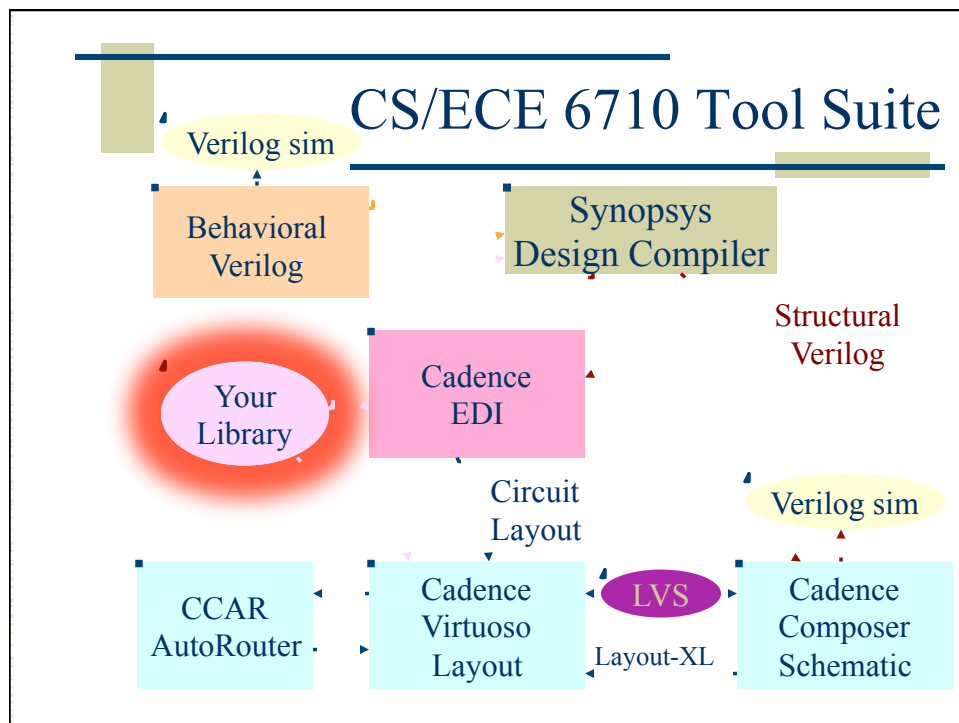


CAD5

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

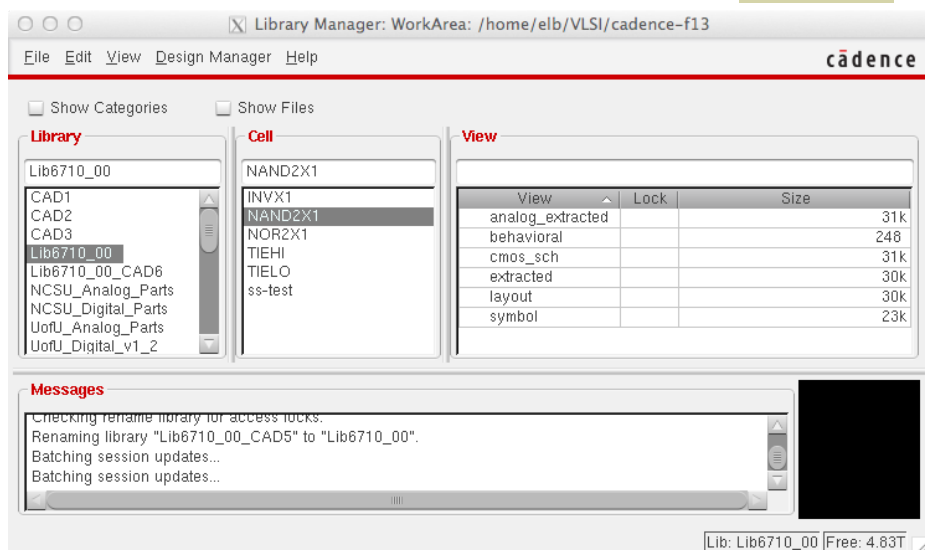


In the CAD Book

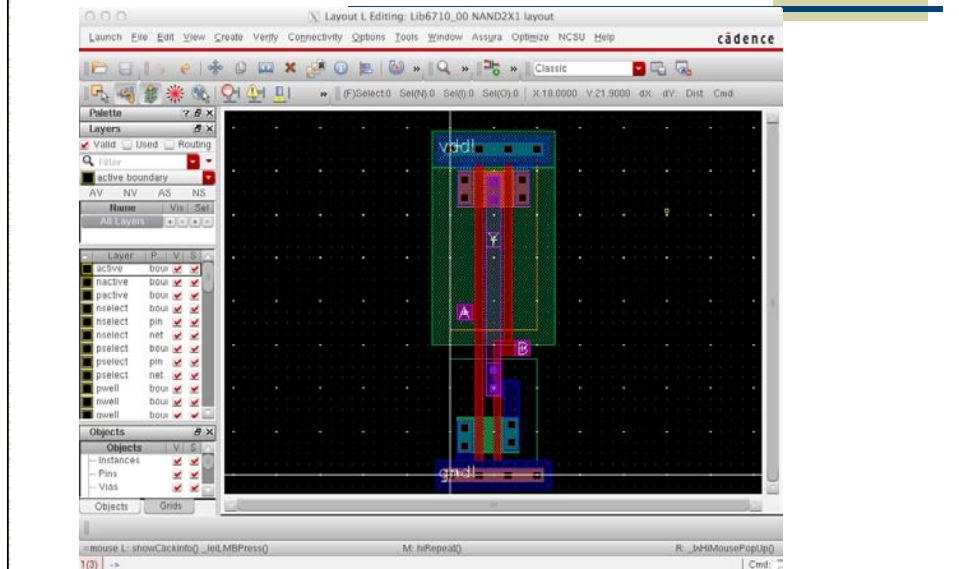
♦ Chapter 8 on Cell Characterization

- 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...)

