Methodology for Automated Layout Migration for 90nm Itanium®2 Processor Design

Kuang-Kuo Lin, kuang-kuo.lin@intel.com
Sudhakar Kale, sudhakar.kale@intel.com
Aditi Nigam, aditi.nigam@intel.com
Intel Corporation, USA

Abstract

Unlike past semiconductor manufacturing processes, nanometer technology has seen an exponential growth of complex design rules and constraints. As a result, direct optical shrink of products between process nodes is becoming less feasible. To facilitate aggressive time-to-market requirements by re-using designs under new process technology, new CAD automation tools and methodologies have been developed. This paper describes a process shifting flow of 130nm custom layout to 90nm. Design challenges at the new process will first be overviewed, followed by EDA-assisted layout migration. Finally, productivity gains together with the design qualities will be shown on the implementation of a next generation Itanium®2 server chip.

1. Introduction

Custom layout design is a craft and is very time consuming. Hence, design houses like to reuse the critical blocks between multiple designs. However, with every new manufacturing process, the design rules become more stringent and hence migrating layout between processes is becoming an ardent task. The layout migration from 180nm to 90nm process [1] proved to be a unique challenge compared to past migrations [2]. Unlike past processes, there is no simple direct optical shrink path due to the complex design rules. In order to optimize the layout for sub-100nm processes, special attentions are needed for critical layer orientations, non-orthogonal routings and small jogs as cited in [3].

This paper describes tools and methodologies developed to assist, through automation, the custom layout process migration. Our study has demonstrated at least a 50% productivity gain on migrating the custom cell layouts using these techniques. The methodology has been applied to next generation Itanium®2 processor with a moderate amount of logic changes. The original source layout is particularly dense and has extensive usage of non-orthogonal routings and devices oriented in both directions. One of the challenges for layout migration is to preserve most of the original layout while meeting the complex migration to 90nm layout design rules.

The paper is organized as follows: Problem Statement section establishes the problem we are solving, CAD Tools and Flows section explains the general migration flow and its integration, and we conclude the paper with Results and Future Opportunities.

2. Problem Statement

Semiconductor Industry Association (SIA) predicts that 50% of new designs in the next two years will be based on previously implemented designs [4]. Inline with those projections, Itanium®2 design has borrowed architectural and physical design from the prior generation. After evaluating the functional changes required in the light of project schedule, scope and resource needs, re-use of our existing design turned out to be the most critical task. For, Itanium®2, re-drawing the layout would take about 1/3 of the total designer effort for the given tape-out schedules. Hence, a new automated methodology was needed for productivity enhancement by porting existing layouts to the new 90nm process. This coincides with the view that layout migration technique is becoming critical for design re-use and process shifting [2] [3]. Key challenges involved in the layout porting from 180nm to 90nm Intel manufacturing process are:

2.1. Complex Design Rules

Some new design rules in 90nm technology are: (a) Transistor gate orientation (b) Non-orthogonal routes (c) Jogged lines (d) Metal coverage/landing of vias (e) APSM and OPC-based rules (f) Gate/Well edge spacing rules (g) Metal and via density rules.

2.1.1. Transistor rotation. The new process dictates specific device gate orientation. In addition to meeting this new design rule, specific cell size has to be satisfied.

2.1.2. Handling non-orthogonal routes. The process has new requirement on non-orthogonal routings. Automation is needed to map non-orthogonal routings in the old process to orthogonal routings.
2.1.3. **Via optimization.** Design rules associated with via isolated pairs and orthogonal rules offer challenges of optimizing the number of vias and vias surrounding another via.

2.2. Chip Assembly

2.2.1. **Preserving cell dimensions.** To reduce the Place-and-Route effort later and to enable easy layout tiling, it is important to keep the ratio of new cell dimensions proportional to the original cell. But, due to the new design rules, there is a tendency for the new cells to grow taller.

2.2.3. **Preserving port locations.** Again, for the effective reusability of layout (especially, hook-up to upper-level routes), it is very important to retain the relative port locations in the migrated blocks.

2.2.4. **Preserving legacy routes.** Since there was already significant amount of time spent during the old chip design to ensure the routings are RV (reliability verification), noise and timing-correct, the effort should not be duplicated. In other words, retaining the proportionality of existing higher layer routing is very important.

2.2.5. **Pitch based compaction.** For both library and custom leaf cells, the cell growth is driven by the metal layer pitch in the particular direction to have a tidy layout. For example, if M2 runs in X direction, then Y growth should be snapped to the nearest M2 pitch grid. Also, the contacts are to be on a particular grid so that when the metal above comes into the cell, there won't be via spacing rule violations.

2.2.6. **Template based porting (abutment at the boundary,½ design rules at the boundary, pitch constraints).** Leaf cells typically abut to one another to share power rails, clock nets and other global nets. Also, there are ½ design rules, which help the spacing between poly, metal 1 to be correct when cells abut to each other.

