APPLICATION NOTE

O K I A S I C P R O D U C T S

Using OKI's Floorplanner : Standalone Operation and Links to Synopsys

September 1995

OKI Semiconductor

■ Contents ■

INTRODUCTION

OKI offers a floorplanning tool for high-density ASIC design. This floorplanning tool supports all of OKI's 0.5-µm and 0.8-µm ASIC product families. The two main purposes for OKI's floorplanning tool are:

- To ensure conformance of critical circuit performance specifications
- To shorten overall design turn-around-time (TAT)

This application note explains the benefit of floorplanning, the scheme of OKI's floorplanner, and the algorithm for estimating wiring capacitance. In addition, this application note provides guidelines for OKI floorplanner users and Synopsys PDEF interface procedures.

Benefits of Floorplanning

In sub-micron technologies, circuit timing is dominated by parasitic capacitance. *Figure 1* shows the relationship between gate delay and wiring delay for different technologies and gate densities.

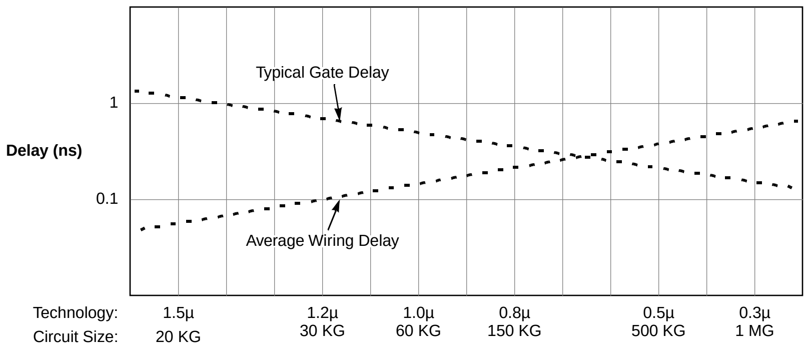

Figure 1. Gate and Wiring Delay for Different Technologies

Floorplanning allows designers to control parasitic capacitance in a circuit by participating in the physical design process. Designers can partition their ASIC circuit in the most efficient hierarchical manner, and/or specify the exact placement of critical timing paths to guarantee high-speed performance.

Floorplanning also allows the reduction of layout iterations, minimizing a design's overall TAT. As parasitic capacitance dominates a circuit's timing in sub-micron technologies, an accurate capacitance estimation is crucial for accurate pre-layout timing simulation. Quite often, designers have to iterate the circuit layout because unexpected post-layout capacitance causes unacceptable circuit performance.

When using a floorplanner, the estimated capacitance is based on the hierarchical region size, defined during the floorplanning session, instead of the overall size of the selected array. For accurate pre-layout timing simulation, the floorplanner generates estimated capacitances and back-annotates these capacitances to OKI's Delay Processor. In doing so, pre-layout timing estimations are highly correlated to actual post-layout values, eliminating the unexpected post-layout capacitance and resulting re-layout and/or redesign of the circuit.

Figure 2 shows the design flow with and without OKI's floorplanner.

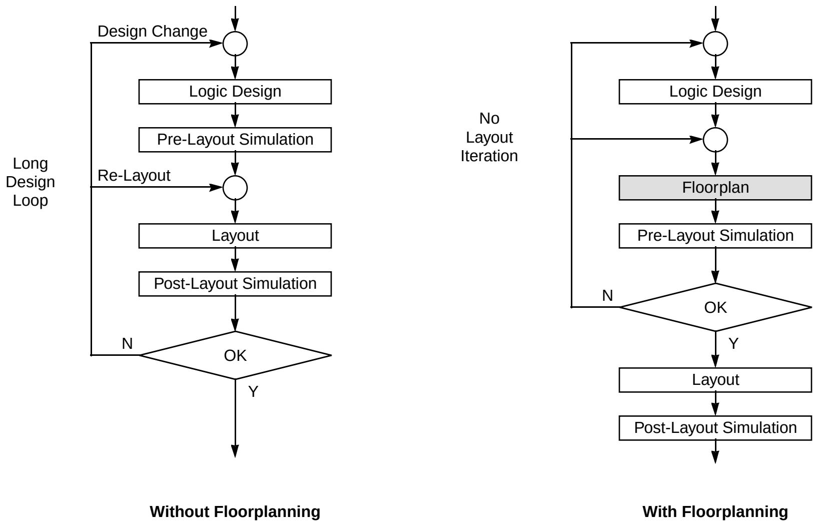

Figure 2. Design Flow with and without Floorplanning

This document is divided into three main parts:

- The first main section, *OKI's Floorplanning Scheme*, describes the overall features, structure, and algorithms of OKI's floorplanner.
- The second main section, *User Guidelines*, describes the FPI file syntax, provides general implementation guidelines, and shows how to calculate the maximum number of nets allowed between adjacent blocks for OKI's 0.5µm and 0.8µm double-level-metal (DLM) and triple-level-metal (TLM) technologies.
- The third section, *Interfacing with Synopsys*, describes how OKI's floorplanner can interface with Synopsys logic synthesis and, optionally, Synopsys Floorplan Management tools.

OKI'S FLOORPLANNING SCHEME

OKI's floorplanner offers a number of features, including:

- Graphical interface
- Sun and RS6000 compatibility
- Simple interface procedures with menu-driven commands and color display of regions
- Automatic gate-count calculation
- Dynamic utilization updating
- Automatic region checking
- Synopsys PDEF interface

The next sections provide an overview of OKI's floorplanner, describe the interface procedure and design flow, and summarize the algorithms used for estimating wire length.

Overview

OKI's floorplanner can be classified as both a *front-end* floorplanner and as a *back-end* floorplanner.

During front-end floorplanning, logic designers use the floorplanner to generate two files: a capacitance file for pre-layout simulation, and a floorplanner interface file for layout.

During back-end floorplanning, the layout engineer transfers the floorplanner interface file to OKI's proprietary layout software, code-named Pegasus. The floorplanner interface file contains information about the placement of blocks and groups of blocks. The back-end floorplanner is automated and is transparent to logic designers.

Figure 3 and *Figure 4* illustrate front- and back-end operation.

