

Basic Information

Introduction

LASI (LAYout System for Individuals(ists))
Version 6.0

LASI has existed in an evolving MS-DOS version since the mid-80's, when PCs started to get useful graphics and sufficient speed.

LASI 6.0 is a new version of LASI that has been completely rewritten to run under Windows 95, 98 and NT 4.0. It is a true 32-bit system written using Win32 functions.

LASI for Windows is expected to run on the upcoming Windows versions "2000" (NT 5.0) and the proposed "Millennium" operating system. Also, since LASI 6.0 is written mostly in C not C++, it should be transportable to other operating systems such as the popular LINUX.

As a warning however, because software and hardware for DOS has pretty much disappeared, expect LASI for DOS to be unsupported in the near future.

Being an entirely new set of programs, also expect more refinement and frequent updates.

LASI 6.0 has been contrived to look and work like the DOS version except that it makes use of Windows facilities that can be handy but confusingly different. For example, LASI 6.0 essentially runs in "cell mode" without going through "system mode". System mode is still there, but it is a popup dialog. The main changes are detailed in "What's New" below.

Actually, LASI 6.0 is full of subtle details that are different from previous versions.

As was true with the DOS version, drawing program commands operate globally, so that editing procedures work on large or small parts of a drawing in a single operation. Strange as it may seem for a Windows program, there are no hierarchical command structures with pull down menus. All commands are essentially random access and may be easily changed or aborted. When a specific input is needed, a dialog pops up.

Most of the original LASI commands are still there, with similar menu buttons at the side of the display. Commands that change the display or non-drawing commands are now at the top of the screen as conventional Windows menu items or toolbar buttons. LASI 6.0 is however enough like previous versions that if you have been using DOS LASI, Windows LASI will have a "gradual learning curve".

LASI was also designed as a "general purpose" drawing system for IC development. Besides layouts, it will draw schematics quite well, and utility programs such as the Spice compiler LASICKT are written to also work with schematics. LASI however is not designed for mechanical or architectural drawing, and does not compete with AutoCAD or similar drawing programs.

At the moment, LASI 6.0 is made up of the following items:

- A main drawing program WLASI.EXE.
- A help and information program WLHI.EXE, which provides online assistance for the main drawing program, the utility programs and operation in general. The new version of this program will also print a basic manual from a help (LHI) file.
- Converter programs WTLC2GDS.EXE and WGDS2TLC.EXE, which convert LASI drawing files to and from Calma Stream Format (GDS or GDSII).

- Converter programs WTLC2CIF.EXE and WCIF2TLC.EXE, which convert LASI drawing files to and from Caltech Intermediate Format (CIF).

- Converter programs WTLC2DXF.EXE and WDXF2TLC.EXE, which translate to and from AutoCAD (DXF) drawing files.

- A design rule check program WLASIDRC.EXE, which uses 2-dimensional bitmap logical operations. The operations are user programmable and therefore not specific to a particular process or technology.

- A SPICE circuit file compiler WLASICKT.EXE, which works from LASI drawings. Besides schematic capture, it also makes SPICE circuit files directly from a layout, allowing more realistic IC simulation.

- A layer logical operation program LASILL.EXE that can be used to make reversals and logical combinations of layers as new cells both for layout and mask making.

- A three odd programs WLASIUUE.EXE, WMAKETFF.EXE and WFLATTLC.EXE that translate to UU code, make text and flatten drawings that may occasionally be used.

Note that programs that are the equivalent of previous DOS programs have a "W" added to their name, to eliminate name conflicts.

As with previous versions of LASI, Windows LASI is intended to be propagated by software alone. Because programs are constantly being updated with improvements and new features. A printed manual would soon be outdated and would add an expense that would require charging everyone for LASI. There is therefore no printed manual.

If you want a printed manual however, the help reader program WLHI.EXE will now print a basic manual.

LASI 6.0 is intended as a good basic CAD system that can be used to teach IC design techniques. Since it will now run under Windows NT, it can be used on systems of networked PCs that many schools are using. LASI teaches the basics, and anyone learning on LASI can go on to more elaborate systems. If a more elaborate system is used, LASI can still be a supplemental system that lets time-consuming tasks be done on a PC and then be transferred to the larger system.

LASI is distributed as shareware, free for non-profit applications. It is now mainly distributed from our web site as a complement to our text book. Anyone may download it, and LASI should be valuable to students, occasional users, or anyone who doesn't have the time or funding for more elaborate systems.

What's New in 6.0

6.00 beta
6.01

At first glance, just about everything, but not really so much that the basics need to be relearned.

- LASI 6.0 now runs under any 32-bit Microsoft Windows operating system.

- LASI is intended to run on VGA (640x480), SVGA(800x600) and XGA (1024x768) displays. Although not well tested, it should also work in 1280x1024 and 1600x1200 screen modes. The display should be set for 256 or more colors. Although 16 colors will work, it is not recommended because it will limit certain intermediate colors.

- Since it runs with a linear memory model (32-bit), there are essentially no memory limitations on the number of objects in cells, as was necessary for the DOS version (16-bit limits). The number of objects in cells (boxes and paths, etc.), are presently set to about twice as many as the DOS version, but can now be changed in the future if necessary.
- Cellnames can now be up to 32 characters long. However if you expect to transfer back to DOS LASI use DOS filename rules.
- The commands that change the display (DRAW, FIT, etc.) are now main menu commands with an associated toolbar.
- A new menu command "CMD" has been added. This executes a command literally from a list of previous commands. This provides a literal command input facility that could eventually be used as a macro command interpreter. Read "Cmd" under Command Information.
- The new CMD command also replaces the old PKE keyboard coordinate entry used in the DOS version. The effect is the same, you just type in the name of a command and then the x,y values.
- The function key (F-key) feature of DOS LASI which allowed commands to be assigned to keyboard function keys was eliminated in the earlier 6.0 beta releases, but has been reinstated due to popular demand. Read the Form File help topic.
- The PRINT command (formerly HCPY) no longer calls a printer specific screen copier and printer driver. PRINT now works through the printer device driver, or using the display driver and then the printer driver. Hardcopies may now be scaled, and printed at high resolution in B&W or color. You are still however limited to the capabilities of your printer driver, that is, you may not get as full a range of colors as the display. For most uses, however, PRINT replaces the old PRNPLOT.EXE program.
- There are five new intermediate display colors defined, descriptively called: lime, turquoise, azure, purple and violet.
- From the nature of Windows, LASI can now be opened on more than one cell at a time in a drawing. More precisely, multiple instances of WLASI.EXE can be run at once in the same drawing directory. This works OK, but can be confusing. You may or may not want to use this feature.

The program icons previously provided in the ICONS directory have been eliminated in favor of individual icons built into each program.

- LASI 6.0 uses different internal files than DOS LASI 5.x. These have extensions .BP6 and .CL6. The .DBD files are also different, CELLS5.DBD is now CELLS6.DBD, and CONSTS5.DBD is now CONSTS6.DBD. The other files FORM.DBD and TFF.DBD are the same.

How do you get an old drawing into LASI 6.0 from an old DOS version? This is easy. Use TLC files. Just run LASI 6.0, click the "Sys"(tem Mode) button, and click the "TLCin" button. This will create new version 6 internal files. TLC files are understood by new and old LASI versions.

One useful trick that you can do with LASI 5.x and LASI 6.0 is to run them both in the same drawing directory. You simply update one version with TLC files from the other version.

These are some major changes, you will also find some commands work a little differently, but do about the same.

What's Newer

These are the latest changes since the initial 6.0 release.

6.02

- LASI now includes a preliminary version a program called LASILL that does logical operations on layers in a cell and then generates a TLC file that contains the logical result.
- The UNDO has been extended to more commands. Originally the undo was limited because it required storage of the whole drawing which would slow down user interaction. Since disk operations have become quite fast under Windows 95, etc. this is not a problem any more and one level of backup has been added to several commands that would be difficult to restore.
- Whenever a cell is loaded or stored, the action is flashed to the title bar. This lets you see when your drawing is moved between memory and the hard drive.
- There are new utility buttons in the System Mode dialog. These now call the smaller less used utility programs that are now fully graphical Windows programs.
- The TLC2GDS program now gives you a choice to abort or fix certain GDS protocol errors, such as open polygons.
- Certain cell selection dialogs now can pop up a list of cells.
- The number of boxes, paths and vertices in a cell can be checked by pressing Ctrl-a.
- LASI 6.0 now contains the elaborate screen clamp that the DOS version used. There should now be no drawing anomalies or strange fill patterns when you zoom in on a small area. Most Windows graphic drivers still seem to be 16-bit and are not to be able to handle the 32-bit graphics used by LASI, although Microsoft defines that they should.
- The SHOW command in Cell and System Mode now operates recursively letting you view the contents of the cells that make up a cell.
- The ATTR popup has been changed to make it more intuitive.
- There is now a text font made from block characters (TFF3.DBD). This is intended for putting permanent text characters on masks.
- The "LASIDRIVE" environment variable is no longer used. When LASI is run from a shortcut, it takes the drive from the command line arguments passed to LASI by Windows. In turn, LASI passes this drive to utility programs and help files.
- Likewise, the "PATH" is no longer used to find LASI programs. If the drive of the \LASI6 directory is established, all programs can be found.
- There is now a long needed CLONE command in the system menu. This lets you easily make new drawings from the current one.
- The HCPY command has been replaced by a PRINT menu item and toolbar button, which do the same thing. HCPY was a remnant of DOS LASI that called a separate screen dump program. The new PRINT command uses Windows facilities to make true scaled plots.
- A FONT command has been added to the side menus that lets you change the drawing text font from a selection of font files.
- The FLATTLC program has been completely revised to work from internal files and not TLC files. It still produces TLC files that can be translated into other formats, or the TLC files can be converted into internal files and passed to LASICKT to make a more thorough Spice analysis.
- When any program creates a new file, The file name is adjusted to conform to the case rules that Windows uses, i.e. DOS names are

lowercase and capitalized, long names are case preserved. All file names from LASI are lowercase and capitalized.

- The "Fit" button has been moved from the main menu to the Setup dialog in the LASIDRC and LASILL programs. This is usually more convenient.
- LASIDRC now has a Lambda parameter that lets you rescale the dimensions in the DRC check file.
- Several utility programs have been revised, TLCIN and TLCOUT in particular. The objective is to "simplify" operations as much as possible without reducing versatility.
- There is now a new top menu item SAVE which lets you update the current cell on the hard drive when working with utility programs.
- Text can now be 64 characters long.
- There is now a fill and dashes patterns popup in the ATTR command.

6.03

- LASI now has an Edit in Place feature. This was relatively easy to add due to LASI's architecture. Read General Information/Edit in Place for more details.
- Period "." was accidentally allowed as a valid character in a cellname. It is now illegal.
- The some of the polygon editing commands have been improved over the DOS versions to make them more intuitive. Mainly:
 - The JOIN command no longer needs to have the first and last vertices of two paths active to join them. JOIN can now reverse a path's vertex sequence if necessary. Only two "beginning" or "end" vertices need be active.
- The Auto Backup Timer has been eliminated. This caused problems with some screen savers and was pretty much unnecessary. All you need to do is click "Save" when you want to preserve your work.
- The "misc=autotlcout" switch in the FORM.DBD file has been eliminated. There is now a check box in the SET command that does the same thing.
- The LASICKT program now has a "Parameter Alias File" which lets you greatly simplify the task of putting Spice parameters on a drawing. Read LASICKT help for more details.
- The MAKE command has be rewritten to interact with the system COPY command so that you can now take objects directly from a cell and add them to any existing cell or make a new cell. This formerly required that you use a "clipboard" procedure.
- The SETUP program will now create a Start Button "LASI 6.0" Program item with sub items for all drawings it installs and for the LASI main help.
- The LASILL program has been improved to make less box objects and to check for undersized boxes. LASILL now also has a more aggressive vertex reducer.
- When LASI starts, it now displays a CELL command popup. This is to indicate that you must first create or load a cell before you can work on it. Click Cancel or press Esc to go to the old startup screen.

6.04

- LASI has been given command button context help, like the old MS-DOS version had. To use it HOLD DOWN the F1 key and click a button. The help information for the button should popup.

- The FORM.DBD parameter "misc=nocellpopup" has been added. This turns off the automatic CELL popup dialog when you start LASI.

- The VIEW,OPEN,LAYR,CLYR and ATTR popups now have high and low pages so that the layer table can now show all 64 layers. Click the ">>" or "<<" button to flip between them.

- The WTLGIN.EXE and WTLGOUT.EXE programs have been eliminated. Instead TLCin and TLCout conversions have been integrated into the main LASI drawing program. Conversions are still done using the buttons on the system mode menu. The effect is the same, except that there are no free-standing TLC conversion programs. Integrating the conversions results in less memory being used and faster execution, since there is no new program instance created.

- The LASIUUE program now contains a "Files" popup window to make selecting a file to convert easier.

- The tutorial help file has been moved into the main help file so that it doesn't need a special setup in the tutorial layout.

- The cell pooling feature, that is, the ATTACH and IMPORT commands have been made easier to use because they now list only cells that fit the operation.

- The LASIDRC program now runs 5-10 times faster. This was done by increasing the bitmaps to a "wide screen" 640x1024 size, and also by reducing the internal bitmaps from 2 to 1 bit per resolution unit. This works with good accuracy for boxes and orthogonal polygons, but increases the roundoff inaccuracies for paths and angled polygon shapes. To improve accuracy on more critical checks just double the resolution.

- The PRINT command now can copy the screen to the Windows clipboard. The clipboard can then be used to insert pictures from LASI into documents, or the pictures can stored as bitmap images.

- The LASIDRC program now has a memory buffered map viewer. Once drawn the bitmap can be moved or expanded with little delay.

- The LASI drawing program CMD command popup now has a Paste button that lets you paste command from the Windows clipboard.

- A new toggle button has been added to the lower right of the screen the lets you show all the vertices if a path or poly.

- When starting in a empty drawing directory, LASI now creates all default DBD files if it can without requiring a restart.

- The Setup program no longer uses functions from Microsoft's Internet Explorer because not everyone has installed the Explorer, particularly those running Windows NT 4.0.

- LASI now comes in a single self-extracting Zip file LASI60X.EXE.

6.05

- LASICKT has been considerably revised so that it now can trace connections through several intrconnecting layers. This greatly reduces the amount of text needed on a layout, and makes the labeling procedure much more obvious. Your old drawings will work with this new program but setup will be slightly different.

- A bug in the Edit in Place that reloaded the previous version of a cell after a draw has been fixed.

- When printing, crosshatches now slope in the same direction as the screen.

- The LASIDRC program now gives a time estimate to completion.
- The LASIDRC program now has two new "gauge" operators GAGR and GAGRS. They do a quick measurement of spacings in the horizontal and vertical directions on a layout. They are almost interchangeable with the expand operators EXPR and EXPRS but are much faster. See the LASIDRC help for more detail.
- The Distance parameter in the LASIDRC check file has been slightly redefined. You need to modify your check files for this. See also the LASIDRC help.
- LASI now has a preliminary "macro" command ability. Read the CMD help topic.
- The limits of objects in a cell have been considerably increased (Press Ctrl-a). This is an anticipation of LASI becoming an efficient system when run on new faster PCs with lots of RAM. Because LASI allocates incrementally by blocks, this does not make the memory usage much larger when used on small ICs.
- LASICKT can now smash certain cells and add the components to the objects being connected in the Spice circuit file.
- LASICKT can now do a much simpler Node List Compare (LVS) that no longer needs nodes to be labeled exactly the same on layout and schematic.
- TLC2CIF and TLC2GDS have been modified so that the Poly Intersect check Modulus parameter has been replaced by a Resolution parameter in actual physical units. This makes setup more intuitive.
- TLC2GDS now reads its scale from the TLC file. Changing scale or physical units from the TLC values is no longer allowed. This caused confusion in the past and could possibly make GDS files that were the wrong size.
- The PKE (parallel keyboard entry) feature that DOS LASI had is back to due to popular demand. After looking at the code for a while, it turned out that the way commands are structured allowed coordinate inputs to be typed in directly using a little Windows trickery.

And many other little changes and bug fixes - large and small.

Problem Reports

This is a listing of problems that have been reported and their possible fix. If the fix doesn't cure your problem try to report it along with any additional useful information.

- A number of people have reported that the right row of menu buttons is cut off. This appears to be due to them using "Large Fonts" in their display. The LASI display is designed to use "Small Fonts" so that more buttons can be put on the page.
To correct this bring up your display setup either under the Control Panel or right click the Desktop. Under Settings choose Small Fonts and reboot as required.
- Drawing and fill anomalies have been reported when a drawing is zoomed in to a small area. This was caused by graphic drivers still being 16-bit. This should have been fixed in this release.
- When the Setup.exe program was run usually under windows NT, Windows reported that it could not find "Shlwapi.dll". This is an Internet Explorer add-on. Use of Explorer functions has been removed from all LASI programs.

Installing LASI

The following information outlines the installation procedures for installing LASI on any PC. If you are reading this, you probably have done most of the installation already. The information is given here so that you can verify that installation has been done correctly.

LASI will be ready to run with just the basic installation. When you are more familiar with LASI's workings you can do the advanced installation.

Installation from a Zip File

If you downloaded from the Internet, LASI should be in a single self-extracting Zip file LASI60X.EXE. Copy this file into a folder optionally named "C:\LASIZIPS" and run it.

The files LASI604.ZIP, SETUP.EXE and a README.TXT should be extracted. The README contains current information and an installation procedure similar as you are reading here.

Run the SETUP.EXE program from the Windows Explorer or from the Start\Run button.

LASI should always be installed using the SETUP program.

If however for some reason SETUP is defective, to install manually (what SETUP really does), do the following:

1. Create a folder named "\LASI6" on your hard drive under the root directory.
2. Copy the Zip files into the \LASI6 directory. Using PKUNZIP or similar program, uncompress the files.
3. Create the subdirectories "\LASI6\TUTOR" and "\LASI6\CHARS" in the \LASI6 directory.
4. Unzip the files TUTOR.ZIP and CHARS.ZIP in their respective subdirectories. Uncompressing the CHARS.ZIP file is optional because there are about 250 small files in the Zip file.

!!! DO NOT UNCOMPRESS THESE FILES IN THE \LASI6 DIRECTORY !!!

NOTE: The main files are listed in the System Contents help topic. If for some reason you obtain LASI as individual files, you should check to see if you have a complete set of files. All files should have the SAME DATE AND TIME to insure compatibility.