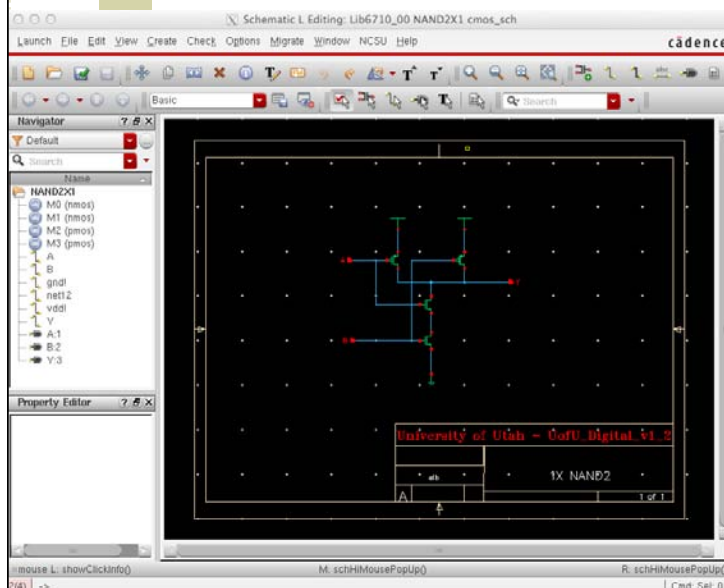
Start with Cells



Layout View



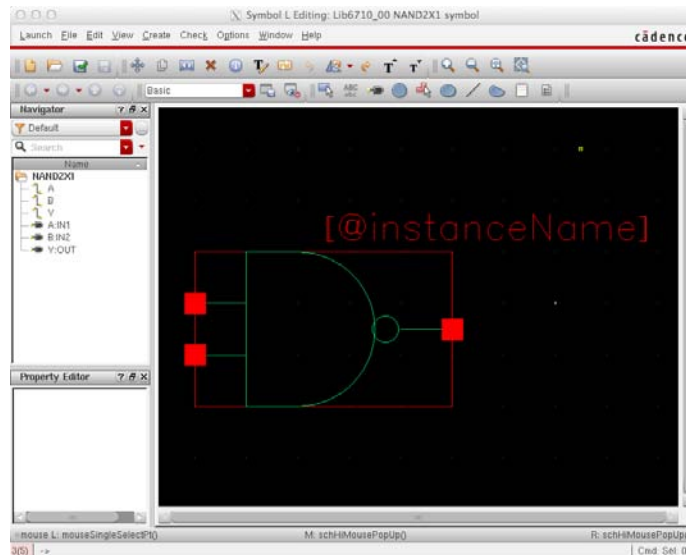
Schematic View



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

```
emacs@lab2-12.eng.utah.edu
File Edit Options Buffers Tools Statements Verilog Help

Verilog HDL for "UofU_Digital_v1_2", "NAND2X1" "behavioral"

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

verilog.v All L1 (Verilog)
```

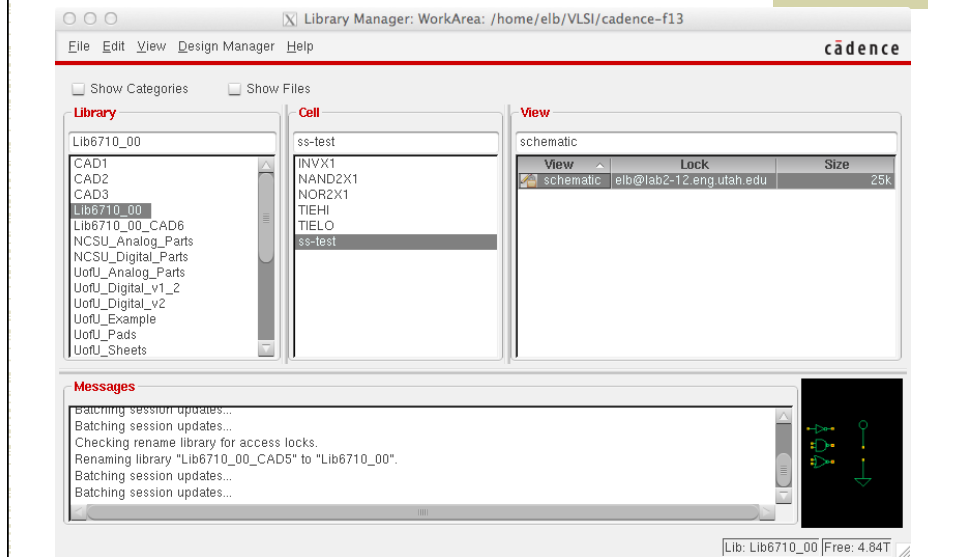
Start with six views of each cell

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

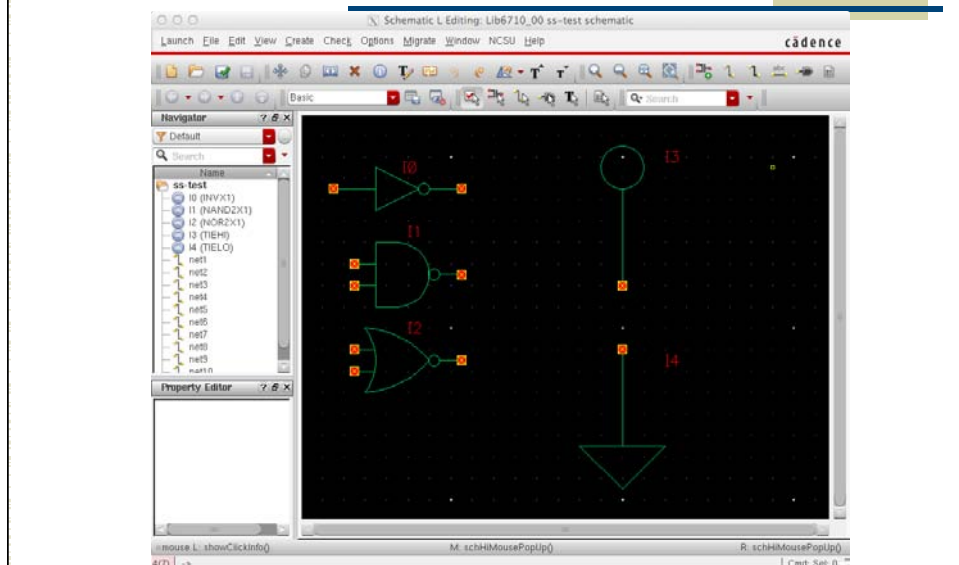
Characterize the cells

- ♦ 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-enc from the bin directory
 - Four steps – step1, step2, step3, cad-elf2lib

