• Designing the first five cells in your library
  • Multiple cell views
• ELC library characterizer
• Abstract generator
• Synopsys database generation
  • Using the cells in synthesis

CS/ECE 6710 Tool Suite
In the CAD Book

- Chapter 8 on Cell Characterization
  - Section 8.2 describes ELC
  - Section 8.4 describes converting from .lib to .db format (used by Synopsys Design Compiler)
  - (.lib is used by other tools…)

Start with Cells
Layout View

Schematic View

Use cmos_sch as the view type for your cell schematics. This is to differentiate them from schematics that use the cells.
Symbol View

Behavioral View

```
module NAND2X1 (Y, A, B);
    output Y;
    input A;
    input B;
    nand _i0(Y, A, B);

    specify
    {A => Y} = {1,0, 1,0};
    {B => Y} = {1,0, 1,0};
endspecify
endmodule
```
Start with six views of each cell

1. layout - make sure to use the template!
2. cmos_sch — use this view type for a schematic
3. symbol — Make them look nice
4. behavioral — Having this view makes simulation go much faster (if you use it)
5. extracted — generated from the extract processes
6. analog_extracted — after LVS

Characterize the cells

- Run a set of analog (Spectre) simulations that builds a table of delay values
  - Input drive vs. output load
  - For all outputs
  - Plus rise and fall times
  - And some power information
- Encounter Library Characterizer (ELC)
  - cad-clc from the bin directory
  - Four steps – step1, step2, step3, cad-alf2lib
Single Schematic with All Cells
Create Netlist of that ss-test Cell

Call it foo.scs (for example)
Convert to ELC format

Run ELC

- **Encounter Library Characterizer**
  - Figures out what each cell is (logic)
  - Generates test inputs for Spectre
  - Runs Spectre
  - Checks output and extracts timings
  - Formats the output in .alf format

- I like to make a new VLSI/ELC directory from which to run this tool…
  - Copy dut.scs into that new directory…
Results from ELC step1

```
... 
  elc> db_gate
  --------------------------------------------
  DESIGN : INVX1
  --------------------------------------------
  - NOT ( Y, A );
  --------------------------------------------
  DESIGN : NAND2X1
  --------------------------------------------
  - NAND2 ( Y, A, B );
  --------------------------------------------
  DESIGN : NORX1
  --------------------------------------------
  - NOR ( Y, A, B );

  Lots of text missing from these highlights...
```

```
... 
  cad-elc -S step1
  
  Results from ELC step1
  
  Terminal — ssh — 97x31
  
  Writing foo.ipdb/HM021.desdb/body/typo    
  Reading foo.ipdb/HM021.desdb               
  --------------------------------------------
  DESIGN : HM021                               
  --------------------------------------------
  - A  B : Y                                   
  --------------------------------------------
  DB000: R 1 : F  DELAY(R)                    
  DB001: R 0 : 1  PWR(R)                     
  DB050: F 1 : R  DELAY(R)                   
  DB051: F 0 : 1  PWR(R)                     
  DB054: 1  F : R  DELAY(R)                   
  DB055: b R : 1  PWR(R)                     
  --------------------------------------------
  - 8 vectors generated
  Writing foo.ipdb/HM021.desdb/simulate/spec
  Writing foo.ipdb/HM021.desdb/simulate/obek
  Writing foo.ipdb/HM021.desdb/boundary/dec
  Writing foo.ipdb/HM021.desdb/body/typo      
  Reading foo.ipdb/HM021.desdb                
  --------------------------------------------
  DESIGN : HM021                               
  --------------------------------------------
  - A  B : Y                                   
  --------------------------------------------
```
cad-elc -S step1

Results from ELC step1

```
DESIGN ( INVX1 )

PORT DEFINITION
INPUT A ( A );
INPUT B ( B );
OUTPUT Y ( Y );
SUPPLY VDD ( VDD );
SUPPLY VSS ( VSS );
SUPPLY VIL,VIH ( VIL, VIH );
SUPPLY VM ( VM );
SUPPLY _NRS ( _NRS );

INSTANCES
INVX1 ( Y, A, B );
```

```
DESIGN ( NAND2X1 )

PORT DEFINITION
INPUT A ( A );
```

