Number Systems

James E. Stine, Jr.

Department of Electrical and Computer Engineering, VLSI Computer Architecture Research Group, Oklahoma State University,
{james.stine}@okstate.edu

September 12, 2017

Outline

- Review
- Traditional
- Redundant
- Conclusion

Review of Basic Number Representation

- General-purpose processors : normal everyday use.
  - Numerical computations
  - Basic operations
  - fixed-point and floating-point
  - IEEE standard
  - vector, SIMD, and VLIW processing
- Application-Specific Processors : specific purpose only
  - Numerically intensive applications
  - Single computation of classes of computations
  - Designed for improving specific application
- New generation : still to be defined
  - Can integrate both into system
  - Morphable
  - Adaptable

ASICs:Application-Specific Processors

- Areas of application
  - Signal processing
  - Embedded systems
  - Matrix computations (BLAS and LINPACK/EISPACK)
  - Graphics, Vision, and Multi-media
  - Cryptography and Security
  - Robotics, Instrumentation
  - Others : follow the money
- Features
  - Better use of technology
  - Improvement in speed, area, and power/energy.
  - Flexible
    - Implementation; decomposition into modules
    - Number systems and data formats
    - Algorithms
- Good EDA tools!

Mentor Graphics Calibre

- Calibre from Mentor Graphics is probably the most popular LVS program in that its been proven to work for small feature sizes (i.e., transistor lengths ($< 45nm$)
- Calibre uses a program to graphicall see the results called RVE (Results Viewing Environment)
- Calibre RVE has several progras to help with verification:
  - CalibreDRC - Design Rules Check
  - CalibreLVS - Layout vs. Schematic (main focus here)
  - CalibrePERC - Electrical Rules Check
  - CalibrexRC - Parasitic Extraction (sometimes called PeX)
  - CalibreDFM - Design For Manufacturing
  - RDB Conversion - convert Calibre Results (RDB) to ASCII for use with place and route tools.
  - RVE SPICE - SPICE viewer

RVE Data Flow Diagram

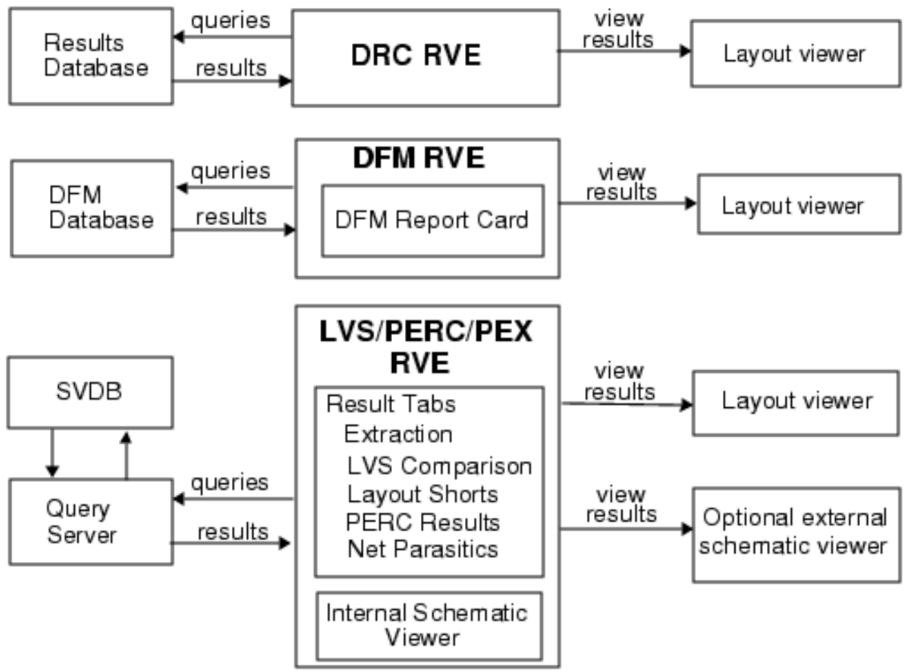

Figure: Output from RVE.