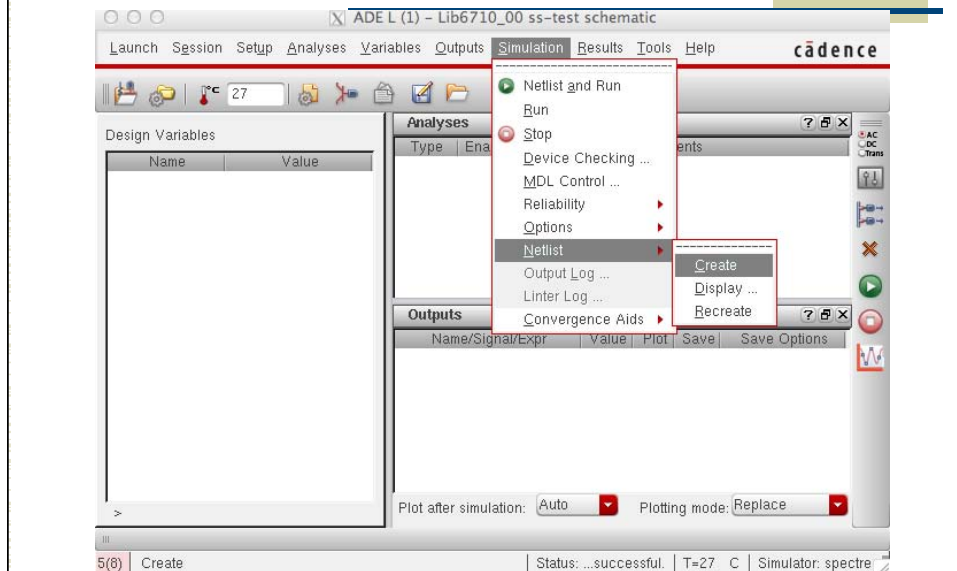
Single Schematic with All Cells



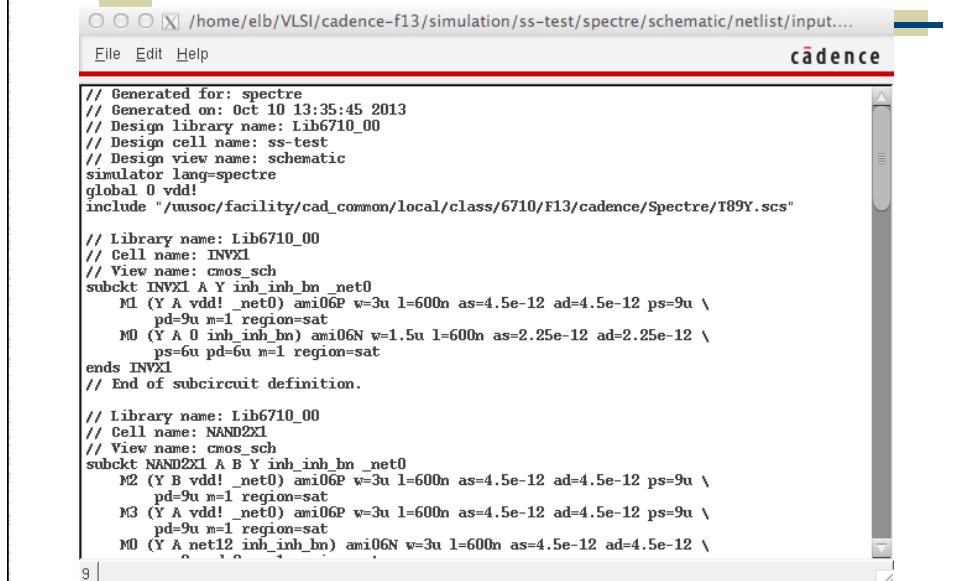
Single Schematic with All Cells



Create Netlist of that ss-test Cell



Call it foo.scs (for example)



Convert to ELC format

```

simulator lang=spectre

subckt INVX1 A Y inh_inh_bn_net0 vdd gnd
M1 (Y A vdd_net0) ami06P w=3u l=600n as=4.5e-12 ad=4.5e-12 ps=9u \
pd=9u m=1 region=sat
M0 (Y A gnd inh_inh_bn) ami06N w=1.5u l=600n as=2.25e-12 ad=2.25e-12 \
ps=6u pd=6u m=1 region=sat
ends INVX1
subckt NAND2X1 A B Y inh_inh_bn_net0 vdd gnd
M2 (Y B vdd_net0) ami06P w=3u l=600n as=4.5e-12 ad=4.5e-12 ps=9u \
pd=9u m=1 region=sat
M3 (Y A vdd_net0) ami06P w=3u l=600n as=4.5e-12 ad=4.5e-12 ps=9u \
pd=9u m=1 region=sat
M0 (Y A net12 inh_inh_bn) ami06N w=3u l=600n as=4.5e-12 ad=4.5e-12 \
ps=9u pd=9u m=1 region=sat
M1 (net12 B gnd inh_inh_bn) ami06N w=3u l=600n as=4.5e-12 ad=4.5e-12 \
ps=9u pd=9u m=1 region=sat
ends NAND2X1
subckt NOR2X1 A B Y inh_inh_bn_net0 vdd gnd

```

--- dut.scs Top L1 (Fundamental) ---

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...

cad-enc -S step1

Results from ELC step1

```
...
enc> db_gate

=====
DESIGN : INVX1
=====
...
NOT ( Y, A );
...

=====
DESIGN : NAND2X1
=====
...
NAND2 ( Y, A, B );
...

=====
DESIGN : NORX1
=====
...
NOR ( Y, A, B );
...
```

```
...
=====
DESIGN : TIEHI
=====
...
- node:NET5 is shorted to node:GND [nmos]
- clamp1 ( Y ) is found..
...

=====
DESIGN : TIELO
=====
...
- node:NET5 is shorted to node:VDD [pmos]
- clamp0 ( Y ) is found..
...
```

Lots of text missing from these highlights...

cad-enc -S step1

Results from ELC step1

```
Terminal - ssh - 97x31
Writing : foo.ipdb/INVX1.design/body/type
Reading : foo.ipdb/NAND2X1.design

=====
DESIGN : NAND2X1
=====
-----
A B : V
-----
D0000: R 1 : F : DELAY(A)
D0001: R 0 : 1 : POWER(A)
D0002: F 1 : A : DELAY(A)
D0003: F 0 : 1 : POWER(A)
D0004: 1 R : F : DELAY(B)
D0005: 0 R : 1 : POWER(B)
D0006: 1 F : A : DELAY(B)
D0007: 0 F : 1 : POWER(B)
-----
=> 8 vectors generated
Writing : foo.ipdb/NAND2X1.design/simulate/spec
Writing : foo.ipdb/NAND2X1.design/simulate/subckt
Writing : foo.ipdb/NAND2X1.design/boundary/port
Writing : foo.ipdb/NAND2X1.design/body/type
Reading : foo.ipdb/NOR2X1.design

=====
DESIGN : NOR2X1
=====
-----
A B : V
-----
```

