Renesas 40nm SoCE 基本デザインフロー補足説明書
慶應大学 天野 2010.9.30

1 はじめに

本ドキュメントは、自動レイアウトサンプルスクリプト使用説明書 (説明書と略す) WEB-00465513-01 を補足し、VDEC 環境で SoCE レベルで Violoation が出ないレイアウトができるまでのフローである。本ドキュメントで参照しているのは全てこの使用説明書である。当方で作ったもので、どうしても解決できない問題について、ルネサスエレクトロニクスの箕輪さん、内藤電誠工業の新保さんに修正していただいてできたものである。

2 基本的な変更

自動レイアウトサンプルスクリプトは、VDEC で開発を行う場合、以下の問題点を持っている。

- 電源配線用のスクリプト 02_proute が欠けているため、自動レイアウトサンプルスクリプトを忠実に実行してもレイアウトが完成しない。
- ツールのバージョンが微妙に異なるため、Violation の擬似エラーが生じてしまう。

まず、環境に以下の修正を加える。

- テクノロジ lef ファイルである top.lef は、2010.9.14 にルネサスが提供したものを用いる。最初に配布したものは用いてはならない。
- 他の所で用いる lef(0.2 参照) とは別に、電源配線専用のテクノロジ lef ファイル lef_proute を以下の手順で作成する。
  - mk_power_lef.csh と mk_power_lef.awk を lef を生成するディレクトリに置く。
  - mk_power_lef と実行

  生成された power_lef は VIA ルールに変更が加えられている。
- SoCEncounter は ver.9.1 を用いる。説明書中の 8.1 の部分は全て 9.1 を用いる。しかし、06_shield の所に出てくる 7.1 のバージョンはこのままにする。
- 説明書中では、7 メタル埋め →8 デカップリング容量埋め →9 フィルセル埋めの順になっているが、これを、8 デカップリング容量埋め →9 フィルセル埋め → 7 メタル埋めの順に変更する。

3 01_floorplan

説明書通りのコア面積の設定を行うと、I/O Pad の Blockage 領域とぶつかってトラブルを引き起こす。このため、コア領域を狭くする必要がある。(もとのルネサスのフローではこれを電源配線時に行ってるらしい) ここでは、

```
if { $chipsize == "2.5" } {
   floorPlan -site nc40_dsc \
             -b -1158.960 -1158.960 \
                 1158.960  1158.960 \
                -1158.960 -1158.960 \
```

```
                1158.960  1158.960 \
                -827.640  -827.640 \
                 827.640   827.640
} elseif { $chipsize == "5.0" } {
```