Shortcuts to Drawings

There are several ways to run LASI under Windows. The easiest is to create a shortcut icon for each drawing on the Windows Desktop as follows:

1. Use the Windows Explorer to create a new folder or "drawing directory" with some appropriate name in some appropriate location on your hard drive. (If you have LASI drawings already, the CLONE command can be used for this.)
2. If you are running Windows 98 or NT 5.0, or you have enhanced your Windows 95 or NT 4.0 with the Active Desktop from the Internet Explorer, and if SETUP has created Start items, simply drag a copy of a drawing menu item from the Start menu to the Desktop (or Taskbar if you like).

If not:

Click the RIGHT mouse button on the Start button and then click Explore. Click open "Programs", then "LASI 6.0", and then drag a copy of a shortcut to the Desktop by holding down the LEFT mouse button on an icon, while also holding down the Ctrl key.

or,

Create a shortcut on the Windows Desktop by clicking the RIGHT mouse button on the Desktop and selecting New/Shortcut. Then add the "Target" line "drive:\lasi6\wlasi.exe". Alternately, you can simply drag the WLASI.EXE icon from the \LASI6 directory with the Windows Explorer to the Desktop.

3. Select Properties of the new shortcut and set the "Start in" directory to the drawing directory created in step 1.

An alternate way to arrange you drawings (if you like to use the Start button hierarchical menu) is to Open the Start folder sequence as in step 2 above and similarly copy a shortcut, but into the same folder. This just adds items to the Start menu.

Note that the SETUP program if used creates Start button items of any drawings that it has installed.

HINT: If you have a large number of drawings or have different groups of projects, first make a folder on the Desktop for a group and then put the shortcuts in that folder. Doing this will reduce the clutter on your desktop.

Starting a Drawing

To setup a drawing, you should follow these steps:

1. Create a "drawing directory" and shortcut as described above.

The drawing directory should be named for the drawing to be made. Each different drawing is kept on the hard drive by isolating it in its own directory. The drawing directory will contain certain DBD files (drawing basic data files) and .BP6 and .CL6 files (internal cell files) for that particular drawing.

!!! NEVER MAKE DRAWINGS IN THE \LASI6 DIRECTORY !!!

2. Run LASI from the shortcut just created.

When run, LASI will first looks for the FORM.DBD, CONSTS6.DBD, CELLS6.DBD files and a font file in the drawing directory. A default version of these files will be written if any are not found. Read General Information/File Types and Form File for more on these files.

3. Start a drawing by creating or loading cells.

The CELL command popup will appear so that you can create a new cell. "none" will also be listed as the current cellname on the title bar.

You will be asked for a cellname. If the cell doesn't exist, which should be the case if a new drawing, then a rank will be asked. If you do not want to create a cell, click Cancel or press Esc. You can then use other commands to load cells already built.

You must always have a cellname on the title bar. If you cancel out of CELL, you must either load cells from some source (using the TLCIN command) or you must create a cell using CELL again.

!!! IF YOU DON'T CREATE A CELL, ANY WORK WILL NOT BE STORED !!!

In general, start by drawing the lowest rank cells that will be used as building blocks for higher ranking more complex cells, and eventually the overall layout. Also, repetitive structures should be made as individual cells and used to build other cells. However, too few boxes or paths can be a waste of a good cell. With experience you will develop a feel of how to partition your cells.

HINTS:

1. If you have a similar drawing in another drawing directory and you have already defined the drawing parameters, you can use the CLONE command in the system menu to make a new drawing. CLONE copies the FORM.DBD and CONSTS6.DBD files into the new folder. This will save you the trouble of defining layers, colors, fills, etc. all over again.
2. When you first begin a drawing you should check the scale using the SCALE command (click SYS first) and decide if it is what you want. The scale really only relates the physical units in which you do a drawing to the basic units in which the drawing is actually stored. If you do not use the appropriate scale, you may create cells that have too much resolution, or worse, too little resolution. As a rule, set the scale to a tenth of the smallest physical dimension you will use.

Naming Convention

Drawing directories and the cells that are used in LASI drawings all have names.

Although Windows will permit most characters in names, it is a good idea to put some restrictions on the length of the names and the characters permitted, because long names are more difficult to display and certain characters may be used for special purposes and therefore create conflicts.

Under Windows LASI, directory (folder) names and cellnames must be no more than 32 characters long and may NOT contain the characters:

" * + \ ' ' , . / : ; < = > ? \ | and SPACE.

Especially, names should not include SPACE. Use underscore "_" in its place.

If for some reason you might want to transfer LASI data back to MS-DOS, use only standard DOS eight character names.

Converting Versions

As stated in What's New, the internal drawing files of version 5 (BP5 and CL5) are incompatible with versions 6, mainly because of the 16-bit (short) header records stored in them. These files have been renamed with the extensions .BP6 and .CL6, so that there will be no accidental conflicts that can destroy a drawing.

If you have any old version 5 cells use version 5 TLCOUT.EXE to convert the old cells to TLC format. The TLCIN command of version 6 can read version 5 TLC files and convert them to version 6 internal files.

The CELLS5.DBD file of version 5 and the CELLS6.DBD of version 6 are compatible. However the cell collection listed in them may be different and the cellnames may be longer than 8 characters in version 6.

The CONSTS5.DBD file of version 5 is TOTALLY incompatible with version 6, and should NOT be copied from old version 5 drawings. When you run LASI 6.0, a new file named CONSTS6.DBD will be written.

The TFF.DBD and FORM.DBD files are compatible. The FORM.DBD file for version 6.0 has actually less variables. These are better explained under General Information/Form File.

Terms of Distribution

The LASI system files can be freely copied and distributed by anyone. LASI can be downloaded from sites on the Internet or can be distributed by floppy disk. Copyright notices on any programs,

whether part of the LASI system files or not, must not be changed without permission. (Press Ctrl-c for copyright.)

The installation program SETUP.EXE uses portions of Info-Zip's public domain source code to decompress files. The LASI60X.EXE file also uses Info-Zip's self-extractor. If you are not able to find Info-Zip on the Internet contact the author.

Internet Distribution

Any university or technical school that maintains an Internet site for CAD or IC development software can distribute LASI.

The Zip file LASI60X.EXE should be distributed in its original form.

If LASI is to be placed on an Internet site, the author should be contacted for the latest Zip version of the system files. Also, the author should be informed of the address and to whom updates can be sent. LASI has been found all over the world, mostly in duplicates or old versions.

Distribution Rules

- Files in a system distribution should NOT be MODIFIED. The files are usually configured to work with each other in that particular release.

- Files should NOT be distributed INDIVIDUALLY. Most programs are intended to work with other files in the same distribution. The help files are a good example of this.

- If extracted and recompressed, which is highly discouraged, the DATES AND TIMES OF THE FILES MUST NOT BE CHANGED.

This is the way

that different (sub)revisions are kept orderly. Notice that the time stamp on a file is actually its release number.

- Any custom files to be distributed with LASI MUST be included on a floppy disk or network SEPERATELY. DO NOT put any additional files into the set of LASI system files.

Finally, a person distributing LASI is allowed to charge a fee that will cover their distribution costs (labor, floppy, packing, postage, etc.). Any fees are ultimately left to the spirit of LASI as being free for personal and educational use.

Terms of Registration

The LASI system has been written for the author's own professional use in developing integrated circuits, and therefore was not intended as profit-making software.

For this reason, LASI is NOT SHAREWARE in the usual sense; that is, everybody is expected to pay if they use the software. LASI is FREE for NON-PROFIT applications by individuals or organizations. There are no copying or installation protections in LASI. It does have the author's copyright notice in it, so that no one can sell it for their own profit.

If LASI is used to OBVIOUSLY MAKE A PROFIT, then a registration fee

PER COMPUTER on which it is installed is requested. The registration fee is \$99 for each installation. This registration is for a period of ONE YEAR and should be made payable to the author. For several installations use the "4Profit" button; for a site license contact the author. Any registration fees should be paid by check drawn on a U.S. bank or by international postal money order.

When LASI is registered, free updates will be provided to the registrant during the registration period, providing that the registrant notifies the author specifically that updates are desired. Otherwise, all registrants will be kept notified of the Internet site where LASI can be downloaded.

The registration fee is not just to ease the conscience of someone making a profit and using LASI for free. It helps to pay for improving LASI with new hardware and software and sending out updates. The fee is not expected to be a compensation for the endless hours of coding and debugging that went into the original DOS version of LASI, and will probably never compensate for the effort that went into writing LASI 6.0 for Windows.

Terms of Liability

The LASI system is constantly being revised and is distributed on an "AS IS" basis. Since the programs are fairly complicated, software bugs may be found, often when someone does something that only rarely would be done. IF YOU THINK YOU HAVE FOUND A REAL BUG, REPORT IT TO THE AUTHOR. The bug hopefully will be fixed, and YOU WILL RECEIVE A CORRECTED VERSION AT NO CHARGE.

Neither the author nor any person distributing the LASI system assumes any responsibility for whomever uses the programs or for how the programs are used. Also, neither the author nor any person distributing the software assumes any liability for any losses monetary or otherwise incurred related to the use of the programs.

Getting Updates

Unlike commercial software which is usually frozen, bugs and all, LASI gets updated often. Because LASI is used in such varied applications, there are many changes and corrections. LASI is updated approximately monthly, or sooner if the situation requires it. If you have a version more than a couple of months old, it is almost certainly out of date.

LASI is distributed principally over the Internet. At the present, the main site is at the University of Idaho's Microelectronics Research Center. This site is kept current as possible by the author.

Since it may be freely copied and distributed, LASI may also be available on the Internet at other sites, mainly at technical universities who maintain a repository for CAD or IC design software. These sites change frequently and the author has little control over these sites. Therefore, the version of LASI at these sites may not be totally up to date.

If no other source is available, anyone using LASI can get an update at no cost directly from the author. The author maintains a database of know users, but it is impossible to send updates to everyone unless requested.

Anyone who got LASI from other sources should be kept updated by those who gave them the software. Updating secondary users is a responsibility that should be taken seriously.

As a general rule, to be sure that you have the most recent software, contact the author directly, by fax or phone.

Final Comments

With the proper software, personal computers have easily become adequate for small to medium sized IC layout applications. LASI can give everyone a personal computer based work station. This is particularly true for students learning IC technology. In fact, it was (and still is) the intention of the author to develop LASI as a complete educational software package for use in IC design courses.

Although it may be used that purpose, LASI was not really intended for doing routine IC or ASIC design. Instead, LASI was developed as a "fundamental" drawing system, which can generate arbitrary shapes that might be needed in any kind of application, new or old.

LASI was also intended for people who think in a particular spatial

way, as anyone working with it will soon realize. Using LASI develops spatial thinking.

LASI was really intended to put some fun and relaxation into doing layout tasks by allowing a person to work when and where they want, and therefore, it is believed, do better more creative work.

Unlike commercial software, that may be designed by closed committee, LASI receives feedback from its users. Since LASI is rather democratic (within limits), people are encouraged to contribute ideas of their own, or even better, to write programs for themselves that expand LASI's applications. Contact the author by e-mail, fax, phone or mail with your ideas and suggestions.

Author of LASI:

Dr. David E. Boyce
68 Bergdorf Rd.
Parish, NY 13131, USA

Fax/Phone: 315-625-7291
E-mail: deboyce1@aol.com or lasicad@aol.com

General Information

About LASI

For anyone interested in technical details, here is some information on how LASI 6.0 works and how it is written:

LASI 6.0 is written using Microsoft's Visual C/C++ 6.0. It is a Windows application written using the Win32 function library. It is actually a C not C++ program. It is written at the lowest level that Windows permits so that the author can control the interactions of the program with the user. It also allows LASI to be more easily ported to other operating systems.

The drawing program runs in a Windows message loop that receives messages from Windows itself. As LASI runs, it receives mouse movement and button click messages from Windows. If the mouse is clicked on the menus or tool buttons, LASI chooses or executes a new command. If the mouse is on the drawing area, it receives coordinate information.

LASI always has what is called a RESIDENT command. Such a command is always looking for coordinate (x,y) inputs. A resident command requests coordinates until it has enough to execute the command.

Each command in LASI self-driven, that is, it contains the routines to recall itself until it is satisfied and can execute. The commands of LASI therefore are not executing haphazardly from different mouse clicks, but are controlled by the command itself. This allows better program organization and literal command calling.

For example, if Esc is pushed or a new menu button is clicked, the command goes through an abort sequence which starts a new command or reinitializes the resident command. Each command knows how to initialize, abort and clean up after itself.

Commands that change the display (WINDOW commands) are slightly different from commands that change the drawing, executing a window command involves doing "push" and "pop" of variables used by the current resident command. This allows LASI to do display window resize and movement within the execution of the other command. Because the resident command has not has its requirements satisfied, it can pick up where it left off after a window change.

All utility programs are also written in C and based on the Win32 program paradigm and function library.

Programs using internal files (BP6, CL6) use a common object memory manager. LASI does not allocate full memory for all the possible

objects (boxes, etc.) in a cell; the manager allocates only what it needs in 8K blocks. LASI leaves RAM available for other executing programs and Windows system functions (print buffers for example).

Graphics are done through the Windows Graphic User Interface (GUI). LASI 6.0 does not require any special graphic drivers (DirectX for example). Speed is not limited by graphics but by data processing.

The programs all operate in a flat 32-bit memory model, so that there is no actual memory limitation, except for the amount of RAM in your PC. Programs all use a coprocessor if available.

Attached Cells

A cell that is kept in a common cell pool that is shared among several drawings is called an ATTACHED cell. The attached cell will be draw normally, but will follow certain rules and will have some special properties:

- Attached cells are protected and may NOT be changed using cell mode.
- Attached cells must have the same physical scaling as the drawing.
- Attached cells must fit within the memory available for allocation by the memory manager for boxes, paths and vertices.
- Attached cells may NOT be renamed with the RENAME command.
- Attached cells are searched for in the pool directory only, while normal cells are searched for in the drawing directory only. If a cell file (BP6) is not found, a "null" cell is drawn. LASI draws a null cell as a dotted outline. A null cell is different from a cell that has nothing in it, in that there is no BP6 file for a null cell.
- Attached cells may be copied from the cell pool and added to the drawing directory in two ways:

ù You can use the system mode IMPORT command and give the imported cell the SAME NAME. This copies the cell's BP6 file to the drawing directory and turns the attached cell into a normal cell.

ù You can use the IMPORT command and give the copied cell a NEW NAME. This is preferred if you are going to modify the cell and use it in a few places.

- Attached cells may be smashed using the SMSH command. This effectively adds an attached cell's boxes, paths and text to the current cell.
- Attached cells may be used in the MAKE command. They will be added to the new cell as would any other cell, but they will still be attached cells.
- The commands OUTL, FULL, INFO and SHOW work on attached cells.
- Attached cells when will have their rank shown as "1+", indicating that they have rank 1 and are attached "+".

NOTE: Cells with rank greater than 1 should not be attached, because only the boxes, poly and paths will be copied and the cell will be forced to rank 1. LASI uses pointers to cells in its cell placement data, and these pointers would be different for different drawings. The TLCIN command must be used to import any cells containing other cells. Usually cells containing other cells are drawing specific and would not be shared with other drawings anyway.

Read the General Information/Pooled Cells topic.

Backups

LASI is designed with protection against drawing loss in mind. It will write your latest drawing files to your hard drive at certain times so that if the computer is turned off or fails, or you simply make a mistake or change your mind, your drawing will be mostly or totally recoverable.

These protections are as follows:

- When you change a cell all files are updated on the hard disk. This includes the BP6 and CL6 files for the particular cell and the CELLS6.DBD and CONSTS6.DBD files.
- The two files CELLBKUP.BP6, CELLBKUP.CL6 store your cell as it was when you enter cell mode. These may be used to return a cell to the same state that it was when you entered cell mode by using the UNDO command.
- The two files MAINBKUP.BP6 and MAINBKUP.CL6 are updated whenever you use certain commands that are hard to reverse. You may recover from an accidental operation by using the UNDO command.

IMPORTANT: Having a current TLC file of your cell is the best way to recover a cell. However, if you have not been making TLC backups, and you have a disaster, the internal files (.BP6 and .CL6) will be current as of the last time those files were written to the hard disk; that is, you saved to the hard drive using the SAVE or SYS commands or changed cells. Just restart LASI and that last version of a cell should be recovered.

Boxes/Paths/Cells

A drawing is built of things called OBJECTS. In a drawing objects are placed in a hierarchy, the position of an object in this hierarchy is called its RANK.

The lowest rank (0) objects are the basic constructions, BOXES, PATHS, POLYGONS and TEXT.

There are 64 layers available for boxes, paths and text, numbered 1-64.

The higher rank (1-15) objects are the CELLS, which are the basic structures of a drawing. Any cell can contain one or more objects of lesser rank.

All the cells in a drawing are called the CELL COLLECTION.

Boxes

Boxes are simple objects that have the properties of four sides, each orthogonal to the adjacent, and a layer. Only the position of the sides and the layer may be changed.

Paths

Paths are a set of vertices in some order which display as a set of endwise merged rectangles, all with the same width, but usually different lengths.

A path with a zero width is called a POLYGON or POLY for short, which is sometimes called a polyline in other CAD systems.

A polygon is special case of a path, and the term "path" can be used to refer to either, except in the case where a zero width path is explicitly indicated.

A polygon need not be closed. A line of zero width however usually does not reproduce anything useful in an integrated circuit, so polygons will usually be closed intentionally.

Vertices may have their position changed, and the vertices of a path may be added or deleted. The width or layer of the path may be changed.

Paths with a POSITIVE width are drawn with their ends INLINE with their end vertices, while paths with NEGATIVE widths are drawn with their ends EXTENDED out half the width from the vertices.

Paths with extended ends should be used if the layout is to be converted to CIF. The TLC2CIF converter program will however change the end type if necessary.

Paths with either end type can be used if conversion will be to GDS, since GDS has a path end type attribute.

A path with width can have no more than 250 vertices. A polygon can have no more than 500 vertices.

Text

Text is a form of path. Instead of displaying vertices, it displays as a string of text characters. Text is stored as a special path, and many of the commands that affect paths work on text objects. Text can have a layer and a size which corresponds to a normal path's width. A single text object can have up to 56 characters.

Cells

All drawing is done within a cell. A cell has a NAME by which it is called for drawing or for insertion into other cells. Under Windows LASI the cellname must be no more than 32 characters and may not include certain characters (Read Naming Convention). A cell is named when it is created, but it may be renamed. A RANK is also assigned when a cell is created, which later can also be changed.