Results from ELC step1

```

Terminal — ssh — 97x31

=====
DESIGN : NAND2X1
=====
DESIGN ( NAND2X1 );
// =====
// PORT DEFINITION
// =====
INPUT A ( A );
INPUT B ( B );
OUTPUT Y ( Y );
SUPPLY0 GND ( GND );
SUPPLY0 INH_INH_BN ( INH_INH_BN );
SUPPLY1 VDD ( VDD );
SUPPLY1 _NET0 ( _NET0 );
// =====
// INSTANCES
// =====
NAND ( Y, A, B );
END_OF_DESIGN;

=====
DESIGN : NOR2X1
=====
DESIGN ( NOR2X1 );
// =====
// PORT DEFINITION
// =====
INPUT A ( A );

```

Results from ELC step2

```
eltc> 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	2	2	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	8	8	foo
NOR2X1	typical	D0000	SIMULATE	foo
NOR2X1	typical	D0001	SIMULATE	foo
NOR2X1	typical	D0002	SIMULATE	foo

Lots of text missing from these highlights...

cad-elc -S step2

Results from ELC step2

DESIGN	PROCESS	#ID	STAGE	STATUS	IPDB
IN VX1	typical	D0000	VERIFICATE	PASS	foo
IN VX1	typical	D0001	VERIFICATE	PASS	foo
NAND2X1	typical	D0000	VERIFICATE	PASS	foo
NAND2X1	typical	D0001	VERIFICATE	PASS	foo
NAND2X1	typical	D0002	VERIFICATE	PASS	foo
NAND2X1	typical	D0003	VERIFICATE	PASS	foo
NAND2X1	typical	D0004	VERIFICATE	PASS	foo
NAND2X1	typical	D0005	VERIFICATE	PASS	foo
NAND2X1	typical	D0006	VERIFICATE	PASS	foo
NAND2X1	typical	D0007	VERIFICATE	PASS	foo
NOR2X1	typical	D0000	VERIFICATE	PASS	foo
NOR2X1	typical	D0001	VERIFICATE	PASS	foo
NOR2X1	typical	D0002	VERIFICATE	PASS	foo

Lots of text missing from these highlights...

cad-elc -S step2

Results from ELC step2

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

Lots of text missing from these highlights...

Results from ELC step3

```
eltc> db_output -report foo.alf.rep -alf foo.alf -p typical

INVX1          typical    2013-10-10 13:55:23 (2013-10-10 19:55:23 GMT) 2 (100%)
NAND2X1        typical    2013-10-10 13:55:23 (2013-10-10 19:55:23 GMT) 8 (100%)
NOR2X1        typical    2013-10-10 13:55:23 (2013-10-10 19:55:23 GMT) 8 (100%)
TIEHI         typical    2013-10-10 13:55:21 (2013-10-10 19:55:21 GMT) 1 (100%)
TIELO         typical    2013-10-10 13:55:22 (2013-10-10 19:55:22 GMT) 1 (100%)

eltc> db_verilog -r foo.v
Reading : foo.ipdb/INVX1.design

=====
DESIGN : INVX1
=====
Reading : foo.ipdb/NAND2X1.design
=====
DESIGN : NAND2X1
=====

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

Results from cad-alf2lib

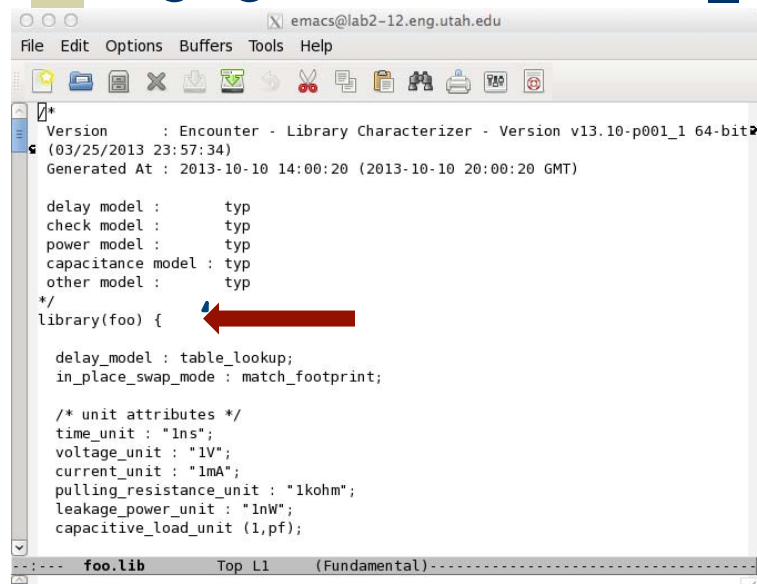
```
CELL INVX1:
  now reading
  now converting
  ***** successful *****
CELL NAND2X1:
  now reading
  now converting
  ***** successful *****
...
CELL TIEHI:
  now reading
  now converting
  ***** successful *****
CELL TIELO:
  now reading
  now converting
  ***** successful *****
---
Total : 5 cells ( successful : 5 failed : 0 )

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

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



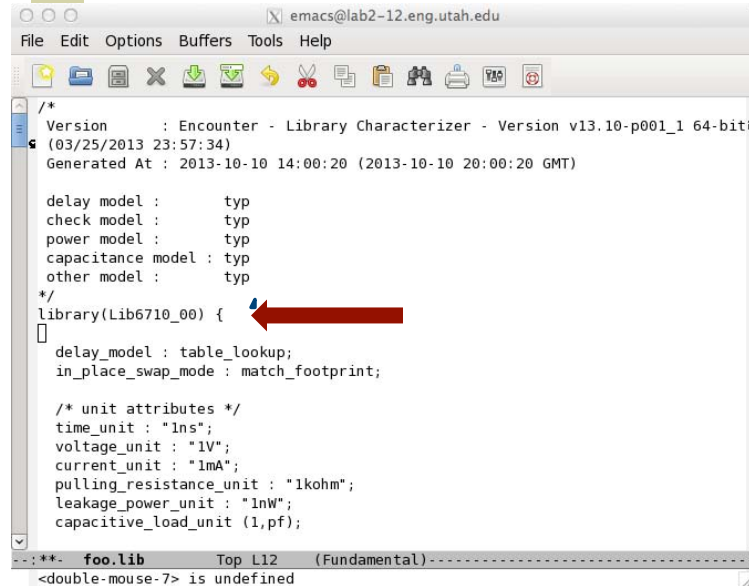
```
emacslab2-12.eng.utah.edu
File Edit Options Buffers Tools Help

