Schematic and Layout Verification for the Elefant Digitizer IC

1.0 Introduction

This document details the schematic and layout verification procedures for the Elefant Digitizer IC. There are four main sections included in this document. This section and section 4.0 are used for introductions and conclusions, respectively.

Section 2.0 summarizes the steps needed to insure an accurate schematic and error free layout are created. An outline of the design flow and a description of the software used is also included. Section 3.0 contains a detailed description of the procedures needed to follow the design flow described in section 2.0. The detailed procedures are mainly targeted for the designers working on the Elefant Digitizer IC. Finally, sample control files, for use in controlling the verification software, are included in the appendix.

Strict adherence to these procedures will greatly increase the probability of obtaining a correct design.

2.0 Elefant Digitizer IC Verification Procedures

The Elefant Digitizer IC will be verified with two verification paths. The first path, layout versus schematic (LVS), is used to insure that the schematic representation of the layouts are accurate. This path is needed to insure that complete, error free, documentation of the Elefant Digitizer IC is generated.

The second verification path for the Elefant Digitizer IC creates the final layout and simulation database. This database is used to guarantee the layout is design rule correct and operates correctly. Both verification paths proceed simultaneously and are intertwined.

All verification is done using Mentor Graphics' Checkmate software. The description of the various programs included in Checkmate will be discussed in the appropriate sections. All steps are performed, first, at the sub-cell level and then, secondly, at the global level.

2.1 Layout Versus Schematic (LVS) Verification

LVS verification will be performed to insure that the schematic matches the layout. However, this is not the main verification design flow. This procedure is mainly undertaken to generate a complete and correct documentation package. The files created during the LVS procedure are collectively called the reference database.

To obtain the reference database the following procedures are undertaken. Also, all of the following procedures are undertaken at the sub-circuit and block levels. This methodology simplifies the integration procedure by insuring that all errors in the sub-circuit level are corrected before trying to generate block level circuits. Note that all Checkmate programs are shown in bold.

1) Netlist a SPICE or LSIM file from the schematic from within Led (The layout/schematic editor).
2) Simulate the netlisted file to insure proper functionality. SPICE is an analog simulation program while Lsim is a mixed-mode simulator. SPICE will generate more accurate results. Lsim, however, can be run in the switch mode and is useful for simulation of very large digital circuits
3) Run **transnets** to generate the reference database for use with **netcheck**.

Transnets takes the SPICE file as input and generates the reference schematic verification database (svdb) for use with netcheck. Netcheck is the Checkmate program used to compare the layout and schematic. Once the reference database is created the extracted database must be obtained. The procedures used to obtain the extracted database are listed below.

1) Netlist the layout file from within Led.
2) Run **maskcheck** to look for design rule errors and to create the extracted database for use with **netcheck**. Keep parasitic extraction parameters turned off.
3) Use **outnets** to create an SPICE netlist or **outnetl** to create an LSIM netlist.
4) Simulate the SPICE file to insure proper functionality.

Netcheck can then be run when both the reference and the extracted databases have been created. If the netcheck comparison is error free the next verification procedure can be carried out.

2.2 Final Layout and Simulation Database

When LVS verification has been completed the final layout/simulation database is written. To insure error free final databases, the same layout file used for the LVS verification must be used. Maskcheck is now run once again on the layout file. The only difference from the procedure as described in section 2.1 is that the parasitic extraction parameters are "turned on."

The extracted svdb is then netlisted to either a SPICE netlist or an LSIM netlist. This netlist now contains distributed parasitic elements. It is up to the designer to determine the best level of simulation to insure proper circuit operation. Ideally, a full SPICE simulation will be carried out. This is, of course, not possible for circuits containing a large number of transistors. Digital, switch level, simulations can be performed using Lsim. Lsim can also be used in the RC timing and event driven analog simulation modes. This allows an easy way to transition between sub-block simulation at the full analog level and global, switch level simulations.

It is also important to carry out full analog simulations on portions of big sub-circuits to measure fundamental timing characteristics. Once again, the designer will have to determine when these simulations must be carried out.

The figure included at the end of this document shows the full verification and design flow procedure.

3.0 Detailed Verification Procedures

This section details the actual procedures used to run the various Checkmate programs. Templates of the control files discussed in this section are listed in the Appendix.

3.1 Generation of the Reference Schematic Verification Database

The reference svdb is created using the transnets program. This program is run on a SPICE file which is created from the schematic within Led. It is important to note that the SPICE file must be edited to include the path to the device models. The transnets program is run as shown below. The ">" character represents the UNIX prompt. Italics represent variables.

> > transnets *spice_file_name top_cell_name* svdb -mixed -fullqual -verbose

A log file is created that gives the status of the run and will be named *spice_file_name*.LOG. After transnets is run a reference database exists in the svdb directory.

3.2 Generation of the Extracted Schematic Verification Database

The extracted svdb is created using the maskcheck program. This program requires a control file to control the extraction process. A sample control file is shown in the appendix. Maskcheck is run as shown below.

> > maskcheck -l *technology_file_directory* -t *technology_file_name control_file_name*

Log files are created and named as shown in the control file. During this phase of verification all parasitic extraction is turned off. After maskcheck is run an extracted database exists in the svdb directory.

3.3 Layout Versus Schematic Verification

When both the reference and the extracted svdbs have been created netcheck is run to perform the LVS verification. A sample netcheck control file is shown in the appendix. Netcheck is run as shown below. Netcheck uses the reference and extracted databases in the svdb directory.

