DOCUMENTATION FOR THE CELL MATRIX™ LAYOUT EDITOR

Nicholas J. Macias, Cell Matrix Corporation

November 2001

Copyright © 2001 Cell Matrix Corporation. All Rights Reserved

1. Introduction

The Cell Matrix Layout Editor is a graphical program for designing Cell Matrix circuits. This document describes the basic use of this editor for creating Cell Matrix circuits. It assumes that the reader is already familiar with the Cell Matrix architecture. More information can be found at http://www.cellmatrix.com

The best way to learn how to use the editor is to experiment with it. This document is therefore relatively brief. It is ordered so that as you read it from the beginning, you get progressively deeper into the editor's use. You should try running the editor and working with the concepts below. There are sample files included to get you started.

The section labeled "Getting Started" should contain enough background information to let you get the editor running. After that, you may work through as much of the document as you like, or throw it out and just experiment. However, the section labeled "Important Details" should be read carefully.

All names and symbols are trademarks of their respective owners.

2. Getting Started

The layout editor (also called the "loader") is a Java program. It has been tested and works well well under Windows 98. Unfortunately, it does not work as well under Linux (font and color problems). I will try and make a Unix-compatible version shortly, but for now, you probably need to run it under Windows.

The attached .EXE file (Portable Loader.exe) contains all the necessary pieces of the layout editor. If you double click Portable Loader.exe and then click "Extract," it will, by default, create the directory "C:\PORTABLE LOADER" and extract the necessary pieces into that directory. If you wish to use a different directory, you may specify one, but you will need to perform some extra steps.

To run the layout editor, you will need the Java Runtime Environment. You will need at least Java 2. Sun's runtime environment, j2re-1_3_1_01-win.exe, is included with this distribution. Double clicking on that file will begin the installation process. Note that when installing this runtime environment, you must agree to the terms and conditions of Sun's licensing agreement, including any export controls placed on the runtime environment. For further information, please visit http://java.sun.com/ and search for J2RE

The archive "Portable Loader.zip" contains the following pieces:

Cell Matrix Layout Editor.PDF	This Document
loader.bat	Main program. Double click to run.
loader.jar	The Java Archive containing the loader code
Acme.jar	Java classes from ACME Laboratories (http://www.acme.com) Used for creating hardcopies
samples	Directory containing sample grids (*.GRD) The ALU and MEM subdirectories contain grids which may be found on the Cell Matrix Website.
libraries	Directory containing libraries (*.LIB). Includes some sample libraries. **NOTE**: You may create subdirectories under this directory, but all libraries must be placed in or beneath this root. **If you intend to share designs and libraries with others,** it is recommended that you place your libraries under a subdirectory with a unique name (such as your own name).

palettes	Directory containing sample grids (*.PAL). These grids contain useful cells, but not in any particularly useful arrangement. They are intended as a convenient place from which to copy cells into other grids. Hence they are called “palettes” instead of grids, though there is no real difference.
j2re-1-3-1_01-win.exe	Sun’s Java 2 RunTime Environment

If you unzip into a directory other than “C:\PORTABLE LOADER” you will need to change the loader.bat file. Currently, loader,bat looks as follows:

```
java -classpath loader.jar;Acme.jar -DLIBROOT="C:\PORTABLE LOADER\LIBRARIES\\" loader
```

The LIBROOT argument must specify where your main “library” directory is located. **The format of this argument is critical!** It must be entirely uppercase, beginning with the drive specification, and must use single backslashes (“\”) throughout the path. The end of the path must have two backslashes (“\\”) Everything else is the command should remain unchanged. Again, it’s easiest to unzip into the directory “C:\PORTABLE LOADER”

To run the loader, simply double-click on the loader.bat icon (or run it from a DOS shell). It will take a few seconds for the program to start. The first time you run the loader, you will see a license agreement. Once you accept the agreement, it will not pop up again.

At this point, you should see two Java windows (and possible a DOS window, which you can minimize if you like). The window labeled “Cell Matrix Layout Editor” is the main window in which you will initially work. The window to its left contains informational messages. You can now begin experimenting with the editor, or read further for more information (but make sure you eventually read the “Important Details” section below).

