Digital VLSI Design

Lecture 4: Standard Cell Libraries

Semester A, 2016-17

Lecturer: Dr. Adam Teman

27 November 2016

Emerging Nanoscaled Integrated Circuits and Systems Labs

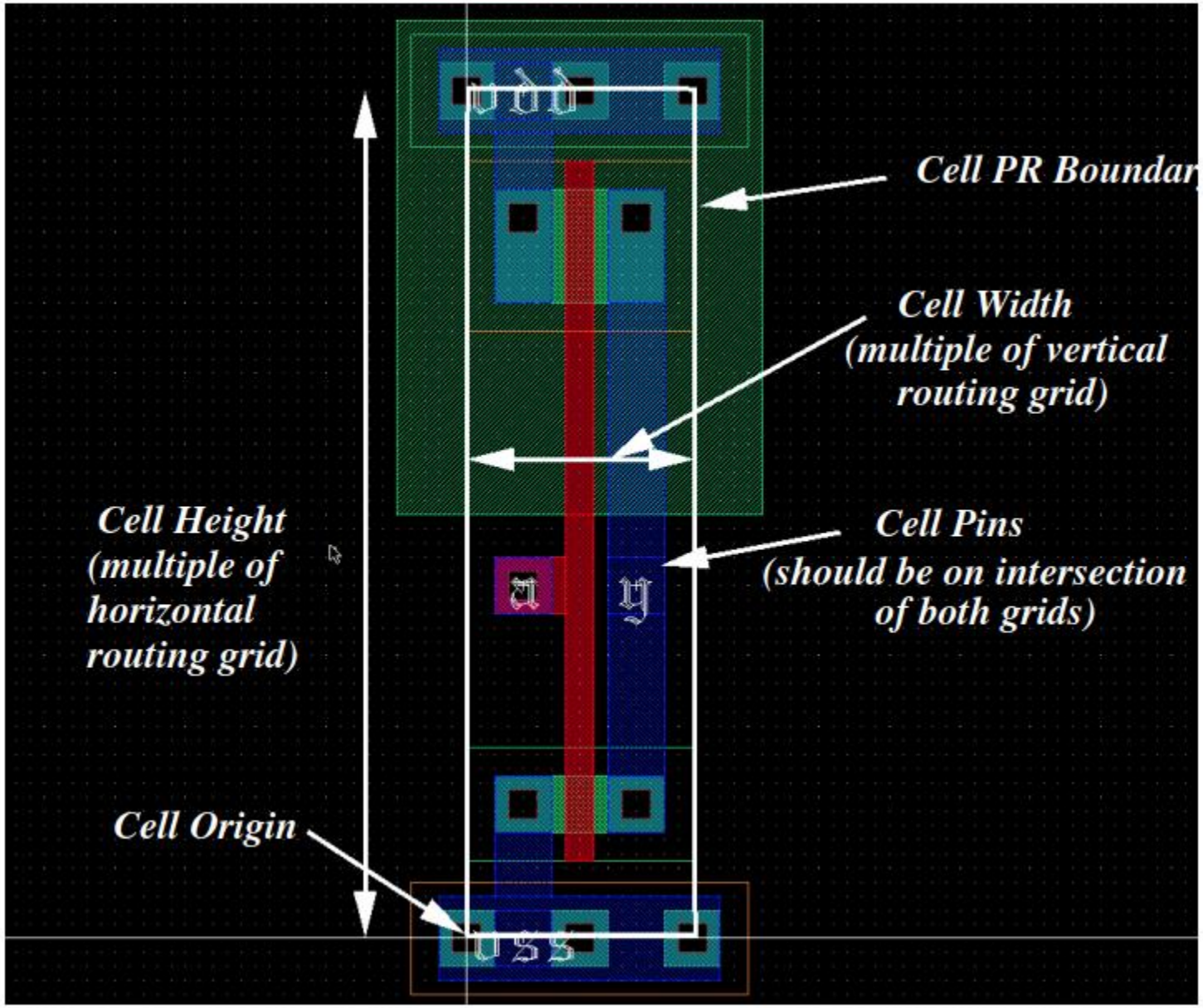
Bar-Ilan University
אוניברסיטת בר-אילן

Motivation

- A standard cell library is a collection of well defined and appropriately characterized logic gates that can be used to implement a digital design.
- Similar to LEGO, standard cells must meet predefined specifications to be flawlessly manipulated by synthesis, place, and route algorithms.
- Therefore, a standard cell library is delivered with a collection of files that provide all the information needed by the various EDA tools.

Example

- **Inverter standard cell layout**
- **Pay attention to:**
  - Cell height
  - Cell width
  - Voltage rails
  - Well definition
  - Pin Placement
  - Metal layers
  - PR Boundary



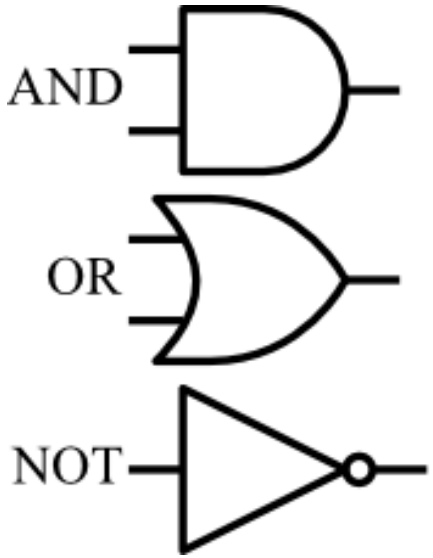
Ideally, Standard Cells should be routed entirely in M1 !

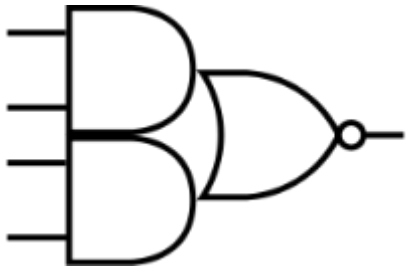
http://www.csee.umbc.edu/~tinoosh/cmpe641/

Standard Cell Library Contents

What cells are in a standard cell library?

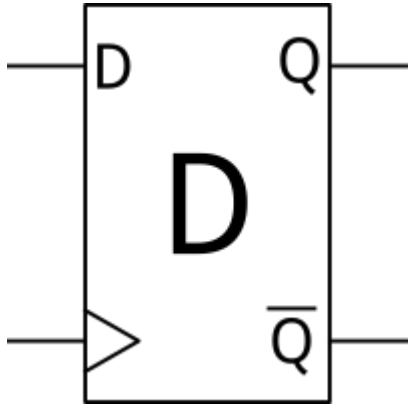
- **Combinational logic cells (NAND, NOR, INV, etc.):**
  - Variety of drive strengths for all cells.
  - Larger variety of buffers and inverters.
  - “Clock cells” with balanced rise and fall delays.
  - Complex cells (AOI, OAI, etc.)
  - Cells with Fan-In <= 4
  - Delay cells
  - Level Shifters
  - ECO Cells





AND-OR INVERT (AOI)