LVS Typical Flow

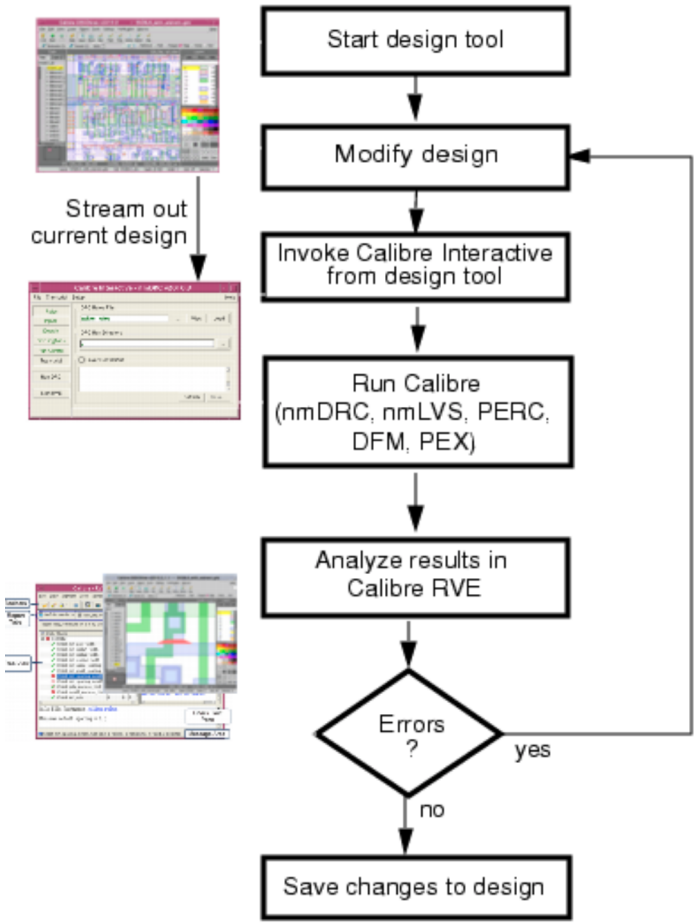

Figure: Typical LVS Flow using Mentor Graphics Calibre.

Batch Basis

- RVE is great, but it takes time to run. Therefore, its convenient to run LVS batch based.
- Calibre runs things batch based through something called Runsets.
- A *Runset* is a Calibre-Interactive-generated file that stores the settings you specify in the GUI windows
- Runset files are typically ASCII files in your current working directory - they are typically called `.runset.calibre.lvs` where the last item is typically the Calibre program it utilizes.

Preliminaries

- Before one starts a LVS, a layout is required to check against a netlist. Most often this is the layout generated from a System on Chip place and route and usually will come with a gds and Verilog netlists.
- To start bringing in the layout to get ready for LVS, a simple SKILL file was created to create a blank layout. This script can be run with the following command:
  `virtuoso -nograph -replay CreateLib.il -log CreateLib.l`
- The SKILL script is as follows creating a library for `tcore` (i.e., any name can be given) and binds it to the GF 32nm design library and its associated rules.

```
(setq libName "tcore");
(setq techFile "cmos32soi");
(ddCreateLib libName);
(setq libId (ddGetObj libName));
(techBindTechFile libId techFile);
```

strmin

- Once the library has been created, your gds layout can be brought in to make sure its ready for LVS.
- This can be done with the `strmin` Cadence Design Systems (CDS) command. The `strmin` command is an updated command that replaces the Pipe-In-Pipe-Out or PIPO command.
- To run `strmin` a template file is needed (as shown on the next slide).
  - The command to run is
    `strmin -templateFile streamOut.templatefile`
- Once the layout is brought in a label needs to be created anywhere on the layout to designate the substrate layer with the name `soisub!`.

