for the Perq1 and Perq2 interchangeable?

A. Yes, to a point. Refer to the previous question. Note: the Perq1's used seperate Ethernet boards, and not EIO combo boards like some T-2s used. The ethernet I/O boards were placed in the I/O option slot of the Perq1s.

Some of the other uses for the Optional I/O slot in the Perq's card cage are as follows. A speech card which would allow a user to issue commands to a Perq by talking (voice) to it; a special keyboard was required for this.

An interface for the streamer tape drive could be placed in either the I/O option slot or the CPU option slot. However, to use it in the CPU Option slot, the ``key'' in the card-edge connector would have to be removed for it to fit in that slot. Refer to the drawing below.

[Refer to drawing in red notebook \& recreate w/Paint]

Keyboard

If, using a Perq1, you get characters such as \^K on the screen no matter which keys you press, it is very likely that the problem is due to a bad PROM in the keyboard.

Keyboard trivia: the keyboard for my T2 survived an approximately 8-foot drop off of a loading platform onto the pavement below. The only damage was a few very small dents in the plastic case and a few keys popped off, whick were easily fitted back on! I just consider this luck and would not advise anyone to attempt to duplicate this happening.

DDS

The PERQ's diagnostic display (DDS) only contains valid information about the system status during the boot process. After the system boot process has completed, it is under the control of user programs and may have other uses defined by the authors of various programs. After a successful boot, the displayed codes cannot be interpreted as they would during the boot process. Some application programs are notorious for causing the DDS to change.

Beware of the DDS bug that can cause the display to be off by one digit during the boot process.

Replacing Disk Drives

On PERQ T-2 systems, the standard hard disk is an ST-506 MFM type. Almost any MFM disk can be used, even if it isn't listed in the file. Only 30 megabytes of any hard disk used with the PERQs can be used with POS unless special software provided by Advent Systems is used.

Any hard disk listed in the disk.params file can be used easily. You may also add disk drives, for similar disks (disks of the same family as those listed in disk.params) to this file. There is one restriction: any disk type added must be added BEFORE the last drive listed in the file; otherwise, the disk.params file may become corrupted and unusable.

Adding a Second Hard Disk

PERQ T-2 systems running Accent or PNX can have up to two hard disk drives. Under Accent, the system treats them as one logical drive. POS will only be able to access the second drive for it's low-level utilities used for formatting, partitioning, etc.

The following illustration explains how to set up a two drive system.

[Place Paint drawing here]

Duplicating Flopies

The use of the Floppy utility to duplicate floppies is considerably time consuming. The preferred method of duplicating floppies is via the Duplicate, DupBoot and PNXBoot utilities. All three utilities work in a similar fashion. It is not necessary to format a floppy before duplicating it with these utilities.

WARNING: be extremely careful when duplicatng a set of floppies! Duplicate, DupBoot and PNXDup must all be restarted (executed) for each different floppy duplicated. Do not insert the next floppy to be duplicated into the drive until you have stopped and restarted the utility. The reason for this is that these utilities attempt to keep duplicating the same floppy, multiple times, until the program is terminated.

* Duplicate *

This works with all floppies, with the following exception: it may not duplicate a FastBoot floppy properly.

* DupBoot *

This is used to duplicate a FastBoot floppy.

* PNXDup *

This is used to duplicate PNX file-system floppies.

Streamer Tape Drive

The streamer tape drive is used mainly to copy entire disks, or partitions, to a tape. However, it can be tricked into copying only a subdirectory if you tell it that a subdirectory is a partition. You cannot use this trick to restore data to a subdirectory on the hard disk.

Each DC300 (or DC300XL) tape will hold a maximum of about 20 megabytes of data, no matter what the length of the tape.

Backing up data using the Accent S5 version of Stut does not work very well at all times. If the partition to be backed up is accessible by POS, then use POS for your backup or re-boot to a different version of Accent for the backup. Otherwise, you may be in for a lot of error messages, such as those buffer allocation problems, errors reading buffers, etc.

IMPORTANT: Prior to using a new tape, you should use the erase command to remove any glitches that may have been put on the tape by stray magnetic fields.

The software used to read from and write to tapes is Stut which is mentioned below.

* Stut *

This is the standard version of stut supplied by the manufactured of the PERQs.

* Stut2 *

This is a version of Stut that I modified to allow one to dump the directory listing of a tape to a file. This is useful since there is no other way, that I'm aware or, that the output of original version of Stut can be redirected to a file.

Booting PERQs, Hard disk initialization, etc.

* Booting a PERQ *

To boot a PERQ from a floppy disk, install the disk in the drive, close the drive door, press the reboot button and hold down the 'A' key. Note, this must be an upper-case A.

A working boot floppy is different for each Perq model. This is because the boot floppy tells the system how many sectors are on the hard disk. Do not attempt to boot a machine with the wrong boot floppy! This wil not boot the machine and it coult wipe out data on the hard disk.

A special autoboot floppy was made by Advent systems. When this floppy is used, it is not necessary to keep pressing on the 'A' key to get the PERQ to boot.

Another useful boot disk that some may have is the fastboot floppy which significantly reduces the amount of time that it takes to boot the PERQ from a floppy drive.

None of the Perqs can be booted without having a hard disk attached; they cannot even be booted from a boot floppy. PERQs did not exist as ''diskless workstations'' (unless, of course, there were no plans to use the PERQ at all until a hard disk was installed).

* Adding New Partitions *

The NewPart utility is used to add new partions to a disk.

* Formatting a Hard Disk *

The POS operating system is always used to format a hard disk, no matter what OS will be used on a PERQ.

The following steps are used to format a hard disk:

1. Boot the system from a floppy if present hard disk is not bootable.

2. Dismount the boot floppy,

> dis f

3. Insert load floppy into drive and mount it.

```
> mo f
```

4. Run the badsector program to check the hard disk's bad blocks. If a bad sector map is not found, you will need to tell badsector which sectors are bad, manually. Use the help command from badsector to see what other commands are available. You will need to have the factory disk defect list to perform the following. You may be able to work around this by selecting additional commands to scan the disk for bad blocks.

```
> badsector

CMD> readmap
```

The above command will read the bad sector map into a buffer.

```
CMD> li
```

The bad sectors will be listed. Compare these, if any, with those in the disk's manufacturer supplied defect list. If they match exactly, then you can quit this program. Go to the last step below. the 'q', quit command.

```
CMD> insert
```

This will allow you to insert bad blocks into the bad sector map from the defect list. You will be prompted for the disk type that you are using. In the following example, a Toshiba MK56 drive was used. If the type of drive that you will be using is not contained in your disk.params file, you will need to press RETURN and enter all of the disk parameters manually.

```
Enter disk model or press <RETURN> to enter other parameters: mk56

Enter bad spots by byte or sector? [B or S]: b
```

You will normally enter 'b' to enter bad spots by byte (4 digits) as listed on most manufacturer's defect lists. You will be prompted further (not shown here) for the bad sectors. After you are finished entering the bad sectors, continue with the following.

```
CMD> li
```

List the bad sectors in the buffer to make sure that you have entered them correctly.

```
CMD> writefile

Writes the bad sectors from buffer to the bad sector map.

CMD> printfile
File: badsectorfile

This writes a file named badsectorfile, that you can read or print
out later.

CMD> q

This will exit the badsector program.
```

4. Run the DiskTest program. Note that running format with checking on is slow; running format with checking off and then using scan acheives the same results, only faster. The 'help' command may be used to obtain a listing of commands.

```
> disktest

Run the disktest program.

COMMAND> area r,1,829

This command tells DiskTest what area of the disk to format. The last
number will vary, depending upon the type of disk drive that you are
using. This example is for an MK56 drive. This is normally the
number of cylinders - 1.

COMMAND> unsafe

Tell DiskTest that is permitted to proceed with something that can
make permanent changes to your disk.

COMMAND> format

Proceed with the format. Your screen will fill up with boxes.

COMMAND> scan

Scan for errors.
```

```
COMMAND> q
```

Quit the program. The disk is now formatted.

5. Run newpart to initialize the disk drive and give it a name.

```
CMD> devinit
[no] y

Enter dev. name> SYS
[no] y

CMD> q
```

6. The disk is now ready to be partitioned. You will again use NewPart to create partitions as necesary.

* Creating New Partitions with NewPart *

The following is an example of using the NewPart program to create a partition. Use the 'help' command to see what commands are available.

```
>NewPart

CMD> createpart/initpages/test boot 10080
```

This will create a boot partition of 10080 blocks. You will repeat this command for each partition that you wish to create. Use whatever mane and size that you wish, with the following restrictions: the partition must contain fewer than 32,768 blocks (for use by wither POS or Accent) and it must be a number which is the ending of a cylinder boundary. Don't worry about this last part about the cylinder boundary; if the number that you have selected is not correct, the program will present you with numbers on a cylindar boundary that is close to what you selected and will ask you which number you wich to select.

You are now ready to make the partion, or partitions, that you created bootable and can load files onto them. Refer to MakeBoot and BindBoot for making a partition bootable.

* Running a Non-destructive Reformat *

A reformat reads information off the disk and writes the data back. Use this every six months or so, more often if you keep your monitor on top of the PERQ cabinet. This will prevent data on your hard disk from deteriorationg. It might not be necessary to run this, but doing so is a good precaution against disaster.

* Using FixPart *

I've never used this 'experimental' program. Please respond if you have used this and understand exactly what it does, etc. and why it is needed.

* Using BindBoot Under POS for Accent *

The BindBoot program under POS may be used to make an Accent partition bootable that is accessible by POS. The following explains what you would enter to create a bootable Accent partition using the 'b' boot.

Existing boot file: Accent.Boot

Hard disk selected.

Which character to boot from: b

System b-boot is unused.

Interpreter b-boot is unused.

System or Interpreter [S]: <press return>

System b-boot disk address=nn

Existing boot file: Accent.T2.MBoot (see table below for other machines)

Hard disk selected.

Which char. to boot from: b

System b-boot disk address=nn

```
Interpreter b-boot is unused.

System or Interpreter [I]: <press return>

Interpreter b-boot disk address=nn

Existing boot file: <press return>
```

In the above example, one question required you to answer it according to the type of machine used. The table below explains what replies to use:

PERQ Type	Answer
PERQ-1	Accent.PERQ1a.MBoot
PERQ-2	Accent.PERQ2.MBoot
PERQ-T1	Accent.PERQ2.MBoot
PERQ-T2	Accent.T2.MBoot

* Using BindBoot Under Accent for Accent *

This is pretty self-explanatory; run the program and you'll see. If further explanation is needed, let me know and I'll add it.

In order to create a bootable partition for Accent that is outside of the range of POS, it is necessary to use this program.

* Avoiding Unusable/Undeletable files under Accent *

1. When installing Accent for the first time, install it from within a partition that is fully accessible from POS.

2. Run scavenger on the partition before installing the Accent files.

3. Install the Accent files and make partition bootable with BindBoot.

4. Create a new partition that is outside of the POS accessible boundaries.

5. Boot Accent

6. Run Scavenger from Accent on the new partition.

You may now use this new Accent partition and make it bootable if you wish, deleting the other Accent partition that was accessible from POS. From now on, you must scavenge each new partition that will be used by Accent by using the Accent Scavenger program. Additionally, you must continue to use the Scavenger from Accent only, for the following additional reason: if you have any bootable Accent partitions that are outside of the boundaries accessible by POS, all boots from these partitions will be deleted by the POS Scavenger program.