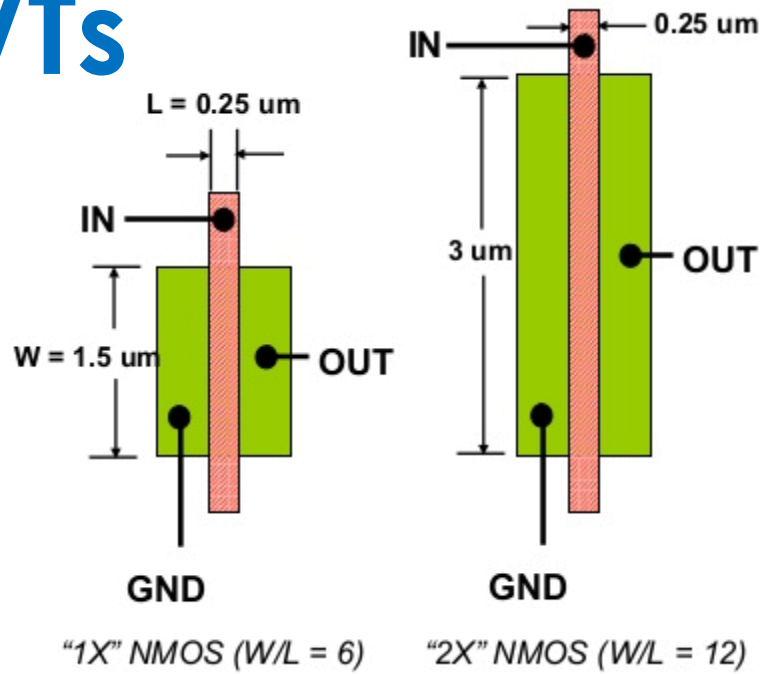
- **Sequential Cells:**
  - Many types of flip flops: pos/negedge, set/reset, Q/QB, enable
  - Latches
  - Integrated Clock Gating cells
  - Scan enabled cells for ATPG.



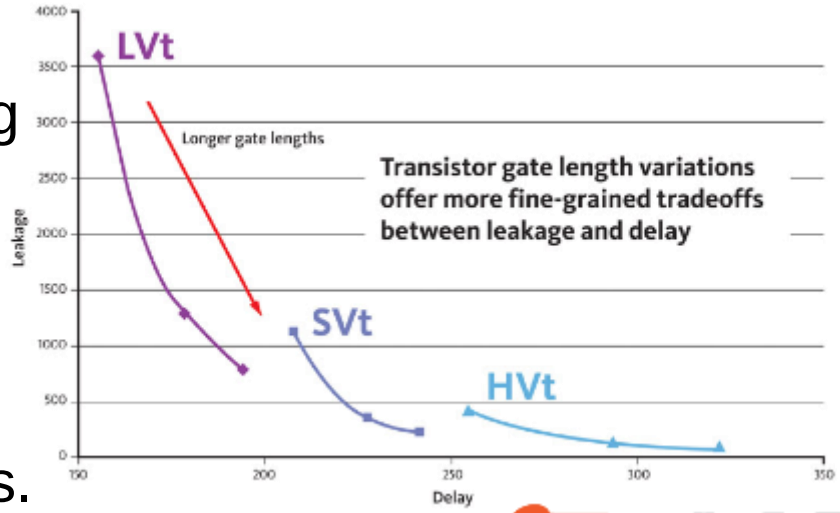
- **Other Cells:**
  - Fillers, Tap cells, Antennas, DeCaps, EndCaps, Tie Cells

Multiple Drive Strengths and VTs

- **Multiple Drive Strength**
  - Each cell will have various sized output stages.
  - Larger output stage → better at driving fanouts/loads.
  - Smaller drive strength → less area, leakage, input cap.
  - Often called X2, X3, or D2, D3, etc.



- **Multiple Threshold (MT-CMOS)**
  - A single additional mask can provide more or less doping in a transistor channel, shifting the threshold voltage.
  - Most libraries provide equivalent cells with three VTs: SVT, HVT, LVT to tradeoff speed vs. leakage.
  - All threshold varieties have same footprint and therefore can be swapped without any placement/routing iterations.

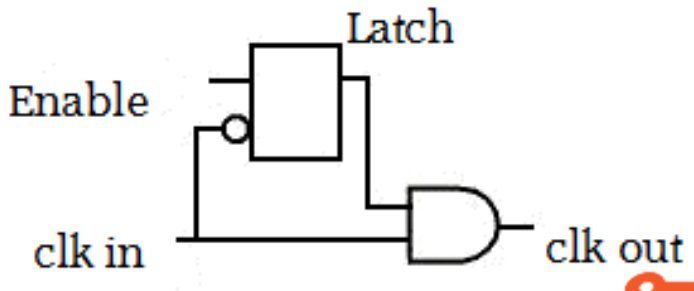


Clock Cells

- **General standard cells are optimized for speed.**
  - That doesn't mean they're balanced…

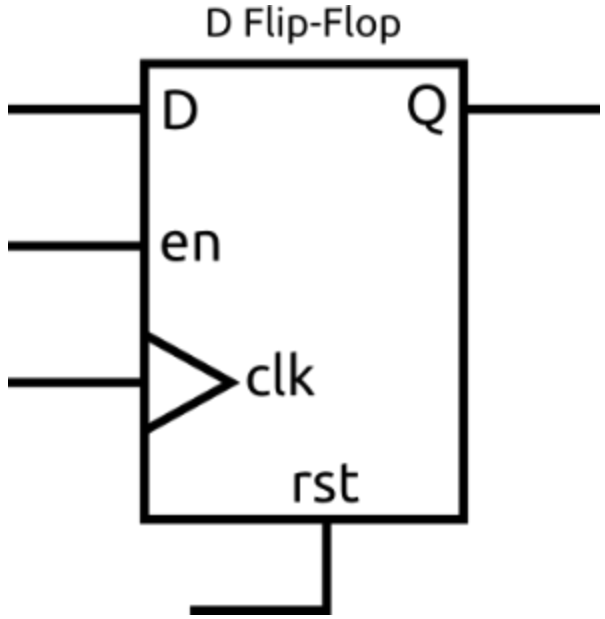
$$\min t_{pd} = \min\left(\frac{t_{p,LH} + t_{p,HL}}{2}\right) \not\Rightarrow t_{p,LH} = t_{p,HL}$$

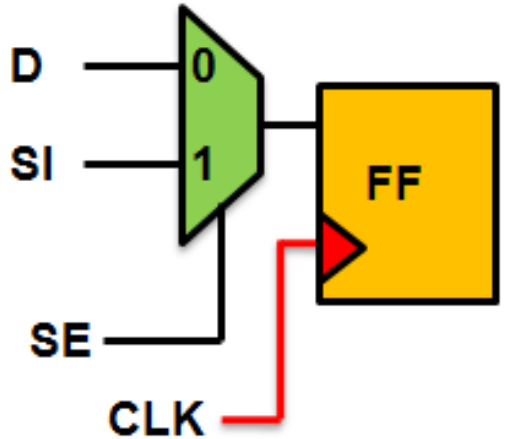
- **This isn't good for clock nets…**
  - Unbalanced rising/falling delays will result in unwanted skew.
  - Special "clock cells" are designed with balanced rising/falling delays to minimize skew.
  - These cells are usually less optimal for data and so should not be used.
- **In general, only buffers/inverters should be used on clock nets**
  - But sometimes, we need gating logic.
  - Special cells, such as *integrated clock gates*, provide logic for the clock networks.



Sequentials

- **Flip Flops and Latches, including**
  - Positive/Negative Edge Triggered
  - Synchronous/Asynchronous Reset/Set
  - Q/QB Outputs
  - Enable
  - Scan
  - etc., etc.