Version : Encounter - Library Characterizer - Version v13.10-p001_1 64-bit
(03/25/2013 23:57:34)
Generated At : 2013-10-10 14:00:20 (2013-10-10 20:00:20 GMT)

delay model : typ
check model : typ
power model : typ
capacitance model : typ
other model : typ
*/
library(foo) {
    delay_model : table_lookup;
    in_place_swap_mode : match_footprint;

    /* unit attributes */
    time_unit : "1ns";
    voltage_unit : "1V";
    current_unit : "1mA";
    pulling_resistance_unit : "1kohm";
    leakage_power_unit : "1nW";
    capacitive_load_unit (1,pf);
}
```

Changing foo.lib to Lib6710_00.lib



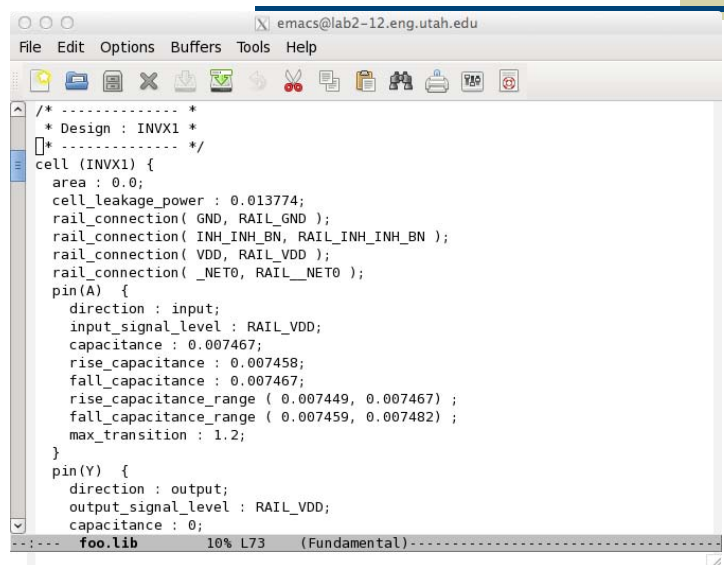
```
/*
Version      : Encounter - Library Characterizer - Version v13.10-p001_1 64-bit
(03/25/2013 23:57:34)
Generated At : 2013-10-10 14:00:20 (2013-10-10 20:00:20 GMT)

delay model :      typ
check model  :      typ
power model  :      typ
capacitance model : typ
other model  :      typ
*/
library(Lib6710_00) {
    delay_model : table_lookup;
    in_place_swap_mode : match_footprint;

    /* unit attributes */
    time_unit : "1ns";
    voltage_unit : "1V";
    current_unit : "1mA";
    pulling_resistance_unit : "1kohm";
    leakage_power_unit : "1nW";
    capacitive_load_unit (1,pf);
}

--:**- foo.lib Top L12 (Fundamental)-----
<double-mouse-7> is undefined
```

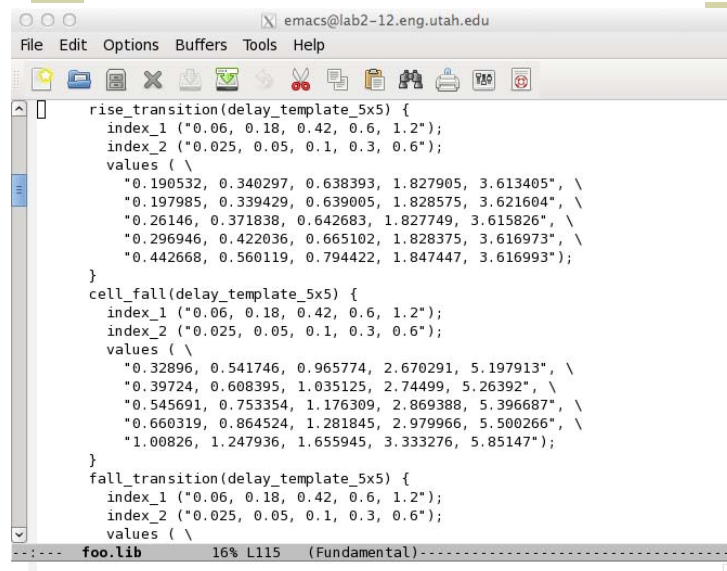
Lib6710_00.lib



```
/* ----- *
 * Design : INVX1 *
 * ----- */
cell (INVX1) {
    area : 0.0;
    cell_leakage_power : 0.013774;
    rail_connection( GND, RAIL_GND );
    rail_connection( INH_INH_BN, RAIL_INH_INH_BN );
    rail_connection( VDD, RAIL_VDD );
    rail_connection( _NET0, RAIL__NET0 );
    pin(A) {
        direction : input;
        input_signal_level : RAIL_VDD;
        capacitance : 0.007467;
        rise_capacitance : 0.007458;
        fall_capacitance : 0.007467;
        rise_capacitance_range ( 0.007449, 0.007467 );
        fall_capacitance_range ( 0.007459, 0.007482 );
        max_transition : 1.2;
    }
    pin(Y) {
        direction : output;
        output_signal_level : RAIL_VDD;
        capacitance : 0;
    }
}

--:**- foo.lib 10% L73 (Fundamental)-----
```

Lib6710_00.lib



```
rise_transition(delay_template_5x5) {
  index_1 ("0.06, 0.18, 0.42, 0.6, 1.2");
  index_2 ("0.025, 0.05, 0.1, 0.3, 0.6");
  values ( \
    "0.190532, 0.340297, 0.638393, 1.827905, 3.613405", \
    "0.197985, 0.339429, 0.639005, 1.828575, 3.621604", \
    "0.26146, 0.371838, 0.642683, 1.827749, 3.615826", \
    "0.296946, 0.422036, 0.665102, 1.828375, 3.616973", \
    "0.442668, 0.560119, 0.794422, 1.847447, 3.616993");
}
cell_fall(delay_template_5x5) {
  index_1 ("0.06, 0.18, 0.42, 0.6, 1.2");
  index_2 ("0.025, 0.05, 0.1, 0.3, 0.6");
  values ( \
    "0.32896, 0.541746, 0.965774, 2.670291, 5.197913", \
    "0.39724, 0.608395, 1.035125, 2.74499, 5.26392", \
    "0.545691, 0.753354, 1.176309, 2.869388, 5.396687", \
    "0.660319, 0.864524, 1.281845, 2.979966, 5.500266", \
    "1.00826, 1.247936, 1.655945, 3.333276, 5.85147");
}
fall_transition(delay_template_5x5) {
  index_1 ("0.06, 0.18, 0.42, 0.6, 1.2");
  index_2 ("0.025, 0.05, 0.1, 0.3, 0.6");
  values ( \
```

syn-dc

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

