Site Reports

The Commodore Amiga: A Magic TEX Machine

Tomas Rokicki

The Commodore Amiga makes an impressive TEX machine, able to compete with computers costing several times its price. In this report on the status of AmigaTEX, features will be discussed that might well be profitably implemented on other machines. Let me tantalize you, first, by mentioning that it is possible with this package to go from your document in your favorite editor to the first page of your TEX'ed document in a preview window in about a second of real time and with a single keystroke or menu selection.

But before I discuss some of the more esoteric features, let me tell you what the basic package contains and requires. AmigaTEX comes on eight floppies including TEX 2.9, iniTEX, LATEX, BibTEX, preview, over 1200 previewer fonts, and some font conversion utilities. Printer drivers are available separately for the HP LaserJet Plus, standard PostScript printers, the QMS Kiss and SmartWriter, the NEC P6/P7 series, the Epson LQ series and MX series, and some other less popular dot-matrix printers. Plain TEX will run on a 512K machine; LATEX requires a megabyte of memory. A hard disk is not necessary; an extra two megabytes of memory is cheaper and much more useful. Two floppy drives are highly recommended. Three megabytes of memory gives you the best environment.

The first question that pops up is, how can a hard disk not be necessary if the package requires eight floppies to distribute? Is it really possible to put an entire TEX environment, including TEX, the preview program, a printer driver, the editor, previewer and laser printer fonts, and all of the system software onto two floppies and still have room for TEX source files? Indeed it is. The TEX software less the previewer and laser printer fonts requires less than 400K, including even the plain format file and an editor. The key feature that makes a floppy TEX environment practical is font caching.

Font caching is based on the idea that, of the thousands (literally) of fonts supplied with the previewer and a printer driver, the typical user will need only a few dozens, or maybe a hundred. The idea is to find which fonts the user needs. These fonts should be made easily available. This is easily accomplished. A directory is assigned to hold the commonly used fonts; this directory resides on a single disk that is always in one of the disk drives when the user is using TEX. This directory is initially empty. As the previewer or a printer driver requires a font and cannot find it in this font directory, it queries the user to insert the appropriate distribution disk on which the floppy was supplied. The driver program then copies it into this cache directory, so the next time it is needed, it is there.

This system works quite well in actual experience. The first few times a printer driver is run, the user has to swap some floppies. But after the fonts are in the cache directory, things go smoothly and quickly. And two floppies are now sufficient for a nice working environment.

But floppies are slow. Even fast floppies are slow. The data rate of standard double density 90 mm floppies is 250,000 data bits per second, or about 31,000 bytes per second maximum. Actual transfer rate is typically around 18,000 bytes per second on good days. Thus, to load the TEX executable image of 128,000 bytes takes seven seconds. This is a long enough delay to get annoying after a period of use.

On a single-tasking computer, there would not be much one could do about this. The Amiga comes with a true multitasking operating system, however, so it is a simple matter to run TEX in a loop mode, where after finishing one document, it hangs around and waits for the next rather than exiting.

The plain format file is almost as large as the TEX executable, so the delay in loading it is as long as the delay in loading TEX itself. But, since TEX is hanging around in memory, a copy of the format file might as well stay in memory too. On the Amiga, as TEX loads the format file the first time, it copies the portions of the data structures that will be destroyed as a document is processed into another area of memory from which it can be quickly restored as soon as the document is done. Since some portions of the format file contain data that does not change as a document is processed, such as the string pool, these areas need not be saved or reloaded after the first time.

Now, actually, it has not been mentioned that on a single-tasking computer, both TEX and the format file can be loaded into a RAM disk, from which loading is quick. Nonetheless, the above tricks require less memory and yield faster operation than a RAM disk alone would provide. In addition, while I've expounded these ideas in the light of floppy drive storage, they are also useful when all of the files reside on a hard disk, although their impact is not as great. The difference is that using the above ideas, TEX is ready and preloaded at the instant you decide to use it.

Still, TEX spends most of its time actually processing documents. Even when run off floppies, the fourteen second load time is dwarfed by processing times of several minutes for sixty page documents. TEX is doing a lot of work, so it is doubtful that this processing time can be cut significantly using the current hardware. A 68020 board can always be plugged in, but the TEXbook will still take a good quarter hour or more. So what facilities can be provided that will allow the user to make the best possible use of the time during which TEX is working?

