Computer-Aided VLSI System Design

EC Lab: Design Verification

==========

Motivation and Importance of Equivalence Checking

Revised models, which are generated with different abstraction level design from synthesis tools based on a golden model, are desired to be functional equivalent with the golden model. Since contiguously simulation on these designs throughout the design flow costs extremely long time, designers prefer to perform equivalence checking between design versions, rather than simulate on every revised version.

Main Objectives of this Lab:

To understand how to use **Conformal LEC** to *formally* check the equivalence between your VLSI designs in the design flow

Lab: RTL Verilog v.s. Synthesized Gate-Level Verilog Designs

Objectives:

1. Understanding LEC command format.
2. Setup environment for simple equivalence checking task.
3. Run LEC in shell mode / script mode.

Related Files:

In this Lab, we intend to perform equivalence checking between two Verilog designs**.** We will use design ***alu.v*** for demonstration. Feel free to use your design in your ***previous HWs***.

Files are listed as below:

Sample File Name	Description
alu.v	Verilog Code
alu_syn.v	Synthesized Gate-Level Verilog
tsmc13.v	TSMC Cell Library Verilog File

Step 0. **Generate a directory to put all the files in.**

Shell Command: *mkdir Lab9*

cp <file> Lab9

Set current directory to Lab9.

Source LEC licence file : *source /usr/cadence/CIC/confrml.cshrc*

Note: Some machines cannot run LEC successfully. Available machines are cad38 and cad43 . (Tested on 06/06)

Step 1. **Start LEC.**

Shell Command : *lec –nogui*

Step 2. **Setup log file. (Optional)**

LEC Command : *set log file Lab9.log*

Step 3. **Read RTL Verilog design as Golden Model.**

LEC Command : *read design alu.v -golden*

Note: You can see some warning and information messages from LEC standard output. Please make sure your design has been read successfully.

Step 4. **Read Gate-Level Verilog design as Revised Model.**

LEC Command : *read design alu_syn.v -revised*

However, you may found some error message outputs from LEC RTL checker.

What's the problem?

➔You should specify cell libraries instantiated in your gate-level design to LEC.

Step 5. **Read Verilog Library File for Gate-Level Revised Model.**

LEC Command : *read library tsmc13.v -verilog -both*

Step 6. **Read Gate-Level Verilog design as Revised Model again.**

As the library file has been read in, we may expect to successfully read in the synthesized circuit. Please perform *Step 4* again and your design should be read successfully now.

Step 7. **Modes in LEC:**

LEC has two modes : ***Setup Mode*** and ***LEC Mode*** for enabling different operations:

1. Setup Mode :

Mainly for design / constraints / rename / black box setup before compare.

2. LEC Mode :

Compare mode for adding compare points and report compare results.

Some commands are available in both modes; you can type *help* to see all commands. As all setup constraints have been applied, now we should change current mode from ***Setup*** to ***LEC*** for further compare:

LEC Command : *set system mode lec*

You can see LEC automatically find compare points and makeup constraints from setup mode specifications, and finally demonstrate the statistical information of mapped points as shown below.

```
SETUP> set sys mode lec
// Processing Golden ...
// Modeling Golden ...
// Processing Revised ...
// Modeling Revised ...
// Mapping key points ...
================================================================================
Mapped points: SYSTEM class
--------------------------------------------------------------------------------
Mapped points     PI     PO     DFF       Total
--------------------------------------------------------------------------------
Golden            11     4      4         19
--------------------------------------------------------------------------------
Revised           11     4      4         19
================================================================================
LEC>
```

Step 8. **Add Compare Points:**

We should specify points in both designs that we want to compare for LEC. In general, all PO and DFF points are desired to be compared. However, you can manually add or delete compare points for LEC on your purpose. In this Lab, we simply add all PO and DFF as compare points.

LEC Command : *add compare points -all*

Step 9. **Compare:**

Since we have set up all indications to LEC, we are now starting to perform EC.

LEC Command : *compare*

You may find LEC compare results shown directly on the screen :

```
LEC> add compare points -all
// 8 compared points added to compare list
LEC> compare
================================================================================
Compared points      PO     DFF       Total
--------------------------------------------------------------------------------
Equivalent           4      4         8
================================================================================
LEC>
```

Step 10. **Exit LEC.**

If your result is equivalent, you have completely proved that your synthesis is sound and the functionalities of synthesized design are the same as those of the original design. Now, you can exit LEC.

LEC Command : *exit -f*

If you have set the log file in *Step 2*, you can see all messages on your screen output in that file. For script generation, you can type as following and all commands related to what you have done will be stored in that dofile.

LEC Command : *save dofile Lab9.script*

Next time you can simply run that script by type this

LEC Command : *dofile Lab9.script*

If you want to run script only, you can also pass dofile as parameter for LEC in *Step 1*.

Shell Command : *lec -nogui -dofile Lab9.script*