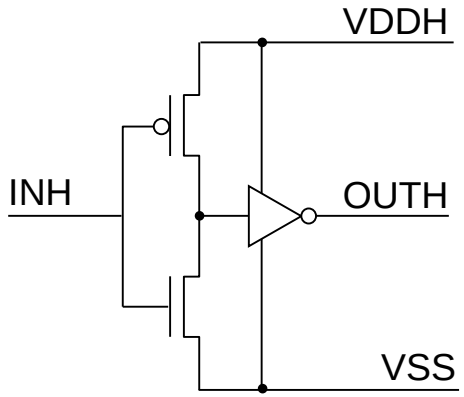




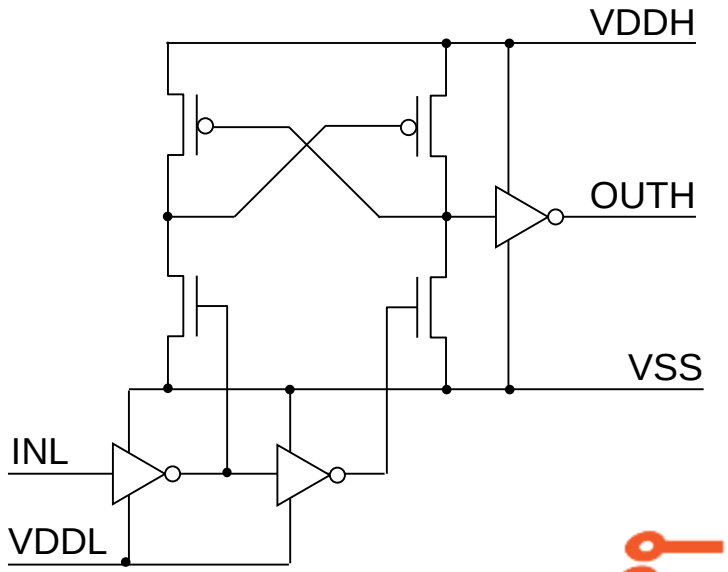
Level Shifters

- **Level shifter cells are placed between voltage domains to pass signals from one voltage to another.**

- **HL shifter**
  - Requires only one voltage
  - Single height cell

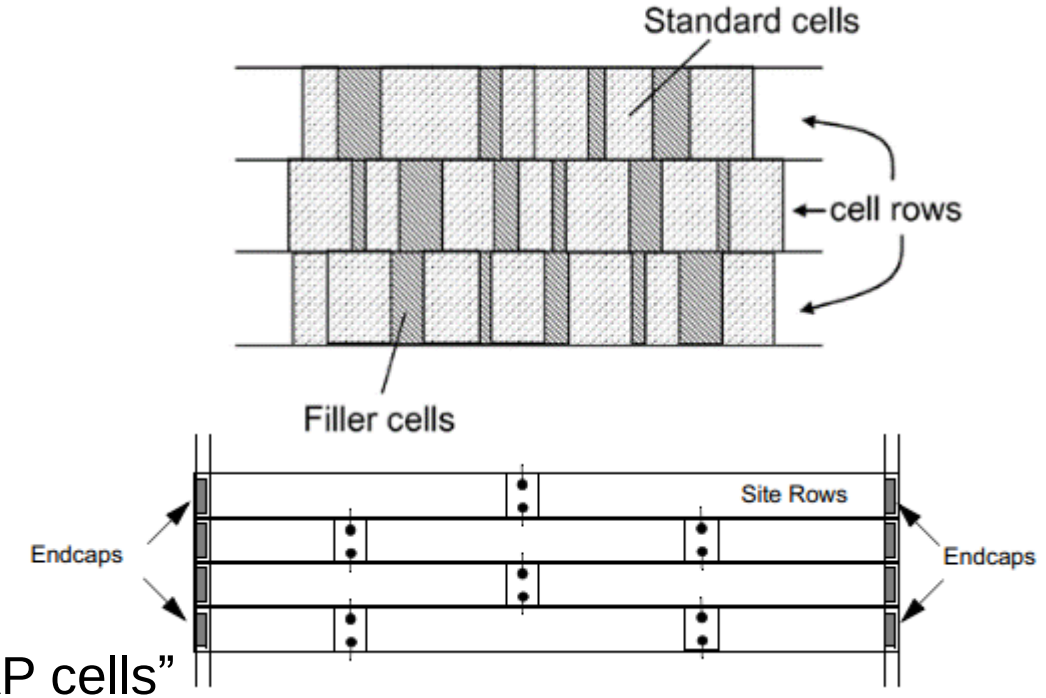


- **LH shifter**
  - Needs 2 voltages
  - Often double height

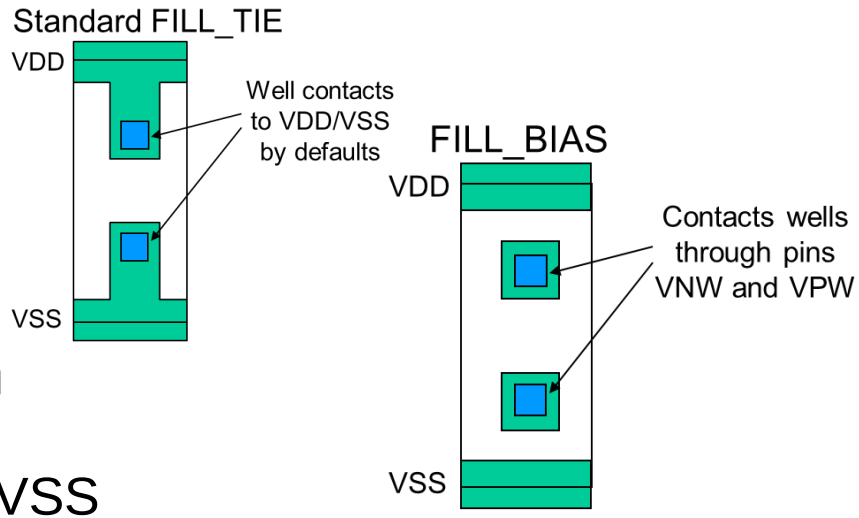


Filler and Tap Cells

- **Filler cells Must be inserted in empty areas in rows**
  - Ensure well and diffusion mask continuity
  - Ensure density rules on bottom layers
  - Provide dummy poly for scaled technologies
  - Sometimes, special cells are needed at the boundaries of rows. These are known as “End Caps”
  - Other fillers may include MOSCAPs between VDD and GND for voltage stability. These are called “DeCAP cells”



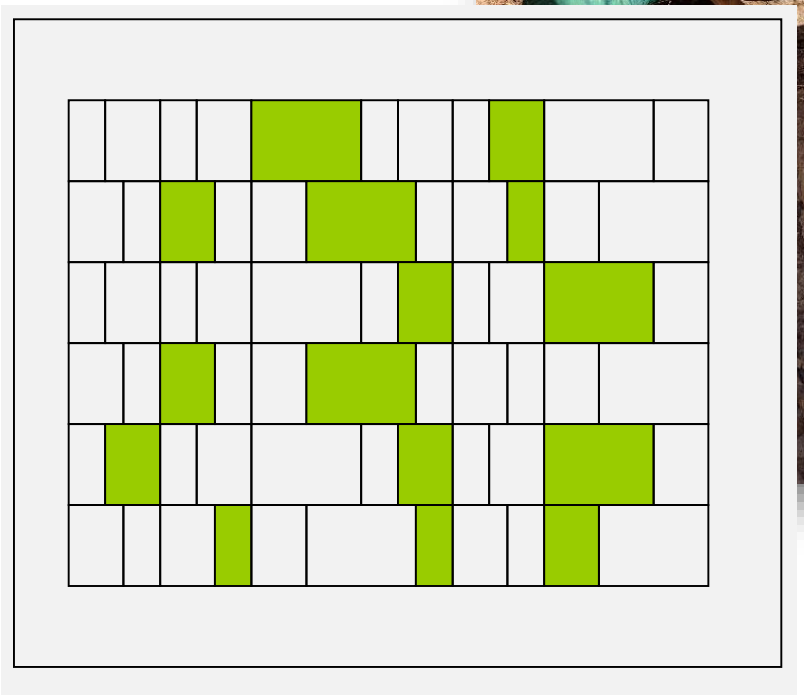
- **Well Taps needed to ensure local body voltage**
  - Eliminate latch-up
  - No need to tap every single cell