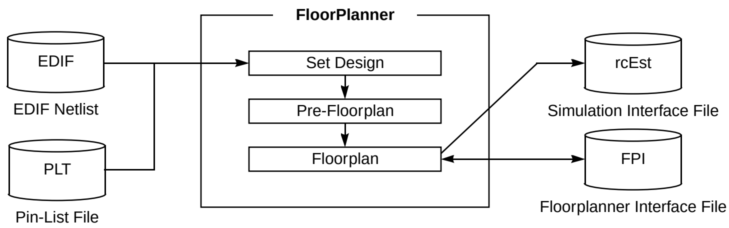

Figure 3. Front-End Floorplanning

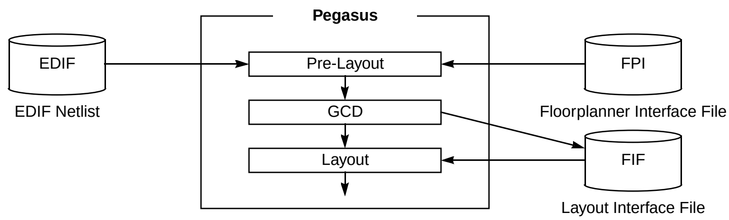

Figure 4. Back-End Floorplanning

Interface Procedures

OKI's floorplanner requires as input an EDIF netlist, a Floorplanner Interface (FPI) file, and optionally a pin-list file. EDIF netlists can be generated on engineering workstations with virtually all current CAE tool sets, including those supplied by Cadence, Mentor, Synopsys, and Viewlogic. The pinlist file is generated by OKI. This application note describes procedures for generating a FPI file.

OKI's floorplanner generates two output files

- An RC delay file for pre-layout simulation
- A Floorplanner Interface File (*FPI file*)

When using the optional Synopys interface, OKI's floorplanner can also generate a PDEf Interface File, a Wire-Load Model File, a netPara file, and a dcScr file.

Design Flow

Figure 5 shows the typical design flow using OKI's floorplanner.

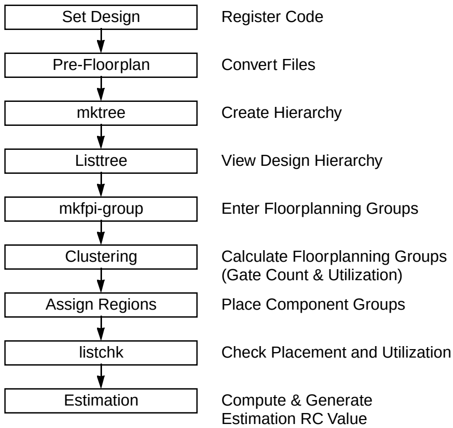

Figure 5. Design Flow

All of the above commands and operations are listed in OKI's *Floorplanner User's Manual*. For more information and operation of each command, please refer to the *Floorplanner User's Manual*.

Wire Length Estimation Algorithm

OKI's floorplanner estimates a wire's capacitance by considering a netlist's hierarchical groups' sizes, relative positions between groups, and the groups' fan-outs. Nets are classified as *closed nets* or *crossing nets*. Nets which are within a defined region are called closed nets. Those nets which cross region boundaries are called crossing nets. Examples follow that show how OKI's Floorplanner estimates the length of routing wires for closed and crossing nets.

Wire Length Estimation for Closed Nets

Figure 6 illustrates a closed net, as well as the estimated wire length for the size of the area in which the closed net resides.

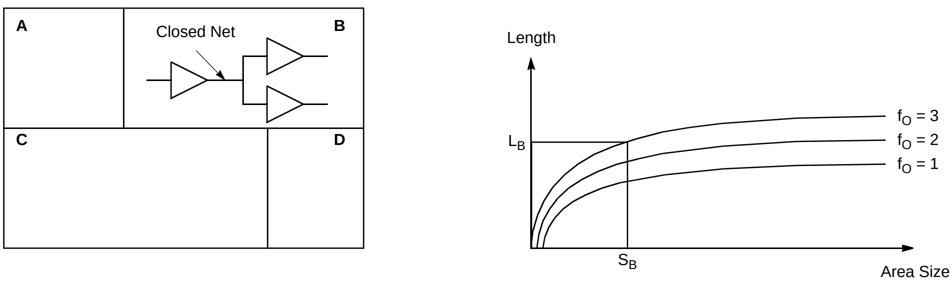

Figure 6. Closed Nets

Referring to the above figure, the length (L_B) of the wire is a function of the size (S_B) of region group B, and the fan-out of the wire, in accordance with the equation:

$$L_B = f(S_B, \text{fan-out})$$

Wire Length Estimation for Crossing Nets

Figure 7 shows crossing nets.

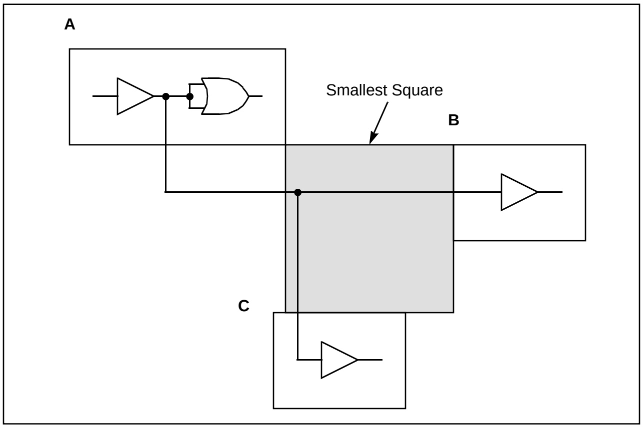

Figure 7. Crossing Nets

Here the length of the wire L_{ABC} is determined by the equation:

$$L_{ABC} = L_A + L_B + L_C + (HP) \times (EWBF)$$

where:

L_A, L_B, and L_C =	The lengths of the routing wires within the A, B, and C regions, respectively.
L_{ABC} =	The total length of the wire *between* the A, B, and C regions (this term includes the wire inside the regions).
X, Y =	The width and height of the *smallest square* between the groups (the smallest square is the shaded area in *Figure 7*).
HP =	The length of half the perimeter of the smallest square (HP = X + Y).
EWBF =	The effective wire bending factor.

The value of EWBF depends on the number of groups, the technology being used for the design and the statistical numbers of previous designs.