The problem of unusable and undeletable files is due to a bug in Accent that is documented in some of the Accent source code. Be forwarned, if you do not follow the above procedures, you could be in for a few unsettling problems.

* Misc. *

WARNING: Track 0, where the boot information block resides, on a PERQ1 with a 14" Shugart drive is close to the center of the disk. On the 5-1/4" drives used on the PERQ T-2, this track is near the outside of the platter. For this erason, you should not use the /wait switch to park the heads of 5-1/4" drives; the heads will be moved to the wrong place!

If the disk information block gets messed up, Scavenger can be run to get information back (even if it can;t find partitions) after going in with newpart.

Sometimes the PERQ1 will get confused and think that the hard disk has four heads insteadd of eight. Disktest can be used to fix this problem.

Be sure to perform a non-destructive reformat every six months or so to avoid the decaying of the magnetic fields on the disk. You should especially do this, and possibly do so more frequently, if you have your monitor sitting on top of the PERQ cabinet. A note in the PERQ T2 Service Guide warns: ''Failure to observe the following can result is system data being corrupted: ... Do not stand the display unit on top of the processor.''

When the laser printer runs out of paper, or becomes jammed, when

performing a screen dump or using certain printing utilities, the PERQ can become totally incapacitated. No sequence of key presses can save you. In such situations, it will be necessary to press the reboot button. The CXPrint software is much less likely to cause you to have to reboot than is the CPrint software. In addition, CXPrint doesn't waste paper like CPrint does.

Installing POS from Tape

After loading files onto the hard disk from tape, it is necessary to link all of the files. You should have a command file named LinkAll.cmd; this will link all of the POS system files.

After linking files, run the MakeBoot program. For example:

MakeBoot system.6/build partition_name

This would have created made a partition named partition_name bootable as your primary POS partition ('a' boot) for POS G.6.

Note that if you have loaded a POS system that was originally used with a portrait display, and you are using a landscape display, you will only have half of a useable screen; the top hale of the screen will be useable. The bottom half of the screen will display graphic memory. To correct this, run the MakeBoot utility as described above.

Installing POS from Floppies

Installing POS from a set of floppies is more involved than installing it from tape. Follow the instructions with your set of floppies; there should be a set of command files to load the system, etc. A command file named AfterLoad should be run after files from floppies have been loaded onto the hard disk.

More information about this is needed.

Reliability

The Perq1 was the least reliable Perq. The most reliable was the Perq2, model T-2. The Perq2 T-1's 8" hard disk was not very reliable.

The least reliable part of a Perq is the I/O board. It may be a good idea to make copies of the PROMs.

Directory Limitations

If you have more than 599 files in a directory, on a system running POS, you will be unable to view all of the file names. Remove a few files and try again.

Software

Some POS binaries, after re-linking, are interchangeable between various PERQ models, This is, however, dependant upon the versions of the operating systems used and the versions of some of the PROMS used in the PERQs. For example, programs written for POS F.1 will not likely run on a PERQ T-2

Logging Off

You should always log off your PERQ runnng POS or Accent by typing bye, with any appropriate switches. WARNING: as explained above in the section on disks, do not use the /WAIT switch with PERQ T-2 systems. Unless you can't avoid it (if the system is hung), always log off properly so as not to leave the filesystem in disarray.

A file appears to exist, but I cannot access it or delete it. What do I do to remedy this situation???

This often happens under Accent when you've recently created a new partition and have not run Scavengr on the new partition before placing new files into this partition. One solution is to re-initialize the partition and run Scavenger then re-install the files. A better way is to just go ahead and run Scavenger (from Accent) and instruct it to rebuild the directories in the partition

that is causing you a problem.

Chapter 40

PERQ vs. NeXT

Similarities betweem the PERQ and the NeXT computer:

"I've started getting into programming on my Mom's NeXT computer! In another message I'll relay some very interesting parallels between the NeXT and what the Perq was becoming... I'm very pleased to say that the spirit of the Perq really does live on in the NeXT!" [CL, June 9, 1992]

"In the NeXT I see the machine the PERQ could have become. The fact that it runs Mach gives it an eerie feeling (but pleasant one!) sometimes, when I read about things in the manuals that Accent was doing before Steve Jobs had even produced the Mac! There are striking parallels between Accent and Mach, as you'd expect, and it makes me feel good inside that the Perq really did make an important contribution. Mach is an excellent operating system kernel, and the layers people have been developing on top of it – bsd Unix, PC and Mac emulators – prove that." [CL, 1992]

"Ironically, I've gort a 400-dpi Canon printer connected to this NeXTcube and guess what? It works just the same way [raw Canon interface]! The only difference is that the Perq never had a PostScript interpreter, so the output was always a bit more crude than the NeXT produced. :-)" [CL, March 31, 1993]

Chapter 41

PNX - PERQ UNIX

- The PNX system's intermediate level instructions are in C-code.
- UNIX was adopted as the SERC Common Base Policy standard
- POS is still used to format a hard disk to be used by PNX (I think this is true, but should be checked out to make sure).
- An implementation of UNIX which would permit a "natural evolution" to future enhancements such as distributed operating systems in which the OS is distributed over a number of processors interconnected by local and, or, wide area networks. Not only special services, but also processing power is distributed around such a network.
- Maintains full compatibility with Version 7 UNIX; a full implementation of V7.
- 32-bit arithmetic
- paged 32-bit virtual addressing with a flat address space
- Satisfied a requirement of the SERC Common Base Policy that all language implementations should be interworkable at the system call level in order to allow mixed language programming and obviates the necessity for translating sections of code from one language to another.
- The extension of facilities which manage display windowing to a multi-process operating system such as UNIX presents a number of problems such as:

- It would be desirable for each UNIX process to be able to create and modify one or more windows, but it may also be desirable for a number of processes to display information in the same window, in a cooperative way.
- It would be desirable to share I/O devices between windows belonging to different processes.
- It would be desirable for "hidden" windows to signal essential information to the user.
- In order to avoid conflicts between the requirements of different processes, a special "window manager" process may be necessary to supervise the creation and movement of windows.

A graphics application, initially developed for POS, the Graphics Kernel System (GKS), was being developed by a collaboration between ICL and SERC, which was at version 7.2 as of April 1983. At this time, Rutherford Appleton Laboratory was installing an earlier version of GKS (V 6.2) under UNIX which was to be ported to the PERQ as soon as full UNIX facilities were available.

"It was not possible for the CCode as it was written to coexist with QCode. I don't think there was enough microcode space, and they both assumed they had 100% control over the machine." [Brian Rosen, mailing list, Dec. 9, 1992]

"...I was dumped out of PNX into some form of low level interface to the PERQ, there was an '@' prompt, I could get the PC contents, but was unable to find the continue command. Would anyone like to enlighten me as to what commands are available?" [Robert Marshall, mailing list, Dec. 21, 1992]

Chapter 42

POS PERQ Operating System

POS provides:

- job management
- virtual memory association
- swapping
- I/O services
- File System
- Window (screen) manager
- Timekeeping functions
- PASCAL routine support
- Operating system modules are accessible at all levels to user programs via exported declarations.
- security is provided with user login/password and multi-level file system security features.
- the operating system is mature and well documented (???)
- replaceable outer level command interpreter which includes support for command files

- The POS system's intermediate level instructions are in Q-code.
- The main program for POS is called SYSTEM. This loads and transfers control to valid user programs.

42.1 POS Memory Organisation

This refers to the way the POS operating system organises the storage of programs and data in memory.

Real memory maximum is 2 Mb.

The system has a virtual store with an address range of 4096 (Mega?)bytes.

Segment: The virtual store is divided into segments which are multiples of 512 bytes and are referenced by segment numbers. The segments are either code segments or data segments.

Code sements: Segments of code for execution by the processor have a maximum size of 64 KBytes; they are accesible on a read only basis.

Data segments: Segments of data have a maximum size of 128 KBytes; they can be written to or read from.

Process: A number of code and data segments are grouped together to form a process (program). A minimum of two segments is required: a code segment and a data segment defined as the process stack segment.

Stack segment: The stack segment (MStack) contains internal segment numbers (generated by the compiler) which link the segments within a process. The stack segment also contains a global data block (GDB) which allows access to data not included in the process.

Process in memory: To run a process it is loaded into memory. The segments loaded are referenced by segment address tables (SAT) in memory starting at address 0. During loading internal segment numbers (SSN) in the segment address tables and by further entries in the stack segment.

Much of the arithmetic of a Q-Code process is performed on operands addressed by means of the GDB and transferred to the CPU expression stack (EStack).

42.2 POS File Types

.SEG - A file containing Q-Code generated by the PASCAL compiler.

.RUN - A file containing Q-Code and the framework to allow its execution, that is, code and data segments linked to form a process.

.MBOOT - A file containing the microcode interpreter.

.BOOT - A file containing the operating system.

42.3 POS File system

- hierarchical, multi-directory file system
- supports search lists and non-contigous files
- random access devices are divided into a number of sections called partitions. Each partition can contain any number of directories and each directory may contain other directories or files. Directories are stored as standard files and can be accessed by any program.
- to maximize disk utilization neither files nor blocks within files need to be on contiguous disk sectors
- PERQs file system stresses reliability
- All critical retrieval information is redundantly stored, and verification checks on disk operations insure the integrity of files
- a special Scavenger program can reconstruct damaged disk structures if necessary
- The boot area may possibly contain a table of offsets for those files referenced by boot letter, so if there's garbage or zeros or the address of a file that doesn't contain valid boot code, the DDS hangs in the 150-160 range during booting. Then for binding accent boot files, BindBoot determines the addresses of the specified files and writes them into that table. [CL]

42.4 POS Display Window Manager

- used to manipulate screen information
- partitions the screen into separate areas of windows
- windows may be moved around the screen, enlarged, or contracted in two dimensions and scrolled, all under user control

- windows can overlap each other and can be as large as the entire screen or as small as a postage stamp
- Provides facilities for the subdivision of the display into a number of different areas, or windows. This permits the display of graphical information in a designated window and provision is made for the creation, deletion and modification of the size and position of these windows. Using a RasterOp procedure, it is possible to develop an enhancement which allows windows to appear to move over each other while preserving the display beneath. The use of windows permits the user to selectively display items from a considerable quantity of textual and graphic information, in much the same way as one would select from material laid out on an office desk.

POS makes very efficient use of the PERQ's unique features and was designed specifically for the PERQ.

Provides facilities for microcode supported line drawing.

Although the graphics primitives available on the PERQ are evidentally very powerful tools, it was felt (by who? ICL?) that there was a need for a graphics application package. A basic graphics package for the PERQ was being developed by ICL (what was the name of this package, GRAFIKS?). In addition, ICL and the SERC were collaborating on the implementation of the Graphics Kernel System (GKS), version 7.2 on the PERQ, as of April, 1983. (a library of graphics routines?)

42.5 POS D.265

Appears to be a fairly standard POS D system, with one exception: it has a CMU-modified streams package - it allows command history recall, command-line editing using Emacs commands (like the Accent shell). When it boots up, it prints almost a half page of commands. The person who built the system used an old-English type of font.

42.6 POS F.2

Required new PROMs on I/O and CPU boards.

Major changes to I/O subsystem:

- Volume system: This subsystem provides the new interfaces to the hard disk.

- A "great deal of functionality has been added to POS RS-232 support"
- New functionality added to GPIB

42.7 U.S. versions vs. U.K. versions

In the U.K., versions of POS were apparently numbered differently. For examle, there were versions of POS such as R.4 and R.5.