- **Back or forward biasing for performance/leakage optimization**
  - N-well voltage different from VDD
  - Substrate or P-well (triple well process) voltage different from VSS
  - Bias voltage routed as signal pin or special power net

Engineering Change Order (ECO) Cells

- **An Engineering Change Order (ECO) is a very late change in the design.**
  - ECOs usually are done after place and route.
  - However, re-spins of a chip are often done without recreating all-masks. This is known as a “Metal-Fix”.
- **ECOs usually require small changes in logic.**
  - How can we do this after placement?
  - Or worse – after tapeout???
- **Solution – Spare (Bonus) Cells!**
  - Cells without functionality
  - Cells are added during design (fill)
  - In case of problems (after processing) new metal and via mask → cells get their wanted functionality
  - Cell combinations can create more complex functions
    - Ex. And,nand,nor,xor,FF,mux,inv,..
- **Special standard cells are used to differentiate from real cells.**



My favorite word… ABSTRACTION!

- **So, what is a cell?**
  - I guess that the detailed layout is sufficient to know (guess) anything and everything about a standard cell.
  - Or it would be easier, if we got the whole Open Access database of the cell…
- **But do we really need to know everything?**
  - For example, does logic simulation need to know if your inverter is CMOS or Pseudo-NMOS?
  - And does a logic synthesizer need to know what type of transistors you used?
- **No!**
  - To make life (and calculations) simpler, we will *abstract away* this info.
  - Each tool will get only the data it really needs.

What files are in a standard cell library?

- **Behavioral Views:**
  - Verilog (or Vital) description used for simulation, logic equivalence.
- **Physical Views:**
  - Layout of the cells (GDSII format) for DRC, LVS, Custom Layout.
  - Abstract of the cells (LEF format) for P&R, RC extraction.
- **Transistor Level:**
  - Spice/Spectre netlist for LVS, transistor-level simulation.
  - Often provided both with parasitics (post-layout) and without.
- **Timing/Power:**
  - Liberty files with characterization of timing and power for STA.
- **Power Grid Views:**
  - Needed for IR Drop analysis.
- **Others:**
  - Symbols for displaying the cells in various tools.
  - OA Libraries for easy integration with Virtuoso.

Behavioral (.v)

Abstract (.lef)

Layout (.gds)

Spice (.spi, .cdl)

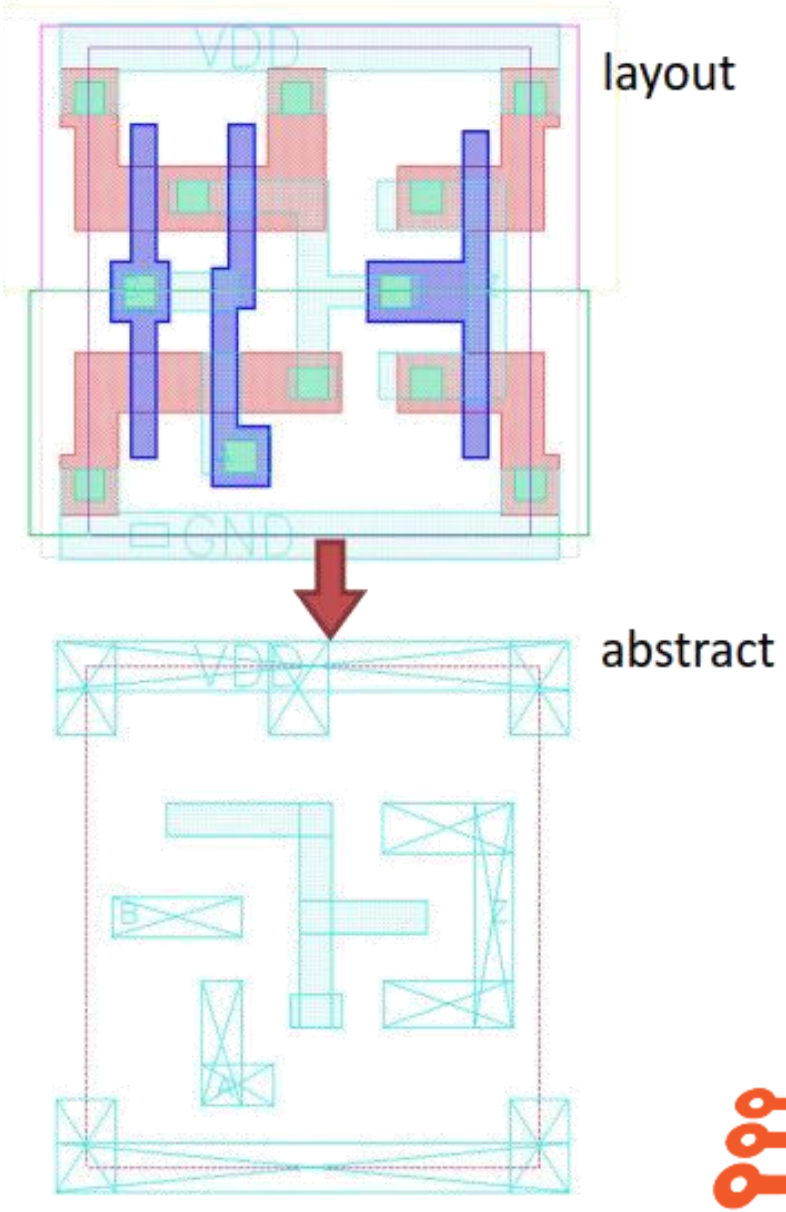
Timing (.lib)

Open Access (.oa)

