Seadali: Your Gateway to Sea-of-Gates Layout

30th September 2019

This appendix deals with *seadali*, the interactive tool at the very heart of the OCEAN layout system. It is worthwhile to read this appendix carefully, since it contains many useful hints on how to click the buttons efficiently. *Seadali* is an interactive program which allows you to:

- View or modify any existing layout cell.
- Create an empty layout cell from scratch with the help of *fish*.
- Place instances in your layout manually or automatically using *madonna*.
- Automatically route your layout using *trout*.
- Verify the correctness of your layout against the design rules and the circuit description.

There are many buttons and options to control this program. But don't worry: you'll get accustomed to them quite soon. We try to describe only the most vital buttons and functions, so you can get started quickly. All trivial or less useful buttons are not described.

In all examples of this, and the following description we assume the *fishbone* image. Some mask names, design constrains and commands work slightly different in the other images.

1 Starting *seadali*

The best way to start *seadali* is by typing:

```
[op5u9/myproject]  seadali &
```

The '&' instructs the operating system (Linux) to start the program in the background. Therefore the prompt returns immediately. After starting a window pops up[1], which shows the main menu on the left. From this menu you can select various sub-menus by clicking the appropriate box. The bottom-most command of any menu always brings you back to the main menu. *Seadali* is also sensitive to keyboard input. Table 1 (page 11) shows all commands.

[1] If starting *seadali* doesn't work and no window appears, remember that you must be in a project directory. You can make a project directory (NELSIS database) using *mkopr*. Also, check whether your DISPLAY environment variable is set to the proper screen (the name of your terminal, followed by ':0').

2 Getting on-line help in *seadali*

In the main menu you'll find the button help. This will start a hypertext on-line help page on OCEAN. You need the browser program *mozilla* too for this help facility. To use another browser, set the `BROWSER` environment variable. Initially it will display the hypertextpage `$ICDPATH/share/lib/seadali/help.h`

3 Reading and writing layout: database

Pressing database brings you in a special menu for reading and writing cells from/into the database. Let's have a look at the buttons:

READ Load a cell from the database into the editor. The list of all cells in your database will be displayed. If this list is empty, your database has no cells. Notice that it is not possible to read or edit imported cells from libraries. You can only have a look at them by loading them as instances in the menu instances (see 5 on page 3). It is possible to start *seadali* with a cell name as argument. For instance:

```
[op5u9/myproject]  seadali adder &
```

causes the cell adder to be read automatically at startup.

Expand all Set the hierarchical level (that is, the amount of detail) you wish to see of the cell. Initially all son-cells are expanded 'to the bottom'. Alternatively, you can just use the keys 1-9 on your keyboard (see table 1 on page 11). Initially all son-cells are expanded 'to the bottom'. You can set this default expansion level for all cells in the `.dalirc`-file (see 9.1 on page 8). Initially all son-cells are expanded 'to the bottom'.

Expand instance Set expansion level of a specific instance. This can be useful to prevent long drawing times on large designs.

show Sub terminals Show the terminals of the sub-cells. The key 's' on your keyboard is doing the same (see table 1 on page 11).

NEW Erase the current cell in *seadali*'s memory to start with a new cell. This only clears the memory of *seadali*. It doesn't remove the cell from the database.

Fish If the design is empty (initially, or after pressing NEW) this button creates a 20×1 array of the fishbone image. This is useful to start a new cell. If the design is not empty, this button will purify your layout (see appendix 11 on page 13).

Fish -i This button performs the same as Fish, except that the image is removed from the cell. Also, the cell is automatically moved as close as possible to the origin. This button can be useful if you want to use the cell as a child cell on a higher level of hierarchy.

WRITE Write the cell into the database. If necessary, you can write it under a new name.

The title bar of *seadali*'s window shows the name of the cell which you are editing now, together with the expansion level.

You can interrupt *seadali* **while it is drawing!** Just hit any key to stop the drawing and regain control. This can be especially useful for extremely large layouts and/or nervous designers. Hit the key r or next (on the keyboard) to redraw the screen, in case you want to complete an interrupted drawing. During input activities (reading, expansion) *seadali* cannot be interrupted.

4 Running other programs: the automatic tools menu

This menu contains all automatic tools which can be run from *seadali*. Most of them are dealt with in other sections. In addition, this menu contains the Print hardcopy-button, which prints the layout which is currently being edited. See section 9.5 (page 9) for more details.