42.8 FixPart

FixPart can be used to repair a corrupt DIB. It knows the names of all the existing partitions, so it should be necessary to only name the device "Sys" again,

42.9 QCode Assembler

There is no seperate Q-code assembler under POS - the Pascal (and Fortran 77) compilers write the ".SEG" files directly, not as a seperate pass through a seperate program. You can, however, use "QDis" to disassemble a compler program and look at the output of the compiler, for whatever good that does. [CL]

42.10 Scavenger

"I used to think that neat pattern from Scavenger was actually something useful, like the contents of its internal data structures being built up in memory as it did its processing or something like that. But if you look at the source code, it's just "eye candy"... It's just to give you something to look at and know it's working. Parts of that display really are a glimpses of memory being updated as the program runs (a display hack unmatched on any workstation I've ever seen :-) but the regular striped pattern is "faked." (More proof of this: look at scavenger.animate with the CursDesign program...)" [CL, Nov. 6, 1992]

42.11 Scrounge not reporting uncaught exceptions

"I once had a machine that never reported uncaught exceptione - ANY error in a program caused it to crash. The problem turned out to be that "scrounge.seg" was corrupted, and so when any program would die and the system tried to invoke Scrounge, that was it - blammo. I reloaded that one .seg file, did a makeboot, and then relinked everything on the system (fun, fun). Things are fine now." [CL, Nov. 6, 1992]

42.12 Shell

"The real problem with POS is that the shell doesn't remain in memory when you invoke a program - the shell has to be reloaded after every user program exits. If the system kept the shell memory resident while user programs run, it would probably be much quicker."

"(One of the hardware enhancements 3RCC planned was a cache/pager unit of some sort - probably an enhancement for the I/O board to cache disk reads and writes, since the system itself doesn't do it. Accent does do some disk caching, I believe, if you turn on "lazydisk"...?)" [CL, mailing list, Nov. 13, 1992]

42.13 QCode - intermediate code for POS

"QCode is the "native instruction set" Pascal and Fortran are compiled into QCode The QCode interpreter, written in microassembler, fetches QCodes and executes them" [Brian Rosen, mailing list, Dec. 7, 1992]

When asked by Malcolm: "Into which of these two levels to the Qcodes fit?":

"The "assembler" of most computer systems" [Brian Rosen, mailing list, Dec. 7, 1992]

When Malcolm asked: "I am also perplexed as to why no-one appears to know of the existance of a native-code assembler for the PERQ-1A. Surely, someone, somewhere must have written one once?":

"Oh yes, there is most definitely an assembler, and a debugger (but that needed two Perq's and a set of boards that allowed on a Perq, running the debugger to control the other)." [Brian Rosen, mailing list, Dec. 7, 1992]

"More on this. QCode is a descendent of PCode, which was invented by the USCD people as the way USCD Pascal worked. The compiler generates PCodes. To port USCD Pascal to another machine, you write a new PCode

interpreter. On most systems, PCode is written in assembler for the native instruction set. On a Perq, the PCode is written in microcode. Now QCode is not exactly PCode, but it was derived from it.

As an interesting sidelight, Microsoft now has an equivalent of Pcode in Windows; PCode is efficient in code size; much more than 80x86 instructions. It is used for code which is not performance sensitive, and is intended to reduce code storage requirements for big programs. The Microsoft C compiler can generate PCode, and Windows has a PCode interpreter.

PNX used a different instruction set from POS and Accent. Some programs loaded in their own microcode which augmented QCode for special purposes." [Brian Rosen, mailing list, Dec. 7, 1992]

At a later time when I posted questions about QDis and an assembler:

"There's something that's been puzzling me about the PERQs running POS: why was a disassembler (QDis) provided when no assembler was provided? Does anyone know of any assembler that was ever written for the PERQs?"

the following reply was posted by Brian Rosen (September 10, 1993):

"QDis exists as a last resort debugger of the compiler. Some code was written in "assembler", but this was done with the inline operations of the compiler."

42.14 Undeleting Files and Scrounge

> The Perq has a very good, very interesting way of doing this. It is also wrapped up in the "scrounge" program which fixes broken disks. The mechanism is very simplem and very non-portable.
>
> On a standard disk, the surface is broken into tracks, and each track is broken into sectors. Sectors are broken into two parts. The first part is a header which tells the disk controller which sector this is. The second part is the data section, it holds one block of data. The sector headers are written once during formatting. The data portion changes on each write.
>
> On a Perq, each sector has three parts, not two. There are two headers, the physical header and the logical header. The physical header looks and acts just like the standard header, and is written once during formatting. The logical header is written the first time a block is written (when the space allocation is made). It has the fileID of the file to which this block belongs and a pointer to the previous and next sectors that have blocks in this file. By reading

the logical headers, it is possible to rebuild the entire disk volume data structures. Sectors that are unallocated are part of a "free" list, which is doubley linked like a real file. The directory structure is redundantly kept by having a "zeroth" block in every file that has a copy of the directory information. By reading all the labels, you can find all the block zero's, and recreate the directory.

The disk controller is designed to do this, and it is custom. This makes it impossible to use a SCSI disk, or a standard disk controller with the POS file system.

I actually don't remember how the undelete command works; ie what structure remembers that there was a file, and which was the starting block (sector). It may be something like the directory entry is left in the directory, but marked as deleted. The files are placed on the end of the free list. Undelete looks at the directory for the file, and fixes it, and then looks at the logical block chain to make sure it is still ok.

The directory structure has the string which is the name, and the starting (zeroth) sector number. The actual directory entry has a redundant list of blocks that comprise the file. In fact, every datum in the file system is replicated in two places in two different structures.

Scrounge works by scanning all the logical headers, putting the chains together, recreating the free list, and recreatng the directory.

[Brian Rosen, mailing list, Jan. 21, 1993]

Chapter 43

PERQ Power Supplies (PSU)

The processor power supply provides power for the ... display unit +55V (except for those with an integral psu). Kriz type monitors will require 24V in lieu of 55V.

Try to find out the specifications for the power-supplies used for all of the PERQs.

The PERQ-1 supplies (at least the Boscherts) have an adjustment screw for calibration; you need to apply a simulated load, power it up, and tweak the screw until you get +5.1 volts (nominal - though 5.0-5.2 is okay). [CL, May 11, 1993]

Chapter 44

PERQ Prices

PERQ1 with 256K, 12 MB disk as of March 1, 1981: $27,000. options:

```
10M Bit Ethernet: $3,600
1M Byte  floppy drive (8"): $2,200.
24M Byte hard disk: $1,600
Memory parity option: $700
4K writable control store: $4,000
Daisy wheel printer: $3,900
Streamer cartridge: $4,800
Mag. tape option 9 (9-track?): $4,500
Source code for POS: $15,000
System software: $340
PERQ I/O wire-wrap board: $320
Button cursor (?): $300
Extender cable (?): $500
Link board set: $5,000
16K writable control store: $8,000
Paging box (?): $6,000
1024KB memory: $5,000
```

Field upgrade costs (in addition fo option price): (for board swapping!)

```
16K Memory: $2,000
1M Byte: $2,000
```

```
Paging box: $1,500
```

PERQ AI (T2) available for shipment in July, 1984 at a price of less than $40,000. (single unit quantities)

The LN3000 with a monochrome display is priced at $25,000 to $35,000 and was available as of May 23, 1984. (single unit quantities) The AI workstation not available until July, 1984.

Chapter 45

Laser printer

Any laser printer with the "raw" Canon CX engine should work with the PERQs.

The line #SID CanonCX needs to be in the default.profile file following #Login in order for screen dumps to work under POS.

"Essentially what you want is the "raw" Canon CX engine. I have two with the Interleaf label slapped on them. They have a 37-pin "D" type connector that is not Centronics compatible. In your PERQ you need a Canon controller (or a combination Ethernet/Canon board, referred to as the Eth-Can board... don't think it ever had an official designation. :-) and the proper cable. Essentially the Perq builds the bitmap for the printer in memory and simply shoves the bits down the wire to the "printer", which is really just the marking engine." [CL, March 31, 1993]

"You see, I've acquired a Canon LBP8A1, with major electronic problems, and Canon will not supply a service manual. However, I've deduced some things about it by looking inside, and one thing that struck me is that the electronics is split between that mounted on the engine chasis, and the controller board in the top cover that contains the 68000 CPU. Now, according to the PERQ schematics, the laser printer port on the PERQ 2 is neither serial (well, not RS232 or anything like that) or parallel. It appears to be a direct connection to the laser engine, and may well output a carefully-timed bitstream that directly modulates the laser." [ARD, mailing list, March 31, 1993]

> CX printers, when bought from Canon, have a "Video" interface, a single bit serial data stream that has the page image expressed as a bit map. It is up to the vendor to turn the page description language into a bitmap. Some kind of board is required, and

it may have various interfaces. Imagen printers originally went from Impress, the Xerox page description language to bitmaps.

Perq never sold Imagen printers. We sold a printer based on a Canon LBP-10 engine, not the CX engine. It also had a video interface. A special board we designed drive it. It created a data stream very much like the way the video is created for the display; i.e. mostly microcode and a little bit of shift register data-path.

[Brian Rosen, mailing list, March 31, 1993]

Chapter 46

PROMs

I had noticed that there appears to have been some sort of PROM burning/reading hardware that could be attached to the PERQs, since I found some software used for this. Brian Rosen replied to my question about this:

"We did build a PROM blower, it was wire-wrapped, and plugged into the Perq somewhere, maybe the link socket. I don't think it was serial or GPIB." [BR, mailing list, Dec. 9, 1992]

Chapter 47

QNIX

- QNIX provides a UNIX System V interface and access to System V system calls and utilities.
- can be integrated into LINQ
- built upon Accent's virtual memory management, file system, and IPC facilities; as a result, QNIX users enjoy all of the benifits of full demand paging, a transparent distributed file system, and access to network resources
- complete UNIX System V environment with over 150 utilities
- provides System V system calls
- use of the network is transparent
- standard UNIX shell
- Because QNIX is built on top of Accent's virtual memory management, file system, and IPC facilities, QNIX users have transparent access to network resources.

Chapter 48

PERQ Quirks and Hints

48.1 Directory Limitations

The directory command cannot list all of the files in a directory if there are more than 599 files in that directory.

48.2 DDS Bug

According to the PERQ T2 service guide: "CAUTION. The DDS hardware has a fault in it, which sometimes causes the number displayed to be 1 less than it should be. When PERQ and DDS are working correctly, DDS pauses at 10 before PERQ loads the bootstrap; if the DDS bug is active, DDS will pause at 9."

48.3 CAUTION PERQ2 Overtemperature

There is no overtemperature sensing. When servicing always check:

- fan tray connector pushed well home
- all fans are running

Note that the Z80 I/O circuitry supposedly contains some sort of temperature sensing; why isn't this used? Was is used for the PERQ1?

48.4 CAUTION Fan types

Fans from both Etrie and Papst are used (PERQ2). All four fans must be from the same manufacturer to avoid beat frequencies since the Etrie and Papst fans have different rotational speeds.

48.5 Screen optical coatings

Some video terminals have a special anti-glare coating applied to their screens. Try to avoid touching these screens. If a screen is contaminated by oils from the skin, use only the recommended cleaning pads to remove the oils and wipe the screen dry with a soft paper tissue or a lint-free cloth. I think this pertains to some of the PERQ1 displays.