The LASI drawing program uses cell ranking to organize drawings so that computer memory usage is better controlled and reentrant constructions (i.e. cells nested in themselves, directly or indirectly) are avoided.

In general, the overall layout drawing will be the highest ranking cell. There however may be many cells of that rank, for perhaps different versions of the same integrated circuit.

Cells have a property known as AREA. The area of a cell is the region within the rectangle that fully encloses all the objects that make up the cell. A cell's OUTLINE is the perimeter of the area, or simply the enclosing rectangle. Cells may be drawn in full, or may be drawn as only their outline. Cells in a part of a drawing that is not currently being worked on may be outlined to improve clarity and to speed up redrawing.

When a cell is drawn as its outline it may be manipulated with certain restrictions as though it was drawn fully. Read the CGET, CPUT, CMOV, OUTL and FULL command topics.

Cells may also have their IMAGE turned on. Drawing of cell images can be toggled by the I key. A cell's image is just its outline that is drawn in addition to any boxes, paths, or other cells. However, the cell is still a normal cell. All the commands that work on normal cells work on a cell with its image displayed. A cell image is independent of which layers (if any) are viewed, and shows exactly where a cell's area is located.

If a cell's image is turned on, its reference location (cell's origin) will be indicated by a small diamond marker. This can be very helpful if you are adding text for the LASICKT program.

Command Help

LASI has command button context help. To use it hold down the F1 key and click a button. The help information for that button should popup. Note however that if you release the F1 key before the mouse button, you will activate the command not the help screen.

Calling button help simply passes the command topics to the LHI help program. Once started, you can navigate around the help just as if you used the "?" help button. You must exit help by clicking "Quit" or using the usual window close "X" button.

Command Menus

LASI 6.0 has a conventional Windows menu and toolbar at the top of the display. The commands at the top are system type commands and drawing window commands.

Most operations are done by selecting a command from a menu on the side of the screen. Clicking the LEFT mouse button activates the command.

There are two different menus. To flip between them, simply click the RIGHT mouse button or click the big Menu # button.

The current resident command is shown at the bottom left of the display.

Command Types

The commands that are found on the menus do many things. The commands all have help topics in Command Information.

Commands either operate immediately, request coordinate points by mouse, or pop up a dialog box to request additional information to be typed in. The commands may be aborted by selecting another command or pressing Esc, or continued to completion.

Commands may be grouped into four types:

- RESIDENT commands are commands that stay in place until you change to another resident command. There is always a default resident command when working on a drawing.

Examples of resident commands are: ADD, GET, PUT, CGET and CPY.

- ONETIME (or ONCE) commands are commands that abort any other commands that may be in progress, but are not retained.

Examples of ONETIME commands are: SORT, FILL, LAYR, and OBJ.

- WINDOW commands are commands that change the window or some parameter of the display. These commands are nestable within a resident command. For example, you can reposition the window an unlimited number of times while you still have the second point of a MOV pending. The command listed at the bottom of the display changes color to tell you that you are in a nested command that is asking for an input.

The window commands are: PAN ARROWS, CNTR, DGRD, DRAW, FIT, GRID, OPEN, RDRW, VIEW, WGRD, RWIN, SWIN, XPND and ZOOM.

- SPECIAL commands are commands that run other programs. An example is the READ command that appears in most LASI programs.

Another special command is the SYS or system mode command. This pops up a dialog box that runs drawing maintenance commands or creates instances of LASI utility programs. This is equivalent to the old DOS LASI system mode.

Common Problems

When LASI is started it checks for the FORM.DBD file, the

CONSTS6.DBD file, the CELLS6.DBD file and the text font file (usually TFF.DBD). If there is a problem LASI should tell you. This may not be perfect and if you crash first suspect the CONSTS6.DBD file. It may be incorrect. Simply erase it from your drawing directory. LASI will make a new one. CONSTS6.DBD files are not guaranteed to work with newer versions of LASI.

The LASI system is a very complex and evolving system. It is possible that situations will occur with different hardware and software environments that will cause difficulties. Once authentic problems are reported they are usually fixed. Consult the author or your source of this software for the latest version.

Drawing Speed

Notice that with LASI 6.0 you may have to redraw once in a while when using multiple windows. Windows seems to save only one level of the display most of the time. LASI does not try to refresh any area that has been overdrawn, but redraws manually. This saves time because a complete redraw sequence is not done whenever the display is changed.

On a fairly slow computer, drawing a large number of cells can take a considerable amount of time. LASI has several features to allow you to work on larger drawings more effectively. These are as follows:

- Pressing Esc causes a drawing sequence to abort. Undrawn cells will be drawn as dotted outlines.

You don't have to redraw fully if you only want to locate a certain area. Just find where the area is, abort further drawing, and then use the ZOOM command.

- Cells may be replaced by their outline, which draws almost instantly. Read the OUTL and FULL command topics.

- Cells that are too small to draw are drawn as their outline (dotted). This is NOT the same as putting the cell in outline, as listed above.

- Objects that are too small to draw may be skipped by setting the resolution. Read the SET command topic.

- Objects that are out of a drawing window are remembered and are skipped for certain commands.

- Finally, the DPTH command can be used to limit the expansion of nested cells as they are being draw. Read the DPTH command topic.

NOTE: If you speed up drawing by pressing Esc, active box sides and active vertices are still marked, and active cells are still drawn as filled outlines. This is to indicate where active objects are located.

HINTS:

1. Run a disk organizing program such Microsoft's DEFRAG on your hard disk frequently. If cell files become badly fragmented, the drawing time for LASI can become incredibly long.

2. When you draw once, Windows buffers cell files in memory (depending on how much RAM you have). The next time you redraw, LASI may not have to go to the hard drive to find a cell and will draw much faster.

Drawing Universe

The drawing area is 16,777,214 units wide since positions are determined by a 32-bit integer. This may not make sense for 32-bits, which is 4 billion units, until you know that LASI actually clips to 24 bits. This drawing space might be called the drawing's universe. This should be adequate for almost any application where LASI is

intended.

If objects are moved they may exceed the signed integer limits of +/-8,388,607 basic units. Since the positions are still 32-bit, objects that move beyond the limits can actually leave the drawing universe, and unless they are still active, they can't be moved back again. The SORT command in addition to ordering objects will move any objects back into the drawing. Read SORT.

Commands affected: CPY, FLP, MOV, ORIG, WMOV, QMOV, CMOV, ROT, STEP

Edit in Place

When editing a lesser cell, you can have LASI overlay the boxes, paths and text of the greater cell that contains the lesser cell. You can then change the lesser cell to match the greater cell. This is a version of what is commonly known as "edit in place".

What you are really doing is normally editing the lesser cell, but the greater cell is always shown as a reference with the correct placement and orientation it has to the lesser cell.

How to Edit in Place:

1. Make a lesser cell active with CGET. Only one cell should be active. If more are active LASI will select the first one it finds in the cell data.
2. Hold down the Ctrl key and click the CELL button or menu item at the top of the window as you would normally do to go to the active cell. This tells LASI to go to EIP mode.
3. Make any changes to the lesser cell, but note the following:
 - ù You are making changes to all instances of the lesser cell.
 - ù The greater cell may be flipped or rotated.
 - ù The cellname in the title bar is the greater cellname with the lesser cellname separated by a period ".".
 - ù The FIT command acts on the lesser cell being edited.
 - ù Using SAVE or going to system mode (SYS) saves and/or reloads the lesser cell just as if you were working on it normally.
 - ù You can go deeper into cell nesting in the lesser cell if it has cells, however, there is no back trace.
4. Clear EIP by using CELL or LIST to go to another cell.

Error Messages

Most error messages are self-evident. Some error messages are less obvious and may be explained as follows:

- "Not a Valid Name". A name for a cell is too long, contains illegal characters, or is a reserved name.
- "Cell Unavailable". A cell is probably an attached pool cell.
- "Cell Present". A cell already exists.
- "Cell Not Present". A cell does not exist in the cell collection or pool.
- "Rank Error". You have tried to add or change a cell with rank greater than or equal to the present cell.
- "Unable to Read BP6 File". Internal box and path file is missing or corrupted.

"Unable to Read CL6 File". Internal lesser cell file is missing or corrupted.

"Utility Failed to Load". A utility program was missing and was therefore unable to be started by Windows.

"Not Enough Space to Smash Cell". When smashing a cell, you would exceed the cell object allocation if smashed.

"Not Enough Memory". General memory allocation error. You may have too little RAM for several running processes. Check running processes and restart LASI.

"File Error". General file reading error. File was probably missing.

"No Object ... Using BOX". The object to be added was not available, default is to box object.

"Unable to Free Memory". General memory deallocation error from the object manager. Restart LASI.

"Too Many Cells". You have exceeded the cell collection limit of about 512 cells.

"Smash Failed ... Use UNDO". While smashing, something went wrong. Use the UNDO command to load the backup.

"Too Few Arguments". A command that required arguments received too few.

"Unknown Command" The CMD command was given a command that doesn't exist.

"Pool Collection Missing". LASI was unable to find the CELLS6.DBD file in a pool directory.

"New Cell Aborted". When trying to create a new cell it was aborted for some reason.

"Cell Not in Pool". A pooled cell was not in the cell collection of the pool directory set in FORM.DBD.

"Cell is Attached". Cell is a legal pool cell and therefore can not be manipulated like a normal cell in the drawing.

"No Printer Installed". You tried to print without LASI knowing anything about the printer.

"Font File is Missing ... Trying TFF.DBD". When booting, LASI could not read the expected font file and will try the default file.

"Font File is Incorrect or Missing". LASI could not find TFF.DBD or other default font file. Check for a good TFF.DBD in the \LASI6 directory.

"Unable to Read or Create FORM.DBD ... Unknown Reason". When booting, FORM.DBD is probably corrupted and should be erased.

"CONSTS6.DBD is Incorrect ... Delete it". When booting, the CONSTS6.DBD file read incorrectly. Allow LASI to make a default.

"Unable to find \LASI6 Drive ... Defaulting to Current Drive". The drive that should contain the \LASI6 directory was not valid.

"Can't Find Reader Program". The reader program listed in the Reader setup was not found. Check its name and path.

"Font Size is Incorrect ... Set Display to Small Fonts". The LASI menu is arranged for small display fonts. Reset the font in the Windows Display Properties.

"Single Path Vertex Limit". You have exceeded the number of vertices in a single path of about 512.

"Unable to Create Folder". The CREATE command was not able to make a new folder for a new drawing.

File Types

The LASI drawing program uses several files. These are:

DBD (Drawing Basic Data) Files

CELLS6.DBD is the master list of cells used by LASI to know which cells it has to use in a drawing. The position of a cell in the file is an index used by LASI to keep track of which cells are used in other cells.

FORM.DBD contains the configuration information used by LASI when it starts. Read the General Information/Form File topic.

CONSTS6.DBD is a file that is maintained by LASI itself, and contains the information (or constants) that determine the present settings of many parameters. Although it is ASCII this file should not be edited.

TFF.DBD, TFF2.DBD and TFF3.DBD are a generic text font files. These files are binary files that contain patterns for the text characters. These files are generated by the WMAKETFF.EXE utility and can only be modified using that utility. Otherwise, the user is free to make his own fonts following the directions under the topic Text Generation.

Cell Data Files

Cell data files are named with the name of a cell with a BP6 or a CL6 extension.

BP6 files are internal binary files that contain information on boxes, paths, text and vertices. These files are maintained by LASI and other utilities and are not to be externally edited.

CL6 files are internal files that contain information on which cells are contained in other cells. Rank 1 cells do not produce these files. These files are not to be externally edited.

Form File

The FORM.DBD file customizes LASI to the specific hardware and software you are using. The FORM.DBD is local to a drawing directory so it can be different for each drawing. You can copy a FORM.DBD file from another drawing, or a default FORM.DBD will be written when you run LASI.

If you need to change FORM.DBD use a simple text editor such as Notepad from Windows. Once you establish a "normal" FORM.DBD file, you can just copy it from drawing to drawing. Read the General Information/Form File topic for a description of the different parameters.

The FORM parameters are keywords ending with "=" followed by a FORM variable. There is no SPACE character between the "=" and the variable. The parameters may be in any order, or be omitted. If a parameter is omitted a default parameter is used if possible.

A comment may be added after any parameter if it is begun with a semicolon ";". A parameter line should not exceed 75 characters.

The present FORM parameters are:

fkey=command string	function key assignment
font=filename	path\name of default text font file
pool=pooldirectory	path\name of cell pool directory

layer=name number	add layer table entry
misc=variable	various variables

- "fkey=" assigns a particular literal command string to a function key (F2 through F12). Command strings are assigned sequentially as they are read in the FORM.DBD file. F1 is reserved for "help", but up to 11 more commands can be assigned.

This works exactly like the "Cmd" command except that pushing a key executes the command. Read "Cmd" under Command Information.

- "font=" is the path\name of a custom text font file. If this parameter is left blank or if it is left out of the FORM.DBD file altogether, the default is "\LASI6\TFF.DBD". Note that this variable must contain a path. This lets you to make your own text font files and put them in any directory.

- "pool=" sets the directory where pooled cells are stored. If the pool directory is on a different logical drive than the drawing directory, precede the directory name with the drive letter. Omit "\" from the end of the pool directory name. If you don't have a pool directory leave this parameter blank or erase it from FORM.DBD altogether.

- "layer=" sets up the layer name table. read General Information/Layer Table for how to use layer tables.

- "res=" sets up parameters for resistor measurement. Read the RES command help topic.

- "misc=" set miscellaneous variables. The only present variable is "nocellpopup" which prevents LASI when starting from asking for a cellname with the CELL command popup.

HINT: Any line not beginning with a keyword will be read as a comment. You can insert your own notes or turn parameters on and off just by preceding a valid parameter with a character such as a semicolon.

IMPORTANT: Be sure to end the FORM file with a newline. If not the last parameter might not be read.

Hardware

The LASI 6.0 will run on almost any PC that runs Windows 95/98/NT. The general rule is: if Windows runs well, so will LASI.

Memory

At least 32K of memory is recommended, because LASI can open not only its main drawing program, but can be running one or more of its utility programs at the same time.

Video with Color Monitor

LASI does graphics using the Windows GUI. Graphics are therefore under the control of Windows and if Windows does its graphics properly, so should LASI. LASI will size its display to the present Windows video resolution. For a small monitor VGA (640x480) would probably be used, for a 15" or larger monitor, SVGA (800x600) or XGA (1024x768) would be used. LASI 6.0 will use more than the standard old 16 colors, and 8-bit or 256 colors is optimum. More colors are unnecessary unless you have hardcopy problems. See the PRINT command topic.

Printer

Hardcopies can be made directly from LASI. Doing this is very handy to examine and keep track of your drawings. LASI 6.0 works through

the Windows device driver you install with your printer. LASI will print up to the capability of your printer or driver. Color inkjet printers should make nice color hardcopies. Dot matrix or laser will print black and white copies.

When you run Setup from the Print command dialog, the information returned indicates just how Windows "sees" your printer. This indicates how many colors, etc. you have available.

If your printer doesn't work, contact the author to see if anything can be done about it. Have technical information available on your printer and be sure that it is capable of bitmap graphics.

Key Assignment

Enter does a DRAW command.

Ctrl-Enter toggles between numbered layers and the layer table
ARROWS pan the drawing window in that direction.

TAB toggles the cursor between a small cross and crosshairs.

W forces the cursor to the working grid.

U forces the cursor to the unit grid.

A toggles the cursor between working and unit grid.

C toggles the path center line on and off.

D toggles the distance marker on and off.

I toggles the cell image on and off.

N toggles the outline name on and off.

O toggles the octogonal cursor mode on and off.

R toggles the 0,0 reference mark on and off.

T toggles the text reference point on and off.

The above keys can also be toggled by clicking the LEFT mouse button on the corresponding button in the toggle button area of the display.

X pops up the PKE box with X edited.

Y pops up the PKE box with Y edited.

Z sets the measurement zero point.

SPACE gives a measurement from the zero point.

Ctrl has several uses:

ù Ctrl causes the mouse cursor to work the same on outlined cells as it does on fully drawn cells.

ù Ctrl changes the way the pan and XPND commands work.

ù Ctrl reverses the VSTP command.

ù Ctrl activates Edit in Place mode.

Esc has several uses:

ù Esc aborts most commands at any stage of completion.

ù Esc aborts drawing at various stages. Read the General Information/Drawing Speed topic for details.

F1 switches command activation to command help.

Local Network

The programs and resources that LASI uses are always loaded locally into memory. This means that files in the \LASI6 directory can be installed on the disk drive of one central server and accessed by a large number of local stations on a network.

The server must be visible to a local station as a normal disk drive having the usual letter designation.

The cell pooling feature of LASI can be used to provide a common set of basic cells over a network. Put the network server drive\path in the "pool=" parameter in the FORM.DBD file. LASI will then load an attached cell's BP6 file from the server.

Doing this may however greatly slow down drawing speed since each time a cell is drawn it must be transferred from the server. You should instead make a permanent copy of a pooled cell from a network using the IMPORT command.

Also remember that for transferring cells with rank greater than 1, you must use TLC format. You can use the TLCIN command in system mode and set the source to the server's directory. Any files transferred by TLC will become normal cells not attached cells.

Layer Table

LASI version 6.0 has two ways to select layers. You can use the numbered layer method LASI has always used, or you can set up a table of names that gives the numbered layers a name. This is usually preferred because it lets you remember more easily which layers are which masks.

To use a layer table, simply write the layer name and number separated by a SPACE as the variable in the "layer=" parameter of the FORM.DBD file.

Rules for the Layer Table:

- If the variable is not a name followed by a space and then a number (1-64), the parameter will be discarded.

- If the parameter "layer=" is left blank, a blank table entry or menu button will be made.

- Layer names are truncated to 4 characters to be CIF compatible.

- Up to 32 layers can be entered into the table; their order in the table is determined by their order in the FORM.DBD file.

Press the Ctrl-Enter keys to toggle between number or table modes.

With the table toggled ON, whenever you click the VIEW or OPEN commands, the layer dialog will pop up. You can click a check box to turn a layer on or off. When a layer is ON the layer check will be shown.

The VIEW command doesn't take effect until you redraw. OPEN takes effect immediately after you exit the menu.

The layer table also works with the LAYR, TLYR and CLYR commands. Simply click the box of the layer you want.

Each cell mode rank has its own table layers that are viewed or opened. These are stored in the CONSTS6.DBD file and returned if you change ranks or if you quit LASI and restart again.

