Schematic capture productivity pack

Productivity tools for OrCAD™ SDT users.

Technical support and feedback

Send questions and suggestions to:

PC Engines GmbH
Flughofstrasse 58
8152 Glattbrugg, Switzerland

+41 43 321 2179
pdornier@pcengines.ch

Installation

Install the program files in a program directory that is in the path (usually `C:\ORCADEXE`):

```
copy a:*.exe c:\orcadexe
copy a:*.drv c:\orcadesp\drv
copy a:*.mac c:\orcad
```

Change the directory names if you don't use the OrCAD default names.

Old OrTie was a DOS program. The new version is a Win32 executable.

Preparing your schematics

These utility programs are designed to read OrCAD SDT version 3.1 and 4.x schematic files. Hierarchical files are NOT supported.

SDT 386+ files are currently not supported. You can use the conversion utilities that come with SDT 386+ to translate the schematic back to the old format, run the utilities, then translate forward again.

No matter which version of SDT you use, you will have to place TEXT lines with the file names of the following pages on the first page:

```
|LINK
| 2.SCH
| 3.SCH
| 4.SCH
```
etc.

The first page of the schematic must have either the same name as the design directory (OrCAD convention), or be named `1.SCH`. For example, if the design directory is `\OR\TEST`, the first page of the schematic must be called `TEST.SCH`.

OrTie - page tie generator

OrTie generates page ties (intersheet references) on flat OrCAD SDT schematics. Page ties make schematics more readable, and are helpful in debugging and service.

Create page ties

Using OrTie is very simple. To create page ties:

- Exit from OrCAD SDT, or suspend from OrCAD ESP.
- Change to the directory your schematic is stored in.
- Make a backup copy of your schematic files.
- Execute OrTie at the command prompt:

```
ortie
```

Note: You can set one of the OrCAD ESP user buttons to run OrTie.

OrTie will run in a few second (most of the time is actually spent on disk access). The program consists of three passes:

- Scan: Scan schematic for module ports.
- Update: Write updated schematics to `.TIE` temporary files.
- Rename: When successful, save the original schematics as `.BAK`, and rename the temporary files to your original file names.

If it doesn't have enough space to place the page tie, OrTie will give the sheet number and net name of the offending module port.

Placement of page ties

Page ties are always placed on the unconnected end of a module port. To ensure that all page ties are visible, it is important to place module ports with enough clearance on the left or right side.

If the placement of the page ties is not acceptable, move the module ports. Don't worry how the page ties look - OrTie will delete the old ones and create new ones. Then run OrTie again.

Page ties are stored as TEXT, with an Escape (ASCII 27) at the end. They should not be edited. Otherwise, the Escape will be lost, and the old page tie won't be

deleted on the next run of OrTie.

Delete page ties

To delete all page ties, enter:

```
ortie /d
```

OrRev - title block update utility

OrRev allows quick bulk updates of schematic title blocks, e.g. to update the revision number.

Update revision number

This is used most frequently, and can be done from the command line. For example, to update the revision number of the design in the current directory to `3C`, enter:

```
orrev 3C
```

Update other title block fields

Start ORREV without any parameters. Then enter the following fields. Leave a field blank if you don't want to change it; enter * if you want to clear the field. Press `^Break` if you made a mistake.

```
Organization name:
Address line 1   :
Address line 2   :
Address line 3   :
Address line 4   :
Title of sheet   :
Revision number  :
Document number  :
Renumber sheets y/n:
```

Note: OrRev does not create backup files, you will have to do this yourself if required.

OGR - component search

OGR stands for OrCAD Grep. This program allows you to search for components by reference designator or part value, and nets / module ports by name. To search for a component with reference designator U10:

```
ogr u10
```

To search for any component starting with SIP, enter:

```
ogr sip*
```

Note: The search is not case-sensitive.

HPLASER.DRV - improved 150 dpi driver

The original `HPLASER3.DRV` driver is "broken": It cannot print a B-size schematic on one page, and needs 4 (!) pages for a C-size sheet. While this can be worked around by using PLOT instead of PRINT, printing time is much longer, and the output quality is lower.

`HPLASER.DRV` is modified to use the full page size available on HP Laserjet and compatible printers. For a larger drawing area, the B page size can be extended up to 15700 x 11500.

To install, make a backup of the original `HPLASER3.DRV`, and copy `HPLASER.DRV` over it.

SPEED.MAC macro library for OrCAD SDT

OrCAD's schematic capture user interface can be a drag on productivity. For example, to place a wire, you first have to get out of the current menu (which can require several mouse clicks), then select PLACE from the main menu, then WIRE from the PLACE menu, then BEGIN from the WIRE menu. Doing this with the mouse takes up a lot of time and detracts from the work at hand.

A better way to use OrCAD is to use each input device for what it does best. Use one hand to move the mouse, which is very good for pointing. Use the other hand to give commands on the keyboard. To place a wire, first go to the main menu, then enter P W B on the keyboard.

Even this is too slow. To improve entry speed, the user can create a keyboard macro file. For example, function key F1 could be defined as place wire, and so on. Samples for this approach can be found in the OrCAD TUTOR directory. Unfortunately, most of these macro files have some fundamental problems.

First, function keys are inherently hard to remember. The accelerator keys in Microsoft Windows are much easier to remember, because their names are an abbreviation of the full command. For example, File Save is Alt-F S.

Second, the OrCAD interface is still modal. The meaning of 'P W B' depends on what menu you are in. If it was the wrong one, results can be unpredictable. The user constantly has to press ESC or the right mouse button to return to the main menu. This can be distracting or even stressful.

