Performance Impact from Metal Fill Insertion

Jayathi Subramanian, Ruiqi Tian, Mike Kobe, Scott Hector

Freescale Semiconductor

Session #4.2 (Physical Design)
CDNLive! 2007

Performance Impact on a design from Metal Fill Insertion

Jayathi Subramanian, Ruiqi Tian, Mike Kobe, Scott Hector
Freescale Semiconductor

Introduction

Variations in dielectric thickness reduce yield and affect performance of circuits. Floating metal fill or dummy features are added into the design to reduce variation in dielectric thickness due to dishing and erosion, as well as to increase planarity and maintain nearly uniform pattern density on the scale of the chemical mechanical planarization (CMP) process.

Metal fill is typically added to design data during chip finishing just before tapeout using Physical Verification tools, although it may also be added after tapeout. At this late stage, the focus is on the lithography requirements and ignores the interconnect capacitance implications. Moreover, the fill added at this stage of the design process must follow conservative rules to not disrupt timing. This is no longer acceptable at geometries of 90nm and below. Dummy metal fills are required to be placed close to signal nets to meet minimum density requirements and may affect circuit performance to a larger degree. Moreover, designs have multi-voltage areas and it gets harder to code this in Physical Verification tools for doing area-based tied fill insertion.

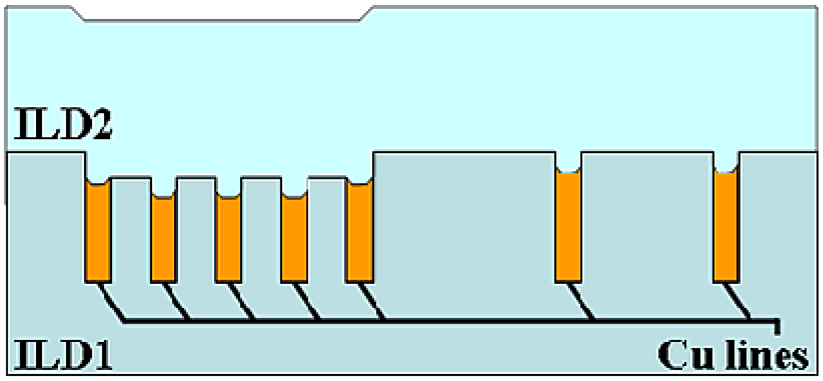
A more robust process would be to add metal fill before timing and power closure, enabling the use of metal fill to help meet power targets as well as allowing the fill to be more aggressive in meeting pattern density targets.

The paper describes a method of dummy metal fill insertion in the SoC Encounter system as a post-route optimization followed by extraction, using Cadence QRC Extraction, in the presence of fills, for use in timing closure iterations. We discuss insertion of Vdd and/or Vss mesh as a post-route timing-aware optimization operation in the SoC Encounter system, but before insertion of floating metal fill. We also discuss using

Cadence Chip Optimizer as an alternate method for metal fill, and detail the tradeoffs between using Cadence Chip Optimizer and SoC Encounter for metal fill insertion.

Background Information

For 130nm and lower geometries, many semiconductor manufacturers moved from aluminum to copper as the interconnect metal of choice because of lower resistivity and higher reliability of copper. Unlike aluminum, for copper process, metal traces are etched onto the interlayer dielectric and copper is electroplated into the newly created trenches. Excess copper is removed in a CMP step. CMP removal rate are a function of local interconnect density. Regions with wide metal lines may experience dishing effects whereas closely spaced lines may experience inter-metal dielectric erosion effects.



To avoid these problems, many manufacturers began to insert dummy metal fill to even out the interconnect pattern density. The dummy metal fill, consisting of tiles in empty areas of chips, is inserted as a post-processing step.

However, metal fill affects chip timing, signal integrity and even functionality. Tight control of planarity requires dummy tiles to be placed in close proximity to functional features. This may cause coupling with functional wires – and create additional parasitics. The existence of the tiles changes capacitance. CMP dishing changes wire resistance, and dielectric thickness variations also change signal parasitics. If not modeled correctly, these can directly affect yield and performance of the circuit.

One side benefit of using tied metal fill is to be able to create a redundant mesh that can carry current and hence help with IR-drop problems. However, as with floating fill, this has to be done with caution so as to not cause timing degradation.