3. Important Details

The following are some important points to note and keep in mind about the Layout Editor:

- This is a very preliminary version of the software. It was designed primarily for internal Cell Matrix Corporation use only. We are making it available as-is, with no guarantees on its behavior. We have had much success in using it internally, but it may frustrate you beyond belief! We are happy to receive feedback, comments and questions (write to us at support@cellmatrix.com), but at this time, we have limited resources for fixing problems, etc.
- The program is rather ruthless. If you ask the program to exit without saving your work, it will do just that. It will not ask if you’d like to save your work first. If you try to close the main window, it will go away, even if you have changes which you haven’t saved. Of source, if the program crashes (which is rare but does occasionally happen), you will lose any unsaved changes. Therefore, **save your work frequently!**
- The program will mostly do whatever you ask it. If you ask it to save a binary image in place of your .GRD grid file, it will happily do so (it will warn you the file exists). If you try and load a non-grid file as a grid, or a non-library file as a library, it will happily do that too (but probably crash as a result). Therefore, be careful what you ask the program to do…
- The program currently works only under Windows. If you really need a Linux version, please let me know. It has worked in the past under Linux, but only with some explicit font and color specification.
- You will need the Java Runtime Environment (JRE) in order to run this program. I have included a copy of Sun’s JRE in this distribution, but have not tried installing it myself, so I have little experience with it. Again, if you’re having trouble getting JRE to work, you can try writing us, but we may not be able to offer much help.
- It is best to unzip the distribution in a directory named C:\PORTABLE LOADER If you prefer a different directory, you will need to edit the loader.bat file (see above for details)

- When entering Boolean expression to describe a cell's behavior, remember that all expressions are entered in Reverse Polish Notation (RPN). In this notation, operands appear before operators. So saying "N and S" would be written "N S and" (or, more precisely, "N S &") Negation (represented by "~" or "!") looks backwards: "Not E" would be "E !" or "E ~" If your cell description is not compiling, make sure you're using RPN.
- Inside a cell's source code, spaces are generally ignored, as is case.
- Grids are stored as references to cells inside libraries. If a cell changes within a library, any grids which use that cell will immediately change.
- Libraries are ordinary files, and therefore subject to the normal filename rules. Case is ignored in library names.
- Cell names can be anything you like (as long as there are no unprintable characters inside). Cell names can contain spaces, special characters, and both upper and lower case. Case is significant.
- Cells have icons associated with them, and there is an icon editor included in the layout editor. Icons are extremely useful for reading grids. In general, you should stick to simply, self-explanatory icons. There is an input/output placement guide available in the icon editor to help you line things up.
- If you're planning on combining multiple cells (using the Edit/Or-Paste command), it is useful to use a white background.
- When selecting or specifying libraries, **do not move above the root of the library tree!** Normally, this root is located at C:\PORTABLE LOADER\LIBRARIES Therefore, while you can create subdirectories under that point, all your libraries should be in a path beginning with "C:\PORTABLE LOADER\LIBRARIES"

4. Cells, Libraries and Grids

The editor is based on the notions of *cells, libraries* and *grids*.

A cell corresponds to a single cell inside a Cell Matrix. A fixed topology of two-dimensional, four-sided cells is assumed in this editor. A cell's behavior, i.e., how it maps D inputs to D and C outputs, is specified by the cell's *source code*. The source code can also contain comments, as well as a cell description which can be read by the simulator. A cell's source code may also explicitly specify the bits of the cell's truth table. Finally, a cell usually has an *icon* associated with it. This icon is displayed by the layout editor to show where the cell has been placed.

A library is a collection of cells. Libraries are basically flatfiles, and generally have a .LIB extension, though any name is acceptable. All library names are converted to uppercase. Therefore, "LIBRARY.LIB" "library.lib" and "LiBrArY.LiB" all refer to the same library. Libraries contain the master copy of each cell. If a cell's definition is changed inside a library, any circuits which use that cell will be changed accordingly. **This is an important point to keep in mind!** If you create a cell X, use it in a design, and then want to use a different version of that cell Y in a new design, you should create a new cell. If you change X to Y, any designs which use X will now be using Y.

A grid is a rectangular arrays of cells. The cells within a grid come from libraries. Cells can be edited, copied, cut, pasted, rotated, or combined. Of course, grids can be written to and read from disk. Grids can also be converted to binary files, which can be read by a Cell Matrix simulator. Creating a grid is a matter of creating or copying the cells which make up small sub-circuits, and placing those cells in the appropriate positions in an empty grid. These sub-circuits can similarly be copied and placed to create larger circuits, and so on. Note that while this has a natural hierarchical structure, grids are essentially flat objects; once a cell has been placed on a grid, there is no notion of whatever circuit that cell may have previously belonged to. Grids are also flatfiles, and generally have a .GRD extension. A grid file is nothing but an ordered list of library/cell pairs, indicating which cell(s) is/are placed in which spot in the grid.