5 How to import cells from a library: the instances menu

Hierarchy is the most important way to handle the complexity of any circuit. In the instances menu you can import other cells into you design as 'son cells' (also called 'instances'). The son cells can be imported from a remote library, which is nothing else than just another project. Obviously you can also add son-cells which are in the current project. *Seadali* displays the names of the instances in your design. It also displays the cell name of each instance between '@' characters. The following buttons are important to play around with instances:

Madonna Automatically place the instances of a circuit. See appendix **??** on page **??** for all the naked details about *madonna*.

ADD instance Add a local instance to your design. After you have selected the instance you must click its left bottom position. It is not necessary to align the instances exactly because *seadali* will automatically snap them to the grid (if the 'image mode' is on). *Seadali* even performs automatic mirroring of small instances, so that they will always match the grid. Also *fish* can take care of the snapping.

ADD imported instance Just like the previous command, but this button adds an imported instance to your design. Use this button to add library cells. *Seadali* shows a list of all imported (library) cells. You can add libraries to your project using *mkopr*.

set instance name Set the name of a specific instance. Notice that an instance which you add by hand using ADD instance has no name yet. This button can be useful to help the automatic router select the correct instance.

DELETE instance Does just what you expect it to do: delete a specific instance.

move, mirror **and** rotate Use these buttons to put the cell on its right place. In 'image mode', *seadali* will shift and/or mirror the cell automatically. On certain places in the fishbone the cells must be mirrored in the x-axis.

Fish Run the layout purifier, which will shift the instances to a proper grid position. See appendix 11 on page 13 for more details.

6 Editing mask layout: the boxes menu

6.1 Selecting active masks

To add a wire or any other mask pattern in *seadali* you first have to activate a mask. The bottom of the window shows the list of available masks. In the sea-of-gates process you must press in to activate the first metal layer and ins to activate the second metal layer. For your convenience the ins-mask is automatically activated when *seadali* is started.

Warning: Only activate the in, ins, con, cop, cps or cos masks. On our sea-of-gates chip it is not possible to add or remove patterns in any of the other masks. If you do so, *fish* will complain and remove them automatically.

6.2 Making wires using APPEND

To add an arbitrary box in the active layer(s), just press APPEND and click at the appropriate positions in the layout. You can change the active masks also while this command is activated. If image mode is activated, the boxes you append are automatically snapped on the grid. Using the keyboard keys you can zoom or pan the window using the keyboard keys (see table 1 on page 11).

6.3 Making wires with wire

Long, continuous wires can be added more easily with the wire button. Just enter the corners of the wire. Click next to start a new wire or ready to return to the boxes menu.

Huh?! I am adding a wire but nothing is happening! The answer is simple: you forgot to activate the layer, or you also activated another more dominant layer (such as cps or other contacts). Also it is possible that you are adding layout over an existing contact. Since the contacts are displayed dominant, they hide other layout which covers them. Finally it is possible that you accidently made a specific layer invisible in the visible masks menu (see section 8, page 7).

Huh?! Some wires are drawn in a lighter color! You cannot edit the layout of con-cells. To indicate the difference between the different hierarchical levels, the layout of son-cells is drawn in a **hashed** style. See section 9.3 on page 9.

6.4 Ensuring design-rule correctness

For Sea-of-Gates layout it is useful to select the image mode in the settings menu or the the `.dalirc`-file. If this mode is enabled, the boxes you append are automatically snapped to the grid of the image. Also, the wires have the proper width, and the vias are aligned precisely. This behavior is disabled by switching off the image mode. See section 9.2 for more details.

Despite the features of image mode, the layout purifier *fish* (see appendix 11 on page 13) should still be used before writing anything to back to the database. *Fish* performs more elaborate design-rule checks on your layout.

6.5 How to make contacts between layers?

In the *fishbone* sea-of-gates image there are basically only two types of contacts:

1. Between the image (ps or od) and metal 1 (in): con, cop or cps
2. Between metal 1 and metal 2 (ins): cos.

To add a contact, use the APPEND command to make a small box at the desired position. For a contact between the image and metal 1, you must activate the mask con, cop or cps. Which of these three you use doesn't matter because *fish* takes care of that. For a contact between metal 1 and metal 2 you must use the cos-mask.

Press Fish to convert the contact mask into a design rule correct pattern.

Warning: There are some restrictions in the placement of the contacts. They may not be stacked and they are not allowed in certain rows. See section **??** on page **??** for more details.

In the *octagon* image there are three layers and therefore masks for contacts:

1. Between the image (ps, nn or pp) and metal 1 (in): co
2. Between metal 1 and metal 2 (ins): cos.
3. Between metal 2 and metal 3 (int): cot.