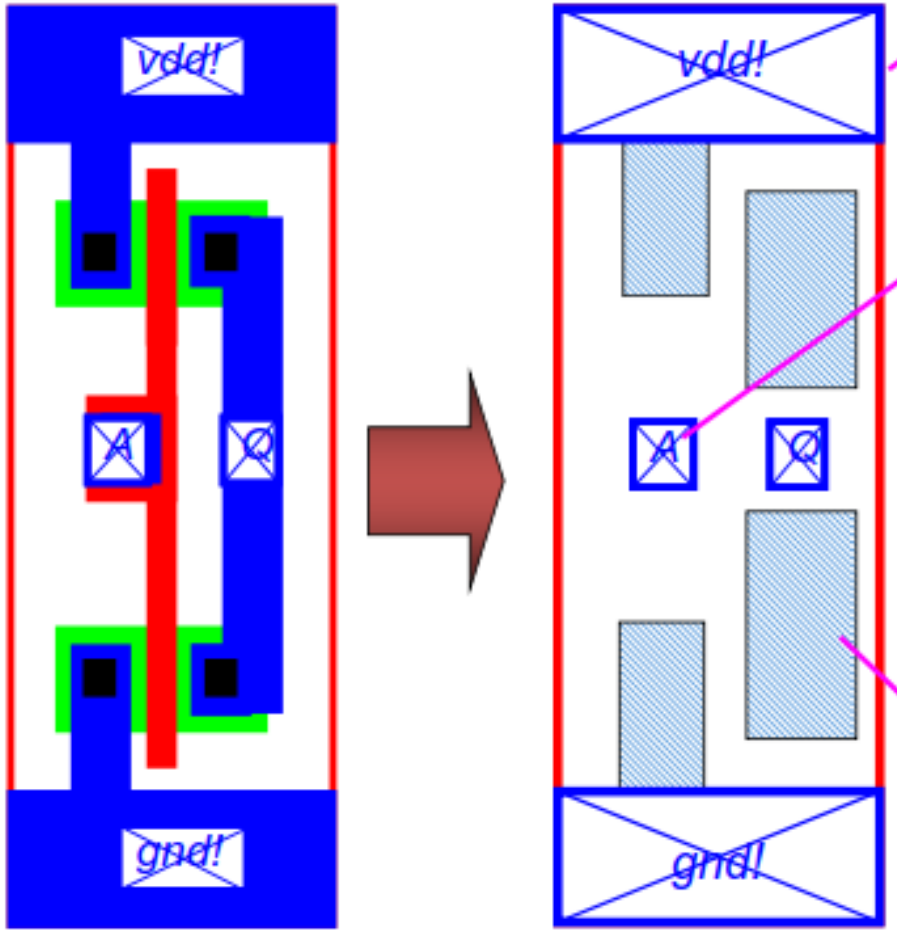
Library Exchange Format (LEF)

Library Exchange Format (LEF)

- **Abstract description of the layout for P&R**
  - Readable ASCII Format.
  - Contains detailed PIN information for connecting.
  - Does not include front-end of the line (poly, diffusion, etc.) data.
  - Contains blockages for DRC.



Library Exchange Format (LEF)



Physical cell size

Terminals with physical placement

Obstructions

LEF

```
MACRO IV
    CLASS CORE ;
    FOREIGN IV 0.000 0.000 ;
    ORIGIN 0.00 0.00 ;
    SIZE 3.00 BY 12.00 ;
    SYMMETRY x y  ;
    SITE CORE ;
    PIN A
        DIRECTION INPUT ;
        ANTENNASIZE 1.4 ;
        PORT
        LAYER metal1 ;
        RECT  0.50 5.00 1.00 5.50
    ;
        END
    END A

    OBS
        LAYER metal1 ;
        RECT  1.90 6.50 2.60 7.20
    ;
        RECT  0.40 4.90 1.00 5.60
    ;
```

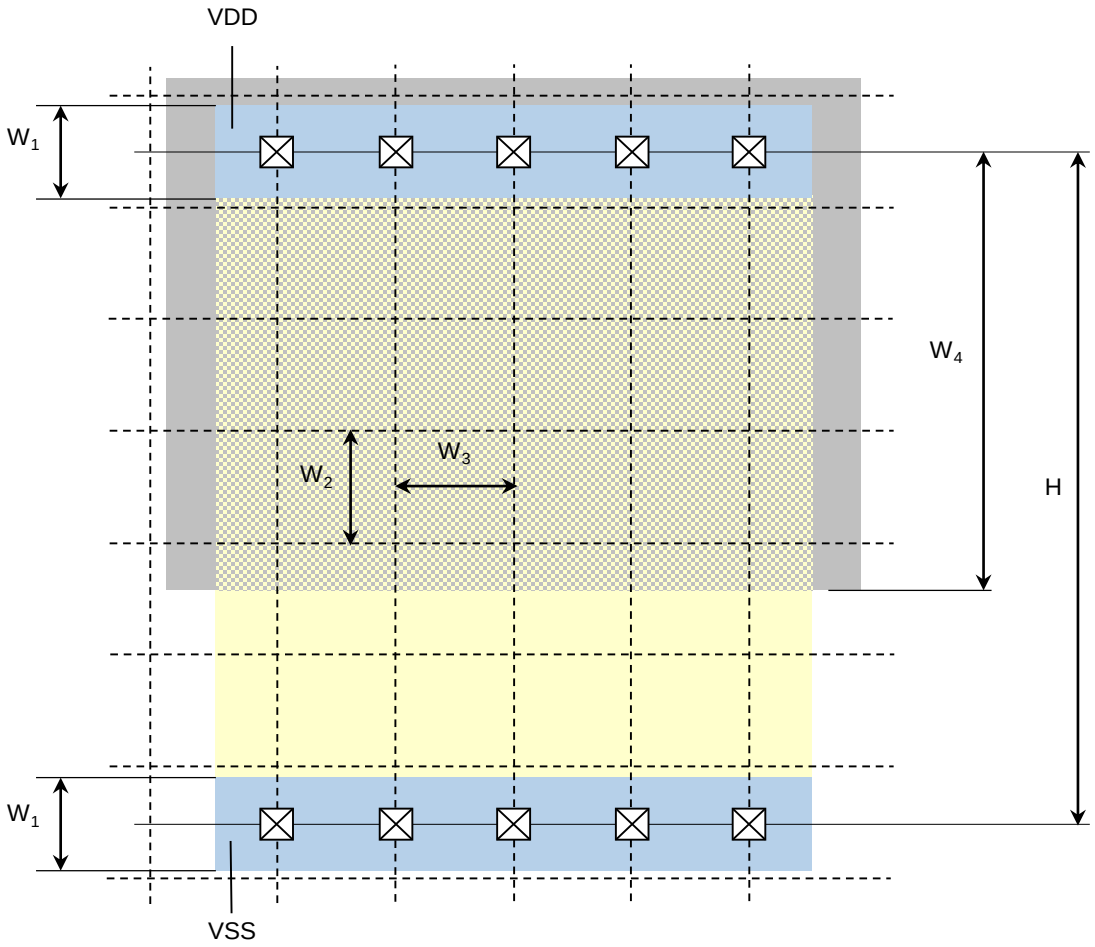
Technology LEF

- **Technology LEF Files contain (simplified) information about the technology for use by the placer and router:**
  - Layers and layer types.
  - Sites (x and y grids of the library) – i.e., double height cells!
  - Via definitions
  - Design Rules
  - Parasitic and Antenna data.

```
LAYER MET1
  TYPE ROUTING ;
  PITCH 3.5 ;
  WIDTH 1.2 ;
  SPACING 1.4 ;
  DIRECTION HORIZONTAL ;
  RESISTANCE RPERSQ .7E-01 ;
  CAPACITANCE CPERSQDIST .46E-04 ;
END MET1

LAYER VIA
  TYPE CUT ;
END VIA
```