```
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
1
```

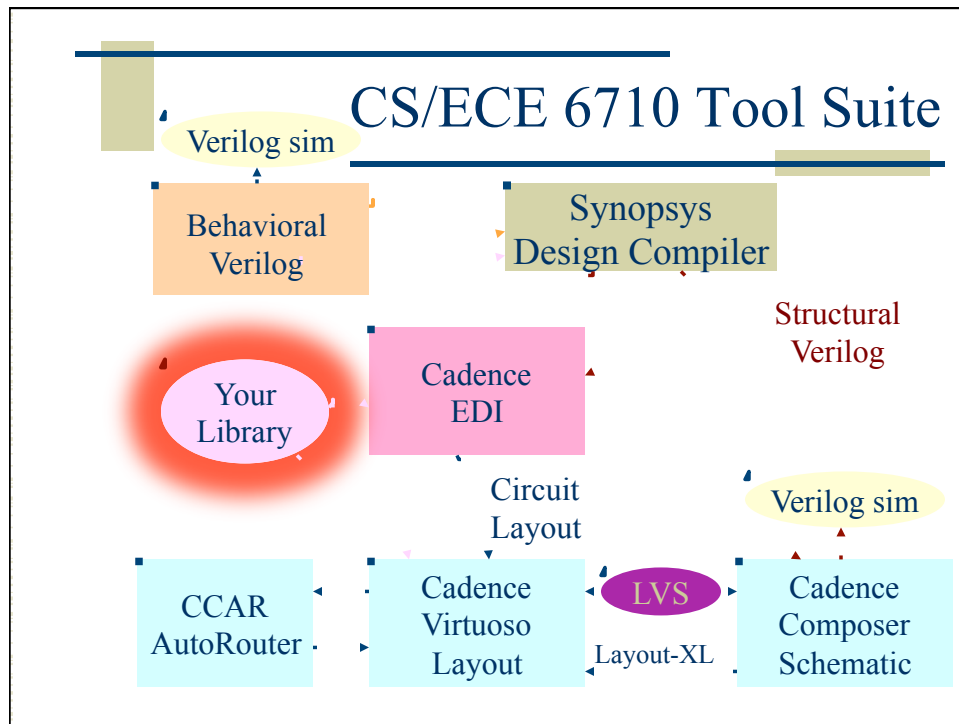
Lots of text missing from these highlights...

Converting .lib to .db

```
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.
1
```

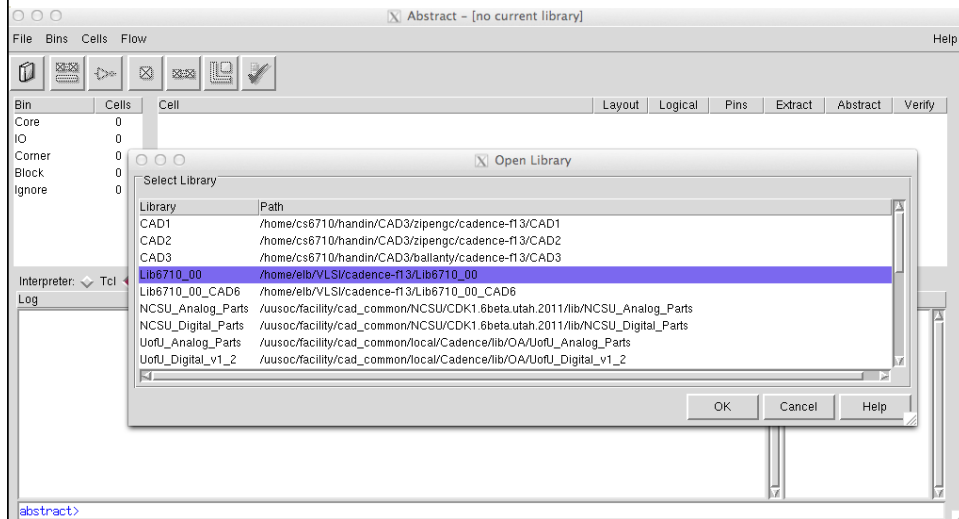
Lots of text missing from these highlights...



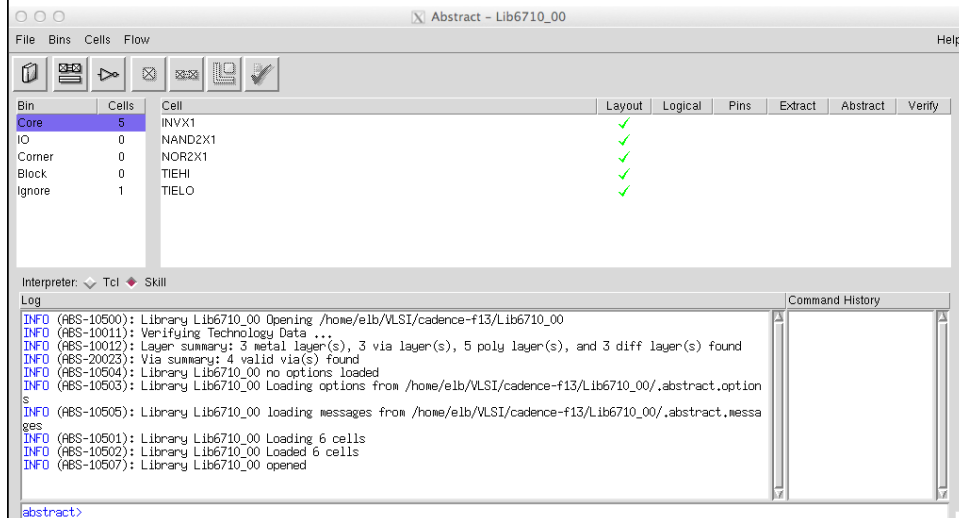
In the CAD Book

- ♦ 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

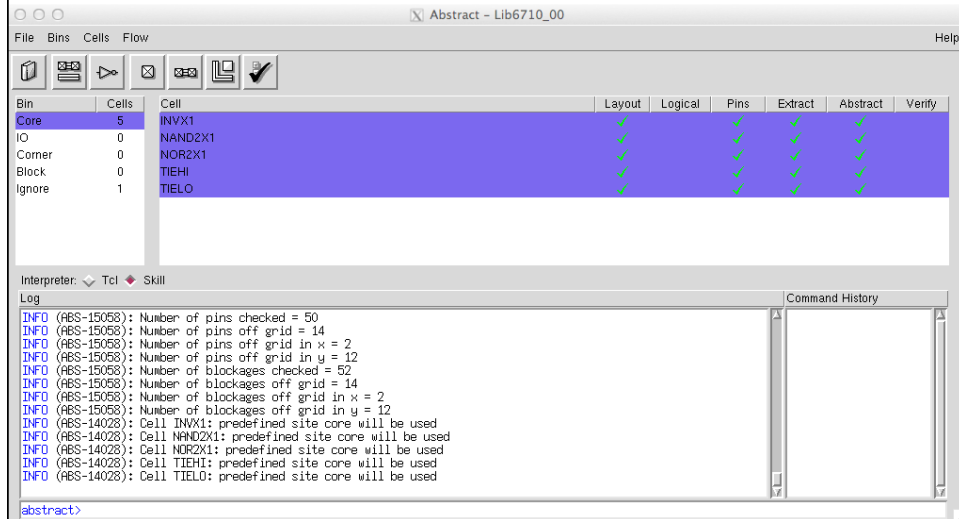
Generating Abstract Views



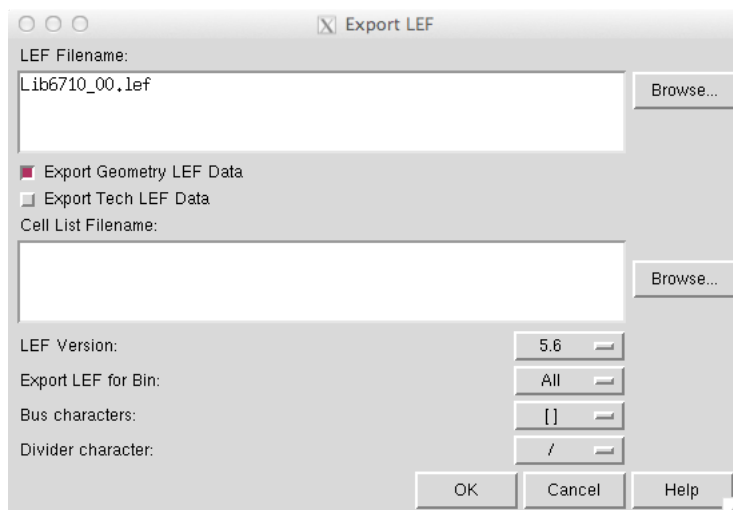
Imported library in cad-abstract



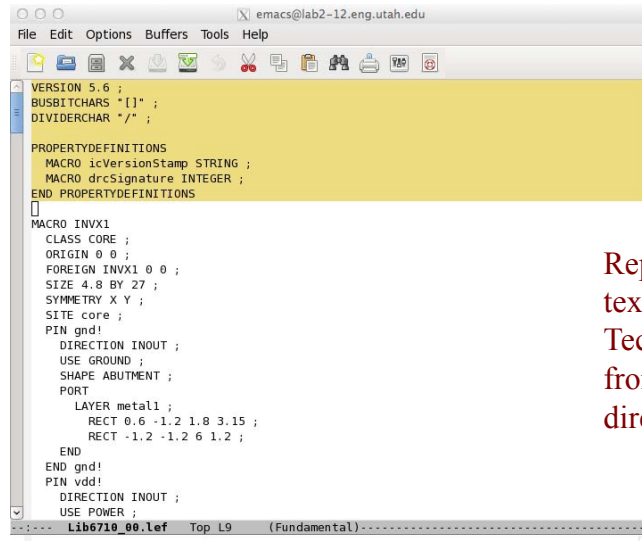
Green checks are good!



Export Lib6710_00.lef file



“geometry lef”