After the routing wire length estimation, a capacitance estimator in the floorplanner converts the wire length to capacitance. The algorithm used for wire to capacitance conversion is:

$$C_{lw} = K_1 \times L_1 + K_2 \times L_2 + K_3 \times L_3$$

The values of K_1, K_2, and K_3 are derived from the process and depended on the technology, while the values of L_1, L_2, and L_3 are based on statistical data gathered from previously routed designs.

USER GUIDELINES

The second part of this application note provides guidelines for users of OKI's floorplanner. User guidelines are divided into six main sections, each containing subsections on various topics as listed below.

- *FloorPlanner Interface File* – Syntax, GROUP data specification, REGION data specification, and example
- *Group Clustering* – I/O macros and omission of components
- *Region Assignment* – Region specification, region boundaries and trunk busses, and mega macros
- *RAM / ROM Placement* – Power and ground bus considerations
- *Connections Between Regions* – Calculation of maximum net count between blocks
- *OKI recommends not to use the entire core area when floorplanning the chip. The suggested utilization is 50% ~ 70%.*

FloorPlanner Interface File

This section provides the FPI file syntax, GROUP data specification, REGION data specification, and a design example.

FPI File Syntax

The syntax of the FPI File is:

```
<keyword> <data_1> <data_2>...
```

Each line has one meaning. Each line is distinguished with a keyword. Spaces or tabs partition each word, and parsing is case-insensitive (all characters are converted to upper-case characters during parsing).

The five allowed keywords are:

- DESIGN – The name of the design
- SERIES – The name of the target array
- REGION – The area in which a group of components is placed
- MEGA – A mega macrocell, RAM, or ROM
- GROUP – A set of hierarchical blocks to be placed in a region. Each group is assigned a single, separate region of its own.

OKI's floorplanner automatically generates lines in the FPI file defining the DESIGN, SERIES, REGION, and MEGA parameters. Users must not manually modify these sections. Users must add lines to the FPI file that define each GROUP in the design.

The keyword format is:

```
DESIGN <design code name>
SERIES <series/frame name>
REGION <group name> <pos_1> <pos_2> <pos_3> <pos_4>
MEGA <group name> <component name> <library name> <pos_1> <pos_2> <width>
<height> <placement flag>
GROUP group <group name> <component name>
```

where:

<pos_1> is the starting column of a region.

`<pos_2>`	is the starting row of a region.
`<pos_3>`	is the ending column of a region.
`<pos_4>`	is the ending row of a region.
`<placement flag>`	is used for floorplan applications. The value is either 0 (not placed) or 1 (placed).
`<component name>`	is a hierarchical block.

FPI File GROUP Data Specification

A GROUP is the name of an instance from the design hierarchy. The user must associate the component name in a GROUP statement with one or more hierarchical blocks from the circuit design, using the following syntax:

```
<cell_name(hierarchy level 1)> /
<cell_name(hierarchy level 2)> / .. / <cell_name>
```

The legal character set of a group's name is:

- Alphabetics: `A-Z`, `a-z`
- Numerics: `0-9`
- Specials: _

The following rules apply to the names of groups.

- The group name must be less than 20 characters long and must begin with an alphabetic character.
- Wildcards are allowed and are a useful way to add more than one component to `GROUP`. However, only the last characters in a component name can be shortened by the asterisk wildcard symbol. For example, `inp/cntl1/data*` is acceptable, whereas `inp/cntl*/data1` and `inp/cntl2/*1` are not acceptable.
- If all components belonging to the same logical branch are assigned to the same group, their names can be abbreviated. For example, if users want to group all components inside the `cntl1 block` under the `inp` hierarchy, users can define a group as `GROUP cntl1 inp/cntl1`
- A component should not be specified in more than one group. In situations where a GROUP statement needs to specify many components, extend the statement over the required number of lines. For example:

```
group <group_1> <component_1>
group <group_1> <component_2>
group <group_1> <component_3>
```

FPI File REGION Data Specification

A region is a rectangular area in which one group is placed. OKI's floorplanner also uses a $ELSE group for all components not included in any user-defined groups. The $ELSE group does not appear in the FPI file. If there is a group defined by a GROUP statement, but not by a REGION statement, the group is merged into the $ELSE group.

FPI File Example

A typical FPI data file follows.

```
STAMP FloorPlan Data file saved at Sat Mar 27 14:47:17 1993
DESIGN MSM91S022        <-- Project Code
SERIES B91S074X074      <-- Frame List
                            GROUP Identifier (points to the second field of the REGION lines)
REGION RX 2 164 459 360
REGION CN 287 5 510 164
REGION TX 2 2 287 164
REGION PM 510 2 614 614
MEGA RX INSTANCE_RX/INSTANCE_DM/1994 RAMS64X8 0 0 74 45 1
MEGA RX INSTANCE_RX/INSTANCE_CD/1698 RAMS16X8 74 0 26 45 1
MEGA RX INSTANCE_RX/INSTANCE_CD/1699 RAMS16X8 100 0 26 45 1
MEGA RX INSTANCE_RX/INSTANCE_CD/1700 RAMS16X8 126 0 26 45 1
MEGA RX INSTANCE_RX/INSTANCE_CD/1701 RAMS16X8 152 0 26 45 1
MEGA RX INSTANCE_RX/INSTANCE_CD/1702 RAMS16X4 178 0 26 29 1
MEGA TX INSTANCE_TX/INSTANCE_RM/I1316 RAMS64X8 74 0 74 45 1
MEGA TX INSTANCE_TX/INSTANCE_IR/I1041 RAMS512X8 464 209 140 143 1
GROUP RX INSTANCE_RX
GROUP TX INSTANCE_TX
GROUP CN I1384
GROUP CN I356/I*
GROUP PM INSTANCE_P*
GROUP PM 12*
GROUP PM 1356/O*
```

(STAMP through the last MEGA line:) This portion of the file is generated automatically by the floorplanner.

Note the use of wildcards in the group definitions.

Group Clustering

This section describes I/O clustering and omission of components from groups.

Clustering I/O Macros in Groups

Primitive macrocells described in a EDIF netlist may be soft macrocells. I/O macrocells that are described in such a manner are automatically expanded into separate I/O and core components during placement, as shown in *Figure 8*.