Technology LEF

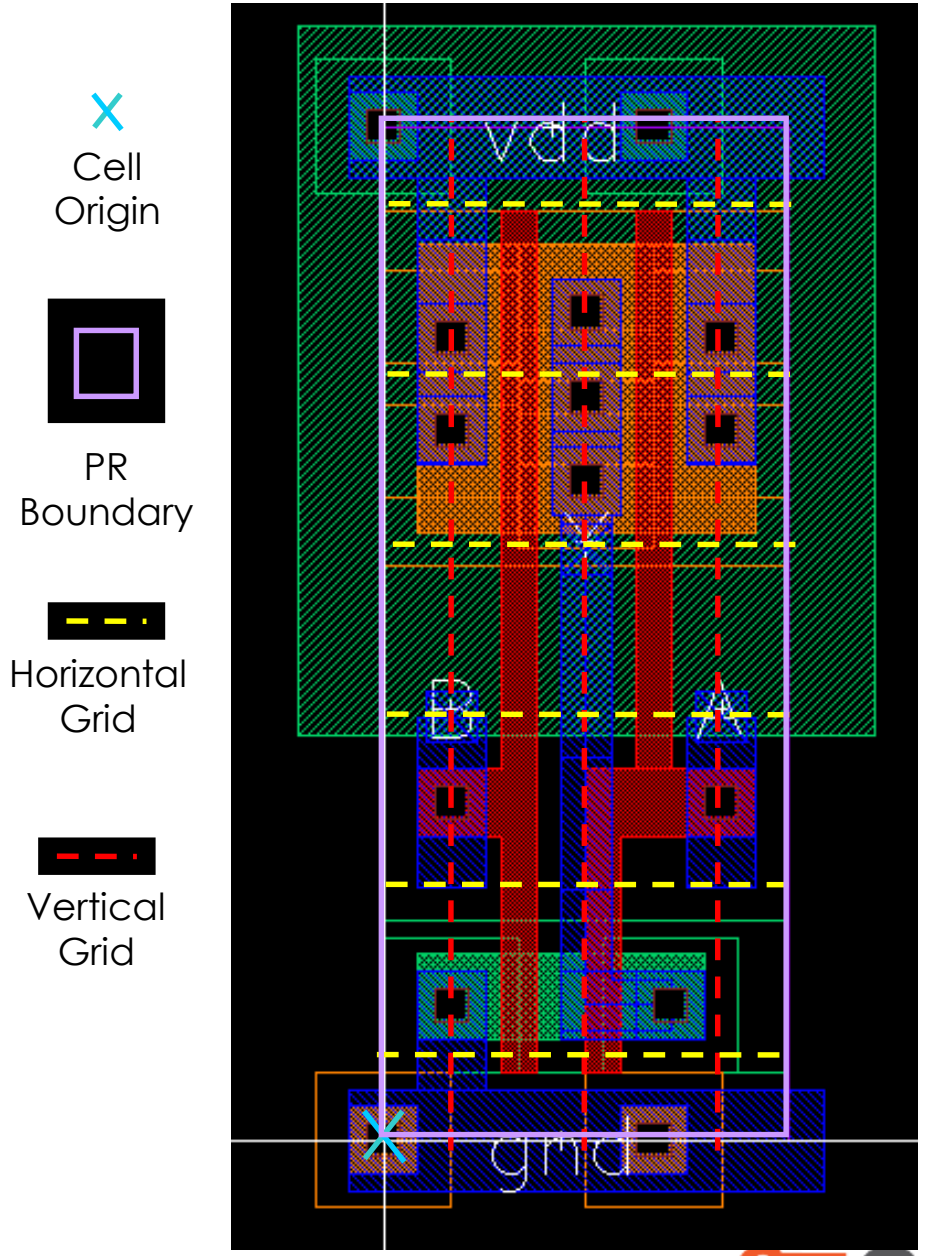
- **Cell height is measured in *Tracks***
  - A Track is one M1 pitch
  - E.g., An 8-Track Cell has room for 8 horizontal M1 wires.
- **The more tracks, the wider the transistors, the faster the cells.**
  - 7-8 *low-track* libraries for area efficiency
  - 11-12 *tall-track* libraries for performance, but have high leakage
  - 9-10 *standard-track* libraries for a reasonable area-performance tradeoff



Parameter	Symbol
Cell height (# tracks)	H
Power rail width	W_1
Vertical grid	W_2
Horizontal grid	W_3
N-Well height	W_4

Technology LEF

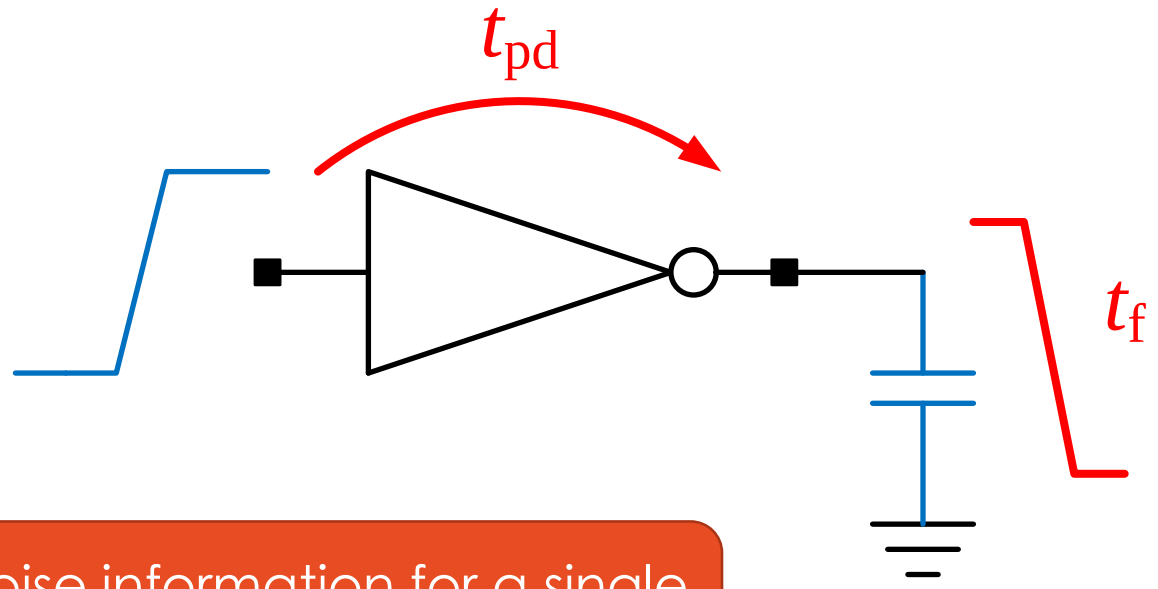
- **Cells must fit into a predefined grid**
  - The minimum Height X Width is called a SITE.
  - Must be a multiple of the minimum X-grid unit and row height.
  - Cells can be double-height, for example.
- **Pins should coincide with routing tracks**
  - This enables easy connection of higher metals to the cell.



Liberty Timing Models (.lib)

Liberty (.lib): Introduction

- **How do we know the delay through a gate in a logic path?**
  - Running SPICE is way too complex.
  - Instead, create a *timing model* that will simplify the calculation.
- **Goal:**
  - For every timing arc, calculate:
    - Propagation Delay (tpd)
    - Output transition (trise, tfall)
  - Based on:
    - Input net transition.
    - Output Load Capacitance

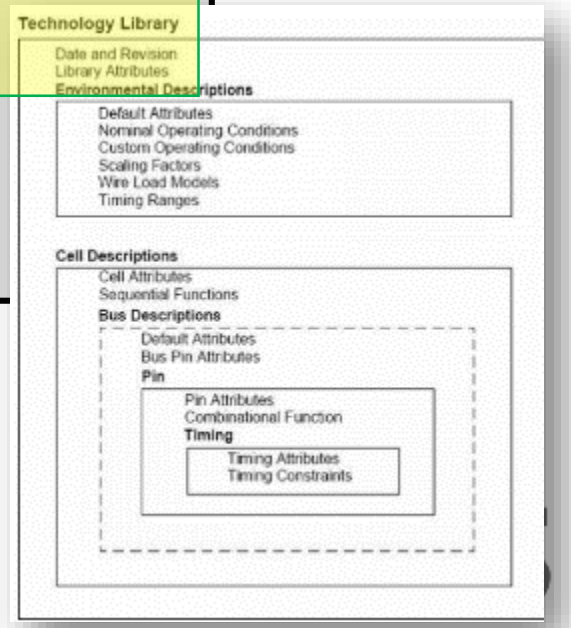


Note that every .lib will provide timing/power/noise information for a single corner, i.e., process, voltage, temperature, RCX, etc.