streamIn.template

```
runDir                  "."
library                 "tcore"
strmFile                "T_CORE_0.calibre.db"
topCell                 "T_CORE_0"
view                    "layout"
refLibList              "refLib.list"
snapToGrid              "true"
runDir                  "."
case                    "preserve"
writeMode               "overwrite"
checkPolygon            "t"
logFile                 "strmIn.log"
summaryFile             "strmIn_summary.log"
```

Running Calibre LVS

- Any Calibre tool must be set up properly, therefore, a proper `.cshrc` file should be utilized to set the path properly.
- To run the Interactive Calibre LVS, you can type the following: `calibre -gui -lvs`.
- To run Calibre LVS as a batch item, type:

  ```
  calibre -gui -app -runset_options_display my_runset -batch >& batch.log
  ```

  where `my_runset` is your Runset for your LVS run and app is `-lvs`.
- Output is typically stored in log files (e.g., layoutLibrary.lvs.report) and graphics output is stored in the `svdb` directory.
- You can call up the graphical output by typing `calibrerve`.

Precursor Files

- Calibre needs some files to get started, which include:
  - Source gds that you want to compare
  - Source netlist (usually as a Verilog file)
- The source gds is important and to make things easier, its convenient to export your layout as a batch-based process, as well.
- Early CDS layouts utilized PIPO or Pipe-In, Pipe-Out, but this has changed to a utility called `strmin` and `strmout` for exporting and importing, respectively.
- This is streamlined utilizing a `templateFile` to set the export capabilities.
  - The command to run is
    `strmout -templateFile streamOut.templatefile`

streamOut.template

```
runDir                  "."
library                 "tcore"
strmFile                "T_CORE_0.calibre.db"
topCell                 "T_CORE_0"
view                    "layout"
snapToGrid              "true"
runDir                  "."
case                    "preserve"
checkPolygon            "t"
logFile                 "strmOut.log"
summaryFile             "strmOut_summary.log"
convertDot              "node"
hierDepth               "32767"
```

strmOut oddities

- There are some items to be aware of with strmout, which is called XStream Out inside CDS tools.
- Any gds output should include the standard-cells, so you need to **leave off** any reference to libraries, which is usually indicated by `refLibList` in the `strmin` templateFile.
- Calibre usually calles its gds that it imports a .db file although it is technically a gds file.
- Although not needed, `snapToGrid` is utilized to avoid any problems with roundoff of any floating-point number.
- To avoid any problems with hiearchy, the `hierDepth` is set to its maximum value of 32767.

Calibre Batch

- To run Calibre in batch, a runset is utilized.
- Any output will be in the `file.lvs.report` file dictated by what is defined in the `Runset`.
- Although you can change the output directory where the results go for the `calibrerve`, it does not seem to work efficiently (despite options in the Runset to change this directory name). Therefore, it is advisable just to store it in the `svdb` directory and move the directory manually, if needed, for now.
- The Verilog netlist must be translated into an equivalent SPICE deck. This is accomplished using the following command:

  ```
  v2lvs -v file.v -o _file.v.sp -w 2
  ```

- The default SPICE deck needed for Calibre LVS requires that the output from `v2lvs` have an underscore as a prefix

Results Viewing Environment (RVE)

- Errors or any output can be easily read through the LVS report, however, its nice to have a method to read output graphically. This is provided by Calibrerve or Results Viewing Environment (RVE).
- To invoke, just type `calibrve`
- You can easily load any results through the directory created by LVS that is typically called svdb or the Mask SVDB directory.
- This directory typically writes Standard Verification Results File information that is standard across Calibre tools.
- RVE can be utilized to identify errors and hopefully fix them.