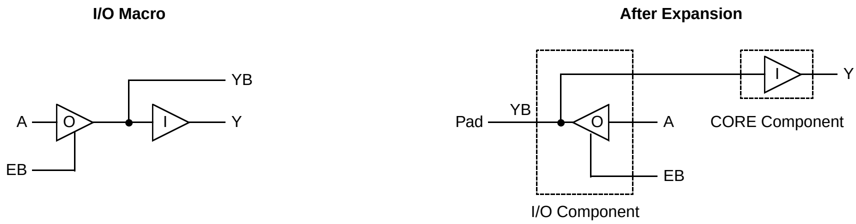

Figure 8. Soft Macrocell Expansion

Since the core component expanded from an I/O macro is not specified to any "keep-in group," there might be a problem if the I/O and core components are placed far away from each other. *Figure 9*, below, illustrates how layout software can separate I/O and core macros during automatic placement.

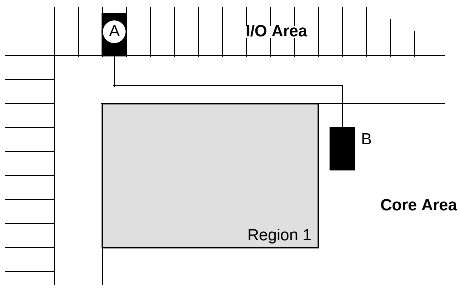

Figure 9. Separation of I/O and Core Macros

There are three solutions to this problem:

1. Include the I/O macrocells in a group. In this case, a core component that is expanded from a soft macrocell is placed inside the region close to the pin area. However, the designer must consider the I/O's position and the region's location in advance, so that the region can be assigned close to the corresponding I/O positions. *Figure 10*, on the next page, illustrates how I/O macros can be included in groups. The different intensities of shading represent different groups.

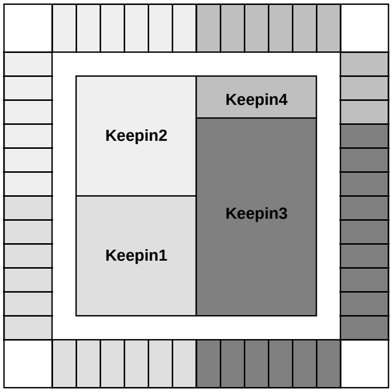

Figure 10. Including I/O Macros in Groups

2. Assign an area for the core component portion of the expanded I/O macrocell around the core area. This is the simplest way to handle the problem, but may increase die size more than necessary. OKI does not recommend this solution. *Figure 11*illustrates this approach, indicating the ring area with cross-hatched shading.

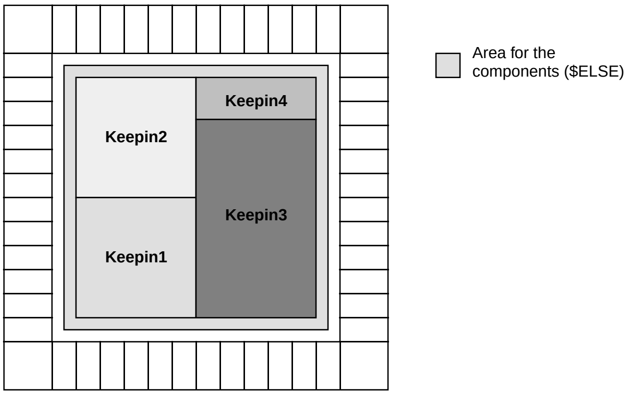

Figure 11. Assigning a Ring Area for Placing I/O Macros

3. Manually preplace the core component portion of the I/O macro before global placement.

Omission of Components in Groups

If some components are not specified in any "keep-in groups," and the core area is filled up with other regions, then the omitted components cannot be placed during the initial placement stage.

The designer must always check the $ELSE group if the core area is completely filled by other regions and make sure that the $ELSE group is empty.

Region Assignment

This section describes how to specify regions, taking account of trunk busses and mega macrocells, without causing placement constraints or incorrect utilization estimates.

Region Specification

Figure 12 indicates the assignment of rows and columns in regions.

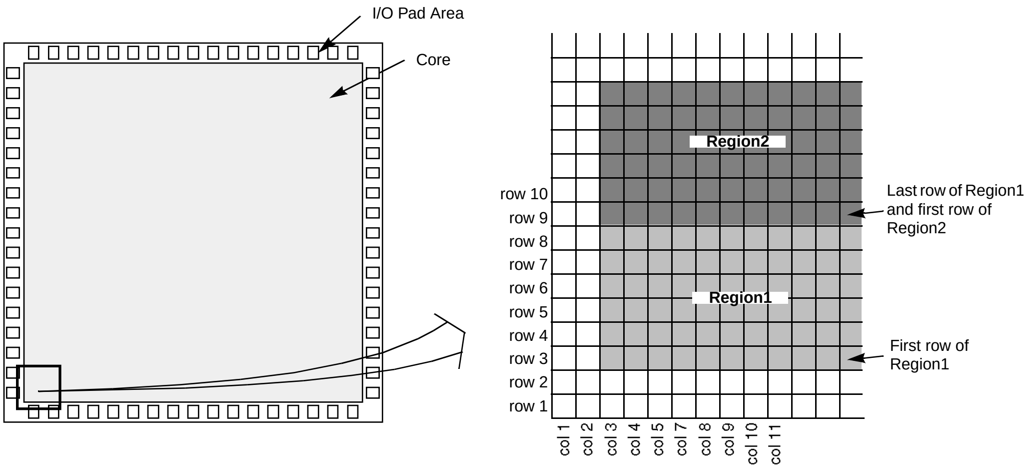

Figure 12. Row and Column Organization in Regions

The following three rules apply to region specification:

- No regions should use the outermost two rows and two columns of the core cell area, as shown in *Figure 12* above.
- Regions should not overlap each other. The first row and column in the region specification is included in the region. The last row and column belongs to the region adjacent to the current region. Regions can thus abut as shown in *Figure 12* above.
- Designers should not leave space between regions because, if the layout software places some random components in the free space, the space may cause routing congestion.

