How to run DRC and LVS using Calibre with 40nm

Masayuki KIMURA*
慶應義塾大学理工学部 天野研究室

平成 23 年 3 月 30 日

1 はじめに

Renesas-40nm プロセスを用いて，DRC および LVS 実行しレイアウトの検証を行う方法を示す.

2 実行前のレイアウトの準備

2.1 必要なファイル

Makefile	DRC,LVS やその準備を行う
CONF_CONST_CTRL_0.gds	GDS ファイル (今回は SMA-2 の CONF_CONST_CTRL_0 を使用)
CONF_CONST_CTRL_0.lvs.v	CONF_CONST_CTRL_0 の LVS 用ネットリスト
CONF_CONST_CTRL_0.LEF	CONF_CONST_CTRL_0 の LEF ファイル
misc_scripts/	テンプレートの入ったディレクトリ
startre40.sh	ArtistLib を使用するための初期設定スクリプト
stdcell_strmin.sh	スタセルの streamin を行うためのスクリプト
run_drc_macro.sh	マクロレベルで DRC を行うためのスクリプト
run_drc_chip.sh	チップレベルで DRC を行うためのスクリプト
make_ed.rb	ED ファイルを作成するためのスクリプト
make_lvsnet.csh	レイアウトから spice ネットリストを生成するためのスクリプト

2.2 ディレクトリ構成

/home/hlab/masayuki/sma-2/verify 上で作業を行うと仮定する.

```
$ls
Makefile  CONF_CONST_CTRL_0.gds  misc_scripts  startre40.sh
stdcell_strmin.sh   rule report
```

2.3 実行前に

Re40 の ArtistLibraryKit を使うためには，まず環境変数の設定を行う必要がある．これを行うのが startre40.sh である．作業を行う前に，これを source する.

startre40.sh(抜粋)

```
export ARTISTLIB='/home/vdec/lib/ux8l/AnalogArtist/UX8'
${ARTISTLIB}/bin/setartlib
export ARTISTCBLIB='/home/vdec/lib/ux8l/AnalogArtist/UX8L_DIGITAL'
cat ${ARTISTCBLIB}/config/cds.lib.UX8LD_9grid_MVT >> ./cds.lib
```

*masayuki@am.ics.keio.ac.jp

2.4 スタセルデータの streamin

icfb による作業を行う前に，UX8L 環境のスタセルを icfb に streamin する．
これを行うのが，stdcell_streamin.sh[1]である．

stdcell_streamin.sh(抜粋)

```
gdsdir="/home/vdec/lib/ux8l/blib/UX8L/wide1_1.1V/gds2/common"
gdsdir_wslash="\/home\/vdec\/lib\/ux8l\/blib\/UX8L\/wide1_1.1V\/gds2\/common"
scr_dir="./scripts_icfb"

for GDS in `ls ${gdsdir}`; do
    rm -f ${scr_dir}/streamin.scr
    sed 's/__CELL_NAME__/'"${gdsdir_wslash}\/${GDS}"'/g' \
        ${scr_dir}/streamin.template > ${scr_dir}/streamin.scr
    pipo strmin ${scr_dir}/streamin.scr
done
```

スタセルの gds があるディレクトリから一つずつ GDS を取り出し，streamin を行っている．
処理が完了すると，./UX8L/というディレクトリが生成され，膨大なスタセル情報が格納される．
この処理を common-cell, foregin-cell, scribe 線-cell に対して行う必要がある (スクリプトではすべて自動的に行っている) なお，この処理には非常に時間がかかるため，帰る前に仕掛けて朝確認するようにすれば良い．
その他にも，すでに他の人が構築した./UX8L コピーしても良い．