RVE Screen

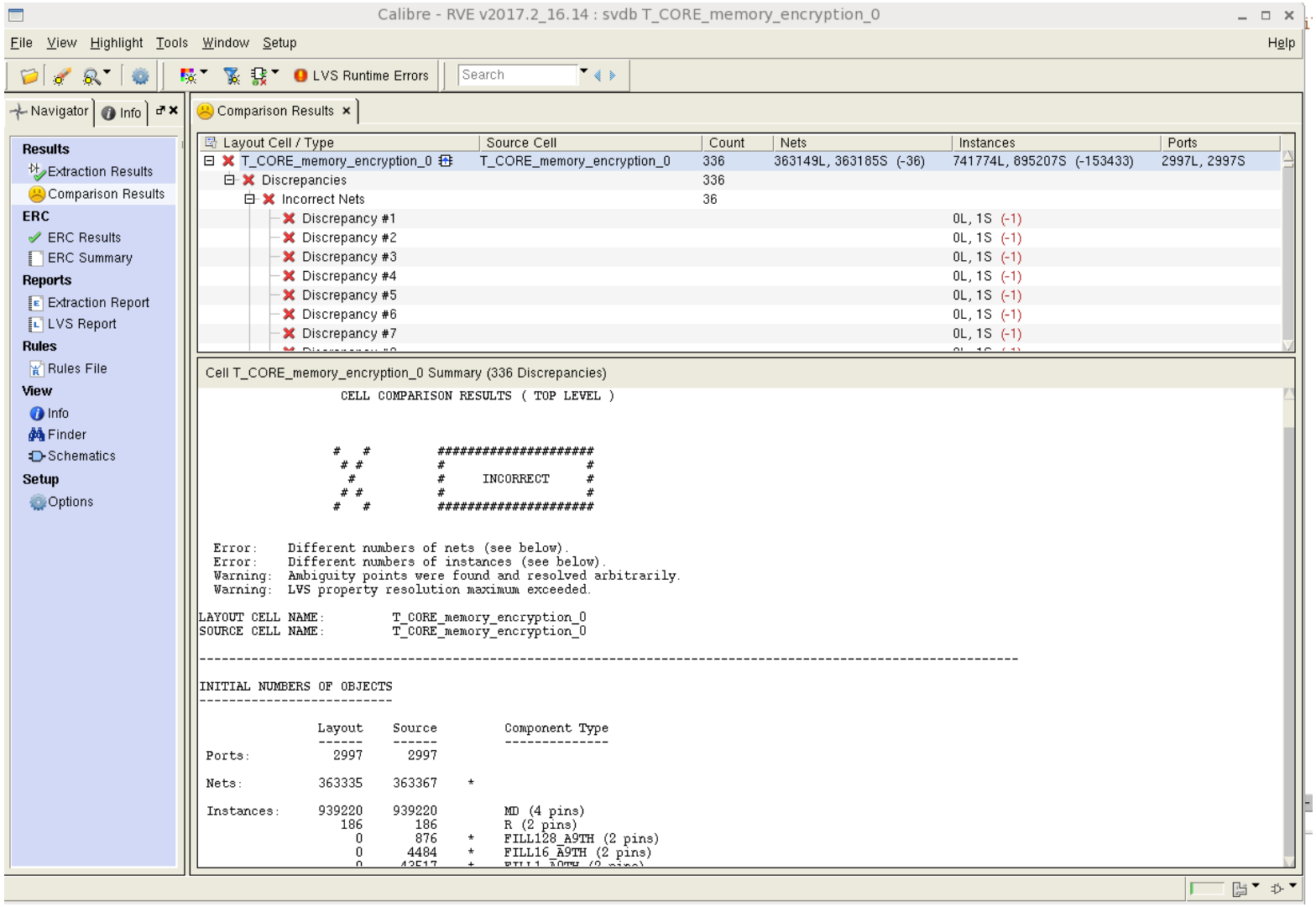

Figure: Calibre RVE Output showing Errors.

RVE Graphical Interaction

- SVDB directories can be loaded within RVE.
- However, sometimes its nice to have some sort of idea what the problems are.
- Although RVE has netlist information, it does not always have information about the layout.
- Fortunately, Mentor Graphics have created a new tool called Calibre DESIGNrev.
  - Calibre DESIGNrev is a layout tool that allows you to pull up layout more efficiently than through CDS's layout editor Virtuoso.
  - The important part about Calibre DESIGNrev is that it has graphical information feedback through RVE (although CDS Virtuoso has this too, but this method is faster).
  - Calibre recommends this tool in their flow for checking and debugging errors as shown on the next slide.
  - To run Calibre DESIGNrev, type `calibredrv`.