```
emacs@lab2-12.eng.utah.edu
File Edit Options Buffers Tools Help

VERSION 5.6 ;
BUSBITCHARS "[]" ;
DIVIDERCHAR "/" ;

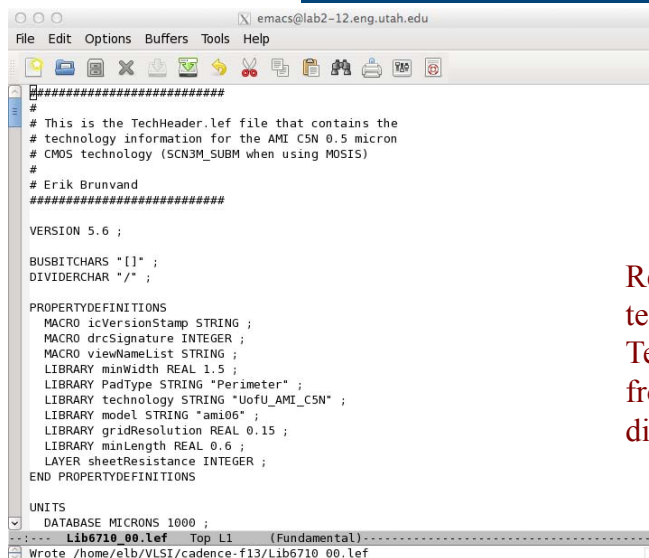
PROPERTYDEFINITIONS
  MACRO icVersionStamp STRING ;
  MACRO drcSignature INTEGER ;
END PROPERTYDEFINITIONS

MACRO INVX1
  CLASS CORE ;
  ORIGIN 0 0 ;
  FOREIGN INVX1 0 0 ;
  SIZE 4.8 BY 27 ;
  SYMMETRY X Y ;
  SITE core ;
  PIN gnd!
  DIRECTION INOUT ;
  USE GROUND ;
  SHAPE ABUTMENT ;
  PORT
    LAYER metal1 ;
    RECT 0.6 -1.2 1.8 3.15 ;
    RECT -1.2 -1.2 6 1.2 ;
  END
END gnd!
PIN vdd!
DIRECTION INOUT ;
USE POWER ;

Lib6710_00.lef Top L9 (Fundamental)
```

Replace highlighted text with TechHeader.lef from class ELC directory

“geometry lef”



```
emacs@lab2-12.eng.utah.edu
File Edit Options Buffers Tools Help

#####
#
# This is the TechHeader.lef file that contains the
# technology information for the AMI C5N 0.5 micron
# CMOS technology (SCN3M_SUBM when using MOSIS)
#
# Erik Brunvand
#####
VERSION 5.6 ;
BUSBITCHARS "[]" ;
DIVIDERCHAR "/" ;

PROPERTYDEFINITIONS
  MACRO icVersionStamp STRING ;
  MACRO drcSignature INTEGER ;
  MACRO viewNameList STRING ;
  LIBRARY minWidth REAL 1.5 ;
  LIBRARY PadType STRING "Perimeter" ;
  LIBRARY technology STRING "UofU_AMI_C5N" ;
  LIBRARY model STRING "ami06" ;
  LIBRARY gridResolution REAL 0.15 ;
  LIBRARY minLength REAL 0.6 ;
  LAYER sheetResistance INTEGER ;
END PROPERTYDEFINITIONS