2.5 レイアウトデータの streamin

次に，実レイアウトデータを streamin する．これにより，SOC Encounter ではスタセルがブラックボックスとして扱われていたものが，スタセルの実レイアウトデータを含むものに変換される．
このレイアウトデータの streamin と後述する streamout は，SOC Encounter によるレイアウトを変更した場合，毎回やり直す必要がある．
天野研究室の環境では，Makefile を用いた自動 streamin 環境を構築している．

```
$ make CONF_CONST_CTRL_0.streamin
```

これにより CONF_CONST_CTRL_0.gds の streamin が行われる．

Makefile(抜粋)

```
%.streamin:
        sed 's|__CELL_NAME__|$*.gds|g' $(SCR_DIR)/streamInKeys_setup.il \
                > $(SCR_DIR)/streamin.scr
        $(PIPO) strmin $(SCR_DIR)/streamin.scr
```

$SCR_DIR(=misc_scripts) 内にある streamInKeys_setup.il をテンプレートとし，streamin にひつような設定ファイルを生成する．そのファイルを利用して，レイアウトデータの stramin を行っている．
streamin が完了すると，icfb 上でレイアウトデータを見ることができるはずである．([File]→[Open]→[Cell Name に CONF_CONST_CTRL]→[OK])
図 1 に，streamin したレイアウトを示す．拡大してみると，スタセルの情報が詳細に埋め込まれていることが分かる．

2.6 レイアウトデータの streamout

streamin により，スタセル情報を埋め込んだあと，GDS にそれを出力する．これが streamout である．
この処理にも，Makefile を用いる．

```
$ make CONF_CONST_CTRL_0.streamout
```

[1]京都大学小野寺研究室の西澤様のスクリプトを参考にさせていただいた

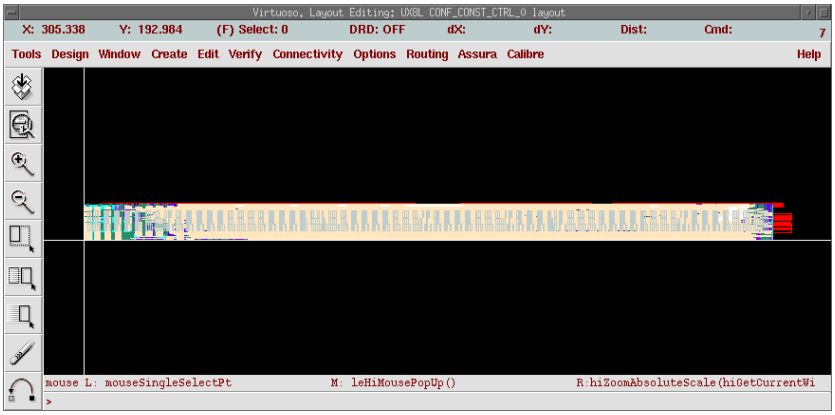

図 1: streamin 後のレイアウトデータ

Makefile(抜粋)

```
%.streamout:
        sed 's|__CELL_NAME__|$*|g' $(SCR_DIR)/streamOutKeys_setup.il > $(SCR_DIR)/streamout.scr
        $(PIPO) strmout $(SCR_DIR)/streamout.scr
```

これにより，CONF_CONST_CTRL_0.gds がスタセル情報を含んだものとなる．

3 DRC(Design Rule Check)

製造上満たさなければならないルールの確認．このルールをパスしないと，製造すら行われない．設計データがこのデザインルールに違反していないかをチェックする．つまり GDS v.s. DesignRule である．

DRC を実行するにも，Makefile を利用する．

```
$ make CONF_CONST_CTRL_0.drc
```

Makefile(抜粋)

```
%.drc:
        ./run_drc_macro.sh $* | tee $*.drc.log
```

実際に実行されるのは run_drc_macro.sh というスクリプトである．もう一つ run_drc_chip.sh というスクリプトも存在するが，これは実行するルールセットが異なっているだけである．