にしたが、これは図 1 に示すように小さすぎるらしくもっと大きくとっていい。
あとは説明書の 01_floorplan に従う。

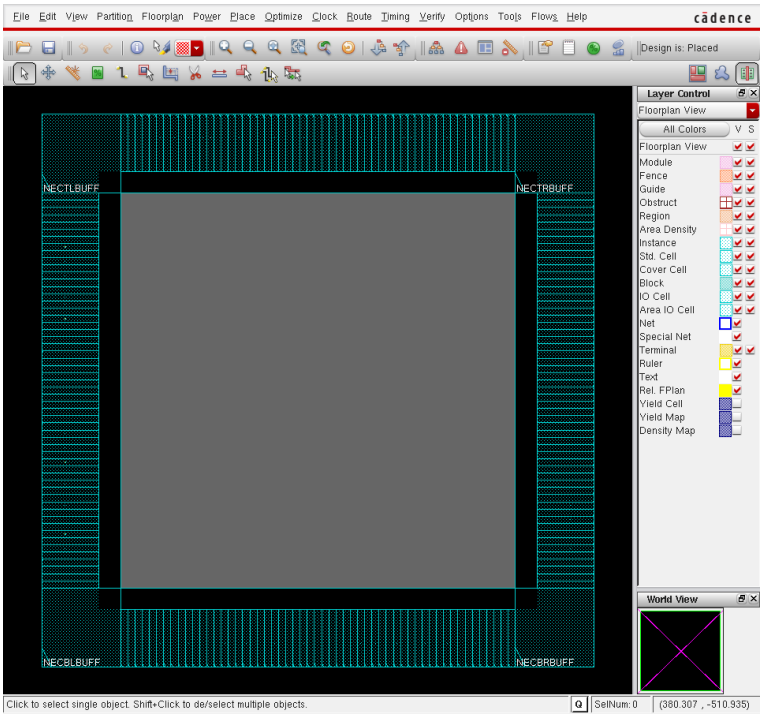

図 1: フロアプラン

4 02_proute

説明書に存在しない電源配線部である。SoCE を立ち上げ、source file/proute_amano.tcl_mod を実行する。

以下、proute_amano.tcl_mod の解説である。e-shuttle などと違って、この部分が非常に複雑になるのは、今回使っている OpenCAD 標準セルが、primitive 用 follow-pin として M1L/M2L 両方を使うことになっていること。普通に電源を引くと、Via の外形が太くなって標準セルとのパターン間で多数の Violation を生じるためである。

まずファイル定義

```
set inputConfig         ../../database/conf/VDEC_soce.conf
set inputVerilog        ../01_floorplan/floorplan.v.gz
set inputDef            ../01_floorplan/floorplan.def.gz

## REL(NDK) chenge lef ---> lef_proute
set inputLef            ../../database/lef/lef_proute
## lef_proute である点を注意！

set inputSDC            ../../database/sdc/POCOP_TOP.sdc
set outputVerilog       ./pg.v.gz
```

```
set outputDef           ./pg.def.gz
set maxLayerNumber      7
set encounterDBS         "encdbs"
set encounterRPT         "encrpt"
createSaveDir ${encounterDBS}
createSaveDir ${encounterRPT}
set defCapMax  "1"
set detCapMax  "1"
set resMax     "1"
# ============================================================================ #
# --- Initial Setting --- #
# ============================================================================ #
if { $TECHNOLOGY == "CB12" || $TECHNOLOGY == "CB130M"|| $TECHNOLOGY == "CMOS12M" } {
        set DBU 100
} else {
        set DBU 1000
}
set numberOfPaths 100
set dbgUpdateDefRows 0
set dbgGPSAutoCellFunction 1
set dbgDefOutUnits $DBU
setImportMode -minDbuPerMicron $DBU
set preRouteSkipApplyGlobalNets 1
setVar dbgAllowPowerDomainMinGapZero 1
set sprUseRowOrient 1
set dbgIPOMaxBufDetourDist 100
set dpgBufInsMinTotCap 0