2.3. Methodology Related

Though reuse methodology claims adopting the existing methodologies seamlessly, having a “development” version of layout view would open changes to the existing methodologies until the migrated layout becomes “production”. Needless to say, whenever there is a new tool, new methodologies around that tool and integration methodologies abound.

3. CAD Tools and Flows

In order to expedite the design migration from 180nm to 90nm, radical new CAD tools and flows are needed. To address the challenges mentioned in the previous sections, the new CAD methodology must produce good layouts under the complex 90nm design rules. The new methodology also needs to be well integrated with other layout tools and the underlying design methodologies. In addition, the new CAD tools/flows should be user friendly to facilitate user controllability and productivity. In this section, we outline the investigations on tool selection, general migration flow and system integration.

3.1. Tools

The first step is to identify/develop the appropriate CAD tools. For the new design, different layout migration tools were investigated that can compact layout in two dimensions [5] and finally a flat migration tool was selected. With the mask designer desire for interactive porting/correcting the cells, flat migration at the leaf cell level (both custom and standard) offers the best compromise between productivity, ease-of-use, and resource-constraint.

The challenge for leaf cell migration is broader than compacting just the cell itself and includes enabling the assembly at higher hierarchies as well. Complex migration constraints, in addition to the new design rules, are needed to preserve the cell footprint with respect to size, internal routing, cell interface and pitch. In addition to the migration tools, a new automatic transistor rotation tool has also been developed to complement the migration process.

3.2. Flows

The next step is the flow development and integration based on the selected tools. The basic migration flow involves organizing the input layout, front-end and back-end data translations, and post-processing of the output data as needed. The basic flow is shown below:

![Figure 1. 180nm-90nm migration flow](image)
Some simple cells may require no cleanup, while individual cell characteristics and the migration flow options desired. The degree of cleanup effort strongly depends on tool is over-constrained and could not complete the migration as cleanup is needed if the user is not satisfied with the result or the tagged with the correct sizes.

The rotation algorithm has these steps – (a) remember the original area occupied by devices to be rotated (b) remove unwanted connections (c) rotate devices (d) shrink devices to fit in original area with routing (e) annotate devices with original size (f) route the connections with less than minimum width wires and vias. When the layout after rotation is passed to the migration tool, it can change device sizes to targeted sizes based on netlist annotations, and resize wires and vias with respect to the process rules.

To accommodate the uniqueness of each custom cell and its unique assembly requirement, special interactive GUI migration directives are developed in the layout editors. For example, special user constraints on cell x/y dimensions and layer width/spacing can be specified. Also, localized migration can be specified in ECO/freeze windows, where migration/no-migration of specified layers is allowed in these windows.

3.2.2. Layout migration. Once the source layout is pre-processed, it is ready for migration. The migration step involves several modular processes, outlined in the sub-steps within the generic migration flow above. First, the source layout is pre-scaled to be close to the target layout. Layer arithmetic operations are then performed to create help layers to aid the migration. Source contacts are merged into population rectangles, which are later re-sized and re-filled after migration with as many vias as can be fit. Devices are recognized and tagged with the correct sizes.

Abutment relationship of the migrating cell can be specified and proper abutment rules (e.g. 1/2 DR) will then be created. The main compaction engine will finally be invoked. After compaction, contacts, notches, and devices are properly post-processed and checked for the target design rules. Also, special migration rules are created to handle transistor rotation, cell size constraint, port preservation, legacy route proportionality (for RV), directional via optimization and etc. Details are discussed in the next section.

3.2.3. Migrated layout post-processing. Post-migration layout cleanup is needed if the user is not satisfied with the result or the tool is over-constrained and could not complete the migration as desired. The degree of cleanup effort strongly depends on individual cell characteristics and the migration flow options. Some simple cells may require no cleanup, while moderate/complex cells will need to remove DRC errors and to constrain cell size/interface. Nevertheless, any post-migration manual cleanup effort is significantly less than manual re-layout effort from scratch. Our productivity study shows at least a 50% gain using the CAD migration flow, the details of which are furnished in the Results section.

3.2.4. Integration. Significant effort was invested in integrating the migration tool/flow with the production layout environment.

First, the custom layout editor was tightly integrated with the migration tool. The integration allows source data to be interactively pre-processed and annotated with special instructions before sending off for layout porting. The layout editor is enhanced to have a migration commands palette so the mask designer can invoke the migration tool directly.

Once the layout is migrated, the data can be imported seamlessly back into the layout editor for clean up. In addition, special scripts were developed to back-annotate migration tool outputs into the migrated layout so that designers can view over-constraints and rules that were violated during the compaction inside the layout editor.

A unique window-based ECO capability has also been implemented and has become a popular feature. The user can draw an ECO window in the layout editor and specify if it is Freeze ECO or Allow ECO. In the case of Freeze ECO, polygons inside the window are not touched during migration, and only the polygons outside the window area are allowed to move. If a polygon cuts the ECO window, then special constraints are imposed while migrating those polygons. It is not necessary that all the layers inside the Freeze ECO be frozen - users have flexibility to identify what layers are to be frozen. Allow ECO works the exact opposite way.