UNITS
  DATABASE MICRONS 1000 ;

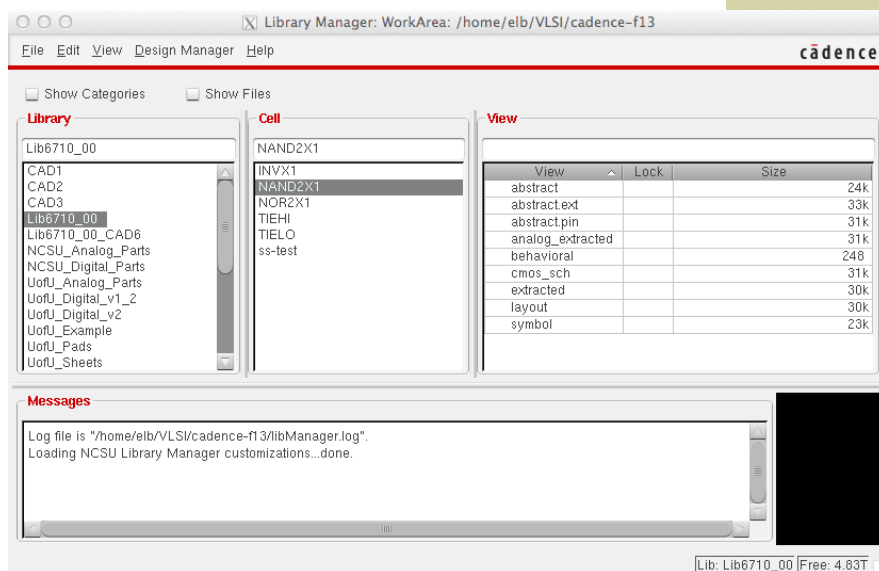
Lib6710_00.lef Top L1 (Fundamental)
Wrote /home/elb/VLSI/cadence-f13/Lib6710_00.lef
```

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

All Nine Views...



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
```

addsub_dc.v.rep

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

Point	Incr	Path

input external delay	0.00	0.00 r
b[1] (in)	0.00	0.00 r
U198/Y (NOR2X1)	0.50	0.50 f
U194/Y (NOR2X1)	0.36	0.86 r
...		
U34/Y (NAND2X1)	0.25	10.16 f
U33/Y (NAND2X1)	0.24	10.40 r
result[8] (out)	0.00	10.40 r
data arrival time		10.40

(Path is unconstrained)		

addsub_dc.v.rep

Library(s) Used:

Lib6710_00 (File: /home/elb/VLSI/cadence-f13/syn-f13/CAD5test/
Lib6710_00.db)

Number of ports:	26
Number of nets:	203
Number of cells:	186
Number of combinational cells:	186
Number of sequential cells:	0
Number of macros:	0
Number of buf/inv:	37
Number of references:	3

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

```
[elb@lab2-12 addsub]$ ll -t
total 132
-rw-r--r-- 1 elb cs-rsrch 1522 Oct 10 2013 addsub_globals
-rw-r--r-- 1 elb cs-rsrch 3814 Oct 10 2013 top.tcl
-rw-r--r-- 1 elb cs-rsrch 2618 Oct 10 2013 mmhc.tcl
-rw-r--r-- 1 elb cs-rsrch 280 Oct 10 16:39 Lib6710_00.v
-rw-r--r-- 1 elb cs-rsrch 22736 Oct 10 16:39 Lib6710_00.lib
-rw-r--r-- 1 elb cs-rsrch 31744 Oct 10 16:39 Lib6710_00.db
-rw-r--r-- 1 elb cs-rsrch 11919 Oct 10 16:39 Lib6710_00.lef
-rw-r--r-- 1 elb cs-rsrch 3413 Oct 10 16:39 addsub_struct.sdc
-rw-r--r-- 1 elb cs-rsrch 11742 Oct 10 16:39 addsub_struct.v
-rw-r--r-- 1 elb cs-rsrch 2016 Oct 10 16:37 verify.tcl
-rw-r--r-- 1 elb cs-rsrch 3005 Oct 10 16:37 route.tcl
-rw-r--r-- 1 elb cs-rsrch 2394 Oct 10 16:37 pplan.tcl
-rw-r--r-- 1 elb cs-rsrch 1940 Oct 10 16:37 place.tcl
-rw-r--r-- 1 elb cs-rsrch 971 Oct 10 16:37 fplan.tcl
-rw-r--r-- 1 elb cs-rsrch 1641 Oct 10 16:37 cts.tcl
[elb@lab2-12 addsub]$
```



```

emacs@lab2-12.eng.utah.edu
File Edit Options Buffers Tools Tcl Help

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

# The following variables are used in fplan.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 usepct or
# larger rowgap or both.
#
# Note that rowgap and coregap should be divisible by the basic
# grid unit of 0.3 that our AMICSN/F process uses.
#
set usepct 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 0.60 ;# aspect ratio of overall cell (1 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 swidth 4.8 ;# power stripe width
set ssize 123 ;# power stripe spacing
set soffset 120 ;# power stripe offset to first stripe
set pownet vdd! ;# the name of the power net
set groundnet gnd! ;# the name of the ground net

--:-- top.tcl 25% L39 (Tcl)

```

Modify the Script Files

```

emacs@lab2-12.eng.utah.edu
File Edit Options Buffers Tools Help

#####
# Encounter Input configuration file
# University of Utah - 6710
#
# Use this file to describe the input files for
# placement and routing.
#
# This file is a ".globals" file for EDI 11
# and above.
#
# Look for terms surrounded by !!...!!
# These are the things you will need to change
#####
#
# Set the name of your structural Verlog file
# This comes from Synopsys synthesis
set init_verilog {addsub_struct.v}
# Set the name of your top module
set init_design {addsub}
# Set the name of your .lef file
# This comes from ELC
set init_lef_file {Lib6710_00.lef}

#####
# below here you probably don't have to change anything
#####
# Set the name of your "multi-mode-multi-corner data file
# You don't need to change this unless you're using a
# different mmmc.tcl file.
set init_mmmc_file {mmmc.tcl}
# Some helpful input mode settings
set init_import_mode {-treatUndefinedCellsAsBbox 0 -keepEmptyModule 1}
# Set the names of your gnd and power nets
set init_gnd_net {gnd!}
set init_pwr_net {vdd!}

--:-- addsub_globals Top L31 (Fundamental)
Beginning of buffer

```

Modify the Script Files