```

次にトラックを引き、グリッドを定義する。flooreplan で作った def を読み込む。

```
setImportMode -keepEmptyModule 1
setDoAssign on -new
generateTracks \
-m1HOffset 0 -m1HPitch 0.132 -m1VOffset 0.066 -m1VPitch 0.132 \
-m2HOffset 0 -m2HPitch 0.132 -m2VOffset 0.066 -m2VPitch 0.132 \
-m3HOffset 0 -m3HPitch 0.132 -m3VOffset 0.066 -m3VPitch 0.132 \
-m4HOffset 0 -m4HPitch 0.132 -m4VOffset 0.066 -m4VPitch 0.132 \
-m5HOffset 0 -m5HPitch 0.132 -m5VOffset 0.066 -m5VPitch 0.132 \
-m6HOffset 0 -m6HPitch 0.264 -m6VOffset 0.066 -m6VPitch 0.264 \
-m7HOffset 0 -m7HPitch 0.792 -m7VOffset 0.066 -m7VPitch 0.792
loadConfig                          $inputConfig 0
set rda_Input(ui_netlist)           $inputVerilog
set rda_Input(ui_leffile)           $inputLef
set rda_Input(ui_timingcon_file)    $inputSDC
commitConfig
if { [ info exists inputECOFile ] } { loadECO  $inputECOFile }
defIn                               $inputDef
```

```
setPreference ConstraintUserXOffset 0.066
setPreference BlockSnapRule 2
setPreference ConstraintUserXGrid 0.132
setPreference ConstraintUserYGrid 0.132
deselectAll
snapFPlan -block
checkMacroLLOnTrack -useM2M3Track
clearGlobalNets
globalNetConnect VDD -type pgpin -pin VDD -inst * -verbose
globalNetConnect VDD -type tiehi -inst * -verbose
globalNetConnect VSS -type pgpin -pin VSS -inst * -verbose
globalNetConnect VSS -type tielo -inst * -verbose

cutRow
clearCutRow
deselectAll
set toplayer 8
set botlayer 1
setSnapGrid -layer { 1 }               -pitch 0.011 0.011
setSnapGrid -layer { 2 3 4 5 }         -pitch 0.033 0.033
setSnapGrid -layer { V12 V23 V34 V45 } -pitch 0.066 0.066
set v_spacing [expr 0.132 * 6]
set h_spacing [expr 0.132 * 6]
```