Managing Objects

The data for LASI drawing objects is mostly kept in memory. The allocation of memory for objects is controlled by an Object Manager. The object manager controls the memory used for box, path, vertex and cell object data records, which are all different.

To simplify matters and reduce memory needed, memory is always allocated in 8KB (8192 byte) blocks, without regard for the type of object. Different cells can have different types of objects, and the object manager will accommodate.

In each block, a certain number of object records can be fitted. The number is the largest number of records that will completely fit in the block.

Each type of object has a maximum number of blocks per cell. The largest number of objects is the (no. records/block)*(no. blocks).

To see the allocated maximum number of objects press Ctrl-a.

Note that each cell rank assigns a new set of blocks to a maximum of 15. Normally you will not make a drawing 15 ranks deep.

LASI 6.0 frees all its data area on certain commands, usually when the drawing data has been stored to the hard disk. Doing this gives maximum memory to run other programs.

Making Measurements

The position of the cursor in the drawing window is continuously read out at the bottom of the screen. The coordinates are either in working grid units or in the smallest possible grid unit, the unit grid. You may switch between these by pressing the A,W or U keys.

There is no ruler, but distances can be measured graphically. The Z key zeroes the measurement reference to the current cursor grid point. The current cursor grid point may be either in the working grid or the unit grid, depending on the resident command, or if the cursor grid has been switched by the A,W or U keys.

If the SPACE bar is then pressed, a measurement from the zero point will be displayed at the bottom of the screen. This second point will be gridded to the present cursor grid.

If the D key is pressed or the D toggle button is clicked, the distance marker will be toggled on and off. A depressed "D" button is displayed in the toggle button area of the screen to indicate that the marker is ON.

NOTE: You can hold down the SPACE bar and move the mouse. This will give a continuous readout of the distance from the reference.

Mouse Cursor

LASI is designed to be very mouse intensive. Positioning the cursor and clicking the mouse buttons chooses the commands and inputs most all the graphical information. The exception being when a literal or numerical input is required.

LASI doesn't use any "clicking and dragging". Each cursor input is a single up and down mouse button click. The reason for this is that LASI always allows you to make drawing window changes within other commands, and therefore must be able to nest mouse button clicks.

The mouse cursor is the standard arrow if out of the drawing area, otherwise, the cursor is a small cross that may have other figures added to it as follows:

- When a command expects a distance displacement, a dotted line or vector will appear.
- When a command expects to select or "get" something, a dotted rectangular cursor window will appear. A point is defined to intersect the window if it is ON or WITHIN the boundaries.
- When a BOX is to be added, the dotted rectangle previews the box.
- When a path is to be added, one or two dotted lines preview or "rubber band" the new path.

The cursor can be toggled between the small cross and crosshair lines by pressing the TAB key. The cursor automatically turns back to the small cross if not on the drawing window.

When a cursor input is expected, the point needed will be shown at the bottom of the screen after the command name. Most commands take one or two points.

The cursor moves in discrete steps. When a command is chosen, the cursor is set to move either in the unit grid or in one of a number of preset working grids. The working grids are entered using the SET command, and are changed with the WGRD command. The cursor's grid type may change from command to command or during a command, but it

may always be toggled from one type of grid to the other by clicking the A,W or U toggle buttons, or by pressing either the A,W or U keys on the keyboard.

Some cell commands use a DOUBLE CLICKING of the LEFT mouse button to

change the way that the command acts on cells. If you click twice rapidly, the cells with AREAS that anywhere OVERLAP the mouse cursor window will be affected. If you wait a bit between clicks, only the cells that have their AREAS fully ENCLOSED by the mouse cursor window will be affected.

The time between mouse clicks is set by the Windows Control Panel.

Octogonal Cursor

The O key or the "O" toggle button turns on the octagonal cursor on certain commands.

The "O" button depresses to indicate that octagonal mode is on.

When octagonal cursor is on, the cursor movement vector is snapped to the nearest 45 degrees. The commands ADD (for paths), MOV, WMOV, QMOV, CMOV and CPY are affected.

The octogonal cursor is used to restrict drawing so that only 45 and 90 degree angles are allowed, which preserves drawing neatness and conforms to certain physical artwork generation requirements.

HINT: If the cursor behaves oddly, first try pressing the O key.

Operating Modes

DOS LASI had a separate mode called System Mode. LASI 6.0 has system mode, but it appears as a popup dialog.

System mode is a housekeeping mode which has commands for overall manipulation of cells and the control of certain drawing parameters.

Cell mode is the mode where actual drawing is done. When in cell mode a cell drawing is opened for creation or modification. LASI 6.0 is in cell mode by default.

Cell mode has the following properties:

- Cell mode has up to 15 ranks.
- The rank of cell mode is obtained from the cell being drawn.
- Each cell mode rank keeps separate parameters, such as window size and position, in order to simplify moving between different ranks of cells while working on a drawing.
- All cell modes are identical except for the ranking.
- Both modes have a menu of commands on the side. The commands are explained under their own help topics.
- When LASI is started it goes to cell mode.
- Using the SYS command that appears in the cell mode menu is the normal way to enter the system mode dialog from cell mode.

A cell may be loaded into cell mode directly by adding the name of a cell to the command line when starting LASI. This may be added to a Windows shortcut "Target" line.

Example: "c:\lasi6\wlsi flipflop" runs LASI and loads "FLIPFLOP".

Read the CELL, LIST and SYS commands.

To add a CELL:

- Enter a single coordinate point at the cell's origin with the mouse.
- The cursor shows a dotted outline of the cell with the cross cursor at the cell's origin.
- A cell is always added in its normal (0 deg) orientation.

NOTE: Use the TEXT command to add text to a drawing.

aGet

AGET (all get) is a combination of FGET and CGET.

- This command acts on cells, boxes, paths and text in the same way as those commands.

AGET is useful if you are MOVing, CPYing or FLPing large sections of a drawing because you can see just which objects are active before the operation, and unlike WGET it will not distort paths.

aPut

APUT (all put) makes all boxes, paths, text and cells inactive.

- This command is for making sure that ALL objects are inactive.

APUT is a handy command to make objects inactive with a single mouse button click.

Arc

ARC calls the arc generator.

To make an ARC:

- To add an arc to an existing path, make a single vertex on the path active.
- To start an arc from nothing, first make a single vertex path and then call ARC.
- Enter the center and end points with the mouse.
- In the dialog box enter the number of segments and the direction (cw or ccw).

NOTES:

1. The radius of the arc is a linear function of the angle and the difference in radius between the starting and end points.
2. Arc internal points are added on the nearest unit grid.
3. Arc center and end points are added on the nearest working grid.
4. When you start an arc the cursor displays one or two dotted lines (rubber bands) which show approximately how an arc will be added.
5. If the active vertex is not the last vertex, an arc will be inserted into the polygon between two vertices. A cursor with two dotted lines will be drawn that gives information on how the arc will be drawn.
6. After an arc is added, the last point generated will remain active.

HINT: It is easy to generate duplicate vertices when you are adding arcs to polygons. When you are finished with one or more arcs, click the UDUP command. This will remove any consecutive

identical vertices.

Attach

ATTACH installs a cell from a common cell pool in a drawing's cell collection.

- ATTACH will first search to find if a cell already exists in the drawing. If one is found you will be ask if you want to continue. If a cell is not found, it will be immediately added to the cell collection.
- ATTACH will turn a normal cell into an attached cell, but the attached cell must exist in the pool directory.
- Higher rank cells may be attached but you will only get the boxes, paths and text of the cell, and the cell will changed to a rank 1 cell.
- If the "pool=" parameter in FORM.DBD is missing or blank, the ATTACH command will not work.

Read the General Information/Attached Cells topic.

Attr

ATTR pops up the layer attribute dialog box.

- The attributes are Color, Fill and Dash.
- The colors are: red, orange, yellow, lime, green, turquoise, cyan, azure, blue, purple, magenta, violet and white. Enter only the first letter.
- The fills are numbered 0-13, 0 is none, 13 is solid.
- The dashes are set by Windows and are number 0-5, 0 is solid and 5 is a dash-dot-dot pattern, with combinations thereof in between.
- The present layer attributes are shown in the "Layer # Attributes" group. These may be edited and then set.
- If you "Close", the present layer attributes will be set.

ATTR works differently depending on the layer table being on or off.

Layer table off:

Write the layer number to be loaded in the box and click the "Set+Load" button. The present layer attributes will be set and new layer attributes will be loaded.

Layer table on:

Click the check box of a layer. The attributes of the present layer will be set and the new layer attributes will be loaded.

All 64 different layers can be displayed in two pages of 32. Click the ">>" or "<<" button to flip between pages.

To see examples of fills and dashes click the "Patterns" button.

You also have temporary check boxes that let you turn on solid fill, no fill and no dashes. These stay in effect until you uncheck them.

Cap

CAP calls the capacitance calculator.

To measure capacitance:

1. When the RES/CAP dialog pops up enter the capacitance in units/physical area unit. This will usually be in pF/um².
2. After you enter the capacitance per unit area, all PATHS (width not equal to zero) with ANY active vertex will be measured automatically, and the total capacitance will be displayed.
3. After all paths are measured, all boxes with ANY active sides will be measured automatically, and the total capacitance of the paths and boxes will be displayed.
4. The command will then go into manual mode. Any polygons or areas in cells that should be included in the capacitance can then be manually measured.
5. When in manual mode, at the command prompt "[/]", you must fill the area to be measured with rectangular cursor areas by clicking the LEFT mouse button at diagonally opposite corners of the areas.
6. An area's capacitance and the sum of previous capacitances will always be displayed at the bottom of the screen.

IMPORTANT: CAP is a RESIDENT command, that is, it recalls itself. To end measuring, click the mouse on the menu button of another resident command. If you click on a non-resident, CAP will recycle itself.

NOTE: Unfortunately at the moment, LASI does not directly measure the capacitance of irregular areas (polygons). To measure an irregular area you must click small rectangular areas until the total area is filled with minimum overlaps. An area once measured is indicated by a lightly dotted fill.

This command may be used as an area parameter calculator. For example, you can calculate the area of an integrated circuit by setting the capacitance per area to 1.

cCel

CCEL (change cell) changes any active cells into the specified cell.

- This command will not work in a rank 1 cell.
- Attempting to change to an unknown cell or a cell too high in rank will abort the command.

Cell

CELL loads a cell or creates a new one.

- A dialog box will pop up and ask for a cellname.

If the cellname is not in the cell collection, a new cell will be created and you will be asked for a rank.

or,

- The first active cell that is found in the drawing becomes the default cellname, and you will automatically exit and load that cell.

This second method also activates the Edit in Place feature if Ctrl is pressed. Read General Information/Edit in Place.

NOTES:

1. To exit from the CELL popup, just click Cancel or press Esc.
2. When LASI is started, it has no cell defined. The CELL command is automatically started so that a cell can be chosen or created.

To turn this off, add the FORM.DBD parameter "misc=nocellpopup".

3. You can go directly to a cell when starting LASI by adding the cellname on the shortcut "Target" line after "wlasi.exe".
4. You can use the LIST command to go to a cell by double clicking the on the name of the cell.

CellSort

CELLSORT sorts the cells in the drawing's cell list (CELLS6.DBD).

- The cells are sorted first into groups of increasing rank and then are arranged alphabetically within the rank groups.

IMPORTANT: DO NOT interrupt the sorting (by rebooting) because the cell pointers may be corrupted and you will have the wrong cells appearing in other cells. This can be fixed by using the CCEL command, but it can take a while.

cGet

CGET (cell get) makes cells active.

- A cell can be made active in two ways:

. If the rectangular cursor window completely ENCLOSES the cell's area, the cell will become active.

. If the LEFT mouse button is double clicked while making the cursor window, a cell will become active if the cell's area OVERLAPS the cursor window. This does not work if the cell is drawn only as an outline.

- An active cell turns bright white on all layers.

- An active cell drawn as an outline turns bright white and is filled with dots.

- There is no dependence on VIEW or OPEN.

- If a cell is not found in the cursor window, a cellname dialog will pop up.

NOTE: If a cell is in outline, only the first method above will work, unless you press and hold Ctrl when using the mouse.

CGET only works on the top level of cells and has no effect on boxes, paths or text.

Clone

Clone makes a new basic drawing from the current drawing.

- A new drawing folder is made if it doesn't already exist.

- The FORM.DBD and CONSTS6.DBD files are copied from the current drawing directory to the new folder.

NOTES:

1. Use TLCOUT to transfer any files that you may want in the new drawing.
2. CLONE does not add any shortcuts or icons to the Windows environment. To access the new drawing, copy a previous shortcut and change the "Start In" to the new folder.

cLyr

CLYR (change layer) changes the layer of boxes, paths or text.

- If the layer table is on, the layer table dialog will pop up.

Click the box of the new layer.

All 64 different layers can be displayed in two pages of 32. Click the ">>" or "<<" button to flip between pages.

or,

- Enter a new layer number from 1 to 64.
- Boxes must have ALL sides active to change layer.
- Paths must have ANY vertex active to change layer.
- Text will change layer if active.

Read LAYR.

Cmd

CMD calls up a list of literal commands and executes them.

- The command in the list can be almost any menu command if executing it literally makes sense.
- The last command is retained at the top of the list if not in the list.

A command must consist of the command name and any coordinate arguments that the command requires separated by comma or spaces.

Example: "get 0,0 100,200"

"Gets" objects open in the area 100 physical units right and 200 units up.

- If there are too few arguments (in coordinate pairs) the command will resort back to mouse input for that command.
- If the command is a resident command the current resident command will be retained.
- The MOV, CPY, QMOV, WMOV and CMOV commands only require a single x,y coordinate pair for the movement part of the command.

NOTES:

1. Some commands have a popup that requires check box or other input. These do not accept coordinates, but will work normally as if the menu button was clicked.
2. CMD cannot be assigned to a function key because the "fkey=" command strings are actually passed on to the CMD command.

Macro Commands

CMD can now open a text edit window where you can write a series of single line commands. To do this click the "Macro" button. These commands will be executed sequentially as they are written.

- Each command should be on a single line terminated by a new line <Enter>.
- The final command must be terminated by a new line.
- Spaces at the beginning or end of a command line are tolerated.

HINT: Keep your favorite commands, for example a series of ADD commands, in a text file, and then open the file with the READ menu command. Copy the block of commands to the clipboard, and then Paste the commands to the macro window.

cMov

CMOV (cell move) moves cells directly.

- A cell can be moved in two ways:
 - . A cell will be moved if the rectangular cursor window completely ENCLOSES the cell's area.
 - . If the LEFT mouse button is double clicked while making the cursor window, the cell will be moved if the cell's area OVERLAPS the cursor window. This does not work if the cell is drawn only as an outline.
- The first two cursor points form the cursor window.
- The next two cursor points determine the distance to be moved.
- When used with CMD or an F-key, CMOV needs only a single x,y pair for the move part of the command.

CMOV only works on the top level of cells and has no effect on boxes, paths or text.

Cntr

CNTR centers the drawing window.

- The new center is a single point from the mouse.
- The display is redrawn.

Read DRAW.

Copy

COPY allows NORMAL cells (not appended) to be copied or appended to other NORMAL cells.

Rules for Copying:

- The source cell must be a normal cell in the cell collection.
- If the destination cell is not in the cell collection, a new cell will be created and the source will be directly copied.
- If the destination cell already exists, the source cell will be appended to the existing cell.
- If either the source or destination cells are attached cells, the command will abort.

NOTES:

1. Inter rank appending is permitted. However, only cells that have rank less than the destination cell will be copied. Boxes, paths and text are always copied.
2. Appended objects will be active objects in the destination cell. This allows you to reposition only the appended objects.
3. If the source or destination cells are incorrect, or if copying will overflow the maximum number of any objects, the copy operation will be cancelled and a message will be shown.

Also read the IMPORT and MAKE command topics.

cPut

CPUT (cell put) makes cells inactive.

- A cell can be made inactive in two ways:

. If the rectangular cursor window completely ENCLOSSES the cell's area, the cell will become inactive.

. If the LEFT mouse button is double clicked while making the cursor window (See SET command), a cell will become inactive if the cell's area OVERLAPS the cursor window. This does not work if the cell is drawn only as an outline.

- This the inverse of CGET, and just works inversely.

- If a cell is not found in the cursor window, a cellname will be asked.

NOTE: If a cell is in outline, only the first method above will work, unless you press and hold Ctrl when using the mouse.

Read APUT.

Cpy

CPY copies active objects.

- The displacement is determined by a 2-point coordinate input.

- The original objects are made inactive and the copies are made active.

- Boxes are copied if ALL sides are active.

- Cells are copied if they are active.

- Vertices of paths are copied only if they are active. To copy a path completely it must be fully active (use FGET). This feature allows you to copy only sections of paths.

- Text is copied if it is active.

- When used with CMD or an F-key, CPY needs only a single x,y pair.

Read STEP.

cSiz

CSIZ changes the size of text.

- Any active text will change its size, and the text size will be adjusted so that it is a integer multiple of 15 basic units.

- The current new size will be shown when the command is called.

Read TSIZ.

Cut

CUT breaks a path into two separate paths at an active vertex.

- This command works if there is ONLY ONE active vertex.

- The last vertex of the first path created at the cut point is made active.

Use this command to break paths and polygons into smaller segments to create new constructions, and to make shorter paths and polygons that will be more manageable for translation into other CAD systems.

HINTS:

1. Since the last vertex at the cut point is left active, you can start adding new vertices immediately with the ADD command.

2. To see the two new paths use the FGET command to highlight each.

cWth

CWTH changes the width of paths.

- Paths with ANY vertex active will change width.

NOTES:

1. CWTH can be used to change the end types of paths. Positive width causes a path to have an INLINE end, with the vertex is on the end face. A negative width gives the path an EXTENDED end, with the end face extended half the width out from the vertex.

2. When the width of a path is changed, the position of the end faces remains constant. This prevents gaps or overlaps when path widths are changed.

IMPORTANT: CWTH has a path end type changing feature. Typing in "+*"

or just "*" for the new path width changes ALL active paths to INLINE end types, but keeps the absolute value of the width. Typing "-*" changes all active paths to EXTENDED end types, but keeps the width. When the end types are changed, the position of the end faces stays the same. This is very useful if you want to change all paths (with different widths) in a drawing from one end type to the other.

Read WDTH.

Del

DEL deletes objects or parts of objects.

- An object's layer must be viewed and opened to be deleted.