In the single-layer *gate-array* image there is only one contact.

6.6 Deleting layout: DELETE

Deleting is simple. Just activate the appropriate layers and click the corners of the box to be deleted. To delete all the wires of you current design, activate all masks and drag the box over your entire design.

Huh?! I try to delete but it won't go away! As usual, you are doing something wrong. First of all, check if you have selected the right active layers. Secondly, remember that it is NOT possible to edit the mask patterns of the son cells (instances). If you want to change that you must edit that particular son cell. It is convenient to use the hashed drawing style for instances to see which wires belong to the son-cells (see section 9.3 on page 9).

Can I undo a deletion or any other command? No, unfortunately you cannot undo any action. Therefore you must be careful. You should also be careful with the masks you activate. Adding a big box in the wrong layer can destroy hours of work. Save your layout regularly to avoid frustration!

6.7 Copying layout: yank and put

Sometimes it is convenient to copy or to move a part of the mask. Just like DELETE, yank puts the indicated box in a paste buffer. The difference is that yank doesn't remove the box. You can copy the contents of the paste buffer to a certain position by pressing put.

Huh?! It doesn't seem to work! Remember that it is not possible to yank the layout of son cells. You can only yank and put the layout of the (father) cell on which you are currently working.

6.8 Purifying and checking your layout: Fish

Pressing Fish makes your manual design design rule correct. All wire segments will be shifted on the grid, and the instances will be properly aligned. *Fish* also shifts sufficient repetitions of the fishbone image under your design. A more elaborate description of *fish* will be given in appendix 11 on page 13.

6.9 Automatically routing your design: Trout

This button will call the automatic router called *trout*. This is a rather complicated command. Pressing it without the proper preparations will most likely result in an error message. See appendix **??** for more details on how to use the router.

6.10 Verifying layout connectivity: Check nets

Pressing this button will verify your layout by comparing it with the netlist in the circuit description. Your layout will be 'fished' automatically if your press this button. The check which is hidden behind Check nets is quite strong. The following errors will be detected:

- All errors which *fish* detects. In this case the errors of *fish* will be logged in the database file `seadif/sea.out`. In contrast to the button Fish, no window with errors will pop up.
- Missing or superfluous instances in your layout.
- Unconnected terminals of nets. The unconnects will be indicated in the layout by a small box with the name of the net and an arrow. The indicator will be placed on top of each terminal that has no connection to another terminal of the same net.

- Short-circuits between different nets. The program reports the names of the nets which form a short-circuit. These nets will also be indicated in the layout in the familiar way: SHORT_no (_no is the number of the short circuit). Notice that the indicators do not indicate the exact spot of the short circuit. The actual short is somewhere on the path between two indicators of the same short.

Huh?! How do I find the indicators in this huge layout? In some cases the scale of the design is so large that the individual indicators are hard to distinguish. One of the ways to find an indicator is by zooming in and moving the window around. For small layouts this is the easiest way. For really large designs, however, this is a hell of a job, a bit like finding a needle in a haystack. The following trick will help you to find that single stacked contact (or unconnect or whatever is indicated). Go to the main menu and press visible masks (see also section 8). Make everything invisible except instances. The indicators will now be visible as little white boxes (or dots). In this way you'll know where to zoom.

7 Adding terminals: the terminals menu

For simulation it is required to indicate the positions of the terminals in the layout. If you routed the layout automatically using *trout* the terminals will also be placed automatically. In the menu terminals you can modify the terminal placement or make one manually.

7.1 How to add a terminal?

Again, you must set the active layer of a terminal by clicking the appropriate mask in the bottom row of the window. A terminal can only be placed in the in and ins layer. Next press ADD terminal and drag a small rectangular box on the grid position of the terminal. Next, *seadali* will ask you to give a name to this terminal. Enter it using the keyboard. The name must be unique. Just give it the same name as in the circuit description you made.

7.2 How can I get rid of a terminal?

That is very simple: use DELETE terminal and point at the terminal you wish to delete. You can also delete all terminals within a specific area using DELETE terms in box. Don't forget to set the proper active layer.

8 Setting the visible layers: visible masks

In many cases it is not so useful to display all the masks of a cell. Especially on our sea-of-gates many masks are not so relevant for the electrical design, and will only make the picture unclear. In the visible masks menu you can switch on or of any mask you want. The default settings of *seadali*

are such that the ps, in and ins plus the contact masks are visible. All the other masks are switched of. You can change the default settings by editing the `.dalirc` file in your project directory (see 9.1).