Region Boundaries and Trunk Busses (for 0.8µm Technology Only)

In the core area, there are power/ground busses and clock trunks that are used for clock-skew management. Trunk busses can sometimes cross over regions defined by the designer.

It is acceptable to have trunks crossing over regions. However, if the trunk busses divide regions into two or more areas, producing a narrow area (labeled “A” in *Figure 13* on the next page), then the trunk bus creates a placement constraint.

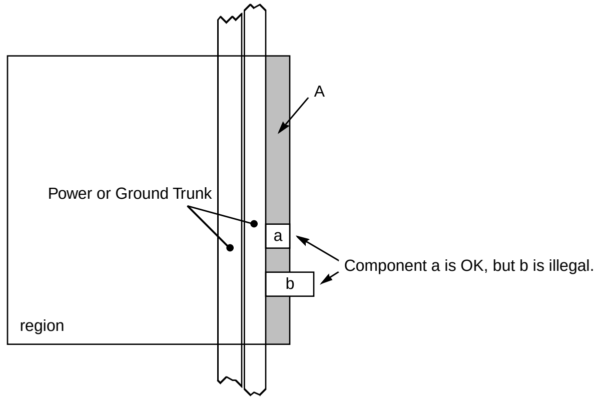

Figure 13. Region Boundary and Trunk Busses Example

To avoid the above problems, place regions so that their boundaries touch the power/ground busses.

Regions and Mega Macrocells

If there is a group containing a mega macrocell, designers must consider the shape of the region and the mega macrocell size. If a mega macrocell is so large that it extends into another region, then the region utilization estimated by the assignment command will be incorrect.

*Figure 14*below shows an overlapping placement that will lead to incorrect utilization estimates, and the corrected placement without the error.

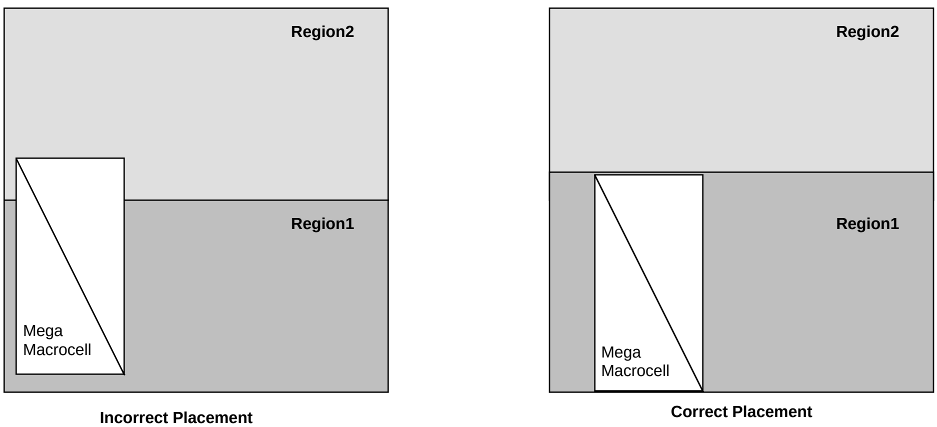

Figure 14. Correct and Incorrect Mega Macrocell Placement

For designs using mega macrocells, define a hierarchy that allows placement of the mega macrocell in a single group by itself. It is preferable to create a single-level hierarchical name for the mega macrocell only, because it is then easier to assign regions for the mega macrocells later during floorplanning. *Figure 15* below shows an incorrect floorplan, with a region containing both logic and a mega macrocell, and a corrected floorplan.

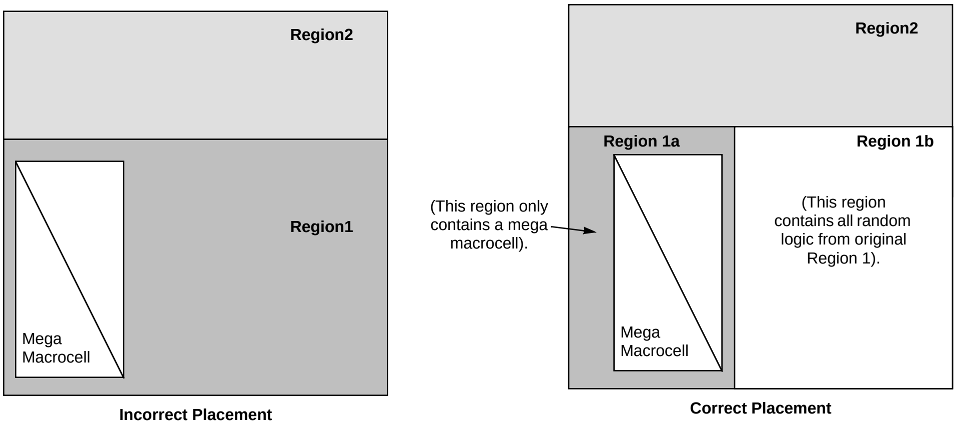

Figure 15. Correct and Incorrect Hierarchical Assignment for a Mega Macrocell

RAM / ROM Placement

Designers using RAMs or ROMs in their designs have to consider the placement locations for the memory cells. Power and ground busses can constrain the placement of RAM and ROM blocks. Guidelines for RAM and ROM placement follow.

- RAMs or ROMs should be placed close to the edges of the die. However, RAMs and ROMs must not block the routing resources for I/O pins.
- Each RAM or ROM is powered by a vertical mega power trunk. RAM and ROM macrocells can be flipped upside down, but cannot be rotated. As a result, for 0.8 μ technologies, the power terminals on the RAM and ROM macrocells are always on both the left and right hand side of the device, while signal terminals are either on the top or the bottom. Similarly, for 0.5μm technologies, both power and signal terminals are either on the top or bottom of the device.
- If the RAMs or ROMs are placed vertically, they can share the same mega power trunk. The designer should leave some horizontal routing channels between the RAMs for routing purposes. If the RAMs or ROMs are placed side by side, the designer should leave some vertical channels free for the mega power trunks.
- If there are multiple RAMs and ROMs of different configurations in the design, the designer must ensure that the mega power trunks can reach the I/O power ring without crossing over the RAMs or ROMs.