RVE Methodology

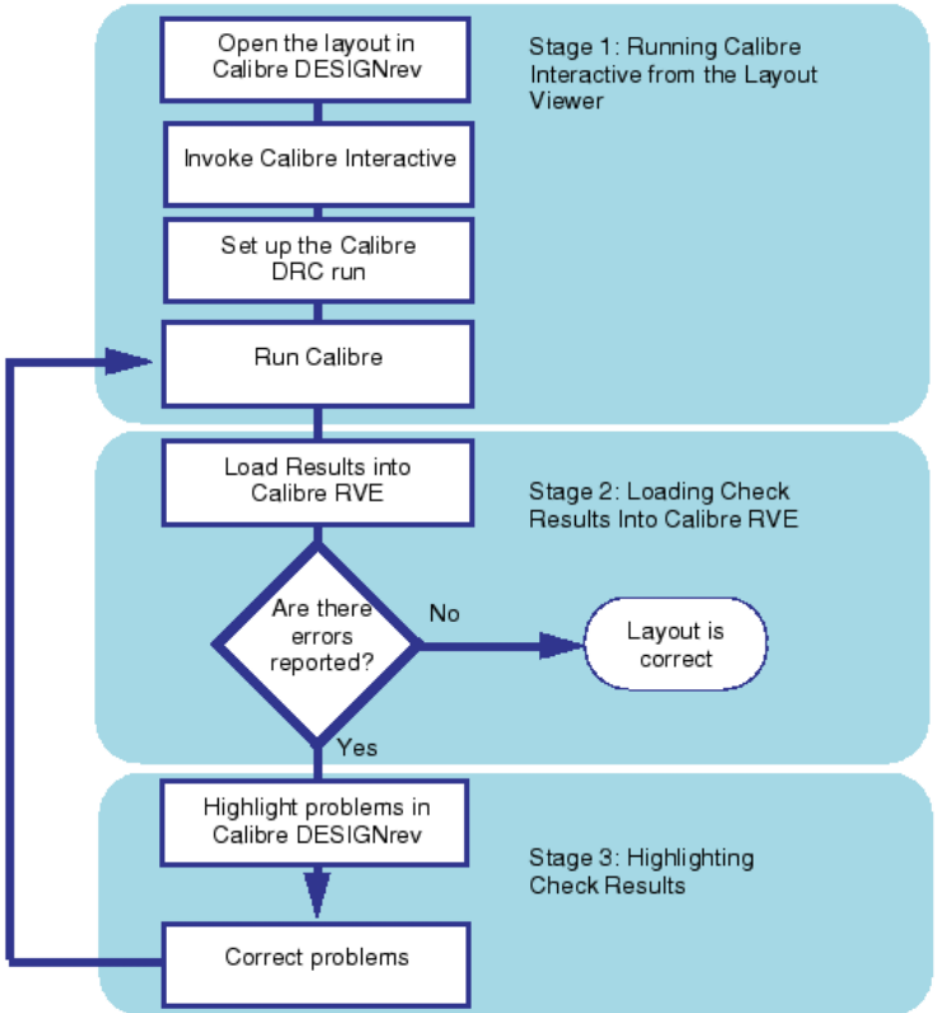

Figure: Calibre RVE FLow.