How can I make the grid of the Sea-of-Gates image visible? You can do this just by clicking show Grid of by pressing g on the keyboard. The dots indicate the grid positions (provided that **image mode** is **on**). The grid will not be drawn if there are too many grid points in sight.

When does *seadali* display the image? *Seadali* will not draw the image if the scale of the layout is too big. In this way long drawing times are prevented. If you zoom in enough the image will become visible again. You can set the maximum amount of image elements which should be drawn in the file `.dalirc` (see section 9.1). You you want to see the image at all times, switch of the **image mode** in the settings menu (see section 9.2).

9 Some other nice features: the settings menu

There are a few ways to customize the behavior of *seadali* according to your own wishes. In this section we'll list a few of them.

9.1 Customizing *seadali*: the `.dalirc`-file

Your project directory automatically contains a file called `.dalirc`. This file is read by *seadali* during startup. You can edit this file to change various settings. Just have a look at it and try change some parameters. Obviously, you must re-start *seadali* for the changes to have any effect.

Huh?! I don't see any file called `.dalirc` in my project!
That is normal. In UNIX all files which start with a '.' are not visible. You can make them visible by typing:

```
[op5u9/myproject]  ls -a
```

If it is still not there, copy the default file into your project:

```
[op5u9/myproject]  cp $ICDPATH/share/lib/process/fishbone/.dalirc .
```

9.2 Switching on the Sea-of-Gates mode: image mode

Seadali has a special **image mode** for working with semi-custom layout. This mode should always be turned on if you are working with Sea-of-Gates. The **image mode** has the following features:

- Layout boxes and instances are snapped to the grid. *Seadali* will warn you if you try to place something at the wrong place. Library cells are automatically mirrored in the x-axis if necessary or possible.

- The position of the tracker is displayed in grid points, rather than in lambda's.
- The grid of the image is displayed if you press show Grid. The grid is only displayed if it is useful to do so.
- The image is not drawn under your layout if the scale of the picture becomes too big. If the **image mode** is switched **off**, the image is drawn always[2].

This mode should generally be turned on if you are working with Sea-of-Gates. Only in some exceptional cases (e.g. to force the mirroring of a certain instance) you may want to turn it off.

9.3 Setting drawing style for instances: draw hashed

In many cases it is not so clear to which level of hierarchy a certain wire belongs. Therefore *seadali* displays the layout of all son-cells in a less bright color than the wires at the current level. The less bright color is achieved by hashing (45 degree lines). You can switch this feature on or off by pressing draw hashed (also in the `.dalirc`-file).

9.4 Setting dominant or transparent drawing style: Draw dominant

There is another drawing mode which can be set by you. In the transparent drawing mode the colors mix. In this way any crossing of the masks results in a mix-color. With many masks this makes the picture rather confusing.

With the dominant drawing style, the layers are drawn as if they are stacked on top of each other. In this case metal 2 (the top layer) obscures any layout below it. The default drawing mode is dominant. You can switch by pressing the button Draw dominant or by setting the parameter in the file `.dalirc`. In the same file you can set the order in which the layers should be drawn.

9.5 Setting the print command for the Print hardcopy-button

The button Print hardcopy in the automatic tools menu causes the current layout to be printed by the laserprinter. It might be necessary to configure the proper print command to perform this action. The proper command sequence can be set using the keyword `Print_command` in the file `.dalirc`. To set the proper command, it is relevant to understand what *seadali* is doing after you pressed this command:

1. The current layout is written away in the NELSIS database using a temporary cell name.
2. Any occurrences of %s in the print command are replaced by the name of the temporary layout cell.
3. The print command sequence is executed by a shell.

[2]Provided its present as a instance in your layout.

4. The temporary cell is removed.

If no print command is set, *seadali* will run the command *playout*. *Playout*, on its turn, will create a PostScript file using *getepslay*, print it, and show the printer queue.

9.6 X-window redrawing strategy: backing store

Seadali asks the X-window server (your screen), to maintain a copy of the window in its memory. This feature (called *backing store*) makes it possible to move, obscure, iconify and then re-expose *seadali*'s window without redrawing the entire picture. Especially for large layouts this saves a considerable amount of drawing time and CPU load. The price we have to pay for this nice feature is some extra memory consumption by the server. In general this is not a problem, but for certain X-terminals without memory management the extra (500K to 1M byte) memory load might cause some problems. Therefore this feature can be switched off in the settings menu. You can also switch it off in the file `.dalirc` by adding the line: "`backingstore off`".