cad-elc -S step2

Results from ELC step2

elc> db_spice -s spectre -p typical -keep_log

```
DESIGN PROCESS #ID STATUS IPDB
--------- ------- ----- ------ ----
INVX1     typical D0000 SIMULATE foo
INVX1     typical D0001 SIMULATE foo
INVX1     typical D0002 SIMULATE foo
INVX1     typical D0003 SIMULATE foo
INVX1     typical D0004 SIMULATE foo
INVX1     typical D0005 SIMULATE foo
INVX1     typical D0006 SIMULATE foo
INVX1     typical D0007 SIMULATE foo
INVX1     typical D0008 SIMULATE foo
NAND2X1   typical D0000 SIMULATE foo
NAND2X1   typical D0001 SIMULATE foo
NAND2X1   typical D0002 SIMULATE foo
NAND2X1   typical D0003 SIMULATE foo
NAND2X1   typical D0004 SIMULATE foo
NAND2X1   typical D0005 SIMULATE foo
NAND2X1   typical D0006 SIMULATE foo
NAND2X1   typical D0007 SIMULATE foo
NAND2X1   typical D0008 SIMULATE foo
NOR2X1    typical D0000 SIMULATE foo
NOR2X1    typical D0001 SIMULATE foo
NOR2X1    typical D0002 SIMULATE foo
```

Lots of text missing from these highlights…
Results from ELC step 2

<table>
<thead>
<tr>
<th>DESIGN</th>
<th>PROCESS</th>
<th>#ID</th>
<th>STAGE</th>
<th>STATUS</th>
<th>IPDB</th>
</tr>
</thead>
<tbody>
<tr>
<td>INVX1</td>
<td>typical</td>
<td>D0000</td>
<td>VERIFICATE</td>
<td>PASS</td>
<td>foo</td>
</tr>
<tr>
<td>INVX1</td>
<td>typical</td>
<td>D0001</td>
<td>VERIFICATE</td>
<td>PASS</td>
<td>foo</td>
</tr>
<tr>
<td>NAND2X1</td>
<td>typical</td>
<td>D0000</td>
<td>VERIFICATE</td>
<td>PASS</td>
<td>foo</td>
</tr>
<tr>
<td>NAND2X1</td>
<td>typical</td>
<td>D0001</td>
<td>VERIFICATE</td>
<td>PASS</td>
<td>foo</td>
</tr>
<tr>
<td>NAND2X1</td>
<td>typical</td>
<td>D0002</td>
<td>VERIFICATE</td>
<td>PASS</td>
<td>foo</td>
</tr>
<tr>
<td>NAND2X1</td>
<td>typical</td>
<td>D0003</td>
<td>VERIFICATE</td>
<td>PASS</td>
<td>foo</td>
</tr>
<tr>
<td>NAND2X1</td>
<td>typical</td>
<td>D0004</td>
<td>VERIFICATE</td>
<td>PASS</td>
<td>foo</td>
</tr>
<tr>
<td>NAND2X1</td>
<td>typical</td>
<td>D0005</td>
<td>VERIFICATE</td>
<td>PASS</td>
<td>foo</td>
</tr>
<tr>
<td>NAND2X1</td>
<td>typical</td>
<td>D0006</td>
<td>VERIFICATE</td>
<td>PASS</td>
<td>foo</td>
</tr>
<tr>
<td>NAND2X1</td>
<td>typical</td>
<td>D0007</td>
<td>VERIFICATE</td>
<td>PASS</td>
<td>foo</td>
</tr>
<tr>
<td>NOR2X1</td>
<td>typical</td>
<td>D0000</td>
<td>VERIFICATE</td>
<td>PASS</td>
<td>foo</td>
</tr>
<tr>
<td>NOR2X1</td>
<td>typical</td>
<td>D0001</td>
<td>VERIFICATE</td>
<td>PASS</td>
<td>foo</td>
</tr>
<tr>
<td>NOR2X1</td>
<td>typical</td>
<td>D0002</td>
<td>VERIFICATE</td>
<td>PASS</td>
<td>foo</td>
</tr>
</tbody>
</table>

Total Simulation : 20
Total Passed : 20 (100.00%)
Total Failed : 0 (0.00%)

Lots of text missing from these highlights…
Results from ELC step3

cad-elc -S step3

Results from cad-alf2lib

cad-alf2lib foo

Lots of text missing from these highlights…

Total : 5 cells ( successful : 5  failed : 0 )
Changing Names

* The ELC scripts make a library named “foo”
* Probably good to rename it Lib6710_00
  - Rename foo.lib to Lib6710_00.lib
    - You have to modify the library name inside the .lib file
  - Rename foo.v to Lib6710_00.v
  - You generate Lib6710_00.db from Lib6710_00.lib
Changing foo.lib to Lib6710_00.lib