次にまず、コア周辺リンクを巻く。新保さんの指導により縦横 M3L を使った。

```
set vlayer M3L
set hlayer M3L
set v_offset [expr 0.132*200]
set h_offset [expr 0.132*200]
set v_width 0.99
set h_width 0.99

addRing -nets { VDD VSS } \
        -follow core \
        -around core \
        -layer_top ${hlayer} \
        -layer_bottom ${hlayer} \
        -width_top ${h_width} \
        -width_bottom ${h_width} \
        -width_right ${v_width} \
        -width_left ${v_width} \
        -layer_right ${vlayer} \
        -layer_left ${vlayer} \
        -spacing_top ${h_spacing} \
        -spacing_bottom ${h_spacing} \
```

```
        -spacing_right ${v_spacing} \
        -spacing_left ${v_spacing} \
        -offset_top ${h_offset} \
        -offset_bottom ${h_offset} \
        -offset_right ${v_offset} \
        -offset_left ${v_offset} \
        -stacked_via_top_layer M6S \
        -stacked_via_bottom_layer M1L \
        -snap_wire_center_to_grid Grid \
        -tr 0 -bl 0 -br 0 -lt 0 -rt 0 -lb 0 -rb 0
deselectAll
```

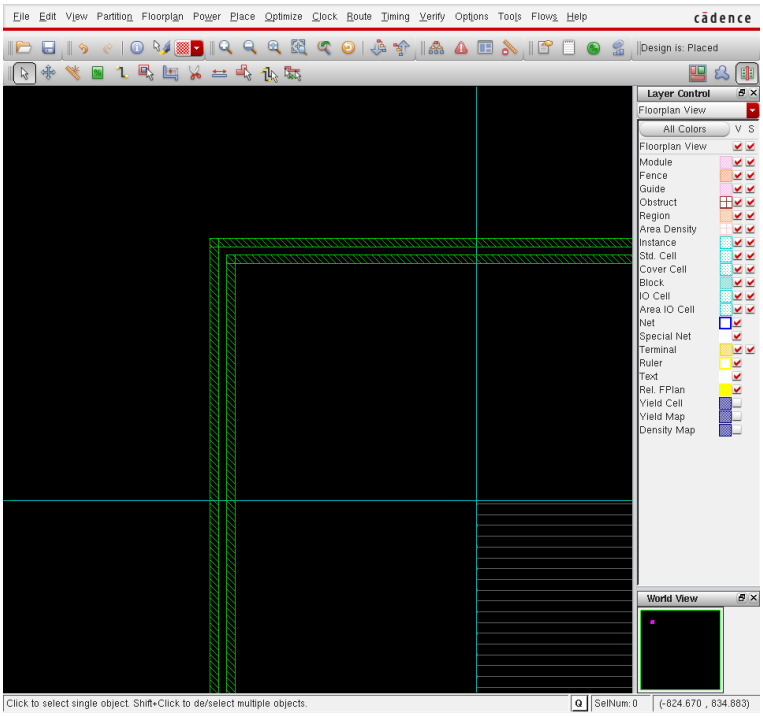

図 2: リング

次に Strip を張る。-padPinLayerRange4 5 で I/O バッファ電源端子-電源リングまでの配線層を制限している。また、split_long_via オプションで、後に出てくる primitive 用の M2L の follow-pin に対して電源リングにまっすぐ通すために 0.264 に設定している。(いずれも新保さんの修正)。

```
setAddStripeOption -remove_floating_stripe_over_block 0
addStripe -nets {VSS VDD } \
  -layer M6S \
    -direction vertical \
    -spacing [expr 0.264 * 20] \
    -width 0.99 \
    -set_to_set_distance [expr 0.264 * 60] \
    -xleft_offset 15.345 \
    -block_ring_top_layer_limit M1T \
    -block_ring_bottom_layer_limit M1L \
```

```
    -padcore_ring_top_layer_limit M1T \
    -padcore_ring_bottom_layer_limit M1L \
    -stacked_via_top_layer PM \
    -stacked_via_bottom_layer M1L \
    -merge_stripes_value [expr 0.264*5] \
        -snap_wire_center_to_grid Grid \
        -via_using_exact_crossover_size 1
```

さらに、I/O から、リングへの配線を行う。同様に配線層を 4 から 5 に制限し、split_long_via オプションを調整している。

```
set connectlayer 4
set con_name VSS
sroute -nets { VSS } \
    -connect { padPin } \
    -padPinAllGeomsConnect \
    -padPinLayerRange { 4 5 } \
    -padPinWidth 0.99 \
    -split_long_via {0.264 0.264 0.264}
set connectlayer 4
set con_name VDD
sroute -nets { VDD } \
        -connect { padPin } \
        -padPinAllGeomsConnect \
        -blockPinTarget { nearestTarget } \
        -padPinLayerRange { 4 5 } \
        -padPinWidth 0.99 \
    -split_long_via {0.264 0.264 0.264}
```

ストリップと I/O との接続後の様子を図 3 に示す。

最後にレールを張るのだが、この時、primitive 用 follow-pin というのを M1L/M2L 両方に用意しないといけないらしい。また、sroute 用コンフィグファイルを使っている。

```
setAddRingOption -extend_stripe_search_distance 0.0
setViaGenOption -optimize_cross_via 1


set srouteExtraConfig_cfg [open ./align.cfg {WRONLY CREAT TRUNC}]
   puts $srouteExtraConfig_cfg "srouteAlignViaOnStripeOffsetFromCenter 264"
   puts $srouteExtraConfig_cfg "srouteNoViaOnWireShape NONE"
   puts $srouteExtraConfig_cfg "srouteTargetVerticalMargin 660"
close $srouteExtraConfig_cfg
```

最初の 2 行が M1/M2, M2/M3 の cross-via を太らなくするための設定、以下が align.cfg を生成するコマンド。ちなみにこの align.cfg で配置する VIA を 0.264um 水平にずらし、縦方向の配線では 0.66um 以内で VIA を発生させないようにしている。以下がその align.cfg の中身である。

```
srouteAlignViaOnStripeOffsetFromCenter 264
srouteNoViaOnWireShape NONE
srouteTargetVerticalMargin 660
```