Previous solution for metal fill insertion

Historically, dummy metal fill had been added to a GDSII database using Physical Verification tools. With shrinking geometries, design rules are more complex. So, coding up the metal fill geometries, especially the tied metal fill is pretty cumbersome. Moreover, since there in no notion of timing/SI in Physical Verification tools, metal fill insertion has to use very conservative rules and place them really far from all signal nets. Finally, parasitic extraction using GDSII is cumbersome and making any design changes (to compensate for timing degradation) at this late-stage is difficult.

Proposed new method

We propose that the dummy fill insertion take place in a P&R tool as a post-route optimization step but before the final timing/power closure runs. Insertion of dummy metal fill as a post-route optimization allows designers to not block any routing channels until design is fairly mature. Pulling in the metal fill insertion step earlier in the design flow will allow designers to adjust the design while taking dummy fill into aacount. Designers will have more control of where the metal fill is placed and it can be done in a timing aware mode.

Using Encounter for metal fill

Metal fill features in Encounter

Encounter has a set of flexible features that enable metal fill optimization:

- Can use square or rectangular shaped tiles: Options are available to pick a range of sizes. The tool starts with the largest tile sizes specified and moves to smaller sizes once the large tiles cannot fit in that space. Options are available to specify tile spacingto active metal segments and tile-to-tile spacing as well, per layer basis. Finally, option is available to specify min, max and preferred density.
- Can add staggered or non-staggered tiles: Staggering metal fill spreads out the effects of cross-coupling capacitance because the staggered pattern ensures that the metal fill does not line up orthogonally on adjacent layers.
- Can be tied-off or left floating: Metal fill segments can be connected (tied-off) to power or ground shapes on adjacent routing layers or left unconnected (floating). For metal fill tie-offs, the software creates vias that fit within the area where the metal segment overlaps with a power or ground shape on an adjacent routing layer. It also has the ability to create a mesh such that the tied metal fill can carry current. Also, there is an option to remove floating metal fill for fill that could not be tied off. However, this will make it more difficult to reach preferred density requirements.
- Floating metal fill is easier to trim when there are violations using the command *trimMetalFill*. If there is tied-off metal fill, they must be deleted manually to avoid problems with vias. Also, ECO in which some layers are frozen will be problematic and will have to be done with care when using tied-off metal fill.
- Can be added in timing aware or non-timing aware mode: When Encounter adds timing-aware metal fill, the software avoids adding fill near clock and signal nets and adds more fill near power and ground nets. Fill is added on cost-basis where the clock nets have the highest cost, signal nets have moderate cost and power and ground nets have zero cost. Timing-aware metal fill is on by default. Other options available are as follows:
  1. Timing-aware mode OFF
  2. Timing-aware mode with CTE run to determine critical nets. In this mode, we need to run a *buildTimingGraph* CTE run and then give a slack threshold to the *addMetallFill* command. Encounter will then partition nets with descending cost-basis as follows: clock nets, signal nets that violate slack

threshold value, other signal nets and finally power-ground nets with zero cost.

- Can do multiple passes by scripting in Tcl: Using *verifyMetalDensity* and *editDelete* command, we can script up multiple passes to be used for Tiling. Windows that fail minimum density violations can be extracted from the report file and all tiles can be deleted from that window before a second pass is applied with a bigger range of tiles.

Software version to use

Encounter 6.1usr3 – dummy floating and tied fill insertion
QRC 6.2 – parasitic extraction with floating fill models

Flow and commands to use

Once you have a routed design, that is relatively clean (drc/lvs), you can add metal fill in power-strapping mode, followed by dummy metal fill insertion and run timing analysis before moving on to chip-finishing phase that would involve addition of artifacts, custom notch-fill, GDSII based via optimization, GDSII based dummy fill run for fixing minimum density violations and adding poly/active tiles and finally GDSII based physical verification drc/lvs run. If ECO's are required during the timing closure phase, you can use *trimMetalFill* command to remove violating metal fill shapes.

These are the commands that are used often in this flow:
setMetalFill : setting fill parameters
addMetalFill –mesh : adding metal fill in power-strapping mode
addMetalFill : adding floating fill
editDelete : deleting fill from an area, use –shapes FILLWIRE
trimMetalFill : trimming metal fill for DRC violations after an ECO
verifyMetalDensity : verifying metal density
verifyConnectivity : check for shorts
verifyGeometry : check for opens and DRC violations,