> > netcheck *netcheck_ctrl_file*

Log files are created and named as shown in the control file.

It is crucial in this phase of verification that the Netcheck control file not be over-specified. This applies in particular to the specification of equivalencies between the reference and extracted database. An iterative solution is best, with the equivalencies between the reference and extracted databases applied only after an initial Netcheck run. This methodology insures that errors between the databases are not masked by specifying equivalencies that don't actually exist.

3.4 Final Simulation and Verification Database

The final simulation and verification database is created with maskcheck as discussed in section 3.2. In this phase, however, the parasitic extraction parameter is "turned on."

3.5 Netlisting

Once the final extracted database has been created the simulation netlists are created with either the outnetl or outnets Checkmate programs. Outnetl creates an LSIM netlist while outnets creates a SPICE netlist. Both programs require a control file. A sample outnets control file is listed in the appendix. Outnet(x) is invoked as shown below.

> outnets *outnets_ctrl_file*

The control file for either outnetl or outnets controls how the devices are netlisted. It is possible to modify device characteristics for more accurate simulation. For example, the reduction in width of the transistors due to lithography can be included in this phase.

3.6 Failsafe Check

When all of the above checks have been successfully completed the GDSII file is created for submission to MOSIS. The failsafe check uses a completely different verification path to insure a correctly functioning Elefant Digitizer IC.

The extracted database, without parasitic information, is used to create the extracted SPICE netlist. The GDSII file is then input to MAGIC for two purposes. The first purpose is to use MAGIC for another level of design rule checking. Then an another netlist, created from within MAGIC, is compared with the netlist created from the extracted database. GEMINI is used to compare the two netlists.

Once it is verified that no errors exist between the two netlists the GDSII file can be submitted to MOSIS.

4.0 Conclusion

By following the above procedures the probability of obtaining a correctly functioning IC is greatly enhanced. These procedures should be strictly followed. Also, any additional steps that the designer thinks appropriate should also be included.

Appendix

Maskcheck Control File

```
OPTIONS
      INPUT LAYOUT_DB              = <file name>.L
      CELL TO PROCESS              = <cell name>
      INTERNAL UNITS               = 0.001 MICRONS
      DATABASE UNITS               = 0.01 MICRONS
      DRC DIAGNOSTICS FILE         = <file name>.drc_diag
      ERC DIAGNOSTICS FILE         = <file name>.erc_diag
      RULES FILE NAME              =
/home/edafiles/gdt/gdtcore_tech/sce3m08hp.dir/sce3m08hp.drc

      CHECK SHORTS ON INPUT LAYERS  = YES
      DATA FORMAT                   = gdt
      DEFAULT DIAGNOSTIC LAYER      = 23
      DEFAULT OUTPUT LAYER          = 24
      DRC ERRORS TO LAYOUT          = YES
      FULLQUAL                      = YES
      LIST DRC ERRORS               = YES
      LIST INPUT SIGNAL LOCATIONS   = YES
      OUTPUT CELL NAME              = cm_drc
      OUTPUT LAYOUT_DB              = cm_drc.L
      OVERWRITE                     = YES
      SAVE PARASITIC FILES          = NO
      SHORTS LAYER                  = 25,25
      SHORTS TO LAYOUT              = YES
      SKIP PARASITICS               = YES
      SUPPRESS ALL CHECKS           = YES
END OPTIONS
```

Netcheck Control File

```
COMPARE
      REFERENCE SV_DB = svdb
      EXTRACTED SV_DB = svdb
END

OPTIONS
      BUILD COMPARE MEMBERS = YES
      BUILD MERGE MEMBERS = YES
      ERROR OUTPUT FILE = nc.errors
      GUESS POTENTIALS = NO
      IDENTIFIED PRINT = YES
      IDENTIFIED PRINT FILE = nc.ident
      LIST EQUATED NODES = YES
      LIST NO CONNECTION NODES = YES
      LIST NON EQUATED NODES = YES
      OUTPUT EQUATES FILE = nc.eq_sigs
      PARALLEL = ALL
END
```

Outnets Control File

```
SV_DB = svdb
OUTPUT DEVICES = YES
OUTPUT PARASITIC DEVICES = YES
SPICE OUTPUT FILE = <output file name>

OPTIONS

INSERT AT TOP
.options accurate post
END

INSERT AT BOTTOM
.include '/home/edafiles/gdt/gdtcore_tech/sce3m08hp.dir/sce3m08hp.bsim'
END

OUTPUT CASE = MIXED
LOCATION IN DEVICE NAME = NO
DEVICE PREFIX = "_"
MAX DEVICE NAME LENGTH = 16
SIGNAL PREFIX = node
MAX SIGNAL NAME LENGTH = 16

PARASITIC REQUESTS
      UNNAMED SIGNAL PREFIX = usp
      OUTPUT PARASITIC = RC
END

PARAMETER REQUESTS
      DEVICE TYPE=NMH PARAM=W NEW VALUE="W - 0.366E-6"
      DEVICE TYPE=NMH PARAM=L NEW VALUE="L + 0.0658E-6"
      DEVICE TYPE=PMH PARAM=W NEW VALUE="W - 0.377E-6"
      DEVICE TYPE=PMH PARAM=L NEW VALUE="L + 0.0896E-6"
END
END
```