48.6 CAUTION T1 PROCESSOR MOVEMENT

Be very careful not to jar the processor cabinet when the 35 Mb disk drive heads are unlocked.

On T1 systems the shipping latch should be locked if the cabinet is to be moved. If this is not possible for final positioning, under a table or against a wall, great care is needed to avoid damage.

T2 systems using the Micropolis 1300 series hard disks have their heads parked automatically.

48.7 CPU Board PROMS/Disk Drive Types

Although the same EIO boards can be used for both 5-1/4" and 8" hard disks by changing a few jumpers, the CPU boards require different PROMS when different size hard disks are used (5-1/4" vs. 8").

48.8 PERQ-1 System Warm-Up

Some PERQ-1 systems need to warm up for a period of several minutes to several hours to let component temperatures stabilize. If not warmed up, disk read errors and uncaught exceptions will arise when booting or during other disk accesses.

48.9 Undeletable Files

What to do about seemingly undeletable files:

There are two ways to deal with this problem which occurs under POS and Accent. Apparently, the problem is a result of a bug in the Accent Kernel, since the following was found in the source code to Scavenger for Accent:

Revision note for version 0.9 of Accent Scavenger:

"The kernel can't deal with 'Bad' segments at all - it refuses to read, write, delete, etc. so we make the "bad" segment be a permanent segment. It really isn't bad - in fact, it's perfectly well formed (except for the random index, which runtime scavenging will add for us). It's only called "bad" because it has the pages of all the segments that WERE bad."

I've found that running the Accent Scavenger on a new partition, which will be used for Accent files, before adding any files to it appears to resolve this problem of files appearing in the directory listing which can't be read or deleted.

When using POS, simply running Scavenger and answering yes to the prompts after it asks if the directories should be rebuilt solves this problem.

48.10 Stut

Never use any versions of this prior to version 1.8. Version 2.0 is preferred. (are these just the POS versions?) [Dave Callen, according to Chris Lamb] Note: there's also a version that I hacked so that it can dump tape directory listings to a file. [RDD]

48.11 PERQ1 vs. PERQ2 Floppy Drives

Although the 8" floppy drives used in the PERQ-1 and PERQ-2 series systems appear to be alike, there is one small difference: the ones in the PERQ2 systems have slightly larger front panels. When attempting to place a drive from a T2 into a PERQ-1, I was unable to put the PERQs front panel back on because the opening for the floppy drive was just a wee bit too small (probably about one-eigth of an inch or so).

48.12 Exploding Capacitors on CPU Board

"I left the front cover of the machine off, (pulled the "Pull to cheat" switch :-) and hit the button.

Something orange and glowing shot out of the card cage.

The system booted and worked file.

Later, upon examination of the CPU board, a bunch of capacitors right around the microsequencer were scorched and burnt. One of them, in a fit or heroism to "save the fuses!" had met its fiery demise.

Say what you will, but the Perq is built like a tank. It just keeps rollin'!"

[CL, Nov. 4, 1992]

48.13 Exploding PERQ-1 Motor-Start Capacitors

This has only been reported as happenning with PERQs running off of 220VAC.

48.14 Slipping Drive Belts

PERQ-1 hard drive belts have been known to slip off.

Chapter 49

S.E.R.C.

- The SERC, Science and Engineering Research Council was (is?) one of the biggest users of computing resources in the U.K.

- Had anticipated a transformation in computing methods, single user systems (SUS) and distributed computing networks, etc., by developing a coordinating plan, the SERC Common Base Policy, which is aimed at avoiding wasteful duplication of effort in the development of basic software for single user systems and providing the hardware and software technology to enable an efficient utilization of such resources.

- The SERC's plan envisaged that by 1983 SERC would have over 200 PERQs distributed throughout the UK at universities and SERC laboratories, and that these systems would be linked to each other, and to existing facilities, by local and wide area networks.

- SERC's Common Base Policy was not intended as a fixed standard, but was expected to evolve in an orderly way, to reflect continuing technological innovation and advances in computing methods.

- Anticipated a transition, which was to take place around the end of the decade (approx. 1990), to computing systems in which the operating system, as well as computing facilities supplied by network servers, would be distributed over the network.

- Adopted UNIX as the SERC Common Base Policy standard.

"Someone who worked for ICL or SERC in those days should be able to give a comprehensive answer, but in the meantime from a user's memory:

c. 1981/82 ICL started selling the PERQ under license in Europe. Mid-1982, the Science and Engineering Research Council recommended the Perq as the workstation hardware element of their Common Base Policy (or whatever its name was then). As a result of this, in the autumn of 1982, the

funding body for computing in universities, recommended that a Perq (then a Perq1 with 4K wcs, 1MB ram, 40MB disk, running the PascalP-code based single-tasking POS op sys) be bought for each of the 33? university computing centres; these were bought at a bulk price of GPB .5M (approx. 33% discount). [I recall that the price we quoted it those days for a somewhat lamer is is still? ;-¿ Sun was twice that.] A year later, system were shipped with 16K wcs which was needed if the system was to run PNX (pronounce how you want, but usually P.N.X.), ICL's v.7 unix.

I assume that eventually there were more than 500 in GB (there were certainly Perqs in NI)."

– Charles Curran, PERQ-Fanatics mailing list, July 30, 1993.

"Actuallly the first one's sold to SERC had 256K RAM (of which the screen took 96K!) and a 12MByte disk. Remember the POS B2 filing system? - but that's another story."

Initially the ICL Perq development team was located in Bracknell and the Perq marketing team in London. Then the marketing team was moved to Slough. Then ICL decided to move development to Dalkeith in Scotland - all the development team bar two or three (which included Nick) left. ICL decided to locate developemnt and marketing together in Kidsgrove. *All* the development team left and formed Spider Systems, OWL etc.. and all the marketing team left to form Advent.

Now you understand why the Perq3 never hit the streets!"

– Graham Underwood, PERQ-Fanatics mailing list, July 30, 1993.

Chapter 50

Serial Input-Output

Two (one for PERQ1) serial input-output (SIO) chips provide four (two for the PERQ1) channels which are used for:

- RS232 interface A (PERQ2 only?)
- Speech output and standard tablet input
- RS-232 interface B
- Keyboard input (PERQ2 only?)

The SIO chips and associated timer circuits are initialied with parameters which define the protocol for each of four channels. The RS-232 interface protocols are defined by PERQ software to the Z80 CPU which loads the SIO and timer registers. Operation up to 9600 baud is supported. The protocols for the speech/tablet and keyboard channels are fixed as:

- Speech output - synchronous, 8-bit characters, sync character 55, continuous syncbytes and transmitted when there is no data. 32 KHz clock.
- Standard tablet - synchronous, about 90 messages a second consisting of a syncbyte of 7E and 4 data bytes. 32 KHz clock.
- Keyboard input - asynchronous, eight data bits framed by a single start and stop bit for each key pressed. 30 KHz clock.

50.1 PERQ RS-232

50.1.1 PERQ1

- full duplex
- high-speed serial data port
- supports asynchronous, bisynchronous and SDLC/HDLC/ADCCP protocols at up to 56K bits per second
- all line and protocol parameters are programmable
- modem control is standard
- PERQ includes a single full bit stream at up to 9600 baud (Note: this ad mentions nothing about the 56K bits per second like the other did)

Chapter 51

PERQ Software

The user writes programs in high-level languages, and his statements are compiled to generate an intermediate level code. This intermediate level code is not executed directoy, but makes entries to sequences of micro-instructions which the central processor executes.

The (micro?) instructions are single byte instructions followed by up to 4 bytes of parameters.

The POS system's intermediate level instructions are in Q-code.

The PNX system's intermediate level instructions are in C-code.

Interrupt requests are sent to the CPU from the memory and I/O boards when they require attention.

Requests remain pending until polled by the microprogram.

Each interrupt causes a unique value (vector) to be input to the micro-sequencer.

The CPU microcode jumps to the routine appropriate to the interrupting device.

The interrupts are not maskable or unconditional.

51.1 Interrupts

The memory and I/O boards send interrupt requests to the CPU when they require attention. The requests remain pending until polled by the micro-program. Each interrupt causes a unique value (vector) to be input to the micro-sequencer. The CPU microcode jumps to the routine appropriate to the interrupting device. The interrupts are not maskable or unconditional.

51.2 Known Operating Systems

```
POS - PERQ Systems
MPOS - PERQ Systems (Multitasking POS)
Accent - PERQ Systems and CMU
QNIX - Spider Systems (UNIX that runs under Accent)
SPOONIX - ?
PNX - ICL
FLEX - RSE
```

51.3 PERQ Pascal

- Upward compatible extension of the Pascal programming language
- "Designed to support the construction of large systems programs."
- Extensions are based on:
  - BSI/ISO Pascal Standard
  - USCD Workshop on Systems Programming Extensions to the Pascal Language
  - PASCAL* [P*] (the * should be superscripted) - Based mostly on this.

When I asked about any influences of Modula-2 on PERQ Pascal (since Niklaus Wirth was at Xerox PARC), Brian Rosen replied:

"Wirth was at PARC around when I was there, but I did not have much contact with him. We were aware of Modula-2 when it came out, but I don't think you can say that Perq Pascal was heavily influenced by Modula-2. There are MESA like characteristics in Perq Pascal. John Strait was the original compiler guru, most of the language constructs are his work. He was a Pascal purist, and was always trying to keep extensions and changes to a minimum." [BR, mailing list, Jan. 5, 1993]

51.4 Microcode/Microprogramming

---

* Microinstructions*

Each microinstruction is a collection of 12 fields with a total of 48 bits.

Each microinstruction is executed by the machine in 170 nS, or 1 micro-cycle.

An instruction is a sequence of phrases.

Microinstructions are produced, from one or more instruction, by the microassembler.

Format of a microinstruction:

```
+--------+--------+---+-+-+-+----+--+----+--------+----+----+
|   X    |    Y   | A |B|W|H|ALU |F | SF |    Z   |CND |JMP |
+--------+--------+---+-+-+-+----+--+----+--------+----+----+

F - Function field: controls interpretation of SF and Z field contents

The ALU field of a microinstruction encodes the function used by the
ALU to combine the A and B inputs to the ALU in the following ways
(there are only 16 functions listed here, shouldn't there be 17?):

A
B
NOT A
NOT B
A AND B
A AND NOT B
A NAND B
A OR B
A OR NOT B
A NOR B
A XOR B
A XNOR B
A+B
A+B+OldCarry
A-B
A-B-OldCarry
```

- OldCarry is the carry from the imediately preceeding microinstruction and is used for multiple precision arithmetic.
- The A-field of a microinstruction contains the select lines used to drive the AMux. It selects: Shifter output, NextOP, IOD, MDI, MDX, UState, XY register at the location specified by the X field and the top of the expression stack.

- The B-field of a microinstruction contains the BMux select line. If B is set, the BMux selects a constant. Otherwise, BMux selects the register specified by the Y-field contents.

- The X field contains the address for the X port. This same field is used to reference a given register in the XY file for a register write operation.

- The Y field contains the address for the Y port. It can also be used as the low-order byte of a constant.

- Microstate condition codes can be read as the Microstate Registers (UState) via the AMux.

- The shift matrix (Shift) can be accesses by loading an item to be shifted via the R bus and reading the shifted result on the AMux.

- Addressed micro-instructions are buffered in a 48-bit micro-instruction register.