Figure 16 below shows incorrect and correct placements for RAMs and ROMs. For the incorrect placement, the mega power trunks for the RAM2 and RAM3 blocks would have to pass either over the RAM1 block or the RAM4 block, which is not allowed. The correct placement rectifies this problem by moving the wider cells to the top and the narrower cells to the bottom. As there are many different possible RAM/ROM placements, please contact an OKI Design Center Engineer to review the floorplanning data.

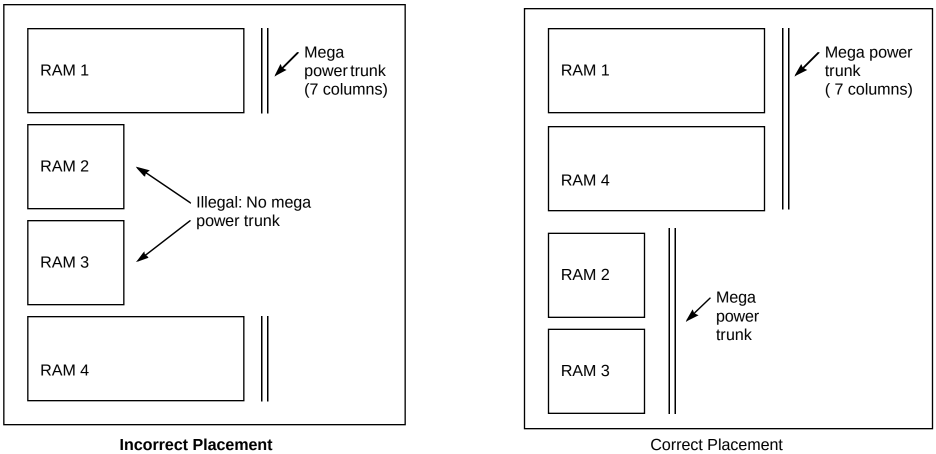

Figure 16. Correct and Incorrect Memory Cell Placement

Connections Between Regions

This section describes how to calculate the maximum number of nets that are allowed between adjacent blocks. Different calculations are used for horizontally adjacent and vertically adjacent blocks.

Maximum Number of Nets Between Horizontally Adjacent Blocks

According to statistics, the maximum number of nets allowed between adjacent blocks must be less than 25% of the total available horizontal routing tracks. *Figure 17* illustrates some such crossing nets.

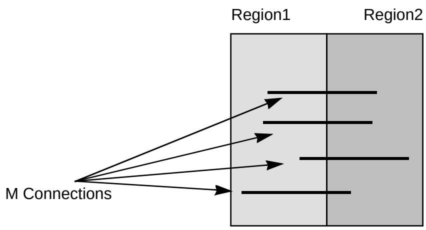

Figure 17. Nets Between Horizontal Blocks

For crossing nets M:

```
M ≤ Horizontal Routing Tracks * 25%
```

For DLM technology:

```
Horizontal Routing Tracks = H * 6
```

For 0.8µm TLM technology:

```
Horizontal Routing Tracks = H * 12
```

For 0.5µm TLM technology:

```
Horizontal Routing Tracks = H * 10
```

Where: M = Maximum number of nets between regions
H = Height of region in rows

Maximum Number of Nets Between Vertically Adjacent Blocks

According to statistics, the maximum number of nets allowed between vertically adjacent (or “stacked”) blocks must be less than 25% of the total available vertical routing branches. *Figure 18* illustrates some such crossing nets.

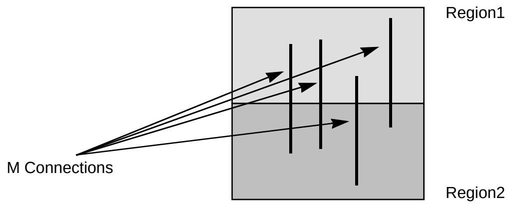

Figure 18. Net Between Vertical Blocks

For crossing nets M:

```
M ≤ Vertical Routing Branches * 25%
Vertical Routing Branches = W * 3
```

Where: M = Maximum number of nets
W = Width of region in columns

Maximum Number of Nets Connected to a Block

To guarantee that each block has enough routing resources to connect to all other blocks, the total number of nets entering and leaving a block must be less than 25% of the total vertical and horizontal routing channels. Equations follow to calculate the maximum allowable nets connected to a block, for DLM and TLM technologies.

```
M ≤ (Horizontal Routing Tracks + Vertical Routing Branches) * 25%
```

For DLM technology:

```
Horizontal Routing Tracks = H * 6 * 2
Vertical Routing Branches = W * 3 * 2
```

For 0.8µm TLM technology:

```
Horizontal Routing Tracks = H * 12 * 2
Vertical Routing Branches = W * 3 * 2
```

For 0.5µm TLM technology:

```
Horizontal Routing Tracks = H * 10 * 2
Vertical Routing Branches = W * 3 * 2
```

Where: M = Maximum number of nets connected to a block
H = Height of region
W = Width of region

The above equations apply to a block that can be accessed from all four sides.

For more information on OKI’s floorplanner, contact OKI Semiconductor’s Sunnyvale design center.

Recommended Utilization

OKI recommends not to use the entire core area when floorplanning the chip. The suggested utilization is 50% ~ 70%.

If the user floorplans the entire core area, there is no flexbility left for the placer.

Designs Using Boundary Scan Methodology

For designs incorporating boundary scan cells, OKI's proprietary layout software automatically guides the placer to put boundary scan registers along the die periphery. Floorplanner users should reserve rows and columns for the boundary scan registers, as shown in *Figure 19*.

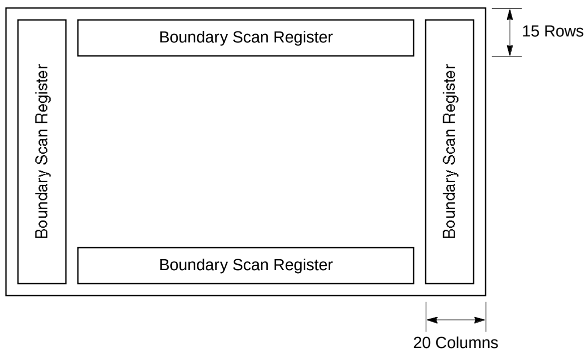

Figure 19. Reserving Rows and Columns for the Boundary Scan Register

INTERFACING WITH SYNOPSYS

In a traditional design approach with synthesis tools, timing violations after pre-layout simulation are fixed by manual editing of the netlist. This process is difficult and time consuming. Also, there is no physical cluster information provided in the synthesis tool, and so it is difficult to synthesize logic using predicted interconnection delay due to wire length. Synthesis tools may therefore create over-optimized results.