5. Main Window

The main window is labeled "Cell Matrix Layout Editor." It appears as a blank grid of squares. Each square corresponds to a cell inside a Cell Matrix. If you move the mouse over the grid, the title of the window will change to reflect the coordinates of the cell you're pointing to. All coordinates are ordered as [row,column], i.e., the first number is the number of cells down from the top, and the second number is the number of cells over from the left.

If you left click on a cell, a blue box will appear around that cell. That cell is now the "selected" cell. If you click and drag over a series of cells, you will select a region of cells.

The initial grid is 32x32 cells. Only a subset of those cells may be visible in the main window. You can scroll across the grid using the scrollbars. Alternatively, if you click and drag to select a region and move past an edge of the window, the window will automatically scroll. Note that there is a built-in acceleration to this scrolling-the longer you remain past the edge, the faster the window will scroll.

6. Description of Menu Items

There are three main menus inside the main window: File, Edit and Cell. Some commands have keyboard shortcuts, which are displayed in the menu itself.

FILE MENU
The File menu is for manipulating files related to the loader, as well as for basic manipulation of the main window.

File/New Grid
This item creates a new main window containing an empty grid. You must specify how many rows and columns the new grid should contain. At any given time, you may have one or more main windows open. Copying from one window will affect what is pasted in another window, i.e., there is a single cut/copy/paste buffer. If you have only one main window open, and you close it, then program exits.

Note that each window has its own set of UNDO buffers, and it's own selection. Each window also has its own file history. For example, if you select a cell from a library, then the next time you try to select a cell (from the same main window), the library will default to the previously-selected one. If, however, you select a cell from a different main window, it will have its own default library. This applied to other file operations as well (unfortunately, this does not work perfectly, and you sometimes find yourself continually re-specifying directories or files-sorry!)

File/Open Grid
This item opens a previously saved grid file. Sample files are located under the "samples" directory (look for files named *.grd). Opening a grid creates a new main window.

File/Save Grid
This writes the current window's grid to a file. If you have never specified a name for this grid, this command functions like the "Save Grid As" command.

File/Save Grid As
The loader prompts you for a file name, and then saves the grid in that file.

File/Hardcopy Grid
Experimental command. Creates a file "c:\temp\gridimg.ppm" which is a Portable Pixmap version of the current window's grid.

File/Print Current Window
Prompts for the name of a .gif file to store an image of the current window. Note that this will create an image from the window as you see it, i.e., whatever set of cells are currently visible in the window.

File/Image to File in Pieces
Useful for printing larger grids. When you specify a filename "x", the loader will create a series of GIF images called "x0.gif" "x1.gif" and so on. Each image contains approximately 50x50 cells.

File/Toggle Grid

Also somewhat experimental. Turns the grid on and off. Selecting cells with the grid OFF does slightly weird things. Useful for making nice printouts though.

File/Write Binary Grid
Creates a .bin file from the current grid. The entire grid will be written, regardless of how many cells are actually in use. If you want a smaller .bin file, use File/New Grid and specify fewer rows and/or columns, then cut and paste into that smaller grid.

File/Zoom In
Makes the displayed cells larger. May produce funny results. If so, minimize and then restore the window.

File/Zoom Out
Makes the displayed cells smaller. May produce funny results. If so, minimize and then restore the window.

There are four levels of zoom. At the largest, each cell is 32x32 pixels. At the smallest, each cell is 4x4. Thus, on a 1024x1280 display, you can display close to 256x256 cells (over 65,000) in a single window.

File/Close
Closes the current window, without question. If this is the last main window, the program will then exit.

File/Exit
Exits the program immediately, without question. All open windows will be closed, even if your work has not been saved.

EDIT MENU

The Edit menu contains commands used mainly for manipulating cells within a grid. These are the commands you'll probably use the most in laying out a circuit. All of the commands have keyboard shortcuts, which greatly speed up the editing of designs.

Edit/Copy
This command stores a copy of the currently selected cells in the global *copy buffer*. The copy buffer can then be pasted into another location on the grid, or into another main window's grid.

Edit/Select All
This command selects all cells within the current window's grid. Identical to clicking one corner and dragging to the other, but much easier.