- Various fields of a microinstruction control:

  - operand selection
  - mill function (what is this?)
  - jump conditions
  - memory access

- Single instructions combine arithmetic or logical operations with conditional jumps.

- Microinstructions control CPU access to the memory, allowing the:

  - Storage of 1,2 or 4 16-bit words.
  - Fetching of 1,2 or 4 16-bit words.

- The words may be accesses in forward or reverse order (MADR:MDI:MDO).

51.5 PERQ1 vs. PERQ1A programming

The assembler reflects features of the hardware specific to the PERQ1 or PERQ1A CPU by restricting usage of the following features:

- The percent sign which signals the use of the base register may only be used on the PERQ1A.
- MQ, RBase, Victim, LeapPop, Goto(Shift), MultiplyStep and DivideStep may only be used on the PERQ1A.
- LatchMA may only be used on the PERQ1.

51.6 Mint

Chris Lamb's stepfather is "close personal friends" with Peter Hibbard, the author of Mint.

Peter Hibbard from CMU now works at Adobe Software.

Mint is extremely "Scribe-like" (CL).

51.7 TeX

A TeX dvi previewer is available (Chris sent Bill a copy); what OS is this for? I haven't received a copy of this yet.

Robert Marshall has a version of initex and virtex (3.141) for PNX, together with the tex and latex format files. As of Feb. 26, 1993, he was "in the midst of 'porting' xdvi". "They run tolerably fast on a perq-1a." He would like to know if someone has a dvi previewer for PNX.

51.8 PERQ POS Software Written/Being Written by Malcolm Shute

(this is was as of Feb. 15, 1993, descriptions by Malcolm)

MJS000: assorted utility programs

- calc.pas (a simple desk calculator), complete.
- create.pas (a note book, to create a file without clearing the screen), complete.

  - wc.pas (line, word and character counter counte for a file), complete.
  - diff.pas (gives a true/false answer for if two files are identical, complete.

MJS001: sed.pas (stream editor), complete.

MJS003: deltree.cmd (delete a whole tree of directories, subdirectories and files), complete.

MJS005: cpa.pas (critical path analysis), complete, but subject to future enhancements.

MJS002: copytree.pas, awaiting inspiration

MJS004: spell.pas (spelling checker), awaiting a program that sorts lines in a file into order and awaiting VMS-to-PERQ conversion.

MJS006: spelling.pas (a spelling and times-tables test for 7 year olds), complete.

MJS007: abotec.pas (a spreadsheet program for engineering design), awaiting SUN-to-PERQ conversion.

MJS008: turtle.pas (a program to draw pictures on the screen by leaving a trail of asterisks, to act as a vehicle to teach pascal programming to 7 year olds), complete.

MJS009: typehit.pas (a program to teach you to touch type, using the medium of a Dungeons and Dragons type game (monsters have home key names on Level-1 and successively harder key combinations on levels 2 to 18, though levels beyond that are also supported), awaiting VMS-to-PERQ conversion and the installation of spell.pas dictionary.

MJS010: kingdom.pas (a boring game in which you have to decide how much grain and land to store, or invest in food and harvesting), awaiting VMS-to-PERQ conversion.

MJS011: mroff.pas (Malcolm's runoff) A pre-processor to any text formatter that is either based on RUNOFF or which can be restricted to behave like it (e.g. as a front-end to enhance basic RUNOFF programs to having something approaching the power of nroff, or de-enhancing programs like TeX to behave as a nroff type formatter... but awaiting SUN-to-PERQ conversion).

Chapter 52

PERQ Solar OS

"Solar was the name of Perq's release of Accent." [Brian Rosen, mailing list, Dec. 7, 1992]

"I liked the one part in a document about the PERQ's I/O as it related to Solar, that described the networking as the "Solar system", as in a universe of interconnected PERQs." [RDD, mailing list, Dec ? 1992]

Chapter 53

PERQ Power supply specs. and system dimensions, etc.

53.1 LN3000

Power requirements:

- 100/115/240 V
- Line frequency: 50/60 Hz
- 880 watts

Environment:

- operating temperature: 15 degrees to 32 degrees C
- relative humidity: 20% to 80%

Dimensions:

- base unit: 26” high x 15-1/4”wide x 30” deep
- portrait display: 16-1/2” high x 11-1/2” wide x 14-3/4” deep
- landscape display: 17-3/4” high x 19-1/2” wide x 18” deep
- keyboard: 1-3/4” high x 8-1/4” wide x 10-3/4 deep
- graphics tablets:

  - portrait: 1/2" high x 8-1/4" wide x 10-3/4" deep
  - landscape: 1/2" high x 12" wide x 10" deep

- digitizing tablet: 9/16" high x 15-1/2" wide x 15-1/2" deep

Chapter 54

PERQ Speech Synthesizer

- A continuously variable slope delta (CVSD) modulator is employed at a 16 kilobaud data rate to provide arbitrary stored speech output.

- Pre-recorded CVSD data is stored on the system disk for voice response, audible signaling, and other speech applications.

- A Continuously Variable Slope Delta Modulator is used to provide arbitrary stored speech output.

- Can be used for a speech synthesizer to go along with voice-input circuitry in order to provide a user interface less conventional than a screen and keyboard.

One of the .spk files include one of a woman's voice announcing the PERQ as a "landmark computer system".

Dave Callen gave me a file with some music by Handel on it. I was told by him that someone in Japan was exerimenting with the PERQs sound capabilities and created the Handel.spk file.

"I think that a guy named Phil Mason, up in Wisconsin (?) somewhere actually *has* one of the microphones they used to record the sounds for the demos, or at least has a schematic for making one work. It's really a very simple interface, so I can't imagine that it's difficule to rig something up."

– ?

54.1 How Sound is Recorded onto a PERQ

"This is all kind of fuzzy. The Perq uses a sound chip that implements an algorythm called Delta Modulation. This is basically a compression technique that lets the bit rate for speech be low. The chip is bidirectional, but I don't recall that we had the input analog circuitry on the board. Maybe we had an off board box that was used to record sounds. Since the algorythm was known, and simple, some folks converted sounds they had to delta-mod form, and some of the music is done that way. However, as implemented, the sound system does not have high fidelity (about 6 KHz or so as I recall), and would not make a good music source." [Brian Rosen, mailing list, Dec. 9, 1992]

Chapter 55

PERQ Tablet

(LN3000)

Tablets with mouse:

- completely flat surface
- active area 8.25” x 10.75” (portrait) or 12” x 10” (landscape)
- electromagnetic operation
- three button mouse
- maximum distance from display 8’ (er, back of display)

Optional high resolution digitizer:

- active area: 11 x 11
- 0.005” resolution
- choice of four button puck or stylus
- maximum distance from base unit 8’

Chapter 56

Technical Miscellany

The PERQ graphic workstations were most unique computing machines, several years ahead of their time. Blazingly fast raster-Op, blit-blit, graphics with 60 Hz refresh, non-interlaced, displays of up to 1024 by 1280 (?) resolution. The ability to be controlled via voice as well as by a keyboard and tablet. Several different graphic user interfaces (GUIs). A CPU which is "microprogrammable on a per-process basis" [RM 25 Sep. 90]; that is, it is possible to have different microcode running as seperate processes. For example, one can be running the Accent operating system while having several processes running seperate microcode, concurrently, in different windows. One can have a LISP machine in one window, the Unix OS running in another window, and an Accent OS command line interpreter prompt in yet another window. Virtual memory. Ethernet networking.

The main concept of the PERQ was that of giving individual users the same benefits that they would gain from using a multi-user mainframe, or mini, computer system. The benefits to the PERQ approach, as compared with a large multi-user system was that each user would have quick response time from the system which would not be affected by any loads that other users might place on a multiuser system. In addition, the user would have a display capable of displaying high-resolution graphics which could be quickly manipulated. By having their PERQs attached together in a network, the users could share files with each other and if one system connected to the network crashed, few, if any, other users on the network would be seriously affected by the crash, since they could still use their own system, whereas if they were using a multiuser system, none of the users would be able to use the system if the system crashed. Another use planned for the PERQs was the role of intelligent terminals attached to larger multi-user systems which

could be used to take some of the processing load off the main system, as well as providing users with high-resolution graphics displays. On the other end of the computing spectrum were small, microprocessor based, personal computer systems which were affordable enough to be purchased for individual use. These conventional personal computer systems do not have the resources necessary to compete with the larger timeshared systems. The essence behind the concept of the PERQ was to provide users with the best of both worlds by providing a user with a very responsive and powerful computing environment that has high quality graphics, large local storage capacity and the ability to connect with other systems through a local area network.

Although some people's recollections of the PERQs are of slow, noisy and unreliable machines, whose primary benefit was the ability to take the chill off of a cold room, these were most often the opinions of people who had never used the newer PERQ T-2 systems. The newer systems were much more responsive, primarily due to quicker hard disks. The speed of the CPU itself never was what one could call slow at the time. Depending on the operating system and the languages programs were written in, the speed of the PERQ ranged from 1/3 of the speed of a DEC VAX 11/750 to the same speed as a VAX 11/750 when the PERQ was running microcode for Spice LISP.

The original PERQ 1 systems were not the most reliable computers ever made; you could not always depend on them to boot up on the first throw of the switch. It was almost as though the machines had to be in the proper mood to be booted, and this tempermentality was enough to make one want to boot a PERQ in a different sense of the word. However, once the PERQ was of the right frame of mind to be booted, or after several attempts at booting it, or after the boards were reseated, it would spring to life ready to manipulate graphics displays with amazing speed and agility. Aside from the afformentioned unpredictable reliability, the primary drawbacks to the early PERQs were the small writable control store (4K), small memory, 256K, although 1 MB was an available option, limited disk space, 12 to 24 Megabytes, using relatively slow and noisy 14" Shugart SA-4000 series hard disks. Although these disks were slow and noisy, they were durable and had a clear cover that let you watch the arm and platters move if you removed the PERQs front cover; this was something that could make the PERQs fun to use, as long as you were careful not to put your fingers in the wrong place and risk getting yourself electrocuted.

The next step in the evolution of the PERQs was the PERQ2, T-1 system. This system had a more solid appearance and feel to it, as well as a

more sturdy cabinet. The major changes were the optional 20" landscape display instead of the original protrait display and increased mass storage space and speed. Insead of the slow and noisy 14" drives, the PERQ2 T-1s could be equipped with up to two Shugart (model?) 8" hard drives, for a total of up to 64 (?) Megabytes of mass storage. However, the cabinet had onkly enough room for one drive, so the second drive would have to be placed outside of the processor cabinet. Although the storage space had been increased and the data access and transfer times had been improved, these drives were less reliable than the drives in the PERQ1s.

The next, and last, PERQ model to be produced and marketed to the public was the PERQ T-2. Along with a redesigned cabinet, the T-2s used smaller, faster and more reliable 5-1/4" hard disks which were ST-506 compatible. Up to two of these drives could be mounted inside of the processor cabinet for a total of up to 280 Megabytes of mass storage space. At this point, the size of the writable control store was now up to 16K in size and mein memory could be increased up to two Megabytes.

Although PERQ had introduced the newer machines, the older ones could still be upgraded to a limited extent, with the exception of changing the type of hard disks. However, even this was not impossible. Internal company documents showed that it was possible to make certain changes, such as replacing the PERQ1s 14" hard disk with one of the newer, albietless reliable, 8" hard disks; this merely required the use relatively simple special interface circuitry. However, although possible, it does not appear that this was ever an option offered to PERQs customers. On the other hand, customers were able to upgrade the CPU and memory of their PERQ1s to 2 Megabytes of RAM and a 16K writable control store.