To minimize these problems, Synopsys proposed a methodology called Links to Layout (LTL). Based on this methodology, OKI developed an interface between OKI's Floorplanner and the Synopsys environment, called Link Synopsys to Floorplanner (LSF).

*Figure 20*below shows the design flow for the LSF system. As not every Synopsys user has access to the Synopsys Floorplan Management tool, OKI had developed the LSF system to support both users who can access Synopsys Floorplan Management and users who do not have access to Synopsys Floorplan Management.

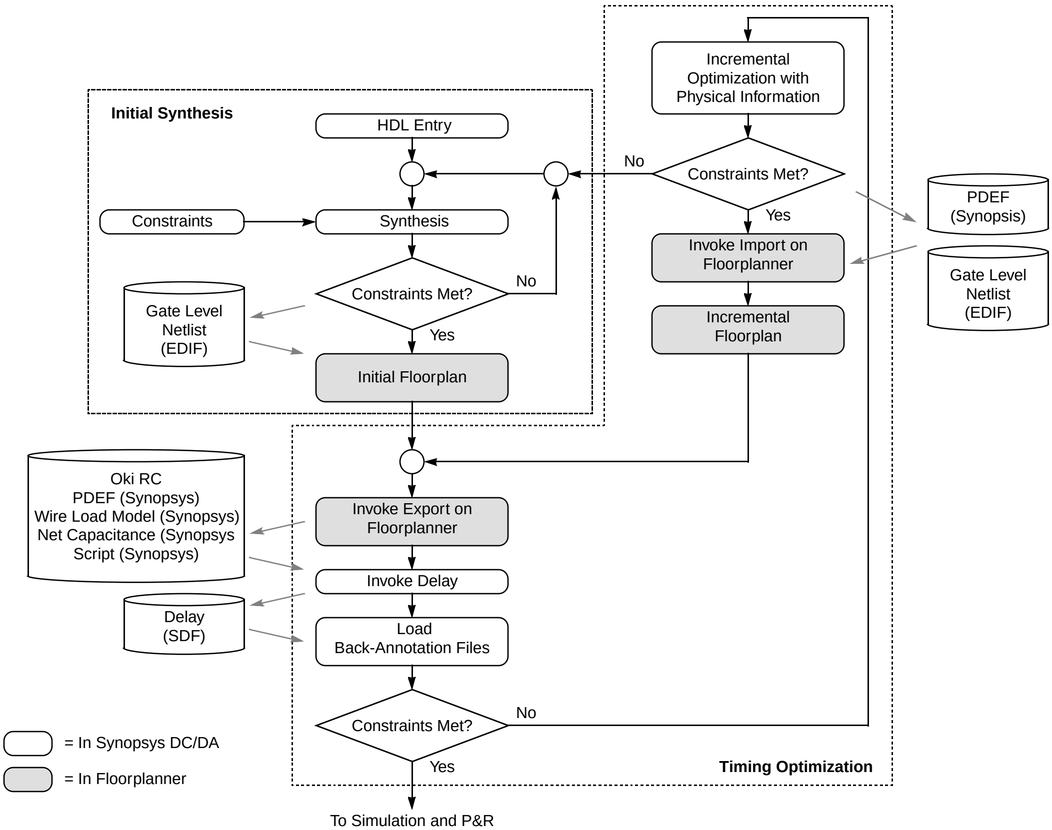

Figure 20. LSF System Design Flow

Using OKI's Floorplanner with Synopsys Floorplan Management

OKI has adapted its Floorplanner to interoperate with the Floorplan Management facilities available in the Synopsys synthesis environment. The Synopsys Floorplan Management facilities allow for netlist optimization in accordance with physical block placement data extracted from an external floorplanner. In principal, design flow iterates between floorplanning and synthesis to provide a smaller or faster design.

There are two major segments to the process:

- Initial synthesis and floorplanning
- verification and reoptimization

Notes on using OKI's Floorplanner with Synopsys Floorplan Management follow.

Initial Synthesis and Floorplanning

First use the Synopsys Design Compiler to create an optimized EDIF netlist.

1. Read HDL and design constraints into the Synopsys Design Compiler.
2. Check timing with the "report_constraints" and "report_timing" commands in Design Compiler.
3. Write out EDIF netlist with the following command:

```
write -hierarchy -format edif -output <LSI_CODE>.edif
```

After EDIF netlist creation, use OKI's Floorplanner for initial floorplanning, and export back-annotated netlist timing data.

4. Perform normal floorplanning, as described in this application note, and generate RC estimates for each net.
5. Use the Synopsys "export" command to create back-annotation files. Answer "y" at the ensuing dialog prompt (see *Figure 21* on the following page for an illustration of the "export" dialog box).

OKI's floorplanner generates the following five files:

- `rcEst` – Estimated parasitic capacitance is contained in this file, which is later used by OKI's Delay Processor to generate an SDF file for Synopsys back-annotation.
- `pdefPega` – Cluster information in this file is used by Synopsys Floorplan Management during the optimization process.
- `wireLoad` – Wire load models for each cluster are available in this file.
- `netPara` – Estimated parasitic capacitance is described as a set of "set_load" commands, used by the Synopsys tools. This file contains the same information as in `rcEst`, but is formatted for Synopsys rather than OKI software usage.
- `dcScr` - This shell script is also produced by the export process to load `pdefPega`, `wireLoad`, and `netPara` into Synopsys.

At this point initial design is complete. OKI's additions to the Synopsys "export" process have generated all the data necessary for verification, reoptimization, and resynthesis.

```
%%BoundingBox: 3 329 539 825
%%Title: export
%%CreationDate: Thu Jun 15 15:55:43 1995
%%Creator: Tgif-2.13J2-p2 by William Chia-Wei Cheng (william@cs.UCLA.edu)
%%Pages: 1
%%DocumentFonts: Times-Roman Helvetica Courier NewCenturySchlbk Symbol
%%EndComments
%%BeginProlog
%
0 0 translate 0.5 0.5 scale

%       Due to bugs in Transcript, the 'PS-Adobe-' stuff is omitted from line 1
%

/tgifdict 132 dict def
tgifdict begin

%
```

Figure 21. Export to Synopsys Command

Verification and Reoptimization

After initial floorplanning, the Synopsys timing analysis tools can identify timing violations and reoptimization can be performed as required.