UX8L では CS202 と異なり，DRC のルールファイルは複数に分かれている．これらを順番に実行し，その結果を summarize するのが run_drc_macro.sh である．

run_drc_macro.sh(抜粋)

```
rule_lists="drc_add.V02R00.cal.ux8_svenc \
    drc_core_beol.7cu_pm_L5S1T1G0.V03R00.cal.ux8_svenc \
    drc_core_feol.V03R00.cal.ux8_svenc \
    drc_pg_latch-up.7cu_pm_L5S1T1G0.V01R00.cal.ux8_svenc \
    extraordinary_figure_check.rule_svenc \
    lvs.7cu_pm_L5S1T1G0.V03R00.cal.ux8l_ld_svenc \
    drc_macro_den.7cu_pm_L5S1T1G0.V02R00.cal.ux8_svenc"
```

rule_lists に，実行すべきルールを追加する．

ルネサスエレクトロニクスの箕輪様のお話では，マクロで DRC を実行する場合は最初の 3 つのみチェックすれば良いらしい．しかし，本スクリプトでは念のためその他のルールについてもチェックしている．

run_drc_macro.sh(抜粋)

```
  sed 's/gds_file/'$gds_file'/g' ./drc.rul | \
      sed 's/rule_dir/'$rule_dir'/g' | \
      sed 's/rule_file/'$rule_file'/g' | \
      sed 's/design_name/'$design_name'/g' > rule/${design_name}_drc_${rule_file}.rul
  calibre -drc -hier -nowait rule/${design_name}_drc_${rule_file}.rul \
      -64 -turbo 8 | tee log/${design_name}_${rule_file}.log
```

./misc_scripts/drc.rul というファイルをテンプレートとし，必要に応じてそのファイルを書き換えた上で DRC を実行する．生成したルールファイルは./rule ディレクトリに保存される．

./misc_scripts/drc.rul(抜粋)

```
LAYOUT PATH  "gds_file"
LAYOUT PRIMARY "design_name"
LAYOUT SYSTEM GDSII

DRC RESULTS DATABASE "./drc_rule_file.gds" GDSII APPEND "_ERR"
DRC MAXIMUM RESULTS ALL
DRC MAXIMUM VERTEX 199

DRC CELL NAME NO
DRC SUMMARY REPORT "./design_name_rule_file.sum" REPLACE HIER

VIRTUAL CONNECT COLON NO
VIRTUAL CONNECT REPORT NO

DRC ICSTATION YES

INCLUDE "rule_dir/rule_file"
```

3.1 DRC 実行後の確認

すべてのルールセットについて DRC は実行されるが，その結果はすべて別のファイルに保存される．
それが$DESIGN_NAME_$drc_rule.sum である．
これらのファイルをすべて確認するのは大変なので，run_drc_macro.sh では，すべての sum ファイルをまとめて，エラーの数を表示するようになっている．それが$DESIGN_NAME_DRC.summary である．

CONF_CONST_CTRL_0_DRC.summay

```
#####
Rule Check drc_add.V02R00.cal.ux8_svenc
TOTAL DRC Results Generated:     0 (0)
#####
Rule Check drc_core_beol.7cu_pm_L5S1T1G0.V03R00.cal.ux8_svenc
TOTAL DRC Results Generated:     0 (0)
#####
Rule Check drc_core_feol.V03R00.cal.ux8_svenc
TOTAL DRC Results Generated:     0 (0)
#####
Rule Check drc_pg_latch-up.7cu_pm_L5S1T1G0.V01R00.cal.ux8_svenc
TOTAL DRC Results Generated:     7461 (221696)
#####
Rule Check extraordinary_figure_check.rule_svenc
TOTAL DRC Results Generated:     0 (0)
#####
Rule Check lvs.7cu_pm_L5S1T1G0.V03R00.cal.ux8l_ld_svenc
#####
Rule Check drc_macro_den.7cu_pm_L5S1T1G0.V02R00.cal.ux8_svenc
TOTAL DRC Results Generated:     1 (1)
```