Although PERQ Systems changed the type of hard drives used over the years, they continued to use 8" floppy drives, Shugart model 801 (?) and appeared to be reluctant to supply streamer tape drives for use by their customers. The tape drives were originally intended for internal company use only; customers gained the use of the tape drives after finding out that they were already in use by PERQ Systems and requested that they be allowed to use them. Aside from the streaming tape drive, there were plans to use other types of tape drives with the PERQ, such as 9-track reel-to-reel tape drives which could be interfaced to the system in several ways: a connection to the GPIB port and through a connection to a multibus board.

Over the years, through various model changes, the basic PERQ architecture remained basically the same: a processor whose heart was a bit-sliced horizontally microprogrammed micro-engine, based on the AMD 2900 chip.

(further technical descriptions of CPU, etc. here)

One of PERQ Systems former employees, Robert Colwell, who was also one of the engineers who worked on the design of the PERQ color display, had the following to say about the PERQs architecture:

"This architecture is like the victorian home in Texas, where they kept adding rooms until there were hundreds; corridors to blank walls, steps that go and come back down without going anywhere, levels, windows and doors scattered hither and yon. The owner was a major-league eccentric. But the tourists keep on coming..."

There had been plans to implement numeric co-processor circuitry using an Intel 8087, but this idea was abandoned. This coprocessor was to reside on the I/O board; although the plans for its implementation appeared to have been abandoned, the evidence of thie remained. Looking at a PERQ I/O board, one can see where the circuitry for the coprocessor was supposed to be; the traces for it are there and the holes for the chips have been drilled; even the outline for the chip has been drawn.

Unfortunately, an effort to continue on into the future with this type of architecture was halted when PERQ Systems could not obtain the financing necessary to produce a scaled down version of the same basic CPU. Functionally, this new CPU would have been basically the same as the original PERQ CPU, with the exception that it would have been much smaller, faster and more reliable. The basic CPU circuitry would have been reduced to only five chips.

Eventually, plans were made by PERQ systems to abandon the traditional proprietary PERQ CPU and to change over to an industry standard microprocessor like some other computer manufacturers such as Sun Microsystems used: the Motorola MC68010 (?).

Chapter 57

PERQ User Groups

57.1 TRUST

"TRUST" was the Three Rivers Users' Society, the company sponsored users group that distributed some PERQ public domain software, which consisted of such utilities as those used to edit fonts and cursors. There is a reference to this group in the POS D.6 manual.

According to Chris Lamb (Apr. 30, 1992):

"In fact, I believe the person at 3RCC who was in charge of the entire "software library" (including the "user contributed" stuff) was Loretta Ferro. She was a good friend of my stepdad, and we saw each other on numerous occasions. I don't know if TRUST had a newsletter or office or anything. I think it was Three Rivers who acted as a clearng house for some of the stuff that users wanted to share, but I could be totally off.

57.2 Oxford PERQ User Group

There was also a PERQ users group based at Oxford University that supposedly published a PERQ newsletter. The person to contact there about it is Charles Curran. (this info. was from Nick Felisiak)

Chapter 58

PERQ uses

There are lot of people who actually believe that PERQs were made to be used only as "forms design worksations"!

Chapter 59

PERQ Virtual Memory System

(note: for more information, refer to the brochure "PERQ Virtual Memory System", TRC-13. Apparently, this refers to MPOS)

- PERQ's virtual memory system features a segmented 32 bit virtual address space mapped onto a 20 bit physical address (1 megaword).
- A subsequent implementation of the PERQ system will implement a paging system with 256 word pages. The new operators will then return a true 32 bit virtual address and the limitations of 64K segments will be eliminated.

Chapter 60

OSLAN and OSLAN Interface Controller

These logic circuits provide the PERQ interface to an ICL open system local area network Transceiver Unit. The EIO board circuits provide data encapsulation and help in the data link management. A microsecond clock-timer is provided to help in sorting out collisions on the network. The timer is set by software and interrupts the PERQ CPU. The DMA controller handles the transfer of data between the OSLAN controller buffers and the PERQ memory. Separate channels are used for transmitted and received data but transfers are not simultaneous.

Open systems (OS) means that different manufacturers' equipment can be connected to the LAN.

Every packet of information has a 48-bit header which specifies a unique destination device address. All devices listen to every packet of information transmitted.

This is a baseband system. The OSLAN transmission rate is 10 million bits per second (10 MBits).

The ICL OSLAN is like Ethernet. It is a contention network using broadcast techniques. The method of accessing the network is called CSMA/CD. That is carrier sense, multiple access, collision detection. CSMA is sometimes called "listen before transmission". CSMA/CD is sometimes called "listen while transmitting" as well as listening before.

Chapter 61

PERQ Haters

Those Dreadfully Nasty CMU (and other) PERQ-Haters...
Andreas G. Nowatzyk, on July 29, 1993, wrote:

```
"Hmm... I really like to know how and why I was added to this mailing list.

I hope that PERQ is not refering to the machine that was produced by
Three Rivers Corp. in Pittsburgh, PA in the early 80s.  That machine,
the Poorly Engineered, Reduced Quality (PERQ) computer was deployed to
the tune of 150 units in our computer science department at the
Carnegie-Mellon university, where it managed to become the most hated
contraption on campus.  The best use we ever found for it was to use
it as a door stop.  Most were sold for scrap metal to the tune of $5 a
piece (some were still in thier original package - never opened).  I
think CMU got a good deal.

So kindly remove me from this list.

  Let the desd rest in peace -- Andreas"
```

Richard Sanzi, on July 29, 1993, wrote:

```
"Andreas is right.  They were nothing more than Space Heaters.  I
remember several being plugged in and turned on, with no keyboard or
display attached, JUST TO HEAT AN OFFICE.  This was unarguably their
best use ever.

UNSUBSCRIBE me as well."
```

Scott Fahlman, on July 29, 1993, wrote:

"Your damned mailing list daemon refuses to let me unsubscribe. It must have been implemented on a Perq. Please remove me by hand, and the sooner the better.

As a parting shot: The Perq had pretty decent graphics, and its Common Lisp environment was actually pretty good for its day, if I do say so myself. But the Perq Systems people made a *lot* of mistakes with that machine, both technical and on the business side, and we at CMU ended up paying in blood, sweat and tears for every one of those mistakes. You won't find a lot of nostalgia for the Perqs on this campus.

Federal property-management regulations even made it impossible for us to enjoy the satisfying catharsis of throwing all the remaining Perqs off the roof. they sat in a hallway for years and were finally trucked away to parts unknown.

Goodbye, I hope.

-- Scott"

Mario Wolczko (Manchester Univ.), on July 30, 1993, wrote:

"Above my PERQ2, when I used to use one some year ago, I had a slide rule mounted, with a sign that said "PERQ Floating Point Accelerator".

I'm another subecriber to the view that "the only good PERQ's a dead PERQ".

Mario Wolczko"

61.1 Replies to CMU PERQ-Haters

"So, the PERQ Fanatics are back, eh? [...]

So the first post I receive is a flame from a CMU guy. How fitting! :-)

I know that the PERQ has its quirks. I've been through enough of those ill-timed, spectacular only-on-the-PERQ crashes and I've muttered a few curses at the beast in my day. But I love the PERQ for the same reason that I'm a NeXT Fanatic now. *Somebody* had to be first, and somebody had to "raise the bar". It is a curious twist of human nature to resist and fear change and innovaton as powerfully as we tend to do, because we are probably the most adaptable creatures on the planet. But maybe I just like cheering for the underdog, or maybe I just like abuse. :-)

The PERQ started the revolution. NeXT continues it. Someone else, down the road, will cary it further. [...]

[...]

In the meantime, perhaps the CMU guys can think on this: Three Rivers had a pretty tight relationship with them (sales of early machines, development of Accent, etc.) and they went under. NeXT had a pretty tight relationship too (choosing Mach as their kernel, investment money from CMU, etc.) and now they've had to cease hardware production and struggle to maintain as a software-only company. Two companies that did bleeding-edge stuff, were tied to CMU, and had/have a very difficult time in the marketplace...

COINCIDENCE? :-)

Perhaps CMU, much as I like the place, is at fault! :-)"

-- Chris Lamb, PERQ-Fanatics mailing list, July 29, 1994

"Andreas, you're right, the original Perqs were poorly designed, and didn't adhere to even nominal worst-case timing conventions. But in the pot-calling-thekettle-black department, the original Sun workstations weren't much better. The difference seems to have been that Sun had a business plan and gradually improved their hardware. Perq, uh, well, didn't."

-- Bob Colwell, July 30, 1994

61.2 Miscellaneous Mutterings

Michael Lawrie, on July 29, 1993, wrote:

```
"Jamie dear boy, it's rude to forcibly subscribe Sun people to a
mailing list about Peras, now be a good chappie and remove him or
I'll put you on a Pr1me mailing list.

And remove that Allistair chappie too,  Mutter mutter.

Would it be a silly question to ask why I am on here?  The most I ever
did with my perq was turn it on, the sight of that disk between my
legs looking like a log saw frightened me to death and it's been in
the garage ever sinde!

I know it's rude to use your list like this, but hell, who cares, I
don't suppose anyone out there has an Apple Lisa still working that I
can nick some startup disks from?

Michael."
```

61.3 New Mailing List Re-Launched

When the new PERQ-Fanatics mailing list was re-launched, after the calcellation of the first new mailing list due to the automagic subscriptions to it of many people who weren't too fond of the PERQs, on August 12, 1993, the following message was sent to those who resubscribed. I couldn't resist adding the lyrics. :-)

```
   The PERQ-Fanatics mailing list  ( Created at this site : 12th August 1993)
   ------------------------------

      [  Think of the song "Hello, Hooray" by Alice Cooper :  ]

     "Hello, Hooray, Yes This List's Begun, Yes We're Ready..."
     "Hoping that this audience does still like PERQs,"
     "Loving every RasterOp, every frustrated scream,"
     "We've been waiting so long for this list to come..."

Greetings from PERQ-Fanatics Headquarters.  This message comes to you
as part of a secret, not too diabolical, mission in which the goal is
```

to get as many past and present PERQ users to subscribe to a mailing list about the PERQs as possible.

This mailing list is a replacement for the previous PERQ-Fanatics mailing list that was maintained by Chris Lamb, and is archived automatically. To receive archive files (each file contains one calendar month's worth of messages), simply mail perq-fanatics-owner with your request, stating which month(s) you require. This facility will hopefully be automated in the future. Full instructions will be given at such a time.

Presently, membership of this list is unmoderated and subscription, as well as unsubscription, is currently automatic. If you don't wish to remain a subscriber to this list, simply send the message "SIGNOFF" to "perq-fanatics-request".

We look forware to seeing you posting to the list!

Robert,
Malcolm,
Jamie.

Chapter 62

Interesting Notes from POS (G.2) Source Code

At the beginning of the change-log for PERQ.QCODE.7:

"Man is born to trouble, as the sparks fly upwards."

- Job, Chapter 5

* * *

At the end of change-log for the I/O microcode for the Krismas system, EIO.Miicro; it was apparently written by Kwok W. Sheh:

"In memory: This code is dedicated to all of those who have lay the foundation for the perfection of the current microcode technology."

* * *

The PERQ boot loaded microcode assumes that if booting from a LINK a PDP-11 is on the other end.