Lib6710_00.lib
Lib6710_00.lib

Converting .lib to .db

[elb@lab2-12 ELC]$ syn-dc
Using setup-synopsys from S13/F13
Assuming your OS is amd64
You are now set up to run the synopsys tools.
Working directory is /home/elb/VLSI/cadence-f13/ELC

Design Compiler Graphical
DC Ultra (TM)
DFTMAX (TM)
Power Compiler (TM)
DesignWare (R)
DC Expert (TM)
Design Vision (TM)
HDL Compiler (TM)
VHDL Compiler (TM)
DFT Compiler
Library Compiler (TM)
Design Compiler(R)

Version G-2012.06-SP3 for RHEL64 -- Oct 23, 2012
Copyright (c) 1988-2012 Synopsys, Inc.

This software and the associated documentation are confidential and
proprietary to Synopsys, Inc. Your use or disclosure of this software
is subject to the terms and conditions of a written license agreement
between you, or your company, and Synopsys, Inc.
Converting .lib to .db

```bash
dc_shell> read_lib Lib6710_00.lib
Reading '/home/elb/VLSI/cadence-f13/ELC/Lib6710_00.lib' ...
Warning: Line 81, Cell 'INVX1', pin 'A', The pin 'A' does not have a internal_power group. (LBDB-607)
Information: Line 571, Cell 'TIEHI', No internal_power information for the 'TIEHI' cell. (LBDB-301)
Warning: Line 578, Cell 'TIEHI', pin 'Y', The pin 'Y' does not have a internal_power group. (LBDB-607)
Information: Line 589, Cell 'TIELO', No internal_power information for the 'TIELO' cell. (LBDB-301)
Warning: Line 596, Cell 'TIELO', pin 'Y', The pin 'Y' does not have a internal_power group. (LBDB-607)
Warning: Line 11, The 'default_fanout_load' attribute is not specified. Using 1.00. (LBDB-172)
Warning: Line 11, The 'default_inout_pin_cap' attribute is not specified. Using 1.00. (LBDB-172)
Warning: Line 11, The 'default_input_pin_cap' attribute is not specified. Using 1.00. (LBDB-172)
Warning: Line 11, The 'default_output_pin_cap' attribute is not specified. Using 0.00. (LBDB-172)
Warning: Line 11, The 'default_cell_leakage_power' attribute is not specified. Using 0.00. (LBDB-172)
Warning: Line 11, The 'default_leakage_power_density' attribute is not specified. Using 0.00. (LBDB-172)
Technology library 'Lib6710_00' read successfully.
```

Lots of text missing from these highlights...

Converting .lib to .db

```bash
dc_shell> write_lib Lib6710_00 -o Lib6710_00.db
Wrote the 'Lib6710_00' library to '/home/elb/VLSI/cadence-f13/ELC/Lib6710_00.db' successfully.
```

Lots of text missing from these highlights...
Chapter 10 – Abstract Generation
- Abstract views are used by place and route
- Eventually they are captured in a .lef file that describes the place and route geometry
Generating Abstract Views

<table>
<thead>
<tr>
<th>Bin</th>
<th>Cells</th>
<th>Cell</th>
<th>Layout</th>
<th>Logical</th>
<th>Phys</th>
<th>Extract</th>
<th>Abstract</th>
<th>Verify</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

Imported library in cad-abstract

<table>
<thead>
<tr>
<th>Bin</th>
<th>Cells</th>
<th>Cell</th>
<th>Layout</th>
<th>Logical</th>
<th>Phys</th>
<th>Extract</th>
<th>Abstract</th>
<th>Verify</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
<td>✔</td>
</tr>
</tbody>
</table>

[Image of CAD abstract view interface]
Green checks are good!

