

Analog Custom Design and Testing Using OCEAN Scripting in Cadence

Mostafa Rahimi Azghadi
School of Electrical and Electronic Engineering,
The University of Adelaide,
Adelaide, SA 5005, Australia
Email: mostafa@eleceng.adelaide.edu.au

1) Introduction

This is a crash tutorial for new cadence users who require post-processing on their simulation data in their design performed in Cadence.

One of the main drawbacks of the ADE is the fact that there is no powerful command or option in its Graphic User Interface (GUI), to perform repetitive tasks. In addition, ADE has no direct solution in order to perform usual design verifications, which are needed for various situations such as various process corners, temperatures, and supply voltage. Instead the user has to run each simulation manually, which seems inapplicable for large scale simulations.

OCEAN that stands for Open Command Environment for Analysis is a powerful programming language that can automate your simulations within Cadence. It is a subset of SKILL language and uses this language to configure the design environment. SKILL is a full language which possesses all features of programming languages such as data types, operators, etc. However, SKILL is not the focus of this tutorial.

OCEAN helps the Cadence Analog Design Environment (ADE) user, to facilitate required simulations for the target design. It allows you to perform the simulation from the command line and control it from start to finish. Utilising OCEAN will save you a lot of time performing post-processing and simulations. The OCEAN script that is fairly complicated can be automatically generated by the ADE. However, you can easily modify it and precisely control all aspects of your simulation.

Here are some of the cases that you could use OCEAN and it helps your simulation and post-processing become very simple.

1) Repetitive simulations

This is usually helpful when there are needs for PVT analysis i.e. multiple Processes, Voltages, and Temperatures.

2) Parametric analysis

Usually you are not able to perform parametric analysis using the GUI of the ADE. Therefore, you will need to do different parametric simulations manually, if you do not use OCEAN.

3) Post-processing your simulation data

Once you need to use commands such as measure, you will find OCEAN very helpful. It will let you sweep the parameters and process your simulation output as you wish.

4) Documenting and saving your simulation results.

This is interesting how you could save your simulation results in a spread sheet format and perform lots of operations on it. As an example, one is easily able to write a script to run the required simulation, save the needed data, and even email them to an address!

There are many other instances that OCEAN can be useful in performing simulations.

2) Generating an OCEAN script using ADE

Open you ADE GUI and set up your required simulation. In this GUI, you are able to determine the simulation type (i.e. DC, AC, Tran), simulation time, and other required details for your simulation. For more information regarding to ADE, the reader is directed to cadence manual.

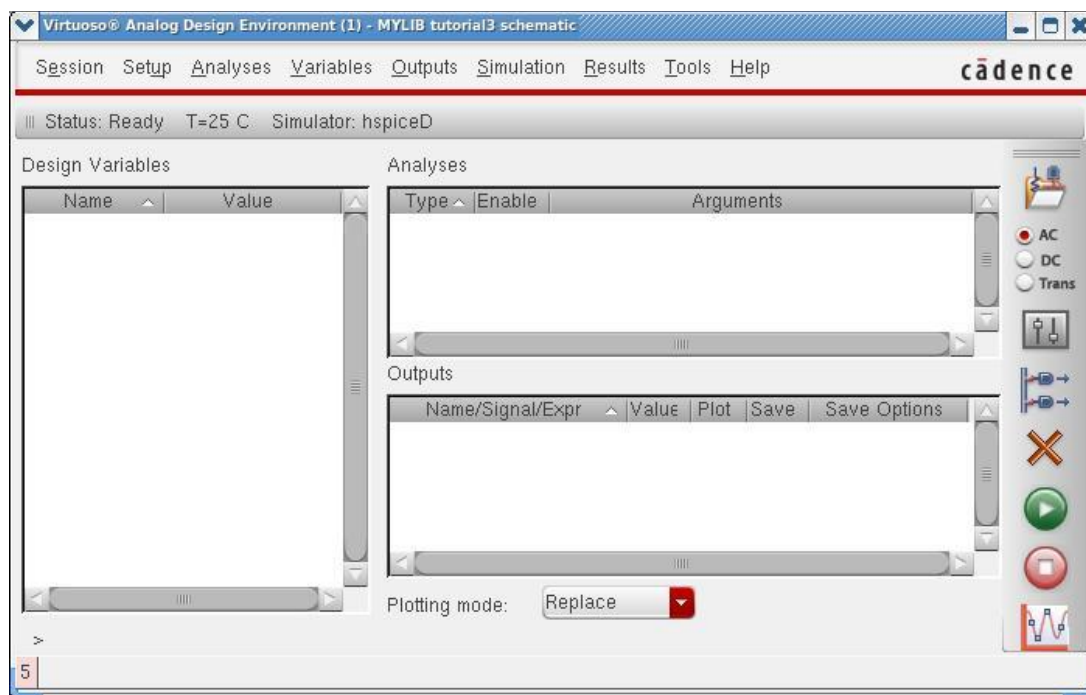


Fig. 1. Analog Design Environment GUI. Using this GUI you can use the right column, or the related options, to set up your simulation as required.