With a multitasking environment, the user can read net news. But more often than not, the user is wondering what the document looks like. With the Amiga, TEX can send each page as it is finished to the previewer. (Of course, the previewer stays around waiting for new jobs just like TEX does.) This way, as TEX is working on page twenty of that forty page report, the user can preview any page up to page twenty, and make changes in his editor buffer as he finds things to change.

It is true that the message-passing executive of the Amiga makes such communication easy to implement; TEX simply sends messages containing the page data as it completes each page, and the previewer actually writes and re-reads the DVI file as necessary. It should be possible to do a similar trick under, for instance, the Unix operating system using sockets.

So, at the moment, Joe User saves his file from the editor, clicks on the TEX window with his mouse and hits carriage return (it's easy to make TEX remember the last file name processed) and, just as soon as that first page is done, it pops up in a previewer window. But, if TEX encounters an error, Joe must find the place where the error happened in his file. In addition, he has no easy way to process just a small portion of his document, say the equation on page twenty-four, to see how his changes look.

At this point, it is not difficult to take care of the problem for Joe. All we need to do is add some small changes to his editor so that it understands a function key or two, and sends the proper messages to TEX. For instance, function key nine might be programmed to take the current cut region, append it to just the top of his file containing macros, saving this file and telling TEX to start processing. Joe uses EMACS, so such hacks are easily made. But Bob uses TxED (by Charlie Heath) and Paul likes vi, and Peter has his own homebrew editor. Source isn't even available for most of these editors, so how do you add such facilities?

Out of the sky appears William Hawes, author of ARexx, an implementation of the REXX language so revered on IBM mainframes. On the Amiga, ARexx is so much more than a script language. It is a general interprocess communication manager, programmable in an interpretive language so simple that anyone can use it. To make any program talk to any other program on the Amiga, all the developer must do is make it talk to ARexx. Then, a set of four-line ARexx macros can be written by either the user or the developer to transfer data back and forth between the applications.

As of this writing, only two editors exist with ARexx ports, so only these two can be used to make an integrated environment with AmigaTEX. But as more and more programs appear with ARexx ports, the versatility of all of them will increase dramatically. For instance, it is now possible to integrate an editor, TEX, the previewer, a terminal communications program and any number of other programs in a single, unified working environment that is remarkable to behold. A similiar situation will soon exist with OS/2 on the IBM PC's.

But enough soapboxing. You have to see it for yourself. If you send me, Tomas Rokicki, a letter, to Box 2081, Stanford, CA 94309, I will send you an Amiga diskette containing a demonstration version of the AmigaTEX package, and pricing and ordering information. For information on ARexx, write Bill Hawes at Box 308, Maynard, MA 01754.

DG Site Report

Bart Childs
Texas A&M University

We are now delivering TEX 2.9. It went in as easy as the previous releases. We are also delivering METAFONT 1.3. (We had been delivering out-of-date 1.0 versions for more than a year.) We now have a stochastic ArpaNet connection and should find it easy to stay current.

IBM VM/CMS Site Report

Dean Guenther
Washington State University

In February I sent the TeX 2.9 VM/CMS distribution tape to Maria Code. This tape includes a few new programs, upgrades, fixes, etc.

SliTeX has been added to the distribution. It has been requested for some time, but has been difficult to get ahold of. Thanks to Barbara Beeton, who sent me a copy of the fonts and macros. The other newcomer to the distribution is DVI3279, a screen preview program for the IBM 3279 or 3179g terminals. It works well on DVI files and METAFONT output run through GFTODVI. Thanks to Georg Bayer at the Technische Universitaet Braunschweig in Germany for this contribution. It is written in WEB, and requires GDDM and FORTRAN. The messages from this program are all in German, so höffentlich ist ihr deutsch gut.

Upgrades on the distribution tape along with TeX 2.9 include LaTeX, thanks again to Barbara. I upgraded DVITYPE to version 2.9, and METAFONT and all of its MF files to version 1.3. GFTOPXL was also modified to output 1K blocks.

Bob Creasy made several upgrades and fixes to the IBM printer support. We worked out a scheme for DVI2LIST and PXLCVT to support larger magnifications, a requirement necessary to support SliTeX. Bob also modified DVI2LIST's absolute horizontal alignment to match DVITYPE. This fixed a problem reported by several, who were unable to get the following to line up:

```
{\obeylines\obeyspaces\tt
I         I         I         I
0123456789012345678901234567890
I         I         I         I
}
```

This now works. Bob also made upgrades to his PFS font building EXECs, including allowing larger magnifications, and support for generating the PXL fonts used by DVI3279.