Implementation details

VDD/VSS metal fill mesh insertion

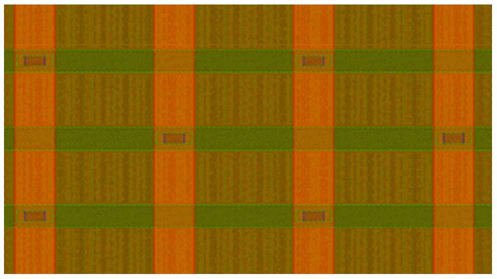
Encounter has the capability to add timing-aware metal fill in open spaces of the design in a mesh mode. This mesh can be tied to power or ground signals. For designs with multi-voltage supplies, area-based meshes can be added to the design. For metal-fill in the mesh mode, it is recommended that long fill lines be used (> 10 microns) and widths about 2-3 times the minimum width. Encounter will add maximum possible number of cuts in vias, based on the intersection area. Some design teams use alternating Vdd/Vss mesh segments to help with dynamic IR drop issues.

The Vdd/Vss mesh procedure uses metal-fill optimization commands in timing-aware, power-strapping mode within Encounter (*addMetalFill –mesh, setMetalFill, verifyMetalDensity*) to add redundant connections between metal-fill segments and existing power meshes.

The steps involved are listed below:

1. Partition the die area into regions corresponding to the voltage domains.
2. Use large window sizes to enable the tool to add long wire segments.
3. Set the minimum density to a high value (50% was chosen).
4. Do a loose vss (or appropriate ground signal) flood.
5. On the same area, do a tighter vdd (or appropriate power signal) flood.
6. Complete all the partitions in a similar manner and then finally flood the entire die area with the generic power and ground signals.
7. Use *verifyGeometry* and *verifyConnectivity* to check for open/shorts and drc violations.
8. Retime design and do any fixes required.

The Vdd/Vss metal fill shapes, as also the VIAFILL shapes connecting tied-off metal fill shapes needs to be mapped to regular drawn metal/via layers when writing out GDSII. This is to ensure that design rule checks are performed on these newly added shapes by treating them as regular signal nets.

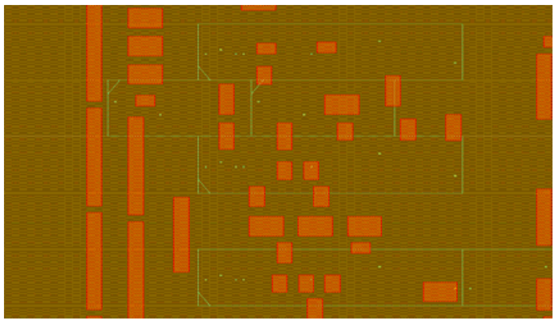


Floating metal fill insertion:
Encounter has the ability to add floating metal fill, squares and/or rectangles, in a staggered manner. This can be done in "timing-aware" mode (by default) and timing analysis can be easily performed after metal fill insertion within Encounter. Controls are available to space tiles from signal nets and to set preferred density for each layer. The common commands used are "*addmetalFill*", "*setMetalFill*" and "*verifyMetalDensity*". Encounter has the capability of performing an ECO in the presence of floating metal fill. Just do the ECO re-route; Encounter will complete the route as if no fill was present. Next use "*trimMetalFill*" command to remove any floating fill that is causing drc/lvs violations with the eco route.

The steps used are as follows:
Using windows-based approach, specify min density, max density, tile-size, shape and distance between active signals and fill shapes using *setMetalFill* command.

1. Start with big square tiles to fill open spaces.
2. Run *verifyMetalDensity* to capture violating windows.
3. Remove metal fill from these violating windows and fill with rectangular shapes.
4. Do multiple passes if required.

Floating metal fill needs to be mapped to a special fill data-type when writing out gds from Encounter. This is to ensure that the design rule checks specific to floating metal fill (tile size, shapes, min area) can be run on the gds out of Encounter.



Parasitic Extraction using QRC:
QRC 6.2.0-p011 has the capability of extracting parasitic capacitance and resistance in the presence of floating and tied-off metal fill.

Metal fill data can be imported into the design with the "input_db" command either through DEF or GDSII file input.

Metal fill that is defined in the "FILLS" section of the input DEF or in an input GDSII file will be considered floating by default. QRC has a command called "metal_fill" that can be used to control how we want the fill in the "FILLS" section to be considered: floating or grounded. Floating metal fill requires special floating metal models in the technology file to allow QRC to evaluate the metal fill objects properly during extraction.

Metal fill can also be specified in the SPECIALNETS section of the DEF input data, with the +SHAPE FILLWIRE construct. This type of fill metal (basically the Vdd/Vss metal fill mesh shapes) is always considered grounded, and is not affected by the "metal_fill" command. Grounded metal fill act as gray data and this allows QRC to evaluate the metal fill objects without the need of any special models.