This macro library solves these problems and can greatly improve schematic capture productivity. For SDT 4.x, install as follows:

Copy SPEED.MAC to the \ORCAD directory (base directory for all designs). Start OrCAD for the TEMPLATE or the current design. Go to the Configure Schematic Tools screen, then page down to Macro Options. Set the draft macro file to:

```
\ORCAD\SPEED.MAC
```

and the Draft Initial Macro to

```
\0
```

The initial macro is run automatically when Draft is started, and will set the following options:

- Disable the error bell. This is required, see below.
- Enable dragging of buses.
- Set left button yes. This saves one mouse click per line segment.

This library is modeless, i.e. each command works no matter what menu you are in. Each macro simply enters ESC four times to get out of the deepest possible menu. Draft will beep at you when you do unnecessary ESC, therefore the error bell has to be disabled.

Control commands are used to place elements. To draw a wire, position the cursor at the start of the wire, then press Control-W (^W). Draw the wire as usual, and end with the right mouse button.

Ctrl-B	Place Bus *	(* = place item at current cursor position)
Ctrl-D	Place Junction *	
Ctrl-E	Place Entry	
Ctrl-J	Place Junction *	
Ctrl-L	Place Label *	
Ctrl-N	Place No-Connect *	
Ctrl-P	Place Module port *	
Ctrl-T	Place Text	
Ctrl-W	Place Wire *	

Alt commands are used to manipulate the drawing. For example, press Alt-G (\G) to get a device. To use the block commands, point to the first corner, then enter the command and click on the second corner.

Alt-B	Block Drag *
Alt-C	Zoom Center *
Alt-D	Delete Object *
Alt-E	Edit *
Alt-F	Find
Alt-G	Get
Alt-I	Zoom In
Alt-L	Library
Alt-M	Block Move *
Alt-O	Zoom Out
Alt-R	Repeat
Alt-S	Save file. Alt-S A = save and exit.
Alt-U	Delete Undo
Alt-X	Exit WITHOUT save. Alt-X will prompt if the file has been changed.

OSYM - Library generator

Creating library devices is tedious work, no matter whether you use LIBEDIT or COMPOSER. There is just too much redundant information to enter.

OSYM accelerates the process by providing a more compact, non-redundant data format. Usage (you may want to make this a batch file):

```
osym sample
composer sample.l sample.lib
```

This takes the input file `SAMPLE`, and generates a library text file `SAMPLE.L`. COMPOSER then generates `SAMPLE.LIB` from `SAMPLE.L`.

The input file is a plain text file and can be generated using any ASCII text editor. The file is not case-sensitive, everything is converted to upper case. It starts with either a `DEVICE` or `PGA / BGA` / (for gridarray) statement. Then specify what side of the chip pins should be placed - `LEFT`, `RIGHT`, `TOP` or `BOTTOM`.

Pin definitions start with `A` (PAS = analog), `B` (I/O = bidirectional), `H` (HiZ = tri state), `I` (IN = input) or `O` (OUT = output). They are followed by the pin name and the pin number(s). Pin names (e.g. `SA0`) are automatically incremented if they end with a number. Pin numbers can be separated either by spaces or by +. This allows faster data entry through the numeric key pad. A + at the end of a line lets you continue on the next line.

A blank line skips a pin location. Use this to separate unrelated groups of pins. If you want to insert comments, start lines with either `;` or `#`.

Finally, power pins are placed in a separate section. Power pins are placed in unused pin locations. This makes life easier when you move signal pins around, the power pins will just go somewhere else.

Example file (SAMPLE):

```
;
; Crystal Semiconductor CS8900 Ethernet controller
;
device cs8900 20 70
left
; Note that empty lines result in an empty position on the symbol.

i sa0           37 38 39 40 41 42 43 44 45 46 47 48 50 51 52 53 54 58 59 60
#
# Pin numbers/names can also be separated by +
# (faster to enter on numeric pad)

b sd0           65+66+67+68+71+72+73+74+
                27+26+25+24+21+20+19+18

i chipsel#      7
i memw#         28
i memr#         29
i iow#          62
i ior#          61
i refresh#      49
i sbhe#         36
i aen           63

i testsel       76

i sleep#        77
i reset         75

i dack#0        16 14 12

h drq0          15 13 11

h irq0          32 31 30 35

h mem16#        34
h io16#         33
h iordy         64
;
right

a res           93

a xtl1          97 98

i eedin         6
o eesk          4
o eecs          3
o eedout        5

o csout#        17
i elcs#         2

o led0#/hc0#    99
o bstatus#/hc1# 78
o led2#         100
```

```
a do-           84
a do+           83

a ci-           82
a ci+           81

a di-           80
a di+           79

a rxd-          92
a rxd+          91

a txd-          88
a txd+          87

power

;digital
p vcc           9+22+56+69
p gnd           8+10+23+55+57+70
;analog
p vcc           90+85+95
p gnd           1+89+86+94+96
```

Notes:

- I recommend setting the device size to be large at first. Once you have entered all pins, run OSYM and inspect the output file to see how much space is really required. Change pin locations until you are satisfied, then set the correct device size.

- Print the output file and compare against the pin number list to verify that the symbol definition is correct.

- When defining large symbols (200+ pins), place large buses (e.g. address and data) on top and bottom, and control signals on left and right. This saves on drawing space, and makes it possible to stay with extended B size drawings.