* * *

Quotation found in CIO I/ microcode, cioio.micro, page 21:

"Important: if there are any micro registers left, store the 4 bytes into them rather than memory, then copy into permanent memory. But for now, what the hack!"

* * *

Around August 1982, PERQ device numbers were changed. For example, the keyboard changed from device 7 to device 10 (octal). Was this one of the changes between POS F and G?

* * *

The Z80 is shut off for a few milliseconds during the boot sequence because "...the hardware has a glitch when first starting up the Writable Control Store RAM, the power surge can screw up the Z80."

* * *

Comment on page 7 of PERQ system loader microcode:

"Good load (we hope)"

* * *

From change log for CIO diagnostic microcode, V1.5; John P. Strait wrote: "Run memory diagnostic over all 2\^20 possible memory words. This is a hack done to make the 1 MByte memory work! The memory diagnostic should really be modified to really test all the memory that is present."

[This appears to have been fixed by W.J. Hansen in the next version, to find the size of the memory and only check that much.]

* * *

In EioEther10.micro, page 6, under

"**** Assumptions, HACKs, and other BAD Things ****"

in section 3, the following appears:

"This feature is found in PERQ, Kristmas and K2. Watch out for other machines."

* * *

In PERQ.QCODE.5:

"Opcode PSW is the unimplemented process swap instruction. Note: this opcode calls UOP. Opcode PBLK is the unimplemented process block instruction."

Are these opcodes that were never implemented, or are they remnamts of MPOS?

* * *

The undoing of MPOS; from the PERQ.Qcodes.Dfs change log:

"Undo changes which the Weekend WOnder Crew made for MPOS".

* * *

Note in change log from PERQ.QCodes.Dfs:

"Note: there are separate definitions of PERQ.QCodes.Dfs for POS and Solar. This version is for POS only!"

* * *

From codegen.pas, page 3:

"Note: PERQ.QCODES.DFS is different for Accent and POS. It is assumed that the Accent version is obtained when the host machine is Accent, and the POS version when the host is POS. With the exception of LSSN and TLATE's, all qcodes generated by the compiler have the same value in both systems. When we read the Accent PERQ.QCODES.DFS, we define missing QCODES to be their (old) values under POS. (This is so that Pascal running under Accent can still generate code for POS.)"

* * *

Jan 6, 1981: First use of standard 3RCC file formats (with Pascal compiler, implemented by Miles A. Barel).

* * *

From procedure ImportDeclaration, file decl.pas, page 2:

Abstract: Parse IMPORT declarations. Only handles IMPORTS line in original program - DeclarationPart actually parses the contents of the imported segment.

```
Design: THIS IS A "HACK".  If a module is imported which contains no
code or variable, and it..."
```

* * *

```
From errormsg.pas, pages 1 and 5:

As of POS G.2, November 13, 1982, the Pascal compiler could not
handles source code longer than 32K.  Resulting error message:

"Implementation restriction - too much code"

Another interesting error message that the compiler can produce:

"Undefined Error"
```

* * *

```
Pertaining to math accuracy, from teh change-log from lex.pas:

"Updated power of 10 table with more accurate numbers" - Michael R.
Kristofic, Dec. 26, 1981.

"Incorporate the modified procedure ConvertReal, which was made
slightly more accurate by John Brodie for Version 7.0 of the compiler
(a version which was produced solely for the F.0 Release)." - Scott L.
Brown
```

* * *

```
Found in change-log of expr1.pas for Bob Fitzgerald's V9.0f revision:

"Cause Write(Character) to use Write.WriteChar instead of File\^:=Char;
Stream.PutC, even when no field width is specified to prevent canvas
from losing the little bugger when its the last thing on a line.
There has to be a better way..."
```

* * *

```
From expr2.pas:

ifNextParam <> nil then Error(126); {Gee! Extra parameters!}
```

* * *

Found in pascal.pas:

"Abstract:

    Based on Zurich P2 Portable Compiler. Modified extensively."

* * *

From pascal.pas change log, Miles A. Barel's 3.1 revision on April 1, 1981:

"Fix Command and Filename parser for the new OS"

Was this for Solar or MPOS?

* * *

Changes made to the Pascal compiler to use a unique type of input file handling intrinsic, as opposed to Pascal intrinsics, were made by Michael R. Kristofic on May 8, 1981.

* * *

Attempted SABOTAGE by someone at CMU? Scott Brown's entry in the pascal.pas change log for V9.9 reads:

''Fix the value of the constant handle_all (it should be true). It seems that someone at CMU carelessly changed it to false for unknown reasons.''

* * *

From page 1 or pascal.pas, Scott Brown's Nov. 13, 1982 entry for V9.3:

"Start a series of bug fixes. This series of fixes is being made to version 9.2, which is "all compilers" before POS and Solar compilers are diverged to have completely separate sources."

* * *

From page 1 or pascal.pas, Scott Brown's Nov. 18, 1982 entry for V9.5:

"Fix the problem of the compiler generating bad code when range checking is changed within a for statement. (This fix compliments of

Dave Golub and Brad Myers, ...the credit/blame is all theirs.)"

* * *

From page 1 or pascal.pas, Scott Brown's Nov. 18, 1982 entry for V9.7:

"This represents a point of divergence of the Pascal compiler. I am taking the current sources of the compiler, modifying some of the constants for conditional compilation, in order to produce a compiler which is to run under POS only! and generate code which is to run under POS only!"

* * *

Page 9 of pascal.pas indicates that 3RCC had considered an Accent to POS cross-compiler, but thought that there was little chance that it would be implemented. However, there appears to be cross compiling from POS to Accent.

* * *

Apparently there was a kluDge in the code for pas0.pas on page 17, according to the following comment:

{we can't really handle wrap around so... chocolate covered fudge}

* * *

From the QDis.pas change log:

John Brodie made a fix to disassemble Solar codes and modified QDis to execute under Solar on Oct. 25, 1982 and Nov. 17, 1982 respectively.

* * *

Again, a mention of the "new OS"; this was from the QDis.pas change log for V1.5 of April 1, 1981:

"Conversions to run under the new O.S."

* * *

From QDis.pas comment, page 28:

writeln(TitleLine); {someday maybe we'll have a solar window title}

* * *

From QDisByteProcessors.pas, page 3:

```
var
   busyBee : integer {counter used to slow down updates to }
                     {ShowProgress -- necessary to prevent bee }
                     {from looking too hyperactive}
```

* * *

From QDisByteProcessors.pas, procedure NextQCode, page 31:

```
{ we don't know about this QCode???? Sigh* }

else
   if doPrtDis then
      writeln(Outfile,'*** Undefined QCode', QC);
```

* * *

The program DiskTest (1983) contains no change-log and no names of those who wrote adn modified it! THis was included with POS G.2 sources, but the only disk supported by this version of DiskTest was the Micropolis 8" drive... strange, since is supposedly the set of floppies used to install POS G.2 on my PERQ-1. Note that later versions of DiskTest are supposed to suport the Shugart 14" drives as well as the newer 5-1/4" drives.

In order to run diagnostics, etc. on the Shugart 14" drives uner POS G.2, the program dtst.micro had to be run from ODTPrq.

* * *

The routine in DTST.micro that reads a scctor was named ‘‘ItsABoy’’. Had someone's wife just had a baby when this was being written? Note: DTST was written by Brian Rosen.

* * *

In Etherhelp.pas, one of the commands was ‘‘Intel’’ which was a mode for use with the ‘‘Intel Development System’’. What was the Intel Development System?

* * *

In high.micro, the ‘‘placer bug’’ is mentioned:

‘‘‘‘if True’’ required by placer bug’’

What was the ‘‘placer bug’’?

* * *

What was the KERNAL program? Was it something similar to ODTPrq?
From comments in mem.micro (test): ‘‘...by typing to ODTPRQ/KERNAL...’’

* * *

An error message form pdm.pas:

‘‘Fatal Error while attempting to report error’’

* * *

Comment in pdm.pas:

‘‘This is a hack since the compiler can’t handle string constants of
length = 1’’

```
Hack := ‘‘<<’’;
Hack[0] := chr(1);
```

* * *

The PDP-11 version of PrqMic was done away with on Sept. 28, 1981
(permic.pas).

* * *

Apparently the RSX-11M O.S. one PDP-11 was used to create microcode
listings in 1980.

* * *

POS streams package uses RSX: as an output device, however, this is
optional, depending on compiler switch settings for conditional
compilation. A file to the RSX11-M OS is opened through the RS-232
port.

* * *

Limited multitasking, sort of, in POS is accomplished via the FullLn function in stream.pas:

‘‘This function is provided in order that a program may continue to do other things while waiting for keyboard input.’’

* * *

Warnings from stream.pas:

*** WARNING ***

It is VERY dangerous to have transcripting on while running the Scavenger or any similar program. No checking is done to insure that this is not done. Caveat EMptor.

*** WARNING ***

Also, it is dangerous to run Typefile while transcripting is on. The data typed out will be wrong (since TypeFile uses ReadDisk) and the PERQ may crash.

*** WARNING ***

From the system.pas change-log:

May 13, 1981 (JPS) - remove vestiges of old system: standard error procedures.

April 6, 1981 (JPS) - Retrofit changes for exceptions to version of system that diverged after March 15, 1981. Change main version from C to D.

March 15, 1981 (JPS) - Put in stuff for exceptions

March 15, 1981 (JPS) - Add code to shrink and expand screen.

*** Caution *** The Shell must never be loaded when the screen is small, because it might allocate a buffer for a command file. This buffer will hang around after the shell returns to the system.

```
                        *   *   *

System.pas change log, Feb, 18, 1983:

‘‘Change system letter to G’’

                        *   *   *

System.pas change log, March 4, 1982:

‘‘Change system letter to F’’

                        *   *   *

System.pas change log, February 16, 1981:

‘‘Change system letter to C’’

                        *   *   *

From system.pas change log (DCF), Feb. 11, 1981:

‘‘Command interpreter made separate from system’’

                        *   *   *

From system.pas change-log (JPS), Oct. 10, 1980:

‘‘Support added for DDS’’

                        *   *   *

From system.pas, Jan 5, 1983:

‘‘Made changes that allow use of Ethernet if system was booted from
floppy.’’

                        *   *   *

From system.pas, August 16, 1982:

‘‘Added new variables to permit selection of either Kriz Tablet of
BitPad.  Permit true relative mode for either.’’
```

* * *

From System.pas, December 1, 1981:

‘‘Make ^C^C clear as ^S’’

* * *

From System.pas, June 1, 1981:

‘‘Do automatic re-enable of swapping if turned off by shell (i.e. for Scavenger).’’

* * *

From System.pas, May 13, 1981:

‘‘Add procedures to enable and disable control C processing’’

* * *

From System.pas, April 6, 1981:

‘‘Virtual memory and timing statistics.’’

* * *

From System.pas, March 28, 1981:

‘‘Changed length of command line to 255 characters’’

* * *

On November 17, 1980, code was added for LastFileName which allows a user to supply the name of a file only once for the editor, compiler and linker.

* * *

System.Pas is basically code which initializes POS and then goes in to a loop alternately running Shell and user programs; it also logs users in.

* * *

From SystemDefs.pas, Nov. 17, 1982 change log:

Planned to eventually introduce a debugging compiler under POS: ‘‘Also, if/when we release a debugging compiler under POS, the modules QMapDefs and SymDefs will need this.’’ Note: ‘‘this’’ was the compiler directive/flag under Accent; set to false for POS.