これにより，ひととおりどのルールにおいて DRC が発生したかを確認する．DRC が発生したルールについては，icfb から Calibre を立ち上げて場所を確認する．
このときに DRC の結果は，.asc に格納されているので，それを Calibre で読み込めば良い．
DRC ごとの対応策であるが，これはプロセスにより異なるし，多くの場合 DRC ルールの解説 pdf を見ながら対応することになる．
天野研究室の環境であるならば，/home/vdec/lib/ux8l/DesignManual/ 以下の pdf ファイルを熟読のこと．

4 LVS(Layout Versus Schematics)

LVS はレイアウトツールにより生成されたネットリストと，GDS から抽出したネットリストを比較する．この二つの透過性チェックを行うことにより，レイアウトに重大なミスが存在していないかをチェックする．
このためには，Layout 側と Schematic 側から情報を抽出する必要がある．

- Layout 側：lvs 用 netlist + スタセル netlist→ レイアウトの spice netlist(cdl)
- GDS 側：GDS → GDS の spice netlist(_net.gz)

4.1 レイアウト側の処理

まず，SOC Encounter で出力した CONF_CONST_CTRL_0.lvs.v と CONF_CONST_CTRL_0.LEF を verify ディレクトリにコピーまたはリンクする．
次に，LEF ファイルを用いて ed ファイルというものを作成する．ed ファイルは，ピン情報のテキストを追加したファイルであり，各ピンに対して，名前，物理的な座標，レイヤを指定する．
この ed ファイルを作成するのが，make_ed.rb である．

```
$ ./make_ed.rb CONF_CONST_CTRL_0.LEF > CONF_CONST_CTRL.ed
```

これにより，自動的に LEF から座標情報が抽出される．

CONF_CONST_CTRL_0.ed(抜粋)

```
LAYOUT TEXT "RST_N"   791.8350    20.8560 14 CONF_CONST_CTRL_0
LAYOUT TEXT "WE_FROM_EXTERNAL"   791.8350    38.4120 14 CONF_CONST_CTRL_0
LAYOUT TEXT "ROMULTIC_BITS_FROM_EXTERNAL[17]"   791.8350    19.8000 14 CONF_CONST_CTRL_0
LAYOUT TEXT "ROMULTIC_BITS_FROM_EXTERNAL[16]"   791.8350    20.1960 14 CONF_CONST_CTRL_0
LAYOUT TEXT "ROMULTIC_BITS_FROM_EXTERNAL[15]"   791.8350    19.0080 14 CONF_CONST_CTRL_0
...
```

次に，spice ネットリストを作成するこれには，make_lvsnet.csh を用いる．
これはルネサス提供のスクリプトを少し改変したもので，基本的な流れは，

1. ダミー RAM 生成
2. ピンばらし
3. def の改造
4. v2lvs の実行

最後の v2lvs がもっとも大事なコマンドで，これにより spice ネットリストが作成できる．
make_lvsnet.csh を実行するには，いつも通り Makefile を実行する．

```
$ make CONF_CONST_CTRL_0.cdl
```

エラーがなくうまく言った場合，最後の v2lvs が実行され，CONF_CONST_CTRL_0.spice が生成される．この中身を確認して，それっぽくなっているかどうか確認してほしい．
また，モジュール内にサブモジュールが定義されているようなレイアウトの場合，lvs.v は，サブモジュールまで出力してはくれないため，make_lvsnet.csh の submodule 変数を変更し，サブモジュールを追加してやる必要がある．

4.2 LVS の実行

次に，LVS の実行を行う．
これも，Makefile を実行するだけである．

```
$ make CONF_CONST_CTRL.lvs
```

Makefile(抜粋)

```
        rm ./misc_scripts/run_lvs.scr -rf
        touch ./misc_scripts/run_lvs.scr
        sed 's/DESIGN_NAME/$*/g' ./misc_scripts/run_lvs.template > \
                ./misc_scripts/run_lvs.scr
        calibre -lvs -turbo 8 -64 -hier -spice $*_net.gz \
                ./misc_scripts/run_lvs.scr | tee $*.lvs.log
```