Fujitsu Announces TeX Port

Fujitsu Limited, a member of the TeX Users Group of Japan, announced that it has completed the porting of TeX V1.0 into their M-series, large-scale mainframe computer. This product, FACOM OSIV TeX V01L10, is now available to users. The command procedure provides support for various printing devices and formats; it allows users to specify execution units (INITeX, DVIwrite, printing), to set printing devices (including display) and printing sizes, and to establish execution environment (batch or TSS). For more information:

Shozo Taguchi
Deputy General Manager
Software Division
Fujitsu Limited
140 Miyamoto, Numazu-shi
Shizuoka-ken 410-03, JAPAN

UnixTeX Site Report

Pierre A. MacKay

The January 1988 upgrade of the UnixTeX distribution is the most important and far-reaching in several years. The changes in TeX itself since the last site report are relatively trivial, and correct bugs that only very advanced users of the program would ever be likely to run into, but almost every other part of the distribution has undergone major changes. TeX has now reached Version 2.9, and the source file has been slightly reformatted with ASCII form-feeds immediately preceding each starred module. This has no particular effect on UnixTeX, except in the change files, where the last starred module before the index:

```
^L@* \[54] System-dependent changes.
```

(the module for system-dependent code) is usually referenced, and will cause an error if the form-feed is not added to the change file.

The most important news associated with the compilation of TeX however, is the successful completion of TeX-to-C, an interpreter which takes the Pascal output from tangle, and converts it into clean C source code, lintable, and so far as we know, completely consistent with the ANSI draft C standard. This interpreter is the result of several years of work by Tom Rokicki and Tim Morgan. Rokicki did the ground work a few years ago, and

Tim Morgan has refined and restructured the whole system so that it now promises to make it possible to eliminate the proliferation of system-specific change files. The code generated is smaller and substantially faster in execution than even the best Pascal compilations, and it makes the generation of a truly immense TEX possible, since C is not subject to the Berkeley Pascal compiler restriction on array sizes. (Berkeley pc restricts array indices to a 16-bit range.) The code can be compiled in two modes, one with standard variables, which seems to work on just about every system tried so far, and one with register variables, which works if your compiler is clever enough, and produces executables that are about 10% faster than the non-register versions. (I note, with some amusement, that the 68010 Sun-2 compiles correctly with register variables under OS 3.2, but the 68020 Sun-3 does not.)

METAFONT is still at version 1.3. There have been some slight changes in the Computer Modern mf files over the past year, and it is probably advisable to compile all fonts again. There are small problems with low resolutions still in CMMI6, CMTT8 and CMTEX8, which produce strange paths when compiled at 118 dpi resolution for the BitGraph and Sun screen previewers. Readers of TEXhax will have followed the recent discussion of mode_specials added to compiled fonts in GF format. The purpose of these is to provide, in the font itself, a record of the settings used when the font was made. A typical group of mode_specials as printed out by the UNIX strings utility is:

```
cmr10
mag:=2.0736;
mode:=RicohFortyEighty;
pixels_per_inch:=300;
blacker:=0.2;
fillin:=-0.2;
o_correction:=0.5;
```

The macros to produce this information are part of the file U_Wash.mf in the directory ./utility-fonts/bases on the UnixTEX distribution tape.

Because we run both write-white and write-black printers at the University of Washington, we have had to think more clearly about the adjustment for write-white devices (discussed by Neenie Billawala in TUGboat Vol. 8, No. 1, pages 29–32). Both the changes to cmbase.mf described there occur in the font_setup macro, and there is really no need to have an entire write-white cmbase_w.mf and a separate preloaded version of METAFONT as we have had for the past year. The ./cmfonts/mf directory now contains the short white_setup.mf file, which is used in place of the write-black font_setup when appropriate through the addition of the line let font_setup=white_setup to all write-white mode_defs. Add the line input white_setup to your local mode_def file. The fonts currently on the tape have all been recompiled with the macros described above.

There has been only one significant change in TEXware since the last distribution. The dvitype program has been changed to reflect a change made in TEX itself. Make sure you have the latest version (2.9) before you try to trip, or you will get some misleading discrepancies in the dvitype output file.