4. Results

A productivity gain study has been performed on the migration flow. Test data from simple to complex cells was examined. Manual re-draw of the layout by the mask designers still generates the highest quality layout in terms of area density. Thus the full manual layout effort is used as the best-case and baseline. Different migration flows with various degrees of automation were compared with the reference manual result.
Table 1. Productivity gain study on a cell from Integer Datapath unit

<table>
<thead>
<tr>
<th></th>
<th></th>
<th></th>
<th></th>
<th></th>
</tr>
</thead>
<tbody>
<tr>
<td>90nm DRC error</td>
<td>4</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LVS error</td>
<td>0</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Area (%)</td>
<td>119</td>
<td>112</td>
<td>100</td>
<td>100</td>
</tr>
<tr>
<td>Total time (min)</td>
<td>4</td>
<td>4 + 100 = 104</td>
<td>23 + 4 + 100 = 127</td>
<td>247</td>
</tr>
</tbody>
</table>

In Table 1 above, a simple cell from Integer unit is presented. A full manual effort took about 247 min (~4hrs) to achieve 90nm-clean design, and to achieve the desired process shrink factor. A full automatic rotation and migration (with manual clean-up) only took ~4 minutes, but resulted in 19% area growth and 4 design rule errors. Another 100 minutes was spent to clean up the port interfaces as well as to reduce the area growth (now to 12%). When manual device rotation (23 mins) was used instead of the automatic rotation in the pre-processing step, DRC/LVS clean layout resulted in same area as full manual re-draw. The total time for this semi-automatic flow was 127 mins. Compared to the full manual effort (247 mins), there was about a 50% productivity gain!

Table 2. Productivity gain study on a cell from Floating Point unit

<table>
<thead>
<tr>
<th></th>
<th>Full Auto rotation/ migration</th>
<th>Full Auto rotation/ migration + Manual Cleanup</th>
<th>Full Manual</th>
</tr>
</thead>
<tbody>
<tr>
<td>90nm DRC error</td>
<td>120</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>LVS error</td>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>Area (%)</td>
<td>179</td>
<td>99</td>
<td>100</td>
</tr>
<tr>
<td>Total time (hour)</td>
<td>0.08</td>
<td>0.08 + 22 = 22.08</td>
<td>40</td>
</tr>
</tbody>
</table>

In Table 2, data from a more complex cell from the Floating Point Unit is presented. A full manual re-draw took 5 full working days (40 hours) to attain 90nm-clean design and is used as the best density baseline. It was an extremely difficult challenge to achieve the desired process scaling given the need to satisfy the new 90nm rules. A fully automated rotation/migration took 4 minutes (0.08 hour), but at the expense of 120 DRC errors and 179% area growth (vs. full manual)! Manual cleanup had to be done to reduce the migrated area, as well as to preserve the cell interface. Total time spent on auto flow with manual edit was 22 hours (2 working days and 6 hours). Compared to 5 days of manual effort, there was again of about 50% productivity gain with slightly smaller cell area!

Additional results are shown in the figures and table below. The migration flow was enhanced with additional abutment and area controls. With the additional options, abutment relationship with respect to neighboring cells was maintained. By constraining dimension changes, the cells could be fit into the original floorplan (e.g. bit-slice structure) in upper hierarchy with relative ease. For example in a simple column buffer cell the migrated cell maintained the same cell dimension and virtually all critical routings/ports in the same locations with only four 90nm DRC errors. For a complex adder-multiplier cell in Floating point unit, the cell was allowed to grow in the X dimension but needed to be fixed in the Y dimension (to match routings/ports in horizontal direction). The migration flow was able to produce a 90nm layout with only 5 DRC errors.

Fig 4. Column buffer (custom leaf) migration results
5. Future Opportunities

A novel interactive leaf-cell layout migration methodology has been developed for hierarchical custom cell design. This methodology has demonstrated that design re-use is feasible with significant productivity gains on an actual 90nm product implementation. As we worked through the migration flows and tools, there are abundant opportunities that we identified that have huge productivity benefits. Listed below are a few candidates for future research.

*Hierarchical migration* can give additional cost benefits. Instead of the process shifting individual leaf cells and re-assembling them to create a new custom layout, hierarchical migration completes the whole layout in a single run. In addition to eliminating the need for re-assembly, hierarchical migration has the full design context so there is no need to assume neighboring cell relationships. However, hierarchical full-block migration has its own set of design challenges that will be discussed in future papers. To further productivity, the migration flows can be further used to draw a loose layout with directives and the migration flow would create the desired target layout.

6. Acknowledgements

The authors would like to acknowledge all the help and suggestions through out the migration flow development. Specifically our thanks go to Eric Fetzer, Kevin Jones, Kao Lee, Kamlesh Kumar, and Blaine Larson during this project.

7. References


