

# **EL 644: VLSI Systems and Architectures**

---

## **SKILL Programming**

**Garrett S. Rose**



**Spring 2007**

# Introduction

- ❑ **SKILL – Cadence scripting language, form of LISP**
- ❑ **Cadence GUI interface is supported by SKILL code**
- ❑ **SKILL code is driven by database syntax**
- ❑ **Anything you can do with the Cadence GUI, you can do with SKILL**
- ❑ **Key to SKILL is a large set of library functions that allow you to manipulate *data structures* such as cells, nets, mask information, etc.**

# Learn from the GUI

- ☐ As you perform operations in Cadence, keep an eye on the ICFB window
- ☐ These operations often show up in the ICFB window as functions that can be used in your own script
- ☐ You can also start with the Analog Environment

# SKILL Simulation

- ❑ Start by setting up a Spectre simulation in the Analog Environment – save the state
- ❑ Now, select Session -> Save Script ...
- ❑ You now have a script version of your simulation



# A DC Sweep

```
simulator( 'spectre )
design("/home/batcher/u1/faculty/grose/cadence/simulation/inverter/spectre/schematic/netlist/netlist")
resultsDir( "/home/batcher/u1/faculty/grose/cadence/simulation/inverter/spectre/
schematic" )
modelFile(
    ("/home/batcher/u1/faculty/grose/cadence/models/spectre/tsmc25P.m" "")
    ("/home/batcher/u1/faculty/grose/cadence/models/spectre/tsmc25N.m" "")
)
stimulusFile( ?xlate nil

    "/home/batcher/u1/faculty/grose/cadence/simulation/inverter/spectre/schematic/netlist/_graphical_stim
    uli.scs")
analysis('dc ?param "VIN" ?start "0" ?stop "2.5" ?step "0.1" )
desVar( "VIN" 0 )
desVar( "VDD" 2.5 )
temp( 27 )
run()
selectResult( 'dc )
plot(getData("/OUT") )
```

**Simulation setup**

**Analysis setup**

# A Routine for Simulation

## procedure(INVSIM())

```
simulator( 'spectre )
design("/home/batcher/u1/faculty/grose/cadence/simulation/
  inverter/spectre/schematic/netlist/netlist")
resultsDir( "/home/batcher/u1/faculty/grose/cadence/simulation/inverter/spectre/
  schematic" )
modelFile(
  ("/home/batcher/u1/faculty/grose/cadence/models/spectre/tsmc25P.m" "")
  ("/home/batcher/u1/faculty/grose/cadence/models/spectre/tsmc25N.m" "")
)
stimulusFile( ?xlate nil
  "/home/batcher/u1/faculty/grose/cadence/simulation/inverter/spectre/
  schematic/netlist/_graphical_stimuli.scs")
analysis('dc ?param "VIN" ?start "0" ?stop "2.5" ?step "0.1" )
desVar( "VIN" 0 )
desVar( "VDD" 2.5 )
temp( 27 )
run()
selectResult( 'dc )
plot(getData("/OUT") )

)
```

- ☐ Create a procedure from your code
- ☐ Save this file with the procedure name
- ☐ To run, you will need to load into ICFB

# Running the New Routine

- ❑ In the ICFB window: enter load(<path/filename.il>)
- ❑ For our example:



The image shows two screenshots of the ICFB (Interactive Circuit File Browser) window. The window title is "icfb - Log: /home/batcher/u1/faculty/grose/CDS.log". The menu bar includes "File", "Tools", "Options", and "Help". The status bar shows "1".

The first screenshot shows the initial state of the window. The command prompt is at the bottom, and the command "load('/home/batcher/u1/faculty/grose/INVSIM.il')" is entered. The log area shows the following text:

```
t
clear
Loading skillDev.cxt
t
```

The second screenshot shows the window after the simulation has completed. The command prompt is at the bottom, and the command "INVSIM()" is entered. The log area shows the following text:

```
simulation completed successfully.
reading simulation data...
... successful.
t
```

# Running the New Routine

- ❑ This little script simulates an inverter
- ❑ Results appear automatically