Calibre DESIGNrev Screen

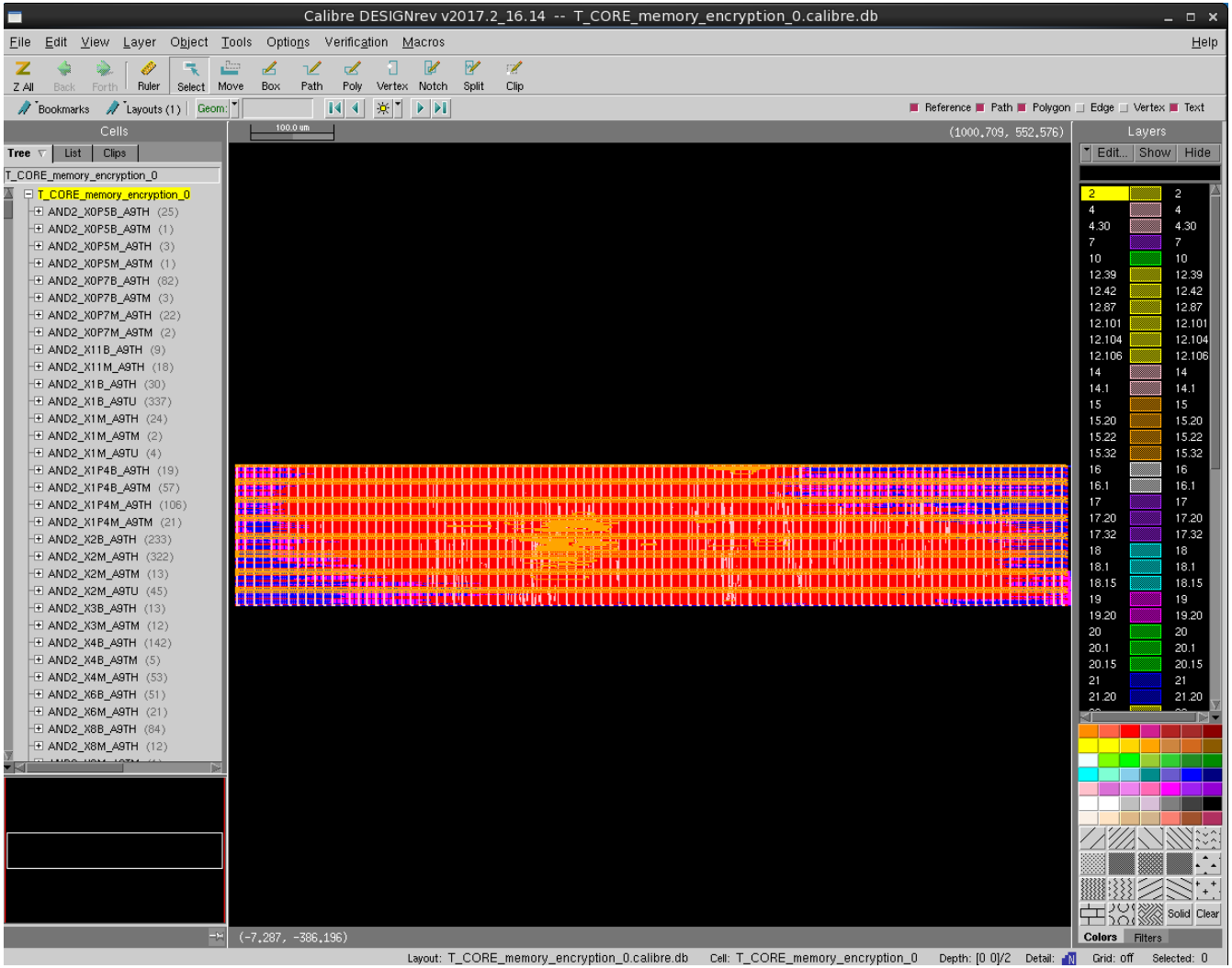

Figure: Calibre DESIGNrev Example Screen.

Calibre DESIGNrev RVE integration

- To run, RVE, first pull up the layout in Calibre DESIGNrev.
- Under the Verification menu in Calibre DESIGNrev, click RVE (interestingly, you can also run any of the Calibre tools from this menu, as well).
- Once RVE is invoked, it will either ask you to enter the svdb directory or it might fill it in for you depending on what information it can extract from the directory.
- Once loaded, any error can be pulled up visually to see any errors.
- To document the problem, a VDD-VSS short is shown in the following slide.