基本的な手法はこれまでと同じ．テンプレートを改変して，ルールを書き換えて実行する．

./misc_scripts/run_lvs.template(抜粋)

```
LAYOUT PATH  "./DESIGN_NAME.gds"
LAYOUT PRIMARY "DESIGN_NAME"
LAYOUT SYSTEM GDSII

SOURCE PATH "./DESIGN_NAME.spice"
SOURCE PRIMARY "DESIGN_NAME"
SOURCE SYSTEM SPICE

MASK SVDB DIRECTORY "svdb" QUERY

LVS REPORT "./report/DESIGN_NAME.lvs.report"

...

INCLUDE "./DESIGN_NAME.ed"
// INCLUDE "./DESIGN_NAME.blackbox"
INCLUDE "./misc_scripts/lvs.7cu_pm_L5S1T1G0.V03R00.cal.ux8l_ld.rft1178_07_source.ilv0029_01"
// INCLUDE "./misc_scripts/lvs.7cu_pm_L5S1T1G0.V05R00.cal.ux8l_ld_svenc"
```

ed ファイルをインクルードしているのが分かると思う．また，ブラックボックスについても一応指定できるようにしているが，メモリもない環境でブラックボックステストはおそらく行わないであろう．
また，ルールファイルに関しては R300 と R500 について両方指定できるようにはしてあるが，R500 は最新版にも関わらずなぜか実行できなかった．

4.3 LVS 実行後の確認

LVS が完了すると，./report/CONF_CONST_CTRL_0.lvs.report に，実行レポートが表示される．
LVS が成功すると，report には次のようなファイルが格納されている．[2]

[2]個人的には非常にムカつく顔だと思う．

./report/CONF_CONST_CTRL_0.lvs.report(抜粋)

```
                ##################################################
                ##                                              ##
                ##         C A L I B R E    S Y S T E M         ##
                ##                                              ##
                ##             L V S   R E P O R T              ##
                ##                                              ##
                ##################################################

REPORT FILE NAME:          ./report/CONF_CONST_CTRL_0.lvs.report
LAYOUT NAME:               CONF_CONST_CTRL_0_net.gz ('CONF_CONST_CTRL_0')
SOURCE NAME:               ./CONF_CONST_CTRL_0.spice ('CONF_CONST_CTRL_0')
RULE FILE:                 ./misc_scripts/run_lvs.scr
RULE FILE TITLE:           WES-00007310-00.60M
UX8G/UX8L/UX8GD/UX8LD/UX8LD(LCM)
Device List and LVS ( Calibre ) 7metals (L=5,S=1,T=1,G=0)  for
 UX8L/LD
CREATION TIME:             Sun Oct 17 20:49:32 2010
CURRENT DIRECTORY:         /home/wasmii3/usr/masayuki/sma-2/verify
USER NAME:                 masayuki
CALIBRE VERSION:           v2010.2_13.12    Mon May 10 22:09:38 PDT 2010

                               OVERALL COMPARISON RESULTS

                         #       ####################       _   _
                        #        #                  #       *   *
                     #   #       #     CORRECT      #         |
                      # #        #                  #       \___/
                       #         ####################
```

もし失敗した場合 (成功，失敗に関わらず)GDS から抽出された spice ネットリストは CONF_CONST_CTRL_0_net.gz を確認するのもよいかもしれない．

5 おまけ

DRC，LVS ともに非常に時間のかかる処理である．筆者は，./../tools/mail.sh というメール送信スクリプトを書き，Makefile の最後などに埋め込むことで，処理が完了すると自動的にメールが送信される環境を構築している (ただし CentOS 限定である)．LVS 実行中に別の仕事をし，メールが来たら確認するという風にして，処理を効率化していったほうが，ストレスが溜まらなくて良い思う．