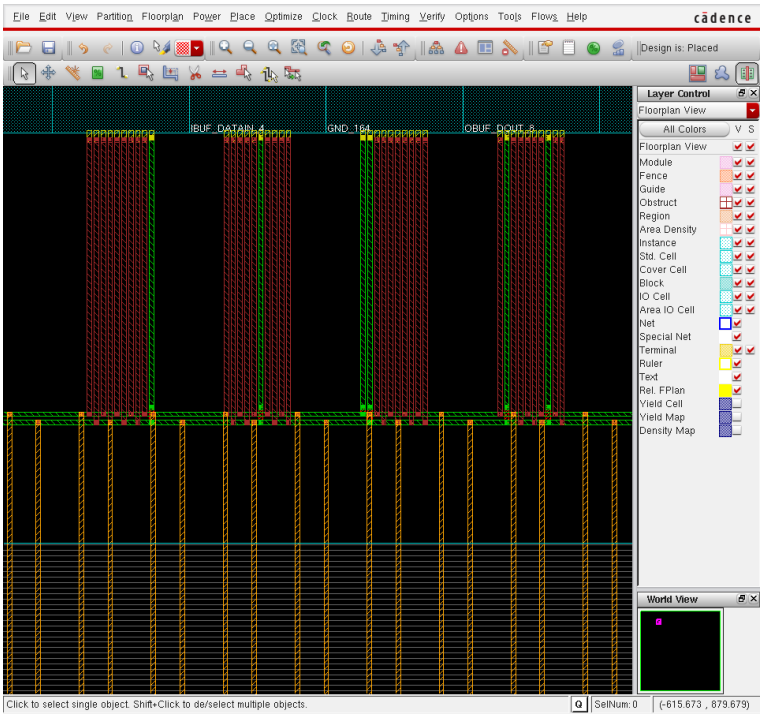

図 3: ストリップ

次に M2L の follow-pin 電源配線を引くための目印として SITE 両端に power-stop セルを配置する。

```
## REL(NDK) added power-stop-cell for route M2L follow-pin
addEndCap -preCap LDL_POWERSTOPL -postCap LDL_POWERSTOPR \
        -prefix LDL_POWERSTOP_
```

以下のコマンドで、M2L の配線対象を設定し、follow-pin を配線する。

```
## REL(NDK) added
globalNetConnect VDD -type pgpin -pin VDD
globalNetConnect VSS -type pgpin -pin VSS
##

## REL(NDK) added for M2L follow-pin
sroute -noBlockPins -noPadRings -noPadPins -noStripes \
        -crossoverViaTopLayer 5 \
        -jogControl { preferWithChanges differentLayer } \
        -nets { VDD VSS } -targetViaTopLayer 5 \
        -corePinLayer 2 \
        -extraConfig align.cfg
###
```

図 4 のように follow-pin 用のレールが引かれる。
最後に M1L のレールを普通に引き、結果を吐き出す。

```
sroute -nets { VSS VDD } \
    -connect { blockPin corePin floatingStripe } \
    -layerChangeRange { 1 8 } \
```