Export Lib6710_00.lef file
```
"geometry lef"

Replace highlighted text with TechHeader.lef from class ELC directory
```
Final CAD5 Files…

- **Nine views of every cell**
  - abstract, abstract.ext, abstract.pin, analog_extracted, behavioral, cmos_sch, extracted, layout, symbol
  - DRC and LVS-checked, and simulated
- **Four versions of the library description**
  - Lib6710_00.lib // timing information
  - Lib6710_00.db // design compiler target lib
  - Lib6710_00.v // interface descriptions
  - Lib6710_00.lef // place and route info
Test with beh2str

- beh2str addsub.v addsub_dc.v Lib6710_00.db

- Results in addsub_dc.v and addsub_dc.v.rep

- NOTE! Your 5-cell library has no DFF, so you can’t synthesize any sequential circuits!
  - Combinational circuits only at this point...

addsub_dc.v

module addsub ( a, b, addnsub, result );
input [7:0] a;
input [7:0] b;
output [8:0] result;
input addnsub;
wire   n26, n27, n28, n29, n30, n31, n32, n33, n34, n35, n36, n37, n38, n39,
n40, n41, n42, n43, n44, n45, n46, n47, n48, n49, n50, n51, n52, n53,
n54, n55, n56, n57, n58, n59, n60, n61, n62, n63, n64, n65, n66, n67,
...
NAND2X1 U33 ( .A(n26), .B(n27), .Y(result[8]) );
NAND2X1 U34 ( .A(b[7]), .B(n28), .Y(n27) );
NAND2X1 U35 ( .A(n29), .B(n30), .Y(n28) );
NOR2X1 U36 ( .A(n31), .B(n32), .Y(n29) );
...
NOR2X1 U216 ( .A(n188), .B(a[0]), .Y(n175) );
NAND2X1 U217 ( .A(a[0]), .B(n188), .Y(n202) );
INVX1 U218 ( .A(b[0]), .Y(n188) );
endmodule
Operating Conditions: typical   Library: Lib6710_00
Wire Load Model Mode: top
Startpoint: b[1] (input port)
Endpoint: result[8] (output port)
Path Group: (none)
Path Type: max

<table>
<thead>
<tr>
<th>Point</th>
<th>Incr</th>
<th>Path</th>
</tr>
</thead>
<tbody>
<tr>
<td>input external delay</td>
<td>0.00</td>
<td>0.00 r</td>
</tr>
<tr>
<td>b[1] (in)</td>
<td>0.00</td>
<td>0.00 r</td>
</tr>
<tr>
<td>U198/Y (NOR2X1)</td>
<td>0.50</td>
<td>0.50 f</td>
</tr>
<tr>
<td>U194/Y (NOR2X1)</td>
<td>0.36</td>
<td>0.86 r</td>
</tr>
<tr>
<td>...</td>
<td></td>
<td></td>
</tr>
<tr>
<td>U34/Y (NAND2X1)</td>
<td>0.25</td>
<td>10.16 f</td>
</tr>
<tr>
<td>U33/Y (NAND2X1)</td>
<td>0.24</td>
<td>10.40 r</td>
</tr>
<tr>
<td>result[8] (out)</td>
<td>0.00</td>
<td>10.40 r</td>
</tr>
<tr>
<td>data arrival time</td>
<td></td>
<td>10.40</td>
</tr>
</tbody>
</table>

(Path is unconstrained)
Summary

- You now have a library that is fully functional
  - BUT – only on combinational circuits
  - No DFF yet!

- Every step of the way requires extreme care to get things exactly right
  - No trick to finding the right answer
  - The point is to practice working with the data & tools

Look ahead to EDI
Modify the Script Files

```c
# configuration file should be named <BASENAME>.globals.
set BASENAME "addsub"

# The following variables are used in fpplan.tcl...
# These set the percent utilization target (how dense should
# the cells be placed), and the gap for routing between rows.
# These are good starting values for small macros. Larger or
# more complex macros will likely need a lowered usept or
# larger rowgap or both.
# Note that rowgap and coregap should be divisible by the basic
# grid unit of 0.3 that our AMECIF/F process uses.
set usept 0.70 ; # percent utilization in placing cells
set rowgap 15 ; # gap (microns) between pairs of std cell rows
set coregap 30.0 ; # gap (microns) between the core and the power rails
set aspect 0.68 ; # aspect ratio of overall cell 1.0 is square,
# 1 is landscape, -1 is portrait

# The following variables are used in pplan.tcl...
# These numbers control the power and ground grid.
# Note that all these numbers should be divisible by 0.3 so
# that they fit on the lambda grid
set pwidth 9.9 ; # power rail width
set pspace 1.8 ; # power rail space
set width 4.8 ; # power stripe width
set space 12.5 ; # power stripe spacing
set offset 12.5 ; # power stripe offset to first stripe
set power_net vdd ! # the name of the power net
ground_ground ; # the name of the ground net
```

Modify the Script Files
Modify the Script Files

Placed and Routed Result