A sub-directory of ./tex82/TeXware contains the CWEB programs, which extend the whole idea of integrated documentation to the C language (which desperately needs it). These programs are not to be confused with TEX-to-C, which starts from the original Pascal-based WEB files. CWEB is the work of Silvio Levy of Princeton.

A few minor bug-fixes have improved the suite of PK utilities in ./mf84/MFware. These, and some C versions of the same programs were provided by Tom Rokicki. A new directory, ./mf84/MFcontrib, includes some PostScript utilities and PXtoGF, which was brought into Unix compatibility by Karl Berry. It was the availability of this program which made it possible to convert the AMS Cyrillic and special symbol fonts into GF format. Tom Rokicki is presently putting the finishing touches on the WEB source for PKtoGF, which will make it possible to send out more fonts in less space without excluding the large number of output drivers which use GF format.

The most important unfinished business in all the above is the extension of the TEX-to-C approach to METAFONT, TEXware and MFware. If all the standard WEB programs could be interpreted into ANSI draft standard C code, as has been done with TEX, it would be possible to eliminate the proliferation of system-dependent change files from the UNIX distribution and to target the large and, so far, inadequately served System V UNIX community. Above all, we need a bootstrap tangle.c.

Up till now, the distribution tape has offered a small range of precompiled binaries of TEX and METAFONT. This makes less and less sense when even VAX no longer means a single architecture and the binaries will soon be dropped. It might be desirable, however, to send out a variety of precompiled tangle executables. I can offer tangle precompiled on a VAX11-750, a VAX8550, a Sun2

(MC68010) and a Sun3 (MC68020). If people will send me precompiled tangle executables for other architectures in btoa format, I will put them on the distribution as I receive them.

Since almost all DVI drivers now use either GF or PK format fonts, the PXL directory has been removed from the distribution. In compensation, a larger range of precompiled GF format files is being sent out, but even these can target only a small number of devices. Lowres and CanonLBP fonts (200 and 240 dpi) are still well represented, but I wonder how many 240 dpi devices are still in use? At 300 dpi, both write-white and write-black fonts are provided (the write-white is tuned for the Ricoh 4080 print engine, and the write-black for the ubiquitous CanonCX). If you have any 300 dpi device at all, one or other of these compilations will serve as a temporary resource, but you will probably want to recompile to get the best out of a print engine that is neither of the above. That unhappy compromise, the 118 dpi font, is also still with us. The AMS Cyrillic and special symbol fonts exist only in write-black versions because they were compiled long ago on a Tops20 machine in old METAFONT-in-SAIL, and I cannot recompile them.

Several of the output drivers have been revised by various contributors to the distribution. Scott Simpson of TRW has completely rewritten the driver for QMS/Talaris, which is now known as quicspool. The basic README file for this is worth reading even if you don't run a QMS machine. The entire ctex system has been reworked, and enhanced by the addition of System V code and routines for a previewer on the ATT 5620 supplied by Lou Salkind of NYU. There is also a new pair of previewers, texsun and texx, running under SunViews and X, respectively, which was contributed by Dirk Grunwald of the University of Illinois. The backbone of the system is the library developed by Chris Torek at the University of Maryland. The ctex library has already spawned a larger number of derived systems than any other, and it seems appropriate to suggest here that drivers written in C might profitably be adjusted to make use of as much common code from ctex as possible. GF and PK interpreter modules are beginning to proliferate, and they all do essentially the same thing. If there is some strong argument of increased efficiency in one of the interpreters outside ctex then surely the techniques could be incorporated into the ctex library.

The LN03 was provided with a new driver by Matt Thomas in September, but his shipment of code got lost in a mail crunch. My apologies for not unwrapping it earlier. In any case it is on the distribution now.

There is no major new development in dvi2ps ready for release as yet. I have increased the array sizes to reflect the fact that GF format permits 256 character fonts, and have changed the meaning of the -d flag so that it can be used to change pixel density. The default remains 300 dpi, but a flag value of -d 600 is available for devices such as the Varityper VT600. The header file tex.ps needs to be completely rewritten to eliminate the dozen or so 300 dpi-isms, and allow for some sort of conditional coding. For now, a 600 dpi version has been added to the distribution under the name tex6.ps. This automatically replaces the default tex.ps when the 600 dpi flag is used.