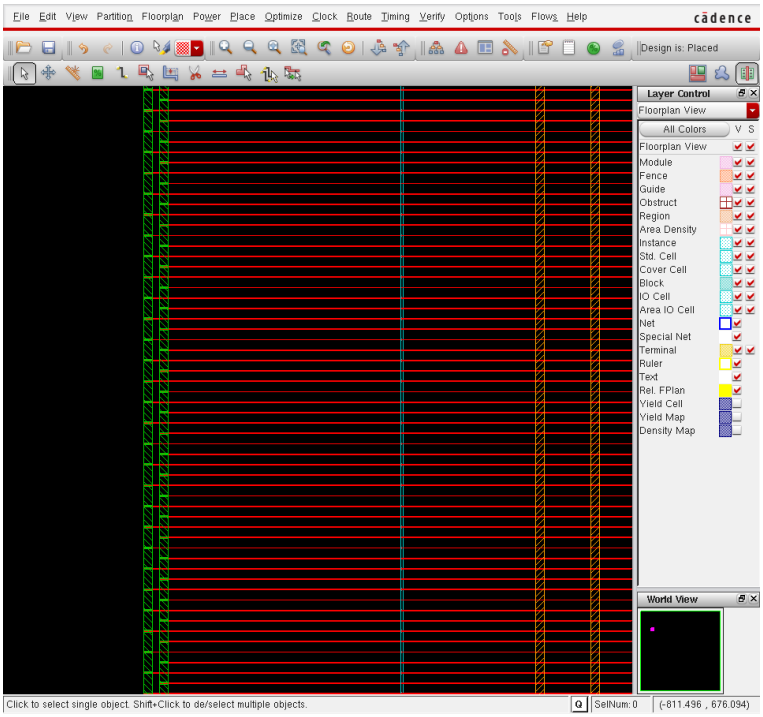

図 4: follow-pin 用レール

```
    -blockPinTarget { nearestRingStripe } \
    -padPinPortConnect { allPort oneGeom } \
    -area {-901.956 -901.956 901.956 901.956} \
    -targetViaTopLayer 8 \
    -targetViaBottomLayer 1

saveNetlist pg.v.gz
defout -floorplan -placement -netlist -routing -noCutRow ${DESIGN}_pg.def.gz
```

こうしてできたレールと Strip の交点の様子を図 5 に示す。

良く見ると、ML1/ML2 両方にレールが張られており、これらの via は太くなっておらず、セルとの干渉が生じないようになっている。上記のスクリプトは新保さんの修正点によるものが多いが、利用されている修正法は SoCE のコマンドと RE40 およびその標準セルライブラリに通じていないととても思いつかない方法である。

上記スクリプトの問題点は、I/O バッファの上位層端子 M7,M8 から電源配線が出ていない点で、これらの端子に Open のエラーが生じる。これらの端子を有効に利用すればさらに電源を効果的に供給できると思うが、これ以上いじる気がしないので、余裕があれば試してみようと思う。

5　これ以降

あとは、最初に記したように、順番の入れ替えとツールのバージョンに気をつけて説明書通りに実行するとほとんど Violation Free のレイアウトを作成できる。今回の例の場合は、M4L で 2 個 Minimum Space の Violation が出たが、いずれも密度を上げるための M4L とフィラーとの間で生じており、単純に密度上げ用の M4L を削除して無くすることができた。ただし、Caliber の DRC は現在テスト中でまだこれで通るのかどうか不明である。

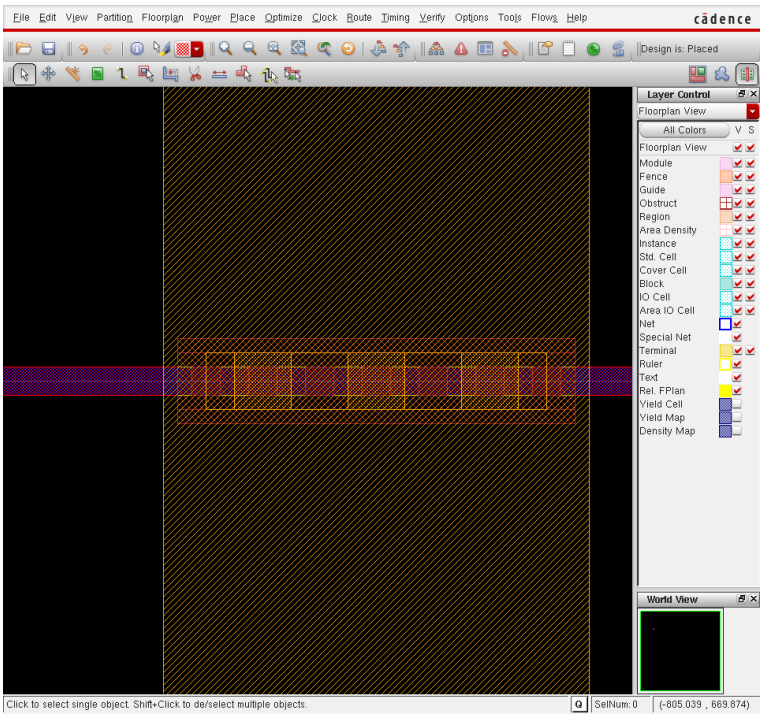

図 5: Strip/rail の交点