- Boxes are deleted if ALL sides are active.

- Active vertices of paths are deleted.

- Paths with ONE or NO vertices on any layer are deleted.

- Text is deleted if it is active.

- Cells that are active are deleted without regard to the layers that they contain and the setting of the VIEW and OPEN commands.

NOTES:

1. Using DEL does NOT give back actual memory that has been allocated by the memory manager. If you make a large number of deletions and want to free memory, exit the current cell and then reenter it. The easy way is to use LIST and click on the current cell marked with the "*".

2. When paths are deleted, the vertex data is resorted and condensed so that there is more space available for new paths. Doing this takes time. Expect a long delete time on cells that have a huge number of vertices. Be patient and watch the percent indicator.

IMPORTANT: Before any object is deleted, the hard drive files MAINBKUP.BP6 and MAINBKUP.CL6 are updated. If you make a mistake and delete the wrong thing, you can use the UNDO command to restore the drawing.

dGrd

DGRD (dot grid) steps through the list of preset dotted grids.

- The dot grids should have been entered using the SET command.

- The present dot grid will be shown at the bottom of the screen.

- Each cell rank may have a different grid from the list of grids.

NOTE: DGRD accepts a numerical argument if you click its button while holding the Ctrl key down. A dialog box will appear and you can set the dot grid to any temporary value.

Read SET and the General Information/Form File topic.

Dpth

DPTH sets the depth level to which nested cells are fully drawn.

- The first level of cells greater than the depth is drawn as dotted outlines.
- The depth may be set at any level between 0 and 15.
- Setting depth to 0 causes the first level of cells to be drawn as dotted outlines.
- Setting depth to 15 allows all levels of cells to be drawn fully.

NOTES:

1. If a cell on the first level is drawn as a dotted outline due to the depth being set to 0, it is NOT an outlined cell and may be edited just like it was a fully drawn cell.
2. Depth of cell nesting is not to be confused with a cell's rank. The rank of a cell determines how deeply cells can be nested within that cell. The depth is how far down in the nesting levels the occurrence of a cell is found. The depth of a nested cell can be no deeper than the current cell's rank minus the nested cell's rank.

HINT: If you want to be sure you are drawing all cells, set DPTH to the current cell's rank.

Draw

DRAW erases the drawing window and redraws the current cell.

- Drawing of boxes, paths and text takes place in ascending layer order.
- In each cell nesting, lesser cells are drawn starting at the lowest rank and are overlaid upward.
- The boxes, paths and text of the current cell always overlay all layers of any lesser cells.
- Pressing Esc while cells are being drawn aborts full drawing of any remaining cells and draws only a dotted outline.
- Pressing Esc while boxes, paths or text are being drawn aborts further drawing.

NOTES:

1. If a cell's BP6 or CL6 files are missing, a dotted outline with the cell's name in the lower left corner will be drawn to indicate that there is something that can't be drawn.
2. Paths with widths near the unit grid limit may appear distorted. This is because all drawing points are set in the unit grid.
3. If you try to draw a path or poly with a vertex count exceeding the limits of 250 and 500 respectively, the path or poly may not draw completely or fill correctly, but drawing will fail gracefully.

Read the General Information/Toggle Buttons, Key Assignment and Drawing Speed topics for useful information.

fGet

FGET (full get) makes a box or path fully active.

- A box is made fully active if ANY side is made active.
- A path is made fully active if ANY vertex is made active.
- FGET is the same as GET for text.
- This command allows the entire box or path to be operated on by any subsequent command if only a part is accessible.

If you want to delete a box or path completely use this command first.

Fit

FIT sets the size of the drawing window so that it will show all of the objects in a cell.

- There is no dependence in VIEW or OPEN.
- The display is redrawn.
- If there are no objects present, fit will expand the drawing area to the maximum size permitted or the whole "universe".

Read DRAW and the General Information/Universe topic.

Flp

FLP flips objects.

- Flipping in either X or Y determined by a 2-point coordinate input.
- The larger dimension of the cursor input determines the flip axis.
- The cursor will show the approximate flip axis.
- Boxes flip if ALL sides are active.
- ALL vertices of a path flip if ANY vertex is active.
- Text will flip if active.

NOTE: FLP and ROT do not commute. That is, the order in which these commands are used is important. Objects FLPed and ROTed in different orders are not the same.

Read ROT.

Font

FONT lets you change the current text font used in your drawing.

- When LASI is started, the FORM.DBD file is read for "font=" parameters. These are installed first in a font list and can be your own custom fonts.
- "font=" names must be a complete path\filename for the font file.
- The (presently 3) standard fonts TFF.DBD, TFF2.DBD and TFF3.DBD are then installed.
- The default font file is the first file in the list.

Read the General Information/Form File and Text Generation topics.

fPut

FPUT (full put) makes boxes, paths and text fully inactive.

- This is the inverse of FGET.
 - A box is made fully inactive if ANY side is made inactive.
 - A path is made fully inactive if ANY single vertex is made inactive.
 - FPUT is the same as PUT for text.
- Use this command to make fully inactive a box or path when only a part of the box or path is displayed.

Full

- FULL removes cells from being drawn as an outline.
- Cells are changed from being drawn as an outline if a cell's area is completely enclosed by the rectangular cursor window.
 - If a cell is not found in the cursor window a cellname will be asked.
- NOTE: You can bypass enclosing the outline completely by holding down the Ctrl key.

Read OUTL.

Get

- GET makes single parts of boxes and paths active.
- The layer must be viewed and opened.
 - Box sides are made active if they intersect the rectangular cursor window.
 - Path vertices are made active if they intersect the cursor window.
 - Text is made active if its reference point intersects the cursor window.
- When made active:
- Box sides turn bright white.
 - Vertices are marked and segments of paths that will move as a unit turn bright white.
 - Text turns bright white.
- GET is the default command when LASI is started.

Grid

- GRID toggles the dot grid on and off.
- The menu button depresses to indicate that the grid is on.
 - The dot grid will not be drawn if the spacing is less than the drawing resolution set using the SET command.
- Read SET and DGRD.

Info

- INFO identifies an active object or gives drawing information.
- Objects are taken in the order: box/path/cell.
 - Only the first active object found is reported.
 - For a box, the layer is shown.

- For a path, the layer and the width are shown.
- For text, the layer and size are shown.
- For a cell, the name, rank, number, position and orientation are shown.
- An attached cell will be identified by rank "1+".

Read SHOW.

Import

IMPORT allows ATTACHED cells to be brought into a drawing as NORMAL cells.

Rules for Importing:

- The source must be an attached cell in the cell collection.
- If the destination cell has the SAME NAME as the source cell, the attached cell becomes a NORMAL cell and replaces all occurrences of its use in all other cells.
- If the destination cell has a NEW NAME, a NORMAL cell will be created with that name.
- The destination cell will always become a rank 1 cell.
- If the source cell is not attached, the destination cell is already present or is an attached cell with a different name, the command will be aborted.
- Cells never export back to the pool.

NOTES:

1. If imported to the SAME NAME, a BP6 file will be copied from the pool to the drawing directory. You would use this procedure if you want to replace all occurrences of an attached cell in other cells with a normal cell that may be modified. You can then RENAME the normalized cell, and ATTACH the original cell again to make other modified versions.
2. If the destination has a NEW NAME, a new normal cell will be created and the attached cell will remain unaffected. You can then modify the new cell and use it where desired.
3. Notice that the difference between these two procedures is that in the first procedure the attached cell is replaced automatically wherever it is used. In the second procedure you generate an independent cell and you must replace any occurrences of the attached cell individually.
4. If the source or destination cells are incorrect, or if importing will overflow the memory available, the import operation will be cancelled and a message will be shown.

Read COPY and MAKE.

Join

JOIN connects two paths to form a new single new path.

Rules for Joining:

- This command works only if there are exactly two active BEGINNING OR END vertices.
- If the active vertices are beginning and end of the SAME path and the path is not already closed, a segment will be added to close

the path.

- If the active vertices are on DIFFERENT paths and the vertices are at the SAME location, the paths will be merged into a single path, and one vertex will be deleted.
- If the active vertices are on DIFFERENT paths and the vertices are at DIFFERENT locations, a segment will be added, and the paths will be merged into a single path.

Kill

KILL removes a cell from the drawing.

All Cells with rank greater than 1 are first checked for the cell to be KILLED. If any cell contains the cell, you will be asked if you want to continue.

When KILLED:

- Cells are removed from the drawing's cell collection.
- Cells are removed from any other cells.
- The cell internal files at the MS-DOS system level are erased.

IMPORTANT: It is recommended that you keep an archive of your cells using the TLCOUT.EXE utility program. This is the only way that you can recover a cell that has been accidentally KILLED.

Last

LAST is equivalent to RWIN 0. It simply returns the last window and redraws.

It however can be very handy when used with ZOOM to view closeups and then return to a previous window.

Layr

LAYR pops up a dialog box that sets the default layer of boxes and paths to be added.

- If in numbered layer mode the value entered is clamped from 1 through 64.
- If the layer table is on, click the layer check box to set a new layer.

All 64 different layers can be displayed in two pages of 32. Click the ">>" or "<<" button to flip between pages.

Read CLYR.

IDrw

LDRW (layer draw) redraws only one layer of the drawing.

- If the layer table is used, click the layer check box on one of the layers.
- If numbered layers are used, enter a layer from 1 to 64.

HINT: This command is useful with solid fills. It shows about what a mask will look like, and usually makes errors easier to spot.

Read DRAW, SDRW and RDRW.

List

LIST lists the cells in the cell collection.

- If you double click the mouse button on a cellname, that cell will

be loaded.

- Attached cellnames appear with the rank "1+".

NOTE: The list of cells has a history feature. The number that appears in front of a cellname indicates the inverse order in which the cells were last edited. That is, number 1 is the last cell you worked on, number 2 the second to last, etc. This allows you to go through a sequence of deeply nested cells, and then randomly return to a higher ranked cell.

Rules for cell history:

- If a cell is opened again, previous events for that cell are purged from the history.
- An asterisk indicates the current cell.

HINT: LIST is the fastest way to change from cell to cell because you don't have to type or even precisely remember a cellname.

Make

MAKE creates a new cell from the active objects or appends active objects to a cell that already exists in the cell collection.

Rules for Making:

- If the cellname is a NEW NAME, a new cell will be created.
- If the cellname already exists, the active objects will be APPENDED to the existing cell.
- Boxes, paths and text are copied if ANY part is active.
- All active cells are copied to a new cell, and the rank of a new cell will be ONE GREATER THAN the rank of ANY copied cell.
- Active cells are copied to an appended cell if a cell's rank is LESS THAN the appended cell's rank.
- If an existing cell is an attached cell, the command will abort.

NOTE: When MAKE creates or appends a cell, it keeps the current position of the objects. If the cell is appended, the copied objects will be active and you can move them to a different position with MOV. If the cell is new, objects will not be active and you should use the ORIG command to reposition the objects.

HINT: When MAKE is used to create a new cell, its name is at the end of the cell list. Use LIST to go to the new cell to make any changes.

Read COPY and IMPORT.

Mov

MOV moves the active sides of boxes, active vertices, active text and active cells.

- The distance is determined from a 2-point coordinate input.
- The cursor will show the approximate move.
- MOVs are always done in multiples of the working grid, unless you press the A or U keys to switch to the unit grid.
- When used with CMD or an F-key, MOV needs only a single x,y pair.

Read WMOV, QMOV and CMOV.

Obj

OBJ lists the name or type of object to be added.

- Double click on the object type to be added.
- Only objects with rank less than the current cell are listed.

NOTE: Boxes, paths and cells are treated equally as far as adding is concerned. However, boxes, paths and text are "basic" objects while cells are compound objects. Generally, there are different commands for these different types of objects.

Open

OPEN pops up a dialog box that sets the layers that can have their boxes, paths and text made active.

- Boxes, paths and text that are not opened are blocked from the GET, FGET, WMOV, QMOV, PBEG and PEND commands.
- Cell commands like CGET, CPUT or CMOV are not affected.
- Each cell rank may have different OPEN parameters.

OPEN works differently depending on whether the layer table is on or off.

Layer table off:

OPEN accepts layer inputs that can be single layer numbers in any order, or a dash (-) can be used to indicate all inclusive layers between the end layers. A dash alone will give all layers from 1 to 64.

Layer table on:

Individual layers can be opened or closed by clicking the check box next to the layer name (defined in FORM.DBD).

All 64 different layers can be displayed in two pages of 32. Click the ">>" or "<<" button to flip between pages.

To temporarily open all layers check the All Layers box.

NOTE: Ctrl-Enter toggles the layer table on and off.

Read VIEW.

Orig

ORIG relocates the coordinates of all objects in a cell drawing so that the single cursor input point is the 0,0 position.

- This command allows objects to be constructed almost anywhere on the drawing window and then be repositioned as desired.
- The working grid is the default grid, but may be changed to the unit grid with the A or U keys.

HINT: Set you cell origin as soon as you make a cell, or you may have to relocate the cell each time it is used in higher ranking cells.

Outl

OUTL sets cells to draw as an outline.

- A cell can be turned into an outline in two ways:
 - . Cells are changed to outlines if a cell's area is completely ENCLOSED by the rectangular cursor window.
 - . If the mouse is double clicked while making the cursor window, cells will be outlined if the cell's area OVERLAPS the cursor

window.

- An outlined cell will be drawn as a white rectangle with the name of the cell in text in the lower left corner. An active cell in outline will be drawn as a bright white rectangle filled with dots.
- The cell's name can be toggled on and off by pressing the N key. If the "N" button is shown depressed names are ON.
- If a cell is not found in the cursor window a cellname will be asked.

HINT: You can greatly speed up an overall redraw by setting the cells in an region where you are not currently working to outline. When a cell is set to outline, retrieval of the cell's files from a disk is unnecessary and drawing the internal parts of a cell is skipped.

Read FULL.

OvSz

OVSZ (oversize) expands the sides of boxes, paths and closed poly by a certain distance.

Rules for Oversizing:

- A negative distance shrinks the sides.
- Only active box sides are oversized.
- All sides of paths or closed polygons are oversized if ANY vertex is active.
- The oversizing algorithm used must identify a polygon as having a rotational angle of +/-360 deg. If this cannot be done, the poly will be unchanged.
- If a poly contains adjacent sides that are folded back 180 deg on each other, the results will be undefined since the direction of folding cannot be identified.

NOTE: Acute angles on closed poly are not oversized correctly. You need to insert an additional vertex at an acute angle to prevent the peak of an acute angle from extending beyond the oversize distance.

Read RESZ.

Pan

Up, down, left and right arrows pan the drawing window.

- The drawing window is moved by a certain fraction of its width (50%) in the arrow's direction.
- If you pan with the Ctrl key down, the window moves only a small fraction of its width (5%).
- The display will be redrawn.

The arrow keys on the keyboard perform this same function to be consistent with other CAD systems.

Read DRAW.

pBeg

PBEG (path beginning) makes the first vertex of a path active.

- This command works if ANY vertex of a path intersects the rectangular cursor window.

- PBEG is useful when the JOIN command is to be used.

Read PEND.

pEnd

PEND (path end) makes the last vertex of a path active.

- This command works if ANY vertex of a path intersects the rectangular cursor window.

- This command is useful both to find the end of a path and to reopen a closed path.

Read PBEG.

pRev

PREV (path reverse) reverses the order of the vertices in a path.

- ALL paths that have ANY active vertex will be reversed.

- The original active vertex remains active.

Reversing the order allows vertices to be added to either the beginning or end of a path. It also allows paths to be JOINed properly.

Print

Printed copies of a drawing or section of drawing can be printed on a printer currently installed in Windows.

PRINT in LASI 6.0 is different from the DOS version of LASI HCPY command. Windows does not allow access to printers except through an installed printer driver.

Printer Info

- Information about a printer is first required. A standard dialog box pops up so that you can choose a printer and change its settings.

- Once a printer is chosen, PRINT pops up a dialog box that lists the default printer and its basic capabilities.

- The printer or its settings can be changed by clicking the Info button. This calls in the printer setup dialog again.

NOTE: The initial values come from the main printer setup that may be accessed through Start/Setup/Printers/Properties.

Color Setup

- The B&W check box forces the palette to a simple 2 colors, black (and white). Set this if you don't have a color printer.

- The Color button lets you change the composition (red, green, blue) of the colors that will be printed. This is useful to redefine the halftone colors (orange, etc.) to look better. Each color is defined by three components for R,G,B using 8 bits or a number from 0-255. For example, if orange prints too yellow, decrease G or increase R.

NOTE: If your printer has only an 8 color palette, set halftone colors to the closest whole color, otherwise the printer driver will choose a color for you.

Printing Method

PRINT has three printing methods: Normal, Direct and Clipboard:

- Normal creates a map in memory, draws to it using the screen display driver and then stretch the map to the printers resolution. This produces a nice solid hardcopy with about 200 dpi resolution, that should have dashes and fills, and full color if your printer supports the colors. This is the default.

- Direct draws directly to the printer. This method is more dependent on the printer driver and may have limited colors, fills that are too fine, or there may be no fills at all. However, if your printer driver supports fills and colors, direct printing will make very finely detailed prints. Click the Direct check box for this method.

Not all printer drivers will work both ways; if one doesn't work try the other.

- Clipboard does not print to any printer device. It is instead a screen copy that puts a bitmap in the Windows clipboard. This bitmap can be inserted into any document that allows bitmaps to be placed within the text. Usually the "Paste" command is used.

The clipboard can be viewed using the Clipboard Viewer accessory and stored as a .CLP file, or the clipboard can also be simply "Pasted" to a program such as the MSPRINT program, where the image can be edited and stored as an actual bitmap .BMP file.

HINTS:

1. Some new printers and their drivers are able to produce a large number of colors (16 or 24-bit). If you are using 256 (8-bit) colors for the display, the normal printing method might not have enough buffer size and fail. Increase the number of screen colors if possible.
2. Try using drivers for similar printers - sometimes you can get better results. Also, try the Microsoft generic drivers for your printer. These usually work pretty well because Microsoft usually knows the intricacies of Windows printing.

Scaling and Position

- PRINT is designed to use your printer in "portrait" orientation. It rotates 90deg automatically, Switching to "landscape" should print correctly, but will give a smaller plot.

- PRINT calculates the print scale from the display window and the paper size. You can change this scale.

- PRINT takes the center of the printed area from the center of the display window. Changing scale just expands and contracts the print area around this point.

- You can move the position printing area on the paper by changing the horizontal and vertical shifts. Normally the print area should be fairly well centered.