10 Interrupting *seadali*: keyboard input

Seadali is sensitive for keyboard input. Table 1 shows all the keycombinations which can currently be used. They serve as a convenient alternative for clicking with the mouse in the menu. If *seadali* is drawing the picture, pressing *any* key interrupts the drawing[3] within one second.

[3]There is one exception: during the reading or expansion process *seadali* cannot be interrupted.

key	function
space /any key	interrupt (stop) drawing
'i' or '+'	zoom in by a factor of 2
'o' or '-'	zoom out by a factor of 2
← or 'h'	pan left
→ or 'l'	pan right
↑ or 'k'	pan up
↓ or 'j'	pan down
'c'	center window around current cursor position
'b'	set window to bounding box
'p'	previous window
'r'	redraw screen
'1','2','3',...,'9'	set expansion level
'0'	expand maximum
's'	show sub terminals
'd'	toggle hashed drawing style
'D'	toggle dominant drawing style
'g'	toggle display of grid
't'	toggle tracker (cursor position display)

Table 1: The key commands for *seadali*.

11 Fish: The Layout Purifier and Design Rule Checker

In some of the menus of *seadali* you've encountered the button Fish. In this appendix we'll describe a bit more about what is behind this button. *Fish* is a layout purifying program for sea-of-gates layout. It reads a layout, processes it, and produces a new enhanced cell. The basic idea is to allow some design-rule errors during the manual design of a gate-array cell. *Fish* corrects these errors and produces a design rule correct result. In this way the process of manual design using *seadali* can be speeded up considerably. *Fish* also prints warning messages for a variety of design rule errors. You can call *fish* in two ways:

1. By pressing the button Fish in *seadali*. This is the most common way of using it. What is actually happening is that *seadali* writes your design as a temporary cell into the database, then it calls *fish*. If everything goes well, *seadali* re-reads the purified cell. This will take a few seconds, depending on the load of the computer.

2. By typing `fish` on the command line:

   ```
   [op5u9/myproject]  fish <cell_name>
   ```

   This purifies the cell named `<cell_name>`. If you want to know more: try `fish -h` for a brief list of all options.

Fish needs an image description file. This file is called `image.seadif` and should be present in the directory `seadif` in your project. It contains data about the positions of grid points, the way in which the image should be repeated, the kind of contact which should be used, etc. If you like you can have a look at it.

12 What errors does *fish* detect?

Fish performs limited design rule checking of your layout. It will report errors for:

- Stacked contacts. The process design rules of the fishbone image do not allow to stack an in to ins contact (= cos) on top of another via. The wire would crack at that point. *fish* will indicate the error in the layout by a small box with an arrow pointing to it. *Fish* does NOT detect a pair of stacked contacts if one of the contacts is in a son-cell.

- Contacts at an illegal position. A cos contact is not allowed in the rows of the polysilicon gate contacts. The error is again indicated in the layout by a small box with an arrow.

- Patterns (boxes) in illegal masks. *Fish* removes any patterns in ps, od, etc. because they may not be used on a semi-custom chip.

- Patterns which are not (entirely) in the first quadrant. In our system all layout boxes must have positive coordinates, that is, it must be to the right-top of the origin. The illegal boxes will be removed.

Anyway, you must keep in mind that *fish* is not too clever. It will not warn you of short-circuits and mis-aligned instances (such as forgotten mirroring in the x-axis). For that you must use the button Check nets (see section 6.10 on page 6). A list of all the errors which were detected by *fish* will be displayed in a pop-up window.

13 How does *fish* handle instances?

Fish snaps instances to the nearest grid point. To perform this snapping, *fish* opens the instance and looks for contacts. The first contact it encounters is used for the snapping. Notice that in this way *fish* assumes that the instances are on grid (that is, fished). If it can't find any contacts *fish* gets angry and it doesn't align the instance.

14 How to create an empty array of image?

Before editing it is convenient to have an empty array to indicate the transistor and grid positions. This can be done by pressing the button Fish in *seadali* when the design is empty. A 20×1 array will be generated. You can also do the same from the command line by creating a new (empty) cell into the database:

```
[op5u9/myproject]  fish -x 20 -y 1 -o nand2
```

The option `-o nand2` instructs *fish* to write the empty array into a cell called `nand2`. You can now start editing your cell by adding wires.

But what should I do if the 20×1 image doesn't fit anymore? Should the underlying image become too small, simply hit the button Fish to enlarge it automatically so that it fits the wire pattern. Alternatively you can enlarge the basic image by pressing press set array parameters in the instances menu. Set nx or ny to the new value. Do not touch the dx and dy buttons!