After setting up your simulation in the GUI, hit simulation run button as shown in Fig. 1. If there is no error your simulation should run and you should see a run window, showing various stages of your simulation. If you wanted to see an output from your simulation, which can be easily picked from your schematic using the GUI, the output should pop up after the simulation ended and be shown on an output waveform.

The good news is you now have your OCEAN script produced by ADE. Simply click on

Session > Save Ocean Script

This option of your ADE and you will save your OCEAN script command under a preferred name. Now you see your OCEAN script file which tells you about most of the facts that you have already determined when running your simulation, such as the simulator, the design directory, your results directory, etc. The sample OCEAN file is shown in Fig. 2.

The listed information in the OCEAN script is as follows:

- 1) `simulator('spectre)` that chooses the simulator.
- 2) `design("/address/to/yourdesign")` that points to the netlist of the design.
- 3) `resultsDir("/address/to/yourresults")` specifies the results directory.
- 4) `modelFile("/address/to/yourmodelfiles")` specifies the model files for your design.
- 5) `analysis('tran ?stop "60m")` lets the user to set the simulation type, start and stop times, step size and accuracy of the simulation.
- 6) `desVar()` allows the user to set all the design variables just like in the GUI.
- 7) `temp(27)` sets the simulation temperature.
- 8) `run()` simply runs the simulation and it has been created because you pressed the simulation button on the GUI.

```
simulator( 'spectre )
design("/address/to/yourdesign"      )
resultsDir("/address/to/yourresults" )
modelFile("/address/to/yourmodelfiles"  )
analysis('tran ?stop "60m"  )

desVar(      "RI" 10K      )
desVar(      "RF" 40K      )

temp( 27 )
run()
```

Fig. 2. Sample OCEAN Script produced by ADE. Different lines in this script perform various tasks.

There are other commands and features in OCEAN which helps the user to process their simulation data.

The first five lines are values from the setup pull-down menu in the ADE. In fact every action that you do in the ADE, has an equivalent function in the OCEAN script file.

3) Executing an OCEAN script

There are a few ways to execute the OCEAN script. All these three ways are equal to pressing the simulation button in the ADE. However, we could modify the OCEAN script and make our required changes in the file to do the needed post-processing, saving, and documentation.

The simplest way to run an OCEAN script is to simply load it from the command line as follows

```
load "thenameofyourscript.ocn"
```

This will run the simulation, just like you hit the simulation button on the ADE.

The second way to run the OCEAN script is from the OCEAN command line. For that you should bring up a shell window and type

```
elecpg_Mostafa> ocean
```

This will bring up the interactive OCEAN command session. Now you could simply load your OCEAN script and run it. So in the ocean command line we type

```
ocean> load "thenameofyourscript.ocn"
```

This will run the simulation in a similar way.

In this command line you have the ability to type and use various commands. Examples of these commands are as follows:

`average (...)` gives us the average of an specific measurement in the circuit.

`value (...)` gives us the exact value of a signal at a specific point of time.

`hardCopyOptions(?hcOutputFile "your_outputfile_name.png")`

`hardcopy()`

These commands sent a simulation output to a printer or a file. For instance the above mentioned commands, we saving the output to a PNG file.

```
reportFile=outfile("your_outputfile_name.txt")
```

```
fprintf (reportFile "output control options" required measured value in your simulation)
```

The above mentioned commands firstly open an output file and assign it to reportFile variable, then using fprintf the user could write all their required outputs which are in match with the "output control options" in "your_outputfile_name.txt" file.

Close(reportFile) closes the output simulation report file after the writing operation was completed.

The structure of fprintf command in this language is pretty similar to C programming language.

4) Examples of useful ocean script

Another important feature in the OCEAN is the ability of implementing loops in a simulation as shown in Fig. 3.

```
i = 1;
foreach(corner corners
openResults(strcat(defaultDir "/" corner "/psf" ))
selectResult( 'tran )
plot(getData("/idut/vdd") )
iavg = average(clip(IT("/idut/vdd") 0.0006
0.0012))*1e9
printf("%10.5f is the average current at %s and
%5.2f \n", iavg, corner,nthelem(i temper) )
i = i +1
)
```

Fig. 3. Sample for loop implementation command in OCEAN Script. The output will be shown on the screen.

The first two commands in the code determine the loop counter and the loop changing value.

The third command opens the results from the specified path.

