* * * * * * * * * * * *

Output Devices

* * * * * * * * * * * *

Editor's note: Not every TEX user has a fast, high-resolution output device at his elbow; it is surprising how modest is some of the equipment that has been made to produce usable, if limited, TEX output. The preceding page is reproduced from output on an IDS-460, an impact printer with a resolution of 84 dots per inch, driven by an Apple II. Its most severe present limitation is a lack of fonts, certainly not the ingenuity of its owner, John Sauter, who describes a recent weekend's work on page 34.

* * * * * * * * * * *

DIRECTORY OF OUTPUT DEVICES

Rilla J. Thedford
Mathematical Reviews

To further the development of TEX, TUG is compiling a directory of TEX output devices. We hope this will be a useful tool.

The directory will identify manufacturers for TEX output devices and required software and hardware for TEX interfaces. It will also include information about fonts, device specifications, contact people, and installation sites.

The membership renewal form will have additional questions for *ACTIVE* TEX users. Please complete and return it to the American Mathematical Society, with your renewal fee. Please feel free to include additional information you feel will help in the purchase and/or installation of a particular output device.

* * * * * * * * * * * *

A VARIAN OUTPUT DRIVER IN VAX/VMS FORTRAN

Jim Mooney
September 8, 1981

A summer project at the Morgantown Energy Technology Center (METC), Morgantown, WV, has resulted in experimental installation of the Oregon Software version of VAX/VMS TEX. As part of this project I have written a DVI file converter to drive a Benson-Varian 200 dot/inch matrix plotter. Since METC lacks a Pascal compiler, this program was written in FORTRAN 77.

We acquired Oregon Software's TEX in July through the efforts of Tom Pierce who was then at METC. As described by Barry Smith, this version is an interim one which still uses (a kind of) TFX font files and produces version 0 DVI files. Also supplied was a pair of conversion programs LVSPOOL and LHSPOOL for a Versatec 200 dot/inch plotter. Fortunately this was very similar to the Varian. A set of character files with extension VRT were included containing the raster character descriptions for the Versatec.

The executable module supplied for TEX ran immediately and we were shortly producing DVI files. However, it was disturbing that the program seems to take between one and five minutes to initialize. (We have a VAX 11/780 with typically 30–35 interactive users.) I sometimes despaired of ever seeing that starting asterisk. I suspect that much of this time is spent in routinely loading the "standard" fonts, and since many of these fonts are often not used I hope the decision to always load them can be repealed.

An inconvenience we encountered here was caused by the OS-TEX SYSDEP module which referred to the directory "[TEX]" explicitly. This prevented using any other name; far worse, it made it impossible to run TEX while the default directory was on a different disk than the [TEX] directory. This could easily be fixed by switching to a logical name such as TEX$DIR, which we would have done if we had a Pascal compiler.

There was also a tendency for the program to abort with "Fatal errors" when it should have known better, e.g., when the installation limit for distinct versions of TEXOUT.DVI was reached.

The Pascal driver program LVSPOOL was supplied in a form which can be linked to a separate assembly language module to issue the actual driver calls. Thus we could have substituted appropriate QIO's for the Varian driver and obtained output. However, there were several advantages to rewriting the driver in FORTRAN. It could be locally maintained, and adapted for other output devices. Moreover, some inefficiencies found in LVSPOOL could be eliminated. For these reasons we wrote a new conversion program, based closely on LVSPOOL, in FORTRAN 77.

The Varian conversion program is called DVITOVAR. It currently translates only version 0 DVI files, but conversion to the newly announced version 1 would be straightforward. Currently at METC, the Varian interface and the [TEX] system are on two different VAX mainframes. For this reason, DVITOVAR actually creates a rather large file (1100 blocks per output page) containing raw raster data, which is transmitted to the

other VAX over DECNET and converted to QIO's by the separate program OUTTOVAR. At an installation with everything on the same machine, this headache can be eliminated by inserting the QIO's directly in DVITOVAR in place of OPEN and WRITE statements. (The peculiar structure of the Varian-supplied driver program does not allow raster plot files to be spooled.)

LVSPOOL set aside almost a full megabyte to hold character raster data, far more than needed. FORTRAN does not allow the preferred solution of dynamic allocation, but we reduced the buffer to 200K bytes which is probably still lots too much. DVITOVAR also defers font loading until a font is actually needed; thus many fonts are never loaded although they are defined in the macros and thus appear in the postamble. This is a considerable timesaver, and reduces even further the buffer size needed.

DVITOVAR is rather verbose in announcing the processing phases it is going through. These messages can be removed if desired. The program has not been adapted to an equivalent of LHSPOOL which produces output horizontally on the page, but such a project should present no difficulties.

DVITOVAR was also adapted into a similar program DVITOLP to drive lineprinter class devices (Yes, many users do need such primitive output). To get this to work I had to construct with trepidation, understanding little of the format, a new TFX file to represent line printer fonts. (Font CMTT which *simulates* such a font was not satisfactory.) All widths in this font are set to 7.2 points (ten pitch); there is no kerning or ligatures; wordspace is set to 7.2 points with zero shrink, and several parameters I didn't understand were left alone. But this font seems to serve the purpose as long as all spacing parameters in the text are appropriately restricted.

Anyone interested in obtaining the programs cited above should contact

Jerry Craig
Morgantown Energy Technology Center
B1-330
Collins Ferry Road
Morgantown, WV 26505
304-599-7178

Technical questions can be addressed to me at

Dept. of Statistics and
Computer Science
West Virginia University
Morgantown, WV 26506
304-293-3607

Meanwhile, I await word of a TEX version which may be adapted to run on our PDP-11/34, which has UNIX v6 and the rather strict ISO standard P from Vrije University, Amsterdam.

* * * * * * * * * * * *

DIABOLIC TEX

Timothy Murphy
Trinity College Dublin

Preamble

Before TEX can be run with a given output de 2 modules must be provided: an input mo consisting of a set of font tables; and an ou module, or driver, which will translate the ". file produced by the main TEX program into ins tions for the output device.

Even for a Diablo, writing these modules prove a time-consuming occupation, at leas amateurs of the computing art like ourselves. our only output device was a Diablo—Versatec Varians being as remote from us as Neptune Pluto—we wrote to all those in the TUG mem ship list under the Diablo heading. The resp was disheartening; the few replies we received from groups in much the same position as ourse viz Waiting for Godot.

This brief account of our own efforts may t fore not be out of place. At the very lea may shame some of the TEXperts who have developed Diablo drivers to share their secrets us beginners.

The Diablo as printer

One can envisage 3 very different ways in w the Diablo might be used as an output device.

(1) The output could be run through the Dia or more times, with different daisy-wheels inst on each iteration, e.g. first with roman, then i then symbol, etc. The driver would of course to be designed so that only those characters i appropriate font were printed on each run.

(2) The output might be sent through the D just once, with a single daisy-wheel, those chara not appearing on this wheel being "made up superposition of existing characters (moved u the right, etc, so as to give the required facsim

(3) All characters and symbols might be mad out of dots, using the graphics mode on the Di In effect this would make the Diablo analogous digitalised type-setter, albeit one of very low re tion.

Our calculations seemed to show that the solution would be impracticably time-consum each page taking more than half-an-hour to put We hope to implement the first solution sho