Writing out GDSII for final Physical Verification:
Usually, Physical Verification decks differentiate between tied-off and floating metal fill. Tied-off metal fill (Vdd/Vss) needs to be mapped to regular drawn metal fill layers. Floating metal fill needs to be mapped to a different layer; data type. This is because foundries have Design rule checks specific for floating metal fill. (e.g min area, tile sizes etc).

In order to be able to achieve this, we need to add Vdd/Vss metal fill mesh on a different data-type. With the current version of Encounter, this is achieved by using a special config file for the "*addMetalFill*" command.

Create a file called fillopc.cfg and add the following line:

```
srouteFillOPC true
```

# the following will create FILLWIREOPC data-type for the mesh (tied-off fill) and FILLWIRE for the floating shapes.

# Add power mesh

setMetalFill -layer {1 2 3 4 5 6} -gapSpacing 0.4 -maxWidth 0.4 -minWidth 0.4 -minDensity 50.000 -maxLength 4000 -preferredDensity 50.000 -activeSpacing 0.6 -minLength 10
addMetalFill -layer { 1 2 3 4 5 6 } -nets { VSS VDD } -mesh -extraConfig fillopc.cfg -removeFloatingFill

# Add floating fill

setMetalFill -layer {1 2 3 4 5 6} -gapSpacing 0.4 -maxWidth 0.4 -minWidth 0.4 -minDensity 15.000 -maxLength 10 -preferredDensity 40.000 -activeSpacing 0.6 -minLength 1
addMetalFill -layer { 1 2 3 4 5 6 }

An Enhancement PCR has been filed **PCR #4022052** (SR #40698070) to address the need where connected fill needs to be output to regular metal whereas floating fill needs to go to another data-type, using a gdsii map file, instead of using the workaround described above.

Experimental Results

Design 1:
Design Stats: 65nm core, 6-layers of metal, 2.65mmx2.65mm, not yet taped out
Flow used: Base P&R->Vdd/Vss alternating metal fill mesh->Floating fill
Machine Stats: x86_64, AMD Opteron, 16G RAM, 4-processor
Run Time:
    Vdd/Vss mesh – **13** hours
    Floating metal fill – 6 hours
DRC :

- Violations for one DRC rule (min #of vias at wide-metal intersection) (PCR filed)
- Min density violations (M3) seen under hard-blocks; fixed with final gdsii based tiling run
- Min density violations (M2) seen in sea-of-gates area; fixed with final gdsii based tiling run

LVS: No issue
Timing:

Step	**Setup WNS**
Base P&R	-365ps
Above + Vdd/Vss Mesh Fill	-367ps

Above + Floating Fill	-378ps

- There is 50ps of clock uncertainity
- Design target of 500 Mhz met

Design 2:
Design Stats: 130nm, 6 levels of metals, 8.8x8.8mm^2, taped out May 2007
Flow used: Base P&R -> Vdd Mesh Fill -> Via Optimization (CCO) -> Floating Fill
Machine Stats: x86_64, AMD Opteron, 16G RAM, 4-processor
Run Time:
Vdd mesh – **13** hours
Floating Fill – 3 hours
LVS: No issues
DRC:
Min density violations in region with overlapping LEF's
Min density violations
Timing:

Step	**Setup WNS**	**Hold WNS**
Base P&R	+6ps	+11ps
Above + Vdd Mesh Fill	-42ps	-11ps
Above + Via Opt	-45ps	-11ps
Above + Floating Fill	**-249ps**	-11ps

- There is 50ps of clock uncertainty
- Timing run at 130MHz, still met design target of 120MHz
- Had this been a critical net, an ECO could have been performed using hi-vt cell-swap.

Static IR drop analysis:
The 20mV improvement brought the voltage above the minimum required

Voltagestorm Analysis	**Without VDD metalfill mesh**	**With VDD metalfill mesh**
Lowest voltage	1.41 V	1.43V

Database size:

	DEF (fill/wo fill)	**GDS (fill/wo fill)**
File size	530MB/170MB	662MB/316MB

Experimental Results Summary
Advantages over GDSII-based approach

- Easily scriptable
- Timing-aware operations
- Ability to specify preferred density for a more uniform tile distribution
- ECO flow available to fix timing violations
- More fill flexibility with respect to spacing/tile-sizes
- Ability to quantify IR drop mitigation with mesh mode
- Acceptable/Improvement of run-time for floating metal fill insertion
- Parasitic extraction is much easier with DEF-based flow