Liberty (.lib): General

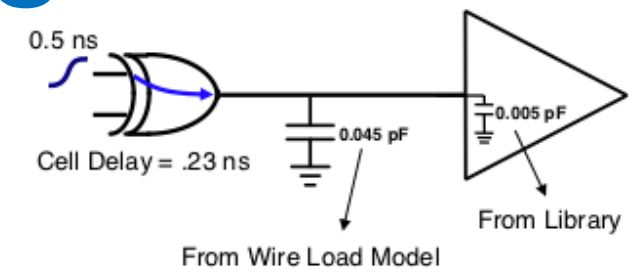
- **Timing data of standard cells is provided in the Liberty format.**
  - Library:
    - General information common to all cells in the library.
    - For example, operating conditions, wire load models, look-up tables
  - Cell:
    - Specific information about each standard cell.
    - For example, function, area.
  - Pin:
    - Timing, power, capacitance, leakage, functionality, etc. characteristics of each pin in each cell.

```
library (nameoflibrary) {
... /* Library level simple and complex attributes */

/* Cell definitions */
cell (cell_name) {
   ... /* cell level simple attributes */

   /* pin groups within the cell */
   pin(pin_name) {
       ... /* pin level simple attributes */

       /* timing group within the pin level */
       timing(){
            ... /* timing level simple attributes */ }
       ... /* additional timing groups */

   } /* end of pin */
   ... /* more pin descriptions */
} /* end of cell */
... /* more cells */

} /* end of library */
```



Liberty (.lib): Timing Models

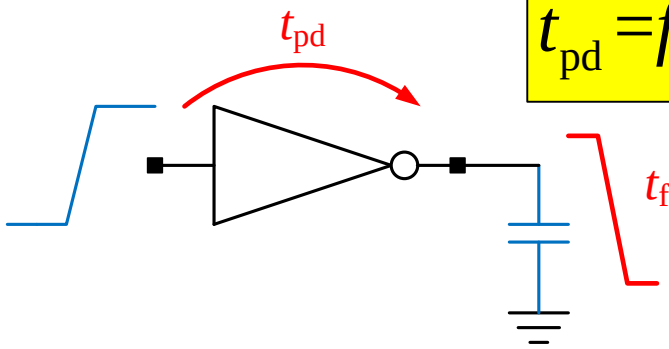
- **Non-Linear Delay Model (NLDM)**
  - Driver model:
    - Ramp voltage source
    - Fixed drive resistance
  - Receiver model:
    - Min/max rise/fall input caps
  - Very fast
  - Doesn't model cap variation during transition.
  - Loses accuracy beyond 130nm



SPICE	Output Load (pF)			
	.005	.05	.10	.15
Input Trans (ns) 0.0	.1	.15	.2	.25
0.5	.15	.23	.3	.38
1.0	.25	.4	.55	.75

Cell Delay (ns)

```
lu_table_template(delay_template_5x5) {
  variable_1 : input_net_transition;
  variable_2 : total_output_net_capacitance;
  index_1 ("1000.0, 1001.0, 1002.0, 1003.0, 1004.0");
  index_2 ("1000.0, 1001.0, 1002.0, 1003.0, 1004.0");
}
```



$$t_{pd} = f\left(t_{input}, C_{load}\right)$$

```
cell (INVX1) {
  pin(Y) {
    timing() {
      cell_rise(delay_template_5x5) {
        values ( \
          "0.147955, 0.218038, 0.359898, 0.922746, 1.76604", \
          "0.224384, 0.292903, 0.430394, 0.991288, 1.83116", \
          "0.365378, 0.448722, 0.584275, 1.13597, 1.97017", \
          "0.462096, 0.551586, 0.70164, 1.24437, 2.08131", \
          "0.756459, 0.874246, 1.05713, 1.62898, 2.44989"); }
```

Liberty (.lib): Timing Models

- **Non-Linear Delay Model (NLDM)**
  - Delay calculation interpolation

Cell Fall

Cap\Tr	0.05	0.2	0.5
0.01	0.02	0.16	0.30
0.5	0.04	0.32	0.60
2.0	0.0[illegible]	[illegible]64	1.20

0.178

Cell Rise

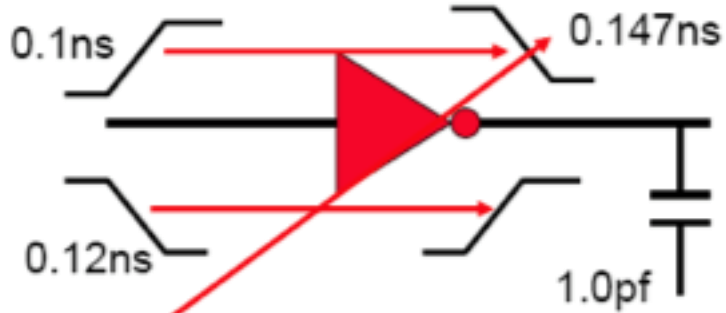
Cap\Tr	0.05	0.2	0.5
0.01	0.03	0.18	0.33
0.5	0.06	0.36	0.66
2.0	0.09	[illegible]72	1.32

0.261

Fall Transition

Cap\Tr	0.05	0.2	0.5
0.01	0.01	0.09	0.15
0.5	0.03	0.27	0.45
2.0	0.06	[illegible]54	0.90

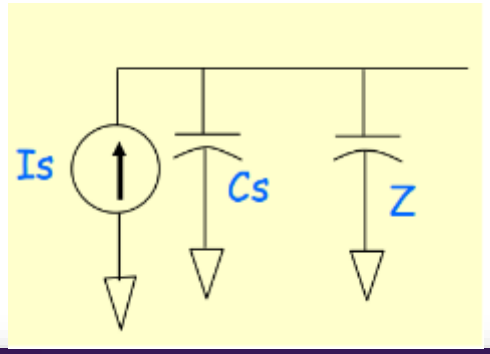
0.147

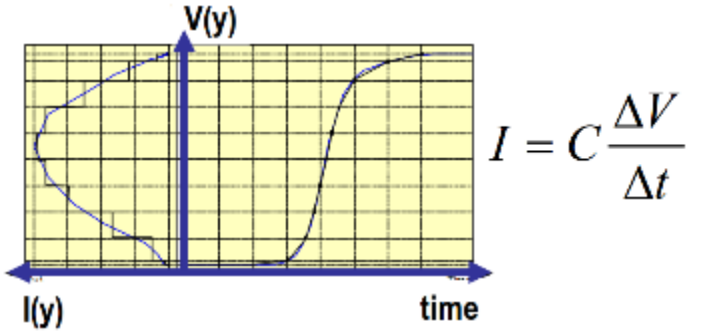


Fall delay = 0.178ns
Rise delay = 0.261ns
Fall transition = 0.147ns
Rise transition = …