1. Invoke OKI's Delay Processor to generate the SDF file and read the SDF file into Synopsys, using the "read_timing -format_sdf" command. Then perform timing analysis using the "report_constraints" and "report_timing" commands in the Synopsys Design Compiler.
2. Use the incremental reoptimization commands in Synopsys to optimize timing. First use the dcScr file generated by OKI's Floorplanner to load the PDEF file, wire- load models, and net capacitance into Synopsys. Then use the following commands in sequence to optimize the circuit.:

```
reoptimize_design -in_place
reoptimize_design -post_layout_opto
reoptimize_design -post_layout_opto -tolerance_to_change high
reoptimize_design -post_layout_opto -tolerance_to_change high -map_effort high
```

3. After fixing all timing violations, generate the EDIF netlist and PDEF information with the following commands:

```
write -hierarchy -format -edif -output <LSI_CODE>.edif
write_clusters -output <LSI_CODE>.pdefSyn
```

4. Copy the new EDIF netlist and put the PDEF file in the floorplanner interface directory. Invoke OKI's Floorplanner to import the new EDIF netlist and cluster information. Answer "y" in response to the dialog prompt, as shown in *Figure 22*below.

Oki's Floorplanner then reoptimizes the physical block layout in accordance with the new PDEF constraints. Verification, reoptimization, and incremental floorplanning can be repeated as necessary.

%%DocumentFonts: Times-Roman Helvetica Courier NewCenturySchlbk Symbol

%%EndComments

%%BeginProlog

%

% Due to bugs in Transcript, the 'PS-Adobe-' stuff is omitted from line 1

%

0 0 translate 0.4 0.4 scale

/tgifdict 132 dict def

tgifdict begin

%

% Using a zero value radius for an ellipse or an arc would result

% in a non-invertible CTM matrix which causes problem when this

Figure 22. Importing EDIF Data into OKI's Floorplanner from Synopsys

Using OKI's Floorplanner Without Synopsys Floorplan Management

If floorplan management is not available within the Synopsys synthesis environment, only wire-load models and net capacitances can be back-annotated, from OKI's Floorplanner to Synopsys, for timing analysis.

To compensate for this reduced accuracy, OKI's Floorplanner generates new wire-load models based on the regions that each net crosses and passes this information to the Synopsys tools. OKI's Floorplanner generates timing estimates based on the chip floorplan, whereas Synopsys Design Compiler estimates are based on statistics. As a result, the wire-load models are more accurate than the defaults used by Synopsys.

Initial Synthesis and Floorplanning

Perform initial synthesis and floorplanning exactly as described in the previous section until the last step, when it is time to export timing data from Synopsys and create back-annotation files. Then, in response to the "export" dialog box shown in *Figure 21*, answer "n" instead of "y". This will produce the following four files (see the previous section for a description of their functions):

- `rcEst`
- `wireLoad`
- `netPara`
- `dcScr`

The `pdefPega` file is not produced at this time because, without Synopsys Floorplan Management, cluster information is not available.

Verification and Reoptimization

Use the following procedure to import region-based timing data from OKI's Floorplanner into the Synopsys environment.

1. As before, invoke OKI's delay processor to generate the SDF file, and read the SDF file into Synopsys using the "read_timing -format sdf" command. Also as before, perform timing analysis using the "report_constraints and "report_timing" commands in Synopsys.
2. If there are timing violations, use the incremental reoptimization commend in Synopsys to optimize the timing. The reoptimization command is:

   ```
   compile -in_place
   compile -incremental_mapping
   ```

3. After fixing the timing violations, generate a new EDIF netlist with the following command:

   ```
   write -hierarchy -format edif -output <LSI_CODE>.edif
   ```

4. Copy the new EDIF netlist to the floorplanner's interfaced directory. Invoke OKI's floorplanner to import the new EDIF netlist. Answer "n" to the "import" dialog box query (shown in *Figure 22* above).

Proper use of these procedures can ensure a far superior design than possible without floorplanning. The increased accuracy yielded in the timing estimates generated by floorplanning can additionally reduce design iterations from routing to synthesis, thus reducing design time and improving time to market.

Contact OKI's Application Engineering Department for more information on OKI's Floorplanner.

Trademark Acknowledgments

Sun is a trademark of Sun Microsystems.

RS/6000 is a trademark of IBM Corporation.

Synopsys is a registered trademark of Synopsys, Inc.

All other products or services mentioned in this document are identified by the trademarks, service marks, or product names as designated by the companies who market those products. Inquiries concerning such trademarks should be made directly to those companies.

OKI SEMICONDUCTOR

OKI Regional Sales Offices

Northwest Area
785 N. Mary Avenue
Sunnyvale, CA 94086
Tel: 408/720-8940
Fax: 408/720-8965

Southwest Area
2302 Martin Street
Suite 250
Irvine, CA 92715
Tel: 714/752-1843
Fax: 714/752-2423

Central Area
2007 N. Collins Blvd.
Suite 303
Richardson, TX 75080
Tel: 214/690-6868
Fax: 214/690-8233

Southeast Area
1590 Adamson Parkway
Suite 220
Morrow, GA 30260
Tel: 404/960-9660
Fax: 404/960-9682

Eastern Area
Shattuck Office Center
138 River Road
Andover, MA 01810
Tel: 508/688-8687
Fax: 508/688-8896

Automotive Electronics
17177 N. Laurel Park Drive
Suite 433
Livonia, MI 48152
Tel: 313/464-7200
Fax: 313/464-1724

FOR OKI LITERATURE:
Call toll free 1-800-OKI-6388
(6 a.m. to 5 p.m. Pacific Time)

OKI Stock No: 030301-002

OKI Semiconductor

Corporate Headquarters
785 N. Mary Avenue
Sunnyvale, CA 94086-2909
Tel: 408/720-1900
Fax: 408/720-1918