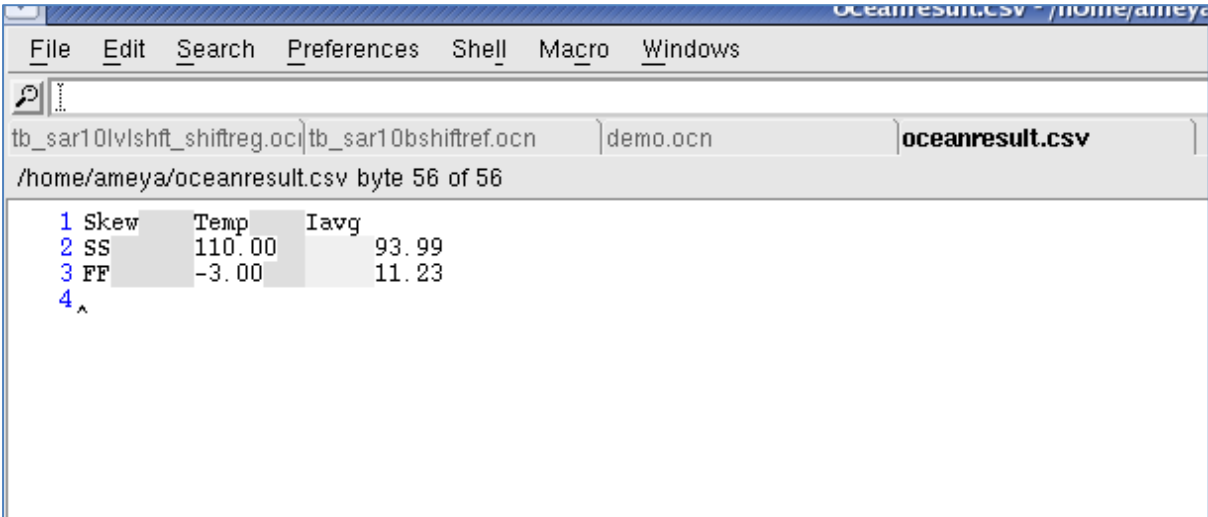
The fourth command is to select the transient analysis results. It could also be DC or AC, depending on the user's needs.

The fifth command plots any stored waveforms.

The sixth command is just a customized function that uses a few predefined functions such as clip and average, to calculate the required value.

The seventh command is the output command that prints results on the screen.

The output of this sample for loop code has been shown in Fig. 4.



```
Oceanresult.CSV - /home/ameya
File Edit Search Preferences Shell Macro Windows
tb_sar10lv1shft_shiftreg.ocn | tb_sar10bshftref.ocn | demo.ocn | oceanresult.csv
/home/ameya/oceanresult.csv byte 56 of 56
1 Skew Temp Iavg
2 SS 110.00 93.99
3 FF -3.00 11.23
4 ^
```

Fig. 4. Output results of the sample ocean script shown in Fig. 3. This is in result of the printf command that outputs the mentioned values in the mentioned order.

Alternatively, the printf command could be a *fprintf* command and outputs the same results into a predetermined file as described above and shown in Fig. 5.

The first command is to open a file in write mode "w", and assign it to the file pointer (fp).

The second command is to write the words Skew, Temp, and Iavg with a tab between them, into the file with the pointer fp.

The loop commands are pretty similar to the previous example. The only difference is, here we write the output into a file, and not on the screen. The output of the code shown in Fig. 5 is shown in Fig. 6.

```

fp = outfile( "/home/ameya/oceanresult.csv" "w");
fprintf(fp "Skew\tTemp\tIavg\n")
i = 1;
foreach(corner corners
openResults(strcat(defaultDir "/" corner "/psf" ))
selectResult( 'tran )
iavg = average(clip(IT("/idut/vdd") 0.0006 0.0012))*1e9
fprintf(fp "%s\t%5.2f\t%10.2f\n", corner,nthelem(i temper),iavg )
i = i +1
)
close(fp);

```

Fig. 5. Sample for loop implementation command in OCEAN Script. The output will be shown in the output file.

The user can simply read from a file in a similar way to C language. The C-like format of the commands should be highlighted that really helps the user to write their own required commands, more simply.

There are also plenty of other useful commands, which could be utilised in your simulation. For a complete list of commands, the reader is directed to the OCEAN Reference provided by Cadence.

The above mentioned commands are not usually used in the OCEAN command prompt, but they are usually saved in a script file and then should be executed in one of the above mentioned ways in Section 3, in order to run the simulation.

	A	B	C	D	E	F	G	H
1	Skew	Temp	lavg					
2	SS	110	93.99					
3	FF	-3	11.23					
4								
5								
6								
7								
8								
9								
10								
11								
12								

Fig. 6. Output results of the sample ocean script shown in Fig. 5. This is in result of the printf command that outputs the mentioned values in the mentioned order into the output file.

One major drawback of ADE is that you could work on one single design simulation at a time and if you do need to run several simulations, you either need to run them sequentially in your single session, or run them in multiple concurrent sessions. Both of these approaches are hard to cope with and very hard to manage.

In order to solve this problem, Cadence has introduced ADE XL, and OCEAN XL. These new features are not the concern of this tutorial.

5) References

[1] <http://www.cadence.com/Community>