- PRINT treats the clipboard scaling differently. It takes the size of the drawing area on the display in pixels and the dots per inch reported from the display and determines a physical size. This is not necessarily the actual size on the screen. The scale that is calculated is this size divided by the drawing window size in drawing physical units. Changing the scale changes the bitmap size proportionally.

Note: The clipboard bitmap takes the number of bits in color pixels from the display. If you are using 18 or 24 bit colors, this can result in large bitmaps. Usually, 8-bit (256 color) is ideal.

When you start a print, the screen erases and then redraws when the printer data has been generated and has been spooled to the printer buffer.

Print tasks are usually spooled by Windows, so you can continue with your work once a print is started. If the printer does not spool, check for the correct switches in the printer setup that may be accessed through Start/Setup/Printers/Properties.

Once the printing process has begun you can abort it from the Task Bar printer icon, or from your printer's own spooler.

Put

PUT makes single parts of boxes, paths and text inactive.

- This is the inverse of GET and works the same way.

Read FPUT and APUT.

qMov

QMOV (quick move) functions like the sequence GET, MOV, PUT.

- Box sides are moved if they INTERSECT the rectangular cursor window.

- Vertices of paths move if they intersect the cursor window.

- Text is moved if the reference point intersects the cursor window.

- Box, path and text layers must be viewed and opened.

QMOV is useful for making small vertex movements and small box side movements when the path or box is only partly accessible.

- When used with CMD or an F-key, QMOV needs only a single x,y pair for the move part of the command.

Read WMOV.

rDrw

RDRW (reverse draw) redraws the drawing with the layers in descending order.

- This draws layers sequentially in reverse (64 -> 1).

- This draws only viewed layers. To speed up drawing, only VIEW layers that you want to see. Using the layer table works nicely.

This command is useful if you want to overlay solid fills so that objects that would normally be covered by higher layers remain visible.

Read DRAW and SDRW.

Read

READ calls up the read dialog that is used in most LASI 6.0 programs.

- It contains two lists that contain viewer programs and files to be viewed.

- The last few viewers and files are retained in the list. A new viewer or file if not in the list is pushed on the top.

- With the viewer set to "Notepad" and the file is set to "FORM.DBD", READ replaces the old DOS LASI FORM command.

This

lets you view and change FORM parameters from within LASI.

NOTE: If you change the FORM parameters, the effect might not be seen until you restart LASI.

ReName

RENAME changes the name of a cell.

- Cellnames may be up to 32 characters, but should conform to MS-DOS filename rules.

- This command will not work if the new name is the name of a cell that already exists in the cell collection or is a reserved name such as "b" or "p".

HINT: To give a cell another cell's name in a cell collection, you must first RENAME or KILL the original cell and then RENAME the cell.

ReRank

RERANK changes the rank of a cell and adjusts all other cell ranks.

The command works in two ways depending on the cellname:

1. If the name of a cell is given as "*", ALL cells will be checked and the LOWEST possible rank will be assigned to them.
2. If the NAME of an actual cell is given, a NEW RANK for that cell only will be requested. ALL cells will then be checked, and new ranks will be assigned according to nesting and the NEW RANK of the NAMED cell. Cells that do not contain the NAMED cell will be unaffected.

Rules for Reranking:

- If a NEW RANK is less than or equal to the rank of any lesser cells within the named cell, the new rank will only go down to one higher than the rank of any lesser cells.

- Attached cells may not be reranked.

- When you rerank, a backup of the list of cells is made. If reranking fails, for example, if ranks are forced to exceed 15, the original ranks will be restored.

- If you manage to bring a cell into the drawing that has lesser cells with rank greater than or equal to the cell's rank (possibly by using TLC) the lesser cells will draw as dotted outlines. Using RERANK should fix this problem.

- Any CL6 files of cells are preserved, except if a cell is rank 1 and the cell has no cells (i.e. it has a null CL6 file). Then, if you enter and exit cell mode with that cell, the CL6 file will be erased. This is to keep you from cluttering your drawing directory with null files.

Res

RES calls the resistance calculator.

The RES/CAP dialog box will pop up.

RES first tries automatic mode. If a resistor is drawn using a path as the resistive element, and the path has ANY active vertex, the resistance will be measured using information from a resistor parameter table that has been setup in the FORM.DBD file with the "res=" parameter.

- Select the table entry number and click the "get" button. You will see the parameters for that entry.

- Up to 9 types of resistors (1-9) can be entered in the table.

- The width parameter is compared to the width of the actual resistor and a warning is given and the measurement is skipped if different.
- A resistor is measured starting at its first vertex and ends at its last vertex. Any extended path ends are ignored.
- Each segment of a resistor is always measured vertex to vertex.
- Each bend in a resistor is compensated by subtracting the corner compensation resistance for each internal vertex from the resistance sum. The incremental resistance shown is the total resistance subtracted.

To set up the parameter table add the following to the FORM.DBD file for each type of resistor:

"res=sheetresistance endcorrection cornercorrection pathwidth"

"Sheetresistance" is the resistance (ohms) per square of the resistor.

"Endcorrection" is the resistance (ohms) used to compensate for BOTH resistor heads. The value may be POSITIVE or NEGATIVE depending on how you make resistor heads.

"Cornercorrection" is the resistance (ohms) used to compensate for EACH resistor corner. This value can be calculated, estimated, or best, empirically measured on actual circuits. The value is usually NEGATIVE.

An example of resistor corners and the corner correction can be found in the tutorial as cell RESCORNR.

"Pathwidth" is the width of the resistor path in physical units.

After any active path is found and measured, RES enters manual mode.

- You must manually enter the resistor width at the "<-->" command prompt. The width of the resistor is measured by a 2-point mouse coordinate input.
- At the command prompt "R-->", you then sum resistances by making 2-point coordinate distance measurements along the length of a resistor. The distance will be displayed by a dashed line, and the incremental resistance and the total resistance will be displayed at the bottom of the screen.

IMPORTANT: RES is a RESIDENT command, that is, it recalls itself. To end measuring, click the mouse on the menu button of another resident command. If you click on a non-resident RES will recycle itself.

NOTES:

1. To automatically measure a path, it must be a path in the present cell. Paths that are visible but are part of a cell within the present cell must be manually measured.
2. When automatically measured, if a path has zero width (polygon), ONE PHYSICAL unit of width will be the default, and the total length of the segments will then be measured.

HINT: This command can be used as a ratio calculator. For example, to measure a MOS transistor size, set the sheet resistance to 1 and measure the gate length as the resistor width and the gate width as the resistor length. If the device has more than one gate stripe, continue to measure and sum the gate widths by clicking at each end.

ReSize

RESIZE multiplies the basic units of all the position data in a cell.

- To change a single cell give the name of the cell.
- To change ALL cells in a collection give the name as "*".
- Changes are limited to multipliers of 0.1 to 10 at a time.

ReSz

RESZ (resize) changes the size of boxes and paths while still preserving their shape.

- The multiplier is 1 for no change, and is limited to ≥ 0.01 to ≤ 100 .
- The geometric center of the box or path is kept constant.
- If a small path is resized or if it is made too small, it may change shape due to the limited resolution of the unit grid.
- RESZ acts on boxes with ANY side active or paths with ANY vertex active.

Use RESZ to make different sizes of complex figures to save drawing time.

Read OVSZ.

Rot

ROT rotates objects.

- Rotation is around a single cursor input point.
- Boxes, text and cells rotate $\pm 90, 180$ deg.
- Paths rotate through any angle.
- Boxes rotate if ALL sides are active.
- ALL vertices of a path rotate if ANY vertex is active.
- Text and cells rotate if they are active.

NOTE: Sometimes a rotated path may not be exactly identical to an original if it has been rotated and then rotated back. This is caused by rounding coordinates to the unit grid each time a rotation is done.

Read FLP.

rWin

RWIN restores a drawing window.

- RWIN pops up a SWIN/RWIN dialog box.
- Windows are numbered 1-9.
- Windows should have been saved using the SWIN command.
- The restored drawing window is immediately redrawn.

Save

SAVE writes the current cell to the hard drive.

- The current version of the cell is written to the .BP6 and .CL6 files on the hard drive.

- The CELLS6.DBD and CONSTS6.DBD files are updated on the hard drive with the current cell outlines and object counts.

This is useful if you are working with one of the utility programs such as LASIDRC or LASICKT that is simultaneously running under Windows and you need to make a change to the drawing. All utility programs read from the hard drive, so that unless the drive is updated, the utility will receive an old cell version.

NOTE: Using the SAVE command rewrites the backup files as if the cell had been newly loaded.

Scale

SCALE is used to set the name of the physical scale units of the drawing and the number of basic units per physical unit.

Rules for setting scales:

- The scale units name can be anything, but for integrated circuits "um" and "mil" are preferred because some utility programs use these units.
- Only the first 5 characters will be accepted for the scale name.
- The number of basic units per physical unit will usually be an integer greater than 1, but fractional scales are permitted.

NOTES:

1. The scale that is set by this command really only relates the basic units in which a drawing is internally done to the inputs that you give to certain questions and to dimensions that you see externally. All inputs and outputs are converted by the scale to and from basic units.
2. LASI 6.0 should have plenty of basic units for any IC to be drawn. It is then somewhat more convenient to set the number of basic units/physical unit first. For example, setting the basic units to 20 or 0.05um resolution results in a "universe" of 838,861um or a little less than a meter.
3. There is a tendency to use too few basic units per physical unit. If you use too few units, you will be limited in the smallest text sizes you may have, because character sizes are multiples of 15 basic units. You may also find that arcs draw too coarsely. However you should only use as fine a resolution that you really need. If you use too much resolution you will slow down some of the LASI utility programs (LASICKT for example) that work at the basic unit level.

HINT: For most IC work (with current technology) 20-100 units/um seems about right.

sDrw

SDRW (sequential draw) redraws the drawing with the layers in ascending order.

- This draws layers sequentially. (1 -> 64) without regard to cell nesting.
- This shows overlays correctly, but is usually much slower than DRAW.
- This draws only viewed layers. To speed up drawing, only view layers that you want to see. Using the layer table works nicely.

This command is useful if you want to overlay solid fills so that you can check coverage by higher layers.

Read DRAW and RDRW.

Set

SET allows you to change certain parameters and the actions of certain commands. The SET dialog will pop up.

Presently these parameters can be set:

- The Size of the preset Working Grids in physical units.

- The Size of the preset Dot Grids in physical units.

Leave a size entry BLANK to have it ignored.

- The Resolution of the drawing.

Minimum size of a figure = size of drawing window / resolution.

ù Boxes are not drawn if both dimensions are less than the minimum.

ù Paths or poly are not drawn if the distance from the minimum vertex to the maximum vertex in both dimensions is less than the minimum.

ù Paths are drawn as lines if their width is less than the minimum.

ù Text is not drawn if its size is less than the minimum.

ù The dot grid is not draw if its spacing is less than the minimum.

ù Cells will not be expanded but will be drawn as dotted outlines if the outline dimensions are less than some multiple of the minimum.

A resolution of 250-1000 is about right for standard VGA. There is usually no need to make the resolution greater than the number of pixels across the drawing window. Occasionally, some small figures may disappear if the resolution is too low. If this is a problem set the resolution higher.

- TLCout when Saving Cells. This calls TLCout in the background and writes the TLC of the cell to the hard drive. This automatically keeps your cells up to date in TLC.

- Enable Global TLC. This allows you to set both the source and destination paths when using the TLCIN and TLCOUT commands. Unchecked the source or destination default to the current drawing directory,

Show

SHOW gives information about an object.

From the System Mode menu command, SHOW gives the following information about a cell:

- The cell's rank, number in the cell collection and if it is attached, indicated by the rank "1+".
- The cell's outline boundaries as found in the cell collection list.
- The number of cells, boxes, paths (includes text) and vertices in the cell.
- A listing of the named cell's lesser cells.
- The number of boxes and paths (includes text) on the layers.

In a drawing, SHOW gives information on the first active object that it finds in the order. box/path/cell:

- For a box, the positions of the sides and the layer are shown.

- For a path, the vertex positions, the layer, the width and an active vertex mark "*" are shown.
- For text, the reference point, the layer, the size and the characters will be shown.
- For a cell, the information is similar to the system mode SHOW, but also includes the position and orientation of the cell.

SmsH

SMSH smashes any active cells, text or boxes into their component objects and adds them to the current cell. The original active objects are deleted.

Rules for Smashing:

- Cells are smashed down one rank at a time. You may have to do multiple smashes if you want to reduce higher rank cells all the way to boxes, paths and text.
- Text is smashed into poly objects that replace the original text. These poly are the same as any other poly, and may be rotated to any angle or given width to make paths. The poly objects are derived from the current Text Font File.
- Boxes must have all sides active. A box will be replaced by a five vertex polygon with the same dimensions.

NOTE: If you try to SMSH any object that will cause any of your object limits to be exceeded, or if you run out of memory, the object will not be smashed but will remain intact and active. You will get a message for each object component that cannot be added. If you are smashing text or boxes you may get a partial poly conversion, but the original object will remain.

IMPORTANT: Before any object is smashed, the hard drive files MAINBKUP.BP6 and MAINBKUP.CL6 are updated. If you make a mistake and smash the wrong thing, you can use the UNDO command to restore the drawing.

Sort

SORT sorts and stores a backup of the drawing.

- Boxes, paths and text are sorted into 16 zones over the area of a cell.
- Cells are sorted by their position in the CELLS6.DBD file.
- BP6 and CL6 files of the current cell are updated on the hard drive.

You may stop a long sort by pressing the SPACE bar. Eventually, when objects are in good order, sorting will be fast.

NOTE: SORT also moves objects that have been accidentally moved out of the maximum drawing area (universe) back into the drawing. This is done by adding or subtracting the maximum window width until the objects coordinates are in the drawing. Boxes and paths can be distorted by this operation and will have to be repaired or deleted.

HINT: Sorting should be done periodically while working on a drawing. This not only cleans out any defective objects, but also improves drawing speed by ordering the cells.

Snap

SNAP moves box sides, vertices and cells to the nearest exact grid.

- The popup dialog will request a dimension in physical unit.
- The default grid is always the minimum grid, so that any accidental use of SNAP will produce no change.
- Each active box side is snapped independently to the grid.
- Each active vertex of a path is snapped independently to the grid.
- Active cells are snapped so that their reference points (0,0) are on the grid.
- Active text is snapped so that its reference point is on the grid.
- Before snapping, a backup of the drawing is made so that you can restore the drawing if you make an error.

IMPORTANT: SNAP is used to precisely grid a drawing. It normally is used to eliminate small errors in placement. It however can be dangerous if you make the snap grid too large. It can cause boxes and paths to be crushed if the nearest grid point turns out to be the same for more than one box side or vertex. Before any object is snapped, the hard drive files MAINBKUP.BP6 and MAINBKUP.CL6 are updated. If you make a mistake and incorrectly snap, you can use the UNDO command to restore the drawing.

Step

STEP steps active objects into an Nx by Ny two-dimensional array.

- The popup dialog will request Nx,Ny and the step distance (Dx,Dy) dimensions.
- The original active objects are copied repeatedly just as though the CPY command was used. The final copied objects remain active, while all previous copies become inactive.
- Nx and Ny are presently limited to 1024 max.
- Stepping can be ended by pressing Esc.

NOTE: Care should be used with this command because you can easily fill up your cell's object capacity if you don't watch your numbers. If this happens press Esc to end stepping.

sWin

SWIN saves a drawing window.

- SWIN pops up a SWIN/RWIN dialog box.
- Windows are numbered 1-9.

The windows saved are global, that is they are independent of the cell mode rank that you are in. The windows are stored in the CONSTS6.DBD file when you quit LASI, and are returned when you rerun the program.

Sys

SYS pops up the System Mode dialog box.

- System Mode allows you to do various drawing maintenance functions and to run utility programs from your drawing environment.

IMPORTANT: System Mode always stores the current cell, reloads it and resets any UNDO backup files. This is intentional because system commands can rewrite cell files or reorganize a drawing, which could make the current cell invalid.

Text

TEXT generates text objects.

- The command first asks for a text reference point. The command will then work in one of two ways:

. If the new reference point is exactly ON the reference point of any old text and the layer of the old text is OPEN, the old text object will be erased and its text string will appear in the input box at the bottom of the display. You can modify the text string as you wish. Any new text will have the layer, size and orientation of the old text.

. If the reference point is not on old text, new text will be generated.

- The text character string may be up to 56 characters terminated by Enter.

- The string of text should appear in the drawing made from polygons. The default layer and size of the characters will be shown at the bottom of the display when the TEXT command is resident.

- New text size and layer are set using the TLYR and TSIZ commands.

- The CLYR and CSIZ commands may be used to change the text layer and size after the text is made.

Text Properties:

- The style or font of text is determined by the Text Font File.

- Text is always left justified and appears on a single line.

- You can make the text active using the GET, TGET or FGET commands.

- You may copy, rotate, flip and move the text using the CPY, ROT, FLP and the several types of move commands.

- Text size will always be in integral multiples of 15 basic units. This is to prevent distortion of the characters.

- A text object is seen by LASI as a single point, the reference point, and behaves just like a one-vertex path object, except that it is not automatically deleted by SORT and DEL commands.

- The T key or "T" button toggles the reference point between visible (a small diamond, same color as text) and invisible. A depressed "T" button indicates that reference points are visible.

HINT: To make text characters that are to appear permanently on any IC masks, first set the font file to TFF2.DBD or TFF3.DBD, make text of the appropriate size and then SMSH the text into polygons. TFF3.DBD should be used on masks that will be generated using the LASILL program. The block lettering is better handled by that program.

Read the Text Generation help category for more information.

tGet

TGET (text get) makes text active.

- This works like GET or FGET, but operates only on text.

- The layer must be viewed and opened.

tPut

TPUT (text put) makes text inactive.

- This works like PUT or FPUT, but operates only on text.

- The layer must be viewed.

TLCin

TLCout

The TLCIN and TLCOUT commands convert between internal and TLC forms of cell data. The commands are a convenient way to move cell data in and out of a drawing without leaving the main LASI program.

- The TLCIN or TLCOUT commands first create a popup that let you enter the source and destination folders and a cellname. Click the "Help" button for information on the parameters required.

- Check Fast Run if you want to run TLCIN or TLCOUT at maximum speed. Normally, either program pauses at certain times so that you can see its progress.

- TLCIN and TLCOUT remember the last source and destination.