* * *

From Virtual.pas:

The PERQ virtual memory manager supervises the segment tables and exports procedures for swapping memory segments. Virtual is the portion of the PERQ memory manager which must remain memory resident at all times.

PERQ physical memory is segmented into separately swappable items (called segments) which may contain either code or data.

* * *

From Virtual.pas, change note of February 24, 1981:

‘‘Allow the ScreenSeg to change sizes. To allow this, the memory manager must prevent system segments (those with RefCount > 1) from being moved into the area of memory which might be used for the screen. This is a hack which guarantees that the system can expand the screen to its original size after returning from a user program.’’

* * *

From Virtual.pas:

If system is booted from a floppy, up to 1 MB of virtual memory is available; if booted from the hard disk, up to 31 MB of virtual memory is available.

When memory is needed, e.g. - a certain size of contigious memory, an attempt is made to find a memory ‘‘hole’’ of appropriate size. If not found, an attempt at compacting memory and then finding an appropriately sized hole in memory again. If this fails to provide needed memory, memory will be swapped out.

Memory compaction is done by moving as many segments as possible towards low addresses. System segments (those with a reference count

greater than one) will not be moved into the screen area, as segments cannot jump over one another.

* * *

A note found in RealFunctions.pas:

DISCLAIMER:

Only the most cursory testing of these functions has been done. No guarantees are made as to the accuracy or correctness of the functions. Validation of the functions must be done, but at some later date.

* * *

From Perq2KetTest.pas:

The break key (on the Krismas keyboard) overlaps the cursor control keys. Thus, it cannot be distinguished from them based on keyboard input.

* * *

From TestEther.pas, page 8, procedure Allocate:

Note:

> This procedure is an incredible hack. The EtherBuffer must be allocated on a one thousand word boundary. To do this, we must do some interesting hacking. Also the segment that is created must be marked as Unmovable. This will allow the memory manager to place it into a reasonable part of memory before it is tied down.

* * *

From arith.pas:

The compiler did not originally support type long and used an interim double precision arithmetic package that originated at CMU in 1980.

* * *

From Code.pas:

```
``Also, added other fields to the zero block (of the .seg files) to
provide access to information needed by fortran.''

                          *   *   *

From ComplexFunctions.pas, March 31, 1982:

DISCLAIMER:

   Since Math identities are utilized to evaluate these functions,
   accuracy and execution speed may be very poor.  Only the most cursory
   testing of these functions has been performed.  No guarantees are made
   as to the accuracy or correctness of the functions.  Validation of the
   functions must be done, but at some later date.

                          *   *   *

From Configuration.pas:

Landscape/portrait logic added on Feb. 2, 1983.

VT-100 compatible keyboard introduced with the PERQ K1 (``Kristmas'')
model in Januart 1983.

An Intel 8087 was to have been used as the floating point co-processor.

CIO - 'Current' IO board.

EIO - Ethernet IO board introduced with the PERQ K1 (``Kristmas'') model
in January of 1983.

                          *   *   *

From Memory.pas change-log:

With help from John Strait, Brad Myers fixed a {\underline serious bug} in
InitMemory where it changes the value of a memory location while
trying to decide what the size of memory is.

                          *   *   *

From Memory.pas, page 20:

The only way to address segments bigger than 256 blocks is using
```

RasterOp.

* * *

From ODTPrq.ps, documenting its lineage:

PERQ version - rewritten with ‘‘lots of neat stuff added’’ by John P. Strait, January 1, 1981.

RT-11 version - Bill Glass, 1978

UCSD version - Miles Barel, 1870\footnote{This appears to be a mistake. :-)}

* * *

Comment found in Clock.pas in part of leap-year code:

‘‘Don’t worry about 100 and 400 years. PERQs won’t be around that long.’’

* * *

From CmdParse.pas:

Command names can be no longer than 80 chars.

* * *

From details.pas:

Microcode can read serial number from PERQ2 but not PERQ1 where the serial number is not in hardware.

* * *

From Editor.pas change-log:

Special RasterOp: DrawByte

* * *

From Editor.pas change-log:

Commend from Scott L. Brown, who aded the ‘‘.FOR’’ extension to FSExtSearch to enable the editor to find FORTRAN files:

```
‘‘I do my small part to help perpetuate the dreaded illness we all know
as FORTRAN.’’
```

* * *

```
From editoru.pas, page 21:

**** This is a HACK.  THis funny Raster-Op entry point is a
**** three address instruction and there is no three address
**** TLate.  We get away with it by:
****    1) Do two TLates.
****    2) Translate the screen address after the other two.
****    3) Count on the fact that the screen is not swappable.
****    4) Count on the fact that the two TLates and the JCS
****       are indivisible, but the forst TLate can take a segment
****       fault of necessary.
*****
***** Note that the E-Stack achieves its maximum depth.
*****
```

* * *

```
From FindString.pas:

Note: a POS carriage return is printed as chr(\#215), chr(\#212).
```

* * *

```
From helper.pas:

CR = Chr(\#015)
FF = Chr(\#014)
```

* * *

```
From login.pas:

When asking for user password:

‘‘Enter the secret password for user ’username’’’
```

* * *

```
From makeboot.pas:
```

```
file types (in file information block):

   type 15 - boot
   type 16 - mboot

                          *   *   *

Note from userpass.pas:

``Passwords get encrypted - yet, still not much security!  Try this:
check to see of user guest can be removed.''

                          *   *   *

From allocdisk.pas:

``When allocating pages, the module updates the PartInfoBlock every
MaxAllocs calls on AllocDisk.''

``Since the system may crash sometime between updates, the pointers and
free list may not be accurate.''

                          *   *   *

From diskdefs.pas:

``Legal unit numbers tp the microcode are 4 bit quantities.  This
allows selection of sixteen drives (Units 0 through 15).''

                          *   *   *

From diskio.pas:

``Eventually diskio will be removed from the system and higher level
software will need to be modified to use VolumeSystem directly; in
particular, this will be required to support more than a single hard
disk and single floppy.''

``This is an implementation of the DiskIO interface which uses the new
VolumeSystem module.  It is provided as a compatibility module in
order to support the K1 hardware with little modification of system
software above the level of DiskIO.''

                          *   *   *
```

```
From diskio.pas change log, March 17, 1981:

‘‘Removed partition kind ’node’.’’

[ What was a ’node’ partition? - RDD ]

                          *   *   *

From initialshell.pas in procedure ProcessCommandEntry:

Abstract:
  Processes a line of the profile and makes a command entry.
  A line must have the form:

   string to  adjust   setdefault   screensize   commandname   helpinfo
   execute
              Bool     Bool         integer      string[25]    string

Parameters:
  Entry - a string read from profile file

Side Effects:
  Appends an entry at the front of CmdList

                          *   *   *

From diskio.pas:

‘‘A directory is an ordinary file which contains SegIDs of files
along with their names.  Directories are hash coded by file name
to make lookup fast.  They are often sparse files (ie contain
unallocated blocks between allocated blocks).  The file name is a
SimpleName, since a directory can only contain entries for
files within the partition (and thus device) where the directory
itself is located.’’

DirEntry  = packed record
              InUse    : boolean; {true if this DirEntry is valid}
              Deleted  : boolean; {true if entry deleted byt not expunged}
              Archived : boolean; {true if entry is on backup tape}
              UnUsed   : 0..\#17777;   {reserved for later use}
              ID       : SegID;
              Filename : SimpleName
            end;
```

[Was some sort of backup program developed which took advantage of the ‘‘Archived’’ field? - RDD]

‘‘The fillers in headers which are used as hints to the new head of the free list when a block is allocated in a partition (by AllocDisk) have been represented as unsigned 16 bit numbers denoting disk relative logical block numbers. For disks larger than 64K blocks this is not possible and a partition relative denotation is used instead on all future disks. FillerSemantics is the type of field in the DIB of a disk which tells which interpretarion applies for that disk.’’

* * *

From filedir.pas:

Maximum directory levels = 9.

* * *

From filedir.pas, page 5:

About the algorythm used to look up and delete files:

‘‘THIS ALGORYTHM MAY FAIL to find a valid entry if there have been a lot of deletes and no enters.’’

[...which helps explain why POS sometimes doesn’t acknowledge the existance of some files displayed by a dirctory listing - RDD]

* * *

From fileutils.pas:

One cannot create a firectory named ROOT.DR because this is a reserved name.

* * *

From iodisk.pas change-log of Nov. 40, 1982:

‘‘Module rewritten for support of multiple disks, including all EIO disks: Micropolis, Shugart (and SMD partially). This code will run on a CIO and EIO system. A new control structure, the DCA, was created for managing multiple disks.’’

* * *

From iokeyboard.pas

Constant used to represent shift-control-c was BlamCh.

```
BlamCh = Chr(\#303)
```

* * *

A quote from iors.pas change-log of March 3, 1983 by Chuck Beckett:

‘‘Exported private variables. This will allow demos and other time dependent stuff to access IORS directly. (Caveat hacker! Importing IORS is really a dumb thing to do unless you have absolutely no other choice!’’

* * *

From FileTypes.pas:

‘‘This module exports Types put into the FileType field of File FIBs. The types are stored as integers.’’

The file types are:

```
UnknownFile = 0
SegFile = 1
PasFile = 2
DirFile = 3
ExDirFile = 4
FontFile = 5
RunFile = 6
TextFile = 7     - for non Pascal text files
CursorFile = 8   - for cursor bin files
BinaryFile = 9
BinFile = 10     - for microcode output
MicroFile = 11
ComFile = 12
RelFile = 13
IncludeFile = 14 - included in a Pascal file
SBootFile = 15   - system part of a boot file
MBootFile = 16   - microcode part of a boot file
SwapFile = 17    - used for swapping by compiler or editor, length not set
BadFile = 18     - created by scavenger
```

```
ForFile = 19     - FORTRAN source file
DatFile = 20     - FORTRAN unformatted data file
PsgFile = 21     - FORTRAN pre-seg file
ExtFile = 22     - FORTRAN external definition file
LibFile = 23     - FORTRAN library file
TempFile = 24    - file created by Temper
```

* * *

From id-Others.pas, page 12:

Supported cursor is 56 x 64 with scan line length of 4.

* * *

From io_private.pas, page 6:

SMD stands for Store Module Technology.

* * *

From io_tester.pas, page 54:

Message returned if user enters an illegal command:

‘‘The command isn’t legal, but you get to try.’’

* * *

From screen.pas change log of Feb. 4, 1983:

ICL played a role in the PERQs screen driver:

‘‘ICL’s version of screen.pas to clear up a few problems. This module is identical to the one received from ICL via August Reinig.’’

* * *

volumesystem.pas, page 1, indicates that there seem to have been plans for the PERQs to use external hard disks with removable packs!

* * *

From volumesystem.pas, page 4:

```
‘‘MaxTotalVols is the upperlimit on the numer of file system volumes
that can be mounted at one time.  MinVolID and MaxVolID delimit the
range of non-nil (i.e. actually mounted) file system volumes.
NilVolID denotes a volume different from any possible mounted volume.’’

Const
   MaxTotalVols = 8;
   MaxVolID = MaxTotalVols - 1;
   MinVolID = 0;
   NilVolID = MinVolID - 1;

                          *   *   *

From volumesys.pas, page 17:

‘‘Note that Disk Operations are directed toward 3 different classes of
devices:

     - CIO Shugart
     - Floppy
     - EIO Shugart, Micropolis and SMD’’
```

Index