The foreign-language ./babel directory is at last beginning to grow. In addition to Portuguese there is now a German section, with a German hyphenation file, a Swedish section with a complete package of macros for Swedish-Language LaTeX, and a Semitics section with a first pass at TeX-XeT. In the near future, I hope to add a machine-independent change file for use with TeX-to-C, which will make TeX-XeT much easier to compile.

TeX to C Converter

Tim Morgan
University of California, Irvine

Tomas Rokicki and I have developed a set of programs, makefiles, shell scripts, and a changefile which can automatically convert tex.web into a C program. We are very glad to say that the entire conversion package has been placed in the public domain, and it is being distributed by the UNIXTeX site coordinator, Pierre MacKay, at the University of Washington. It's also available for anonymous ftp from Internet host "ics.uci.edu".

There are still a number of developments in progress as of this writing. Several people are working on adapting the conversion software to work for METAFONT, and this effort is nearing completion. We are currently looking for someone to convert the TeXware and METAFONTware programs to C as well, preferably via an automated process. Once *tangle*, in particular, is available in C, a site with only a C compiler will be able to bring TeX and METAFONT up starting with only sources on the UNIXTEX distribution tape.

The C source which is generated is fairly efficient and portable. It passes through *lint* with no unexpected complaints, and I've tried the code on a number of different UNIX systems. In all cases, and with versions of TEX from 2.2 to 2.9, the resulting executables easily pass the trip test on the 4.2BSD-based machines available to me for testing. The environments in which I've tested the code include a VAX-11/750 running 4.2BSD, SunOS 3.2 on a Sun-3, Sequent Dynix 2.1.1, and Integrated Solutions UNIX version 3.07. The code turned up minor bugs in the VAX, Sun, and Sequent compilers, but workarounds were found in all cases. I was also able to compile the C sources and pass the trip test using the System V compilers supplied by Sun and Sequent. As long as there is sufficient memory available, I believe that this code can be easily ported to any C environment. It has been in production use at the University of California, Irvine, for over six months. Although improvements in the conversion process are still being made, the code which is generated remains almost the same, so I consider it to be highly stable.

Obviously, the most important thing is that TEX is converted automatically into C. This feature makes it easy to track new versions of TEX as they become available. A changefile is used to handle many of the necessary changes, and therefore a working *tangle* is required. The output from *tangle* is mainly PASCAL, although with a C flavor. It is then converted into C and split into multiple modules by a pipeline of other programs. The entire process is automated, so all the user need do is type "`make`".

VAX/VMS Site Report

David Kellerman
Kellerman & Smith

There is a new version of the VAX/VMS distribution available from Kellerman & Smith. TEX is at version 2.9 and changing quickly, METAFONT is at version 1.3. Most of the other software, most of the macro packages, the font sets, and the packaging of the release have also changed since our last release.

The repackaging is perhaps the biggest change to the new distribution. Over the years, our conversations with new users of TEX have made it clear that this enormous and complex package has all the intrigue of a giant puzzle for many users, but has, well, the frustration of a giant puzzle for many others. To help make TEX more accessible to the increasing number of users who hold the latter view, we have rearranged it into pieces that can be installed selectively with the standard VMSINSTAL mechanism. This should be of particular benefit to users who want to get TEX or LATEX going quickly, who need to fit a system into very little space, or who are working in a cluster environment. (You old pros can still install it by hand if you want to.)

The font sets, based on new Computer Modern METAFONT sources, have been expanded to magsteps 0–8 for both the XEROX XP-12 (QMS) and the Canon CX (IMAGEN). They are distributed in PK format and come with a new utility program, XXtoXX, that converts between GF, PK, and PXL formats (all combinations) and converts from one RMS record format to another. The XXtoXX utility is fast, it can convert multiple files in one execution, and we have tested it more thoroughly than some of our earlier changes to font conversion utilities.

The LATEX and SLITEX macros are updated. Also, we now provide all the fonts they require, and the XXtoXX utility can produce SLITEX "invisible" fonts to match any normal font.

The contributed software on the release includes the LN03 driver from Flavio Rose, the PostScript driver and screen previewers from Andrew Trevorrow, and the MWEB software (Modula-2 WEB) from Wayne Sewell.

The software is now available on either 2400′ magtape at 1600 bpi or TK50 cartridge, and costs $200.00 (U.S.) including shipping within the U.S. and Canada. Add $50.00 (U.S.) for air parcel post shipment to other countries.