TLCIN and TLCOUT accept cellnames containing the wildcards "*" and "?" as the first named cell, and convert cells that match. Cellname matching works exactly the same as DOS or Windows matches wildcards to file names.

WARNING: Wildcards should be used with TLCIN if the source contains ONLY cells you want in your drawing. It is very easy to fill you drawing with unwanted cells that have to be removed individually using the KILL command.

Read the TLC Conversion help topic for more detailed information about TLC.

tLyr

TLYR (text layer) sets the default layer of text to be added.

- The value entered is clamped 1 through 64.

- If the layer table is on, click the layer box to set the text layer.

Read CLYR.

Toggles

Toggle buttons control special drawing functions. They depress to indicate that a function is toggled ON. Clicking a button with the LEFT mouse button is the same as pressing a key.

"W U" indicates that the working or unit grid is in effect. This resets after each command is completed. Pressing "W" or "U" sets, or pressing the "A" key toggles the setting of this button.

"C" indicates that dashed path center lines will be drawn.

"D" indicates that a small diamond shaped marker will be drawn at the end point when you measure distances.

"I" indicates that a cell's image will be drawn in addition to the cell.

"N" indicates that if a cell is being drawn as an outline, the name of the cell will be drawn in the lower left corner of the outline.

"O" indicates that octagonal cursor is ON.

"R" indicates that a 0,0 reference mark will be drawn.

"T" indicates that the small diamond shaped reference mark for text will be drawn.

"V" indicates that all vertices of a path or polygon will be marked. The mark is similar to the text reference mark and is drawn with the same color as the path. Only the paths in the current cell are marked not those in lesser cells.

Read General Information/Key Assignment for more key functions.

tSiz

TSIZ (text size) sets the default size of text to be added.

- The size will be adjusted so that it is a integer multiple of 15 basic units.

Read CSIZ.

uDup

UDUP (unduplicate) removes duplicate objects in a cell.

- Boxes are removed if the layers and sides coincide exactly.
- Boxes are removed if TWO or MORE sides coincide.
- Paths or poly are removed if they have the same layer and all vertices coincide in the same order.
- Sequential vertices with the same coordinates are deleted.
- Cells that have the same name, position and orientation are removed.

Undo

UNDO aborts the current cell and restores a backup.

- This command will restore the cell as it existed when the cell was last LOADED or before a MOV, QMOV, CMOV, WMOV, ROT, FLP, DEL, SNAP or SMSH command.
- There is presently only ONE level of undo.

NOTE: Using the SYS command to go to System Mode always reloads a cell and rewrites the backup files.

View

VIEW sets the layers that will be drawn on the display.

- Boxes, paths and text that are not viewed are blocked from the GET, FGET, WMOV, QMOV, PBEG and PEND commands.
- Any active boxes or paths will be drawn as bright white line segments or as bright white vertex marks regardless of the VIEW setting.
- Each cell rank may have different VIEW parameters.

VIEW works differently depending on the layer table being on or off.

Layer table off:

VIEW accepts layer inputs that can be single layer numbers in any order, or a dash (-) can be used to indicate all inclusive layers between the end layers. A dash alone will give all layers from 1 to 64.

Layer table on:

Individual layers can be viewed or not by clicking the layer check box. To temporarily view all layers click the All Layers box.

All 64 different layers can be displayed in two pages of 32. Click the ">>" or "<<" button to flip between pages.

NOTE: Ctrl-Enter toggles the layer table on and off.

Read OPEN.

vStp

VSTP (vertex step) steps an active vertex along a path or poly.

- There must be only one path (or poly) with one vertex active or the command will not operate.
- Each time the VSTP button is clicked the active vertex moves one segment.
- The active vertex normally moves forward in the path, however, if the Ctrl key is pressed while the button is clicked, the vertex moves backwards.
- The active vertex stops at the beginning or end of the path, depending on the direction.

This is a handy command. It can be used to determine the order of a path vertices, but can also be used to make vertices active (for moving or deleting) that are coincident with other vertices on the same path.

HINT: To step from beginning to end of a path, first use PBEG. To step from end to beginning, use PEND first.

Wdth

WDTH sets the default width of paths to be added.

- The pathwidth is clamped to internal limits.
- A width of 0 is acceptable. This produces paths of zero width or polygons (poly).

IMPORTANT: Paths with a POSITIVE width are drawn with their ends INLINE with their end vertices, while paths with NEGATIVE widths are drawn with their ends EXTENDED out half the width from the vertices.

EXTENDED paths should be used if the layout is to be converted to CIF.

If conversion will be to GDS either can be used.

Read CWTH.

wGet

WGET (window get) is a combination of GET and CGET.

- Cells, boxes, paths and text are affected the same way as those commands.

WGET is useful if you are MOVING large sections of a drawing because you can see just which objects are active before the operation. WGET makes single sides of boxes and single vertices of paths active, which are then pulled with active cells when MOVING is done.

Read AGET.

wGrd

WGRD (working grid) steps through the list of preset working grids.

- The working grids should have been entered using the SET command.

- The present working grid will be shown at the bottom of the screen.

- Each cell rank may have a different working grid.

NOTE: WGRD accepts a numerical argument if you click its button while holding the Ctrl key down. A dialog box will appear and you can set the dot grid to any temporary value.

Read SET and the General Information/Form File topic.

wMov

WMOV (window move) moves boxes, paths and cells.

- The sides of boxes are moved if they are ENCLOSED by the rectangular cursor window.

- Text and vertices of paths move if they intersect the cursor window.

- Cells are also moved, but only if the cursor window completely encloses the cell's area. The cell may be drawn fully or be in outline.

- The first two coordinate points form the cursor window.

- The next two coordinate points determine the distance to be moved.

- Box and path layers must be viewed and opened.

- When used with CMD or an F-key, WMOV needs only a single x,y pair for the move part of the command.

WMOV is useful for stretching large sections of drawings orthogonally, for selectively doing box side movements, and for quickly doing small vertex movements.

Xpnd

XPND expands the width of the drawing window.

- The new window width is expanded by a factor (2 or 100%).

- If you expand with the Ctrl key down, the window width is expanded by a smaller factor (10%).

- The display is redrawn.

Continuously XPNDing will eventually cause the window to expand to the maximum size permitted or the whole "universe".

Read DRAW and the General Information/Universe topic.

Zoom

ZOOM reduces the drawing window size.

- The new drawing window will be the best fit of the rectangular cursor window.

- The display is redrawn with the new drawing window.

- ZOOM works over a 1000:1 range.

HINT: To ZOOM to a minimum window at a certain position, click the mouse on that same position twice. This will zoom to the default minimum window size centered at the cursor position.

Read DRAW.

Text Generation

Introduction

Text is generated by LASI version 6.0 by storing strings of ASCII characters as though they were a type of path. A reference point is stored as the first vertex, and then subsequent vertices contain up to 8 ASCII character bytes. When a string of text is drawn, it is expanded much like a path, except that the characters are drawn in sequence from predefined polygon patterns. A string of text can be thought of as a single path segment, where the character size is equal to the path width and the length is determined by the number of characters.

Each character is defined in the Text Font File. TFF.DBD, TFF2.DBD and TFF3.DBD are the generic files supplied with the system. LASI 6.0 uses the same files as previous DOS versions. These files may have any name, but the DEFAULT name must be installed the FORM.DBD file using the "font=" parameter. Otherwise the default "TFF.DBD" is assumed. Read the General Information/Form File topic.

A text font file is a binary file which contains a single record of 128 bytes for each of 96 characters. The characters may be drawn and modified using LASI itself. This file is loaded permanently into memory when LASI is started, so that text drawing can be done without slow disk access.

When entering text, a position is requested and the size and layer are taken from the current text size and text layer default settings. The text string is then simply typed in and terminated by Enter. Like paths, text layer, size, orientation and position can be changed using the same commands that are used for paths.

The Text Font File

A text font file is ninety-six 128 byte records long. Each record corresponds to an ASCII character starting at decimal 32 (space) and ending at decimal 127 (delta). When LASI draws a character, the correct record is found and the information is used to draw polygons in the drawing.

The records may be randomly written using the WMAKETFF.EXE program by giving the name of the TLC cell file which contains the character drawing and then simply typing the desired character key.

The vertices of the path pattern are written in a single byte of a record in the form of the X-coordinate as the upper 4 bits and the Y-coordinate as the lower 4 bits (nibbles). The coordinates may then be only the numbers 0-15. The coordinate combination (15,15) is however prohibited because it is used to indicate the end of a path.

A record therefore contains vertex coordinate bytes from 0-254, with 255 decimal (or FF in hex) bytes marking the end of a path. The remaining bytes in a record are always filled with FF bytes to prevent further paths being made as the record is read.

The maximum number of vertices that a record can hold is dependent on the number of independent paths, but can be determined from:

$$\text{total no. vertices} = 128 - \text{total no. paths}$$

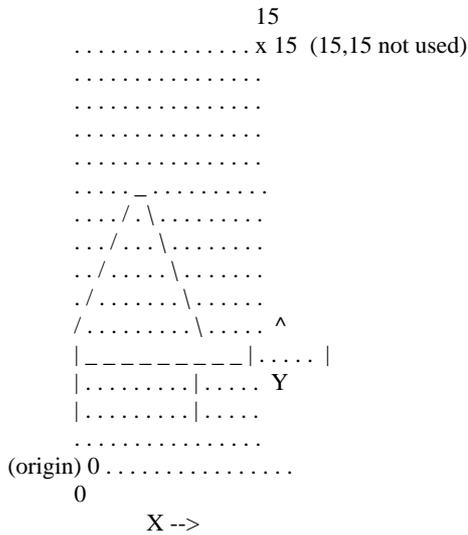
There can be any character drawing in the font pattern corresponding to a given typed character. In fact, different text font files (with different names) can be kept for different uses, and they can be conveniently "turned on" by naming them in the "font=" parameter of the FORM.DBD file.

The Character Field

Characters start as little cells containing a drawing of a character. The data format of the character drawing is quite

different from a cell. A character is drawn as a rank 1 cell using paths (0 width) on a 16x16 (0-15) field of basic drawing units. The paths may be on any layer. The lower left corner of the field is position 0,0. The vertices of the paths fall on the basic unit grid points. The vertices may be on any grid point except 15,15. The cell can contain boxes that can be used as drawing aids but will be ignored when the character is installed. For proportional character spacing to work correctly, the character drawing should be left justified.

23



Character Drawing Field in basic units (Letter A Shown)

Once a character cell is drawn, the program WMAKETFF.EXE is used to install the character in the text font file.

NOTE: The CHARS.ZIP file that should have been included with LASI distribution contains TLC files of all the characters in TFF.DBD, TFF2.DBD and TFF3.DBD.

Usefulness of Text

Text is usually used to mark your drawings with information such as the names of cells or component names and values. Text will be translated with the other drawing data into text in other drawing systems.

Text may also be used as a pattern for putting writing on IC masks by smashing the text using the SSMH command. The characters in the text font files TFF2.DBD and TFF3.DBD have been designed to smash into closed polygons that can be reproduced by most mask generators. The characters in font file TFF.DBD smash into open polygons that must be given width (with the CWITH command) to be reproducible. The TFF3.DBD file should be used with the LASILL program.

Text objects have a very useful property. Text can be considered as a NAME with a LOCATION. You can therefore use text to mark connection points to cells or subassemblies. You can then search for those points by NAME and obtain a physical location on a drawing. This can be used to do automatic cell placement and interconnection. This means that you can do actual layout from wiring lists used by SPICE type programs.

Inversely, you can also write programs that locate NODES by POSITION in LASI layouts or schematic drawings and associate TEXT with the NODE. You can therefore write wiring lists for SPICE directly from drawings that contain NODE NAMES and DEVICE NAMES written ON the drawing. The utility program LASICKT.EXE works this way.

TLC Conversion

Introduction

By design, LASI is a cell oriented CAD system, that is, each cell is more or less an individual entity. Standard cells can be "transported" between different drawings or layouts. Which is essentially the way ICs are made.

TLC (Transportable LASI Cell) is a form of LASI cell data that is used for drawing cell data storage and interchange. TLC files are ASCII sequential files written in a well documented form. The information in a TLC file is "line oriented", that is, you can write TLC with a text editor and you can read it in a simple line input procedure. This makes TLC files easily readable by programming languages including BASIC and C. It also makes them very tolerant to catastrophic errors since data can be fairly easily recovered or repaired by a text editor.

TLC is convenient for writing special programs to be used with LASI such as drawing transformation utilities or translators from other drawing systems. (WCIF2TLC.EXE, WFLATTLC.EXE and WGDS2TLC.EXE are examples.)

A TLC file contains information on the boxes, paths, text and cells in a larger cell. It does not contain information on how to make any cells that may be used within a cell. To build a complete drawing you need a COMPLETE set of TLC files.

Cell files in TLC are called EXTERNAL files, while cell files used by LASI directly (with ".BP6" and ".CL6" extension) are called INTERNAL files. TLC files always have the extension ".TLC". Internal files are designed for the computers convenience, while external files are designed for human convenience.

To bring a cell in TLC form into a LASI drawing, the cell must be "registered" into the drawing by TLCIN conversion. Using TLCIN is the ONLY WAY that cells can be installed in a LASI drawing from an external source. Internal files use a pointer reference to cells. This pointer is the position in the CELLS6.DBD file of a given cell's basic information, which normally cannot be changed without changing all cell references. TLCIN uses this referencing to install cells correctly.

Using TLCOUT to convert an internal file to an external TLC file removes this referencing by placing the actual cellnames in the TLC file.

Since each cell has its own TLC file, and a TLC file may not really contain all the information to construct a cell if it has lesser cells within it, the TLC files for the lesser cells will have to be present for conversion to internal form using TLCIN. When converting to external form, the TLCOUT is smart enough to make the TLC files of any lesser cells automatically.

When conversion from external to internal is being done, the lesser cells will also be made in internal form if they are not already present in the drawing cell collection. You may optionally replace any cells that are already there with new ones.

Packaging Cell Files

Some thought went into the question of making TLC files that contain a complete drawing. This may change, but it was decided that it was easy and efficient to just compress cell files into a Zip file using your favorite Zip program (PKZIP 2.5, 2.6, WinZip or Info-Zip).

You should put the TLC files for a drawing in a separate folder so they can be zipped all at once. This produces a Zip file containing a set of files that can totally reconstruct a drawing. The Zip file also compresses the TLC files to a fraction of their original size.

TLCOUT will create a folder if it does not exist. To create a folder just below your drawing directory, use the destination "./name".

Most new Windows Zip compressors allow a whole subdirectory of files to be compressed at once into a single Zip file. In general you should set any options so that only the files are placed in the Zip file, and not the directory structure (usually called recursion).

The Zip files may be used as drawing archives, or they may be sent by floppy disk or modem to other locations to share drawings between LASI users.

Running TLCIN and TLCOUT

Both TLCIN and TLCOUT run from the LASI drawing program system mode.

Clicking the TLCIN or TLCOUT button initiates a popup dialog. For a brief description of TLCIN and TLCOUT options, click the popup "Help" button.

Automatic TLC

Version 6.0 of LASI has a feature that lets you automatically converting internal files to TLC files. This keeps your TLC files up to date. The feature is optional and may be activated by setting the switch in the SET command dialog. LASI will write TLC files of a cell in the drawing directory when the cell is stored.

Attached Cells

When converting internal files to TLC files, TLCOUT will look for an ATTACHED cell that may be used in a drawing in the POOL directory as it is listed in the FORM.DBD file. If it doesn't find the cell, it will produce a token TLC file consisting of the name of the cell in text. It will not look for an attached cell in the drawing directory.

When converting from TLC files, TLCIN will put ALL cells in the DRAWING directory since it has no knowledge (nor should it) of any cells that may have come from a pool. Keep this in mind if you are converting into a drawing; any pooled cells may get replaced.

The CELLS.REF File

As TLCOUT makes it, the CELLS.REF file is just a copy the CELLS6.DBD file with cell outline parameters changed.

You may change the CELLS.REF file to convert modified lists of cells into internal form.

To remove a cell, delete the cell's name and the data line that follows it in the CELLS.REF file.

To add a cell, insert the cell's name on a single line and then the cell's rank on the next line. You don't need any additional data on the second line.

The entry on the first line of the CELLS.REF file is the number of cells in the file. If you add or delete cells, you should adjust this number.

When TLC converts to internal form using CELLS.REF, it counts either to the number of cells it reads on the first line or to the end of the file, whichever comes first.

The LAYERS.REF File

When TLCOUT makes a CELLS.REF file it also makes a LAYERS.REF file.

This is a useful list of the "layers=" statements copied from the FORM.DBD file in the source drawing directory. You can copy this list into your FORM.DBD file when reconstructing a drawing from a complete set of TLC files. The layers are never automatically inserted in the FORM file.

TLC Cell Outlines

The outline of a cell is normally not used when a cell is in TLC format. This is because TLC format presumes no knowledge of the details of other cells, and hence doesn't know how to include lesser cells in a cell's outline.

However, the TLC header record contains four entries that are the left, bottom, right and top boundaries of the cell outline. These parameters exist because the outline might be useful information once in a while. Normally, the outline of a cell is only expected to be correct when it is internal data that is used by the LASI drawing program.

By definition, these outline parameters MAY NOT and NEED NOT be correct in a TLC file.

It is the job of TLCIN to provide a correct outline when it converts a TLC file to internal. A new outline is ALWAYS recalculated from the object information in the TLC file and the lesser cell outline information in the drawing's CELLS6.DBD file.

When TLCIN converts several cells, either by name or from the list in CELLS.REF, it sorts by ascending rank, so that lesser cells have their outlines established first. This insures that a cell's outline is calculated from correct lesser cell data.

TLCOUT follows these rules somewhat in reverse. When TLCOUT writes a file, it puts cell outline parameters in the TLC file header record and in the outline parameters of the CELLS.REF file.

However, because TLCIN calculates outline parameters, when writing a program that creates TLC files, you only need to put token parameters in the header record. For example, GDS2TLC and CIF2TLC only write token parameters to the the TLC header record. This greatly simplifies writing TLC files, because finding the outline of a cell can be difficult if not impossible when not enough is known about the other cells in a drawing.

For future use, any token parameters written to a TLC file header should indicate an invalid outline with NO AREA. For example, Left=0,Bottom=0,Right=1,Top=0 qualifies.