Calibre RVE Error

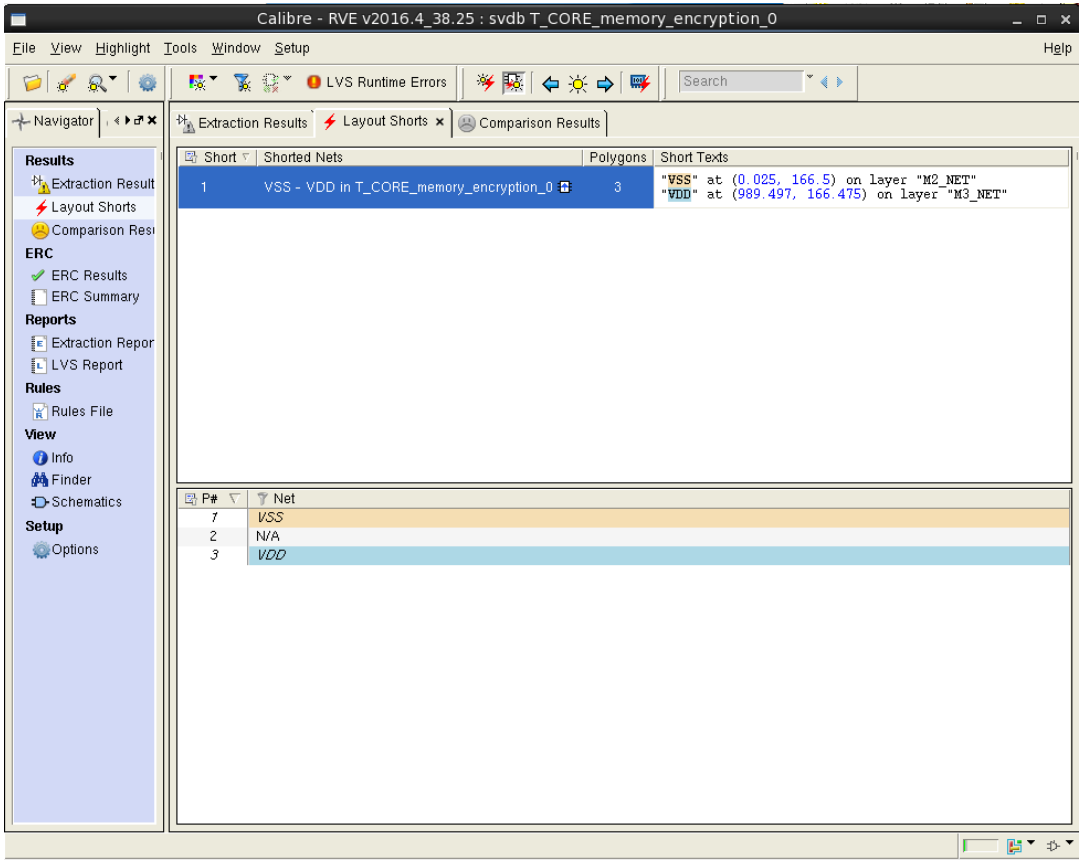

Figure: Calibre DESIGNrev Error showing VDD-VSS Short.