Edit/Cut
Similar to Edit/Copy, but this command also deletes the selected cells from the grid

Edit/Paste
Copies the selected cells from the copy buffer into the current grid, as follows:

- If no cell is currently selected, then this command does nothing
- If a single cell is currently selected, then the entire copy buffer is copied into the current grid. The upper left corner of the copy buffer is placed at the selected cell.
- If two or more cells are selected, then the copy buffer is copied into the selected cells. The selected shape need not match the shape of the copy buffer. If necessary, the copy buffer's contents will be repeated in both directions to fill the selected area.

Note that any cells already in the grid (and within the region to be pasted onto) will be overwritten by the cells from the copy buffer (but see the next command for another option).

Edit/Or-Paste
Identical to the Edit/Paste command, except that cells already on the grid are not overwritten. Instead, they are combined with the cells being pasted, creating a *composite cell*. The exact method of combining two cells is to simply logically-OR the bits of their truth tables. The effect of this is to overlap their functions. **NOTE:** In general,

it only makes sense to OR two cells whose outputs are independent of each other. For example, if two cells both specify non-zero outputs to their North, then ORing them probably doesn't make sense (this is called an *Output Conflict*). If, however, one cell outputs to the West and the other to the North, then ORing them creates a composite cell with outputs to both the West and the North (and those outputs are specified by the respective functions of each cell).

Up to 9 cells may be ORd in this way, though in reality, 3 or 4 is the most you're likely to use. Composite cells can be selected, copied, cut and pasted, but can not be edited (see Edit/Explode Composite Cell below).

When writing a binary simulator file from your grid, if any composite cells have output conflicts, a warning will appear in the informational window.

When cells are ORd in this way, their icons are also ORd in what is usually a reasonable manner. As noted above, using a white background for icons seems to work best in creating composite cells.

Edit/Rotate
This command rotates the selected cells by 90 degrees clockwise. If multiple cells are selected, they are each rotated. The icon is rotated, as is the cell's truth table. Thus, for example, a cell with equation "Data North=West And South" will rotate to "Data East=North and West"

Edit/Region Rotate
Similar to the Edit/Rotate command, this command not only rotates individual cells, but also rotates their position within the grid, i.e., the rectangular region currently selected is itself rotated 90 degrees clockwise, and the re-positioned to preserve the location of its upper left corner. This command is best understood by experimenting with it.

Note that if the region being rotated is not square, then the shape of the region will change following this command. This will sometimes have the undesirable result of erasing cells outside the selected region. This would happen if, say, you're trying to rotate a non-square region 180 degrees by using this command twice. If this is a problem, first open a new main window, paste the region into their, then rotate it in that new window, copy it back, and paste it into the original window.

Edit/Explode Composite Cell
If a single composite cell is selected, this command will create a new window containing a small grid (called the *exploded grid*). The exploded grid will contain each of the single cells comprising the composite cell. The orientation of each cell will be preserved in the exploded grid. This command thus allows one to examine composite cells, and to change their composition by re-ORing cells from the exploded grid.

Edit/Undo
A very handy command. Mainly for undoing cuts and pastes. 5 levels of undo, and no re-do option.

CELL MENU
The Cell Menu's commands are used for loading and editing cells within the grid. These commands only work if exactly one cell is currently selected.

Cell/Edit
Opens a *Cell Editor Window* and loads it with the source code for the currently selected cell. If an empty cell is currently selected, it creates an empty cell editor window. See the next section for details on the cell editor window.

Cell/Load
Used to load a cell from a library into the currently selected cell, overwriting any cell already present in that location. One the first load, a Library Select window will appear, and the user must select a library (*.LIB) Do not move above the root of the library directory (normally C:\PORTABLE LOADER\LIBRARIES).

After selecting a library, a Cell Selection window appears. This window lists each of the cells contained in the library. Double clicking a selection or selecting a cell and clicking OK will load that cell into the grid. Clicking Cancel will terminate the load command. Clicking "New Library" will again bring up the Library Select window.

7. Cell Editor Window

The Cell Editor Window is invoked using the Cell/Edit command from a main window. The cell editor window displays the cell's icon, as well as the *source code* for the cell. In general, spaces are ignored in a cell's source code, and case is irrelevant.