Conversion Errors

It is possible that files may be missing, incorrect, or there may not be enough memory when converting. TLCIN and TLCOUT will try to construct a drawing as best as possible by making a TOKEN cell. A token cell is a cell containing only the name of the cell enclosed in double angle brackets ("`<<name>>`"), as text on layer 64. The rank of a token cell will always be 1, so that it will always appear in other cells. If a token cell is made, TLC will indicate the error, and continue as well as it can.

Since it has rank 1, the token cell will contain no other cells. However, if the error is corrected, and TLCIN or TLCOUT is used to overwrite the token cell, normal cell nesting will be constructed. The token cell may be thought of as a placeholder for the missing cell where it is used in other cells. TLCIN or TLCOUT will replace the token cell with the actual cell in the drawing cell collection, once the correct information is available to make the actual cell.

TLC Format

TLC Cell files have a simple record structure. There are presently five kinds of records: Header, Cell, Box, Path and Text.

The records for the objects in the cell are preceded by a "keyword" consisting of two characters: the first is "=" and the second is "H", "C", "B", "P" or "T" for Header, Cell, Box, Path or Text respectively. Each keyword is terminated by a newline <nl>.

A TLC reading program then determines the type of object to be made from the keyword and acts accordingly. If at some time in the future, other objects are added to LASI, similar code will be used.

The TLC format is designed so that data can be resynchronized if an error occurs. If an error is encountered during translation, it is possible that one or two objects may be incorrectly reconstructed. The Header record contains an object count. If this is compared to the actual number of objects made, any miscounts may be detected by a TLC reading program.

When writing programs that read TLC files in C, read the file line by line using fgets() and then parse the line using strtok(). Reading a line at a time allows you to resynchronize if there is a data error.

All records, including the Header, may be written in any order in a file.

Records in Detail

Header Record: <nl>= carriage return + linefeed <sp>= space

```
=H<nl>
Name of Cell (filename)<nl>
Version of LASI (literal)<nl>
Version of TLC (literal)<nl>
Basic Units per Physical Unit<nl>
Name of Physical Unit (literal)<nl>
Date of Cell Conversion (literal)<nl>
Time of Cell Conversion (literal)<nl>
Rank of Cell<sp>
Left Outline Boundary in basic units<sp>
Bottom Outline Boundary in basic units<sp>
Right Outline Boundary in basic units<sp>
Top Outline Boundary in basic units<nl>
Number of Boxes<sp>
Number of Paths<sp>
Number of Vertices<sp>
Number of Cells<nl>
```

NOTE: Versions must begin with a numeric character

Cell Record Entries:

```
=C<nl>
Name of Lesser Cell<nl>
Orientation (see below)<sp>
X Position in basic units<sp>
Y Position in basic units<sp>
reserved (presently zero)<nl>

Property Number: (0-15)
16 bit integer, all bits 0 except:

Bit 4,
0= draw cell fully
1= draw cell outline

Orientation:
Bit 3,
0= cell not flipped
1= cell is flipped in X before rotation

Bits 2 and 1,
0,0 = no rotation
0,1 = 90 deg rotation CCW
1,0 = 180 deg rotation CCW
1,1 = 270 deg rotation CCW
```

Box Record Entries:

```
=B<nl>
Layer of Box<sp>
X of Lower Left Corner in basic units<sp>
Y of Lower Left Corner in basic units<sp>
X of Upper Right Corner in basic units<sp>
Y of Upper Right Corner in basic units<nl>
```

Path/Poly Record Entries:

```
=P<nl>
Layer of Path/Poly<sp>
Width in basic units<sp>
No. of Vertices in path/poly<nl> (512 max.)
Vertices in basic units in the form:
```

```
X1<sp>Y1<sp>X2<sp>Y2<sp>X3<sp>Y3<sp>X4<sp>Y4<sp>X5<sp>Y5
<nl>
```

```
.....Xn-1<sp>Yn-1<sp>Xn<sp>Yn<nl>
```

(Groups of 5 coordinates separated by <sp> ended by <nl>, always with <nl> after Last Coordinate Pair)

Text Record Entries:

```
=T<nl>
Layer of Text<sp>
Size in basic units<sp>
No. of Vertices used by text (includes Ref Point)<sp>
Orientation 0-7 (same as cells)<nl>
X Ref Point in basic units<sp>
Y Ref Point in basic units<nl>
ASCII character text string<nl> (u/l case, up to 64 characters)
```

NOTES:

1. The keywords must always be followed by a <nl>. No additional characters are permitted on the line.
2. Keywords must be unique, any program reading TLC must be able to scan a file, line by line, and read all keywords correctly.
3. In text records, the text strings "=H", "=C", "=B", "=P" and "=T" are not allowed since they could be read as keywords. In the very rare situation that these strings occur, a LEADING SPACE character should be inserted.
4. You can add comment lines BETWEEN records and not corrupt a file.

System Contents

System Contents

```
README.TXT    Installation instructions
SETUP.EXE     Installation program
WLASI.EXE     LASI main drawing program
WLASIDRC.EXE  LASI design rule checker utility
WLASICKT.EXE  LASI SPICE circuit file compiler
WLHI.EXE     LASI help and information reader
WFLATTLC.EXE  Flattening and smashing utility
WLASIUUE.EXE  LASI version of UU encoder/decoder
WLASILL.EXE   Layer Logical operations and reversal utility
WMAKETFF.EXE  Text Font File builder utility

WTLC2GDS.EXE  TLC to Calma Stream (GDSII) converter
WGDS2TLC.EXE  GDSII to TLC converter

WTLC2CIF.EXE  TLC to CIF converter
WCIF2TLC.EXE  CIF to TLC converter

WTLC2DXF.EXE  TLC to AutoCAD DXF converter
WDXF2TLC.EXE  AutoCAD DXF to TLC drawing extractor

LASI60.LHI    Main help and information text file
```

LASIDRC.LHI LASIDRC help text file
 LASICKT.LHI LASICKT help text file
 LASILL.LHI LASILL help text file
 FLATTLC.LHI FLATTLC help text file
 CNVRTGDS.LHI GDS conversion help text file
 CNVRTCIF.LHI CIF conversion help text file
 CNVRTDXF.LHI AutoCAD DXF conversion help text

TFF.DBD Generic text font file (line characters)
 TFF2.DBD Generic text font file (polygon characters)
 TFF3.DBD Generic text font file (block characters)

TUTOR.ZIP Tutorial files collection
 CHARS.ZIP Text character cell collection

LASI files that may be included:

MCLAYER.EXE Layer mover/copier
 MCLAYER.C Sample source code
 SITES.TXT LASI Internet sites
 LASIDSS.LHI Data structure standard help text file

Non-LASI files that may be included:

CKSUM.EXE MOSIS checksum calculator
 UUEN.EXE MOSIS compatible binary file encoder

Tutorial

Starting the Tutorial

The layout of a small IC is used to familiarize a new user with LASI. The circuit is a simple bipolar opamp, but not necessarily a good one. It is mainly designed to show drawing techniques.

When you installed the drawing files from a ZIP file, you should have created a subdirectory on your hard disk called \LASI6\TUTOR. This is the drawing directory for the tutorial IC.

If you used SETUP to install LASI, SETUP will have created a "Tutor" Start Menu item under "Lasi 6.0".

If for some reason this has not been done, create a shortcut on the Windows Desktop with a "Target" of "drive:\LASI6\WLASI.EXE", and with the "Start in" directory of "drive:\LASI6\TUTOR", where "drive" is the hard drive where you installed LASI.

If you are not running LASI on the tutor layout, close this help screen and restart the tutor from a shortcut above. Restart the LASI main help system by clicking the Help button and go to this tutorial topic again. The nice thing about running under Windows is that you can restart LASI help more than once and flip the helps from the Task Bar. You can start help from the Help button, or you can use the command button context help by holding down F1 and clicking a LASI button. Just don't open too many successive help screens.

When in the help system, read a few of the topics. If you are new to LASI, don't expect to learn the details immediately. There are many subtleties in LASI.

Basic Transistor Cells

The opamp is made from bipolar transistors based on a simple bipolar IC process. There are only 8 masking layers used, which are in the usual terminology:

- 1 N-type Buried Island
- 2 P-type Isolation
- 3 N-type Sinkers
- 4 P-type Base
- 5 P-type Resistor
- 6 N-type Emitter
- 7 Contact
- 8 Metallization

The schematic is on layer 32.

Layers 60-63 contain text that is to be used by the WLASICKT.EXE program which constructs SPICE code from the layout and schematic. To learn more about this run LASICKT and click the Help button.

There are different types of NPN and PNP transistors, some resistors and miscellaneous bits and pieces. These are all made as rank 1 cells.

Going to a Cell

1. If you are not there already, load the NPN1 transistor by clicking your LEFT mouse button on the LIST command and then double click the LEFT mouse button on the name "NPN1" in the listing. The cell should appear.

The cell is made from "basic objects": boxes, polygons and text.

If the transistor does not show completely in the display window click the FIT command button with the LEFT mouse button.

2. Use the commands ZOOM, XPND and the pan arrows to examine the cell more closely.

3. There are two menus of commands that can be flipped by clicking the RIGHT mouse button. The commands are always selected by clicking the LEFT mouse button on a menu button. The LEFT mouse button is always used to make cursor points or windows to input graphical information. Flip the menus to see the different commands, and if you are curious, hold down F1 and click a command button. This brings up the help for that command.

The Grid Systems

When you work with LASI you will discover that your mouse cursor jumps between two grid systems: the unit grid and the working grid. This may seem confusing, but its purpose is to control the placement of drawing objects to certain discrete positions. As we know, IC layouts are not really drawings, but are precisely placed items.

You use the SET command to program working grid sizes, and then click the WGRD command to step through them. The number and size is up to you. This tutorial is set up for 3 convenient sizes of 1um, 5um and 25um. The working grid is indicated at the bottom of the display.

The DGRD and GRID commands control a grid of white dots on the screen. DGRD steps through a set of dot spacings, and GRID turns the dots on and off. The dot grid spacings are also programmed by the SET command. The dot grid is a drawing aid and has no control over object placement.

HINT: Pressing the A key flips the mouse cursor between working and unit grids.

The Layer Table

The tutorial has been setup to use a feature of LASI6, a Layer Table. The table is a list of layers by names and numbers (set in the FORM.DBD file). With certain commands, the table appears in a dialog box. When the table appears, click the check boxes with the LEFT mouse button.

LASI 6.0 also works with layer numbers, just as did previous versions of LASI. To toggle between the layer table and numbered layers, press Ctrl-Enter.

Getting Objects

1. You will find that the transistor is mostly drawn with box

objects. First use the GET and FGET commands by making a cursor window that crosses a box side. Notice that the command and the points are indicated at the bottom of the screen and that the cursor turns into a dotted rectangle. When you use GET, a side of the box turns bright white. This is called "active". Using FGET (full get) makes the entire box object active.

2. Click the INFO or SHOW commands if you want information on an object. SHOW is like INFO except that it gives more information on a new screen.
3. The base of the transistor is made from a poly(gon) object. To "get" a poly object think of it as a series of vertices and make a cursor rectangle around one or more vertices. A diamond symbol will appear at an active vertex, and a segment line will turn bright white if adjacent vertices are active. GET makes a single vertex active. FGET makes an entire poly or path active. Use the INFO and SHOW commands again on the poly.
4. Try the PUT and FPUT commands. The "put" commands are the inverse of any "get" commands.

HINT: You can start a command over again by relicking the command or by pressing Esc.

Moving Objects

1. Now try the MOV command. First make one or more box sides active with GET and then click the MOV command. Notice that the mouse cursor now jumps in discrete working grid distances and requires two input points. Notice also that the cursor turns into a dotted vector to indicate the movement.
2. Click the APUT (all put) command to make boxes inactive again.
3. Make a vertex active on the base poly. Try MOV again and notice that vertices move discretely in any direction.
4. Click APUT again to make all objects inactive.

Adding Objects

1. You can now try the ADD command. To do so, first set the type of object using the OBJ command. Double click on "box". Make your boxes by clicking at diagonally opposite corners. The boxes will be on the layer indicated at the bottom of the screen. Notice that boxes will be "snapped" into the working grid indicated at the bottom of the screen.
2. Use the LAYR and CLYR command to set and change the layer of boxes. These commands call the Layer Table if it is turned on. For CLYR to work a box must have all four sides active.
3. Make polys by setting the object to "poly/path" and clicking the cursor at different points to add vertices. You will produce a sequence of line segments with the last vertex active and marked by an "x". By adding the last vertex at the same point as the first vertex, you produce a closed polygon. Notice like boxes that the vertices are set at working grid points.
4. Try the WPTH and CPTH commands to set and change a poly's width. Setting width greater than zero makes a poly into a "path".

In general, the term "path" refers to both poly and paths; a poly just has zero width.

5. There are special commands ARC, VSTP, PREV, PBEG, PEND, VSTP, CUT and JOIN that act on paths only. You can try these if you wish

after you have determined their function by consulting help.

You would use closed polygons to make non-rectangular areas in a transistor. (In this case, a base area with clipped corners, or PNP emitters and collectors as you will find in the PNP1 cell)

6. Now add some text. Choose the TEXT command and click a reference point, and then type in some characters terminated by Enter. Flip the menu and notice that the LAYR and WPTH commands are replaced by the TLYR and TSIZ commands. Text is a form of path, and the TLYR, CLYR, TSIZ and CSIZ work on it. Size for text is the same as character height. Text is like a one vertex path, you "get" text by its reference point, which is always in the lower left corner of the character string. You can toggle this reference point on and off with the T key.

More Commands

1. You should try the CPY, ROT and FLP commands on the boxes, paths and text that you have made. These commands save time when symmetries occur in a drawing.

CPY will copy individual active vertices, so you can copy pieces of paths. Boxes copy only if they have all sides active.

ROT will rotate boxes paths and text. Boxes and text rotate in 90 deg increments. Paths rotate to any angle.

FLP flips boxes, paths and text around a horizontal or vertical axis.

2. You can also try the WMOV and QMOV commands. These are composite commands that combine the actions of "getting" and "moving". The action of these commands on boxes, paths and text is the same except that the WMOV command requires that the cursor window completely enclose a box side. (WMOV also acts on whole cells as you will see later.)
3. There are two commands, VIEW and OPEN that restrict the layers which are shown on the display and limit the action of commands on different layers. These commands call the Layer Table or accept a series of individual layer numbers. Layers that are not VIEWed are never OPENed.
4. There are additional commands like AGET, WGET, STEP, SMSH, RESZ and OVSZ that act on basic objects. These are more rarely used and you will find their purpose once you have worked with LASI for a while.

Cleaning Up

1. You should use the DEL command to delete all of the boxes that you have added yourself. To delete a box you must have all sides active. Remember that the FGET command "gets" all four sides.
2. Delete all the paths that you have added. When deleting paths you may delete vertices individually if they are active, or a complete path if all vertices are active. Remember, FGET will make all vertices of a path active if a single vertex is made active.
3. Delete any text that you may have added by first making it active.

HINT: If you have made a mess of the transistor while trying commands, instead of trying to delete and repair, use the UNDO command. This will restore the original version of the cell.

More Cells

Use the LIST command to go to some of the other cells and see how

they are made. Remember that the cells are intended to demonstrate drawing, not necessarily good device design.

Going to the Main Cell

1. Click the LIST command and load OPAMP. The overall IC layout should appear. Notice that it is rank 2 and is made from the rank 1 cells along with boxes, paths and text. The transistors are set in convenient locations and are interconnected using paths on the metallization layer.

This is basically the way that you would layout an IC, although the techniques vary with the individual.

2. Use the FIT, ZOOM, XPND and direction commands to examine the layout.

The opamp layout is a very simple example. More elaborate ICs may be constructed by using subsections (like our opamp), and then assembling these subsections into higher rank cells... and so on. This is how LASI can handle large if not huge ICs.

Cell Commands

1. Try some of the commands that act on cells only. Select the CGET command and rapidly double click the LEFT mouse button on a single transistor cell. The cell should redraw in bright white. This is now an active cell.
2. Use the MOV, ROT and FLP commands on the cell. Cells rotate in 90 deg increments.
3. Make the cell inactive again by clicking the APUT button.
4. Try the OUTL command by rapidly double clicking the mouse on a cell. The cell should be replaced by an outline. Now try the FULL command on the cell by making a cursor window completely around the cell. Outlining allows you to draw cells as a simple rectangle outline only. Putting cells in outline can greatly improve drawing speed if you have a large complex layout and a slow computer.
5. Also try the CMOV command which moves cells directly. Rapidly double click on the cell and then click two points to move the cell. This too is a time saver.
6. There are commands that work on both cells, boxes, paths and text such as AGET, WGET and WMOV. These do more things at a time, but are also more difficult to use. Try these commands. You will find that they are useful for operations like moving a whole section of a drawing.

HINT: By quickly double clicking the LEFT mouse button while making a cursor window in cell commands, you change the sense of the "get". The commands then act on cells if there is any overlap of the cell in the cursor window at all. If you wait a moment between clicks, the "get" window must completely surround a cell.

To reverse this "get", hold down the Ctrl key when clicking.

Adding Cells

Now you can add a cell. To do so:

1. Click the OBJ command button and make the object the name of a cell such as NPN1.
2. Select ADD and click the LEFT mouse button in some open location to give a reference point. Your cell should appear and you can work with it just like any other cell.

Note that the reference point is the 0,0 point in the added cell.

Deleting Cells

When you want to eliminate a cell, simply make the cell active and click DEL to remove it. Once a cell is gone, it is gone, but cells are easy to put back.

As with lesser objects, you can also use UNDO to restore the original drawing, or reload it completely from TLC form.

The Schematic

There is a schematic diagram named OPAMP_SCH independently living with the IC layout. It is similar to the IC layout except that the cells are symbols that are drawn and interconnected using zero width paths or open polygons, whichever way you may want to think of them. Different layers are used for the schematic and the layout, so that there will be no conflict with the layer attributes of color, dashing and filling.

Notice also that in this example schematic cellnames have "_SCH" as their last characters so you can easily tell a symbol from a device.

There are in effect two main cells, one for the IC, and one for the schematic. There actually can be as many different cells on a rank as memory capacity will allow.

You may want to use the schematic as a practice for working with open polygons. You will find that the techniques are slightly different from working with the IC layout.

Summary

You have been introduced to working with LASI. There are many more things that you can do using the many commands, and many subtleties in the way that the commands work that you will only learn by experience. When help is needed click the help button, or for a particular command, hold down F1 and click the command button.

You will soon learn that LASI is really pretty simple in the way that it operates.

Exit LASI by clicking the Windows Close button "X". This stores the current cell on the hard disk, and updates the drawing basic data files.