Calibre DESIGNrev Screen

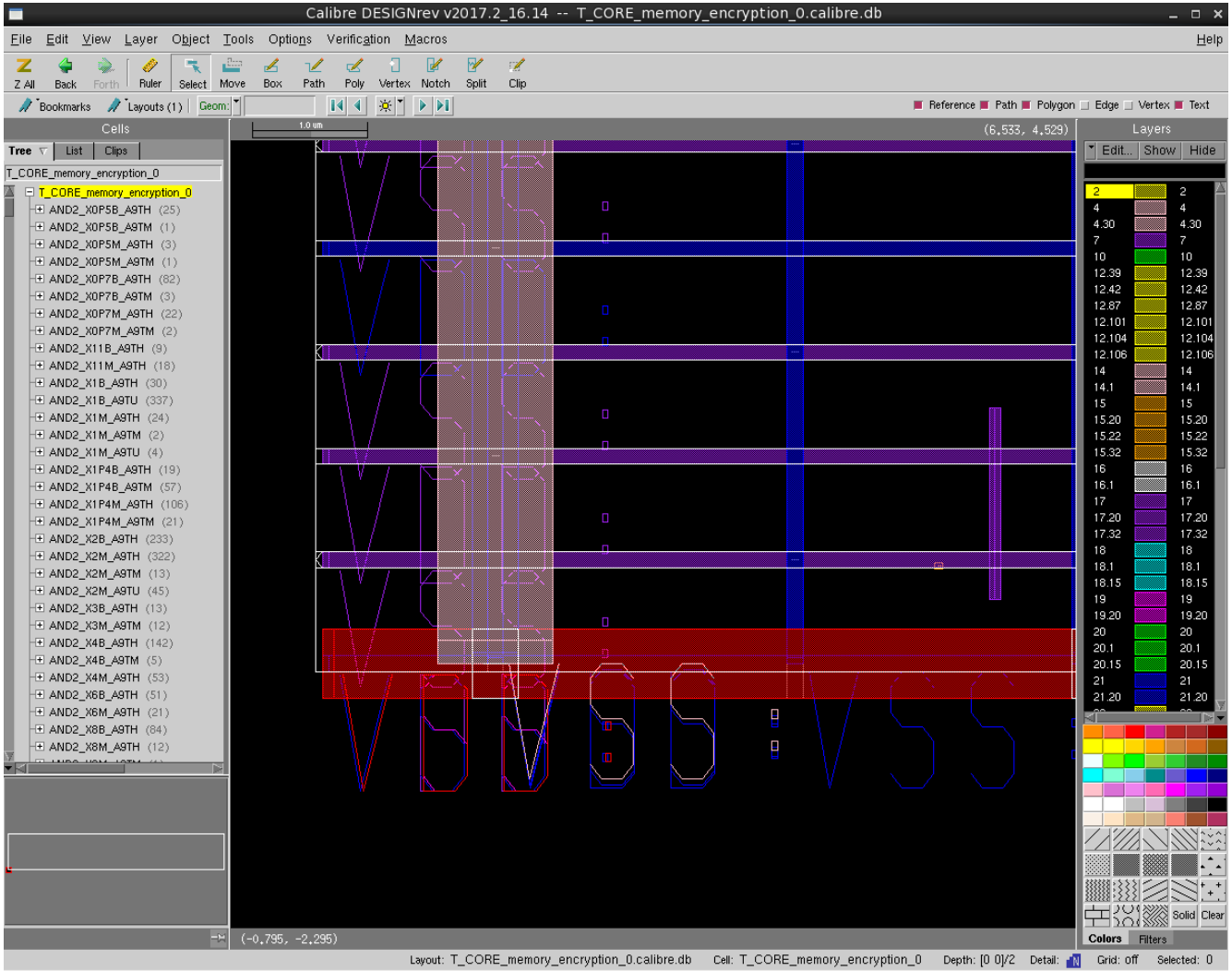

Figure: Clicking RVE screen invokes Calibre DESIGNrev to show error.

Methodology

- The following are the procedure for running the LVS
  1. Locate the gds and netlist you want to compare. This will usually come from CDS' Innovus.
  2. First import for checking[1],
     `strmin -templateFile streamIn.template`

[1] Make sure *soisub!* label is present on layout using SXCUT GDS number 62:10

Summary

- A repeatable batch-based LVS design flow has been introduced.
- Many Mentor Graphics Calibre tools are utilized to help figure out any errors with layout.
- Other helpful scripts to read in and out gds files as well as other small scripts have been introduced.
- All scripts and this presentation are at `/import/vlsi4/IBM_PDK/cmos32soi/LVS_Stream`
- Any feedback is welcome to this process.