Disadvantages over GDSII-based approach

- Run-time for VDD power strapping-mode is on the higher side. PCR has been filed for enabling multi-threaded feature for *addMetalFill* command. Expected to be present in Encounter 7.1 release.
- 2-5X Output file size for DEF, SPEF, GDS

Metal Fill using Cadence Chip Optimizer

Cadence Chip Optimizer is a space-based tool that can be used to perform post-route optimizations on a finer grid than Encounter. Freescale collaborated with Cadence to develop methodology to add Vdd/Vss metal fill mesh into a post-routed design. A feature has been added to the CCO tool to be able to extract connectivity of power/ground signals inside the gdsii views of the hard-macros. Tied-off metal fill straps added at chip-level can tap down and connect to the Vdd/Vss straps inside the hard-blocks. Currently this just works in the interactive mode.

Cadence Chip Optimizer works natively on Open-Access. Encounter database will have to be converted to OpenAccess using def2oa and lef2oa translators.

Feature	CCO	Encounter
VDD metal fill mesh can strap to power grids inside gdsii of hard-blocks.	Yes. (currently only interactive supported)	No
Hierarchical Fill	No	No
Net-based fill	Yes	No
GDS layout swap	Yes	No
POLY/ACTIVE fill insertion	No	No

ECO after metal fill insertion	Under development	Has established flow
Pushing metal fill into lower levels of hierarchy	No	No

Advantages of using CCO over Encounter for metal Optimization

- Works on manufacturing grid, so can add more shapes
- Ability to swap gds layout views for LEF abstracts for better density optimization.
- Can extract connectivity into the gdsii views for hard-blocks. Vdd metal fill segments can strap into the Vdd connections of lower-level blocks.

Advantages of using Encounter over CCO for metal optimization

- No database transfer required, avoiding data-fidelity checks
- Well established ECO flow

ECO Flow using CCO

Due to differences in connectivity model and use of manufacturing grid for polygon placement within CCO, Nanoroute is unable to perform an ECO route on a CCO database. Nanoroute ends up ripping out lots of shapes when performing an ECO of a CCO database.

Recommendation from Cadence is to use CCO space-based router to do ECO routing. Since CCO does not have a placement engine, the CCO database will have to be brought into Encounter (lef/def or Open-Acess). All netlist changes and/or placement changes and optimizations should be completed within Encounter. The data should then be exported back to CCO to do the final ECO route within CCO.

The above recommended flow is still not completely usable because of the following reasons:

CCO router has the following limitations:

- It is not SI-aware, so post-SI ECO flow is a problem.
- Unable to handle metal fill features during an ECO route

Cadence is currently working on the CCO router issues for an ECO flow using a test-case from Freescale and this is expected to be ready by September 2007.

Issues/Future work

- GDSII based approach still required for regions around and over hard-blocks
- ECO Flow using CCO needs to be tested
- Test out CCO floating metal fill flow
- Develop flow to use Cadence CMP Predictor for thickness topography prediction and fix hot-spots
- Thickness variation models taken into account during QRC extraction

Summary & Conclusion

Encounter has the ability to add either tied-off or floating fill to a design. Recommended flow is to use "mesh" mode for tied-off fill with long fill segments, followed by floating fill insertion. Fill should be added in "timing-aware" mode which is on by default in Encounter. Timing analysis should be performed after metal-fill insertion using QRC (sign-off extractor) and CTE, and any ECO's performed to offset timing degradation. Metal fill in Encounter does increase the size of gdsii and DEF. However, it does not have a significant effect on the run-time for QRC extraction and/or streamout from Encounter. Timing is affected by addition of floating metal fill, especially if dummy fill is placed close to active signals. Timing impact based on dummy fill insertion is dependent on how dense the design is and also on tile size/spacing information used during insertion. IR drop problems can be mitigated to some extent using the Vdd/Vss mesh technique described above.

References

SoC Encounter (6.1) User Guide, Command Reference Manual
Cadence Chip Optimizer (6.2.1) User Guide, Reference Manual
http://www.edadesignline.com/howto/196500175
http://www.ece.northwestern.edu/~debjit/files/Sinha_VLSI07.pdf

Acknowledgments

We would like to thank the design teams, especially Javier Saldana, Anis Jarrar, Cong Do and Greg Ward who worked with us to establish the flow. Also, we would like to thank Cadence CCO AE’s who are working with Freescale to iron out CCO flow issues.