Source Code

The most important part of a cell's source code is a set of Boolean expressions which describe the behavior of the cell. The left had side of each equation corresponds to an output, and must be one of:

> DN (Data North), DS (Data South), DW (Data West), DE (Data East), CN (Control North), CS (Control South), CW (Control West) or CE (Control East)

This is followed by an = sign, and then a RPN expression describing the combination of inputs which drive the given output. Since only D inputs are used in determining outputs, there is no need to specify (for example) DN; the single letter N will suffice to indicate "Data North Input." The right hand side is thus a combination of the following:

- N, S, W or E for North, South, West or East Data inputs, respectively
- & for logical AND
- + or | for logical OR
- ! or ~ for logical NOT

Thus, for example, the line "DE=NS+W~&" means

> Data East Out = ((Data North In) or (Data South In)) and (Not (Data West In))

A cell's source code may also contain comments, preceded by a semicolon (";"). Comments may appear most anywhere inside the source code.

There are two special directives which can be used in a cell's source code. One is the ".desc" command. The syntax for this command is:

```
.desc
one or more lines of text
.end
```

This stores a description of the cell inside the binary grid file (produced from the File/Write Binary Grid command). The description may be viewed in the *simulator's* Cell Window (opened by double clicking on a cell in the simulator's main window). Additionally, when a cell is loaded into the simulator, the first line of the description is displayed in the corresponding square of the main window's grid. This is also best understood by experimenting.

The second directive is ".hex" This one-line statement has the format

```
.hex xxxx xxxx xxxx xxxx xxxx xxxx xxxx xxxx
```

where the x's are hexadecimal digits. The resulting 128-bits completely specify the cell's truth table.

Source code is compiled only when the cell is saved in a library. If there are compile errors, a window will pop up and tell you this (though it won't tell you where the errors are).

FILE MENU

There is a single menu item in the cell editor window, the File Menu.

File/New

Used to open a blank cell, i.e., erase the icon and source code. As usual, if you have not saved changes to a cell's source code and you say File/New, your unsaved changes will be lost.

File/Open Cell

Specifies or changes the cell being edited in the cell editor window. As with the main window's Cell/Load command, you may select a library, followed by a cell within that library.

File/Save Cell
Used to store a cell's source code, truth table and icon into a library. It is only while saving a cell that its code is compiled. Again, if a cell name has not been specified, this command is the same as File/Save Cell As

File/Save Cell As
Identical to File/Save Cell, but forces you to specify a cell, and possibly a library. If you are not asked for a library (because you've previously specified one), you may change libraries using the "New Library" button in the "Save Cell As…" window.

File/Delete Current Cell
Removes the cell being edited from the library in which it is contained. This operation can not be undone, although if you have not closed the Cell Editor Window, you could always save it right back into the library.

File/Icon Editor
Opens an editor which allows you to edit cell icons (see next section).

File/Load Icon
Reads the icon currently in the Icon Editor Window, and loads it into the current Cell Editor Window. If the icon editor is not currently open, it opens it up, and loads an empty icon (all white) into the Cell Editor Window.

File/Save Icon
Stores the Cell Editor Window's icon into the icon editor (which it opens if it's not already open).

Remember, Load Icon loads **from** the icon editor. Save Icon write **to** the icon editor. If you get these reversed, it's easy to lose the icon you've just edited.

File/Close
Closes the cell editor window (without asking about unsaved changes).

8. Icon Editor Overview

The icon editor allows you to create and modify 32x32 pixel icons. The basic operation within the editor is to simply click on squares, which changes them to the current *foreground color* (initially black). Note that the Icon Editor's window can be resized to make the squares larger (or smaller) for easier use.

The foreground color can be changed by clicking on a color outside the 32x32 grid.

Lines can be drawn by clicking on the square at one desired endpoint, then clicking on the square at the other desired endpoint, then hitting the "L" key (lowercase). Lines are always drawn horizontal, vertical, or at 45 degrees.

Rectangles can be drawn by selecting two opposite corners (as above for lines), and then hitting the "R" key.

Filled rectangles can be drawn by hitting "R"

By convention, C inputs and outputs are drawn in red, while D inputs and outputs use black or some other color. In the icon editor's window, black and red arrows are drawn around the perimeter of the 32x32 grid. If you stick to placing C and D inputs and outputs where the arrows are drawn, then the I/O lines of your icons will line up nicely.

The Icon Editor Window has no menu options. Icons are loaded into and out of the editor via menus in Cell Editor windows. The Icon Editor can be closed by clicking the upper-right corner's X, or the editor can be left up (only one copy of the editor will appear at once).