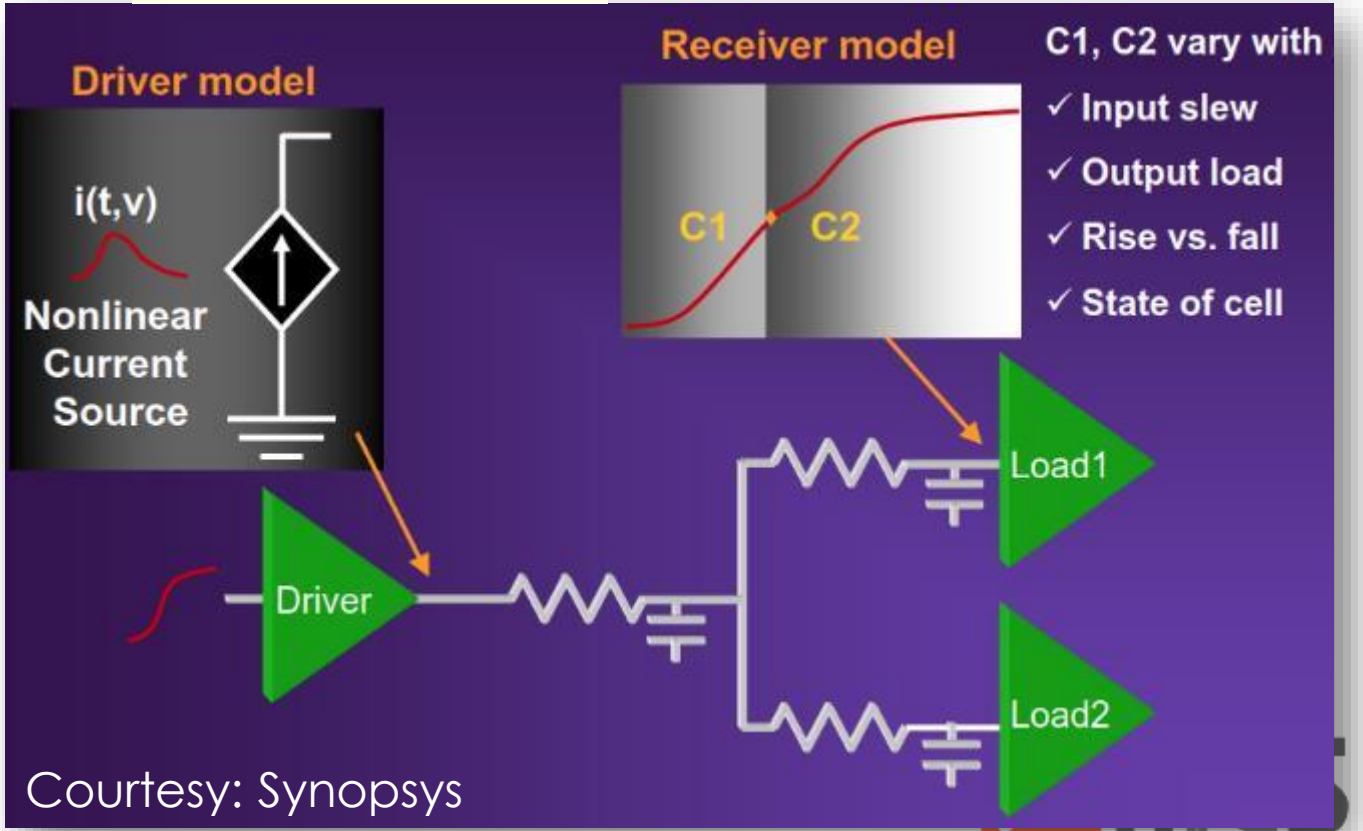
Liberty (.lib): Timing Models

- **Current Source Models (CCS, ECSM)**
  - Model a cell's nonlinear output behavior as a current source
  - Driver model:
    - Nonlinear current source
  - Receiver model:
    - Changing capacitance
  - Requires many more values
  - Requires a bit more calculation
  - Essential under 130nm
  - Within 2% of SPICE.





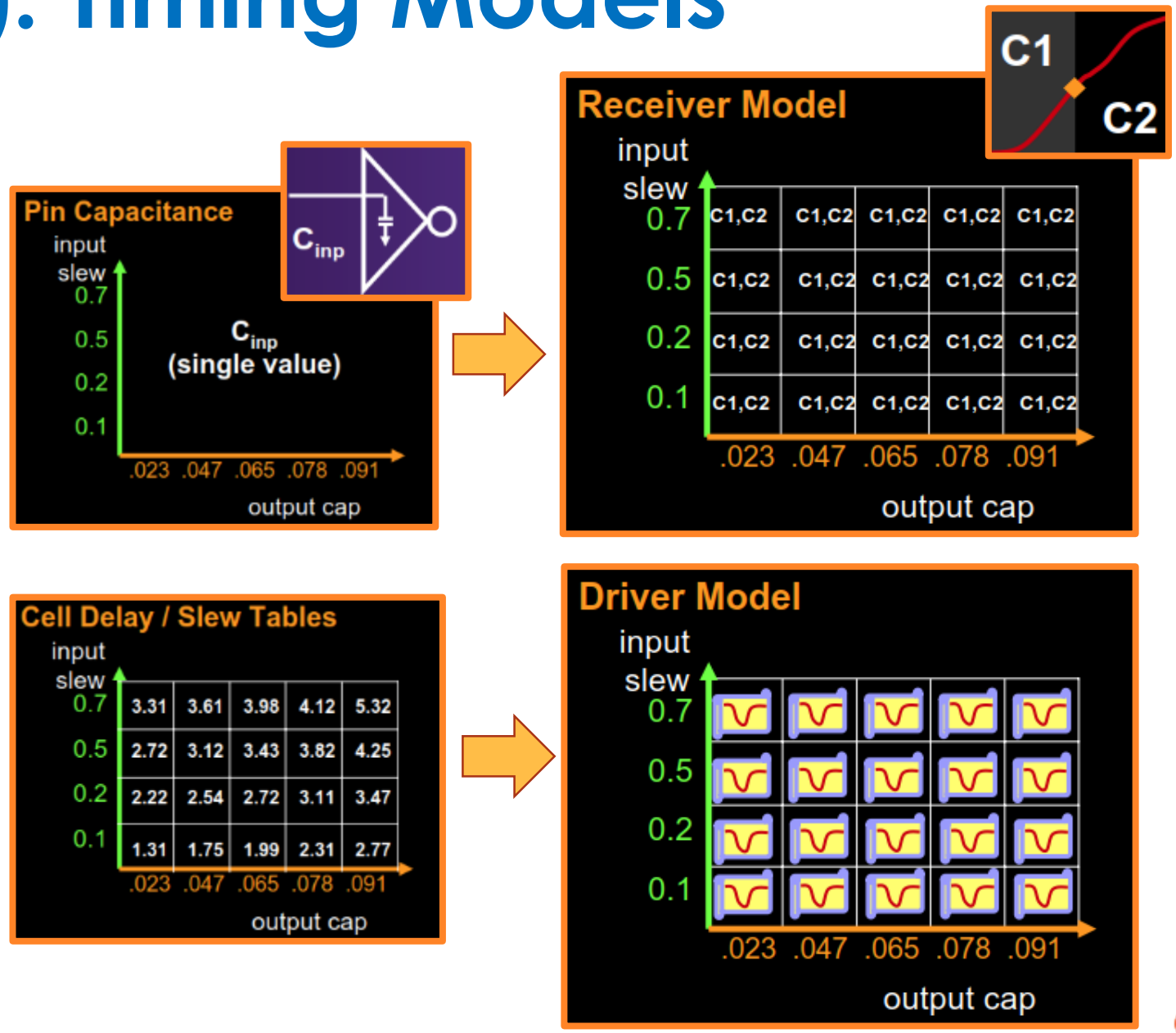
Courtesy: Cadence



Courtesy: Synopsys

Liberty (.lib): Timing Models

- **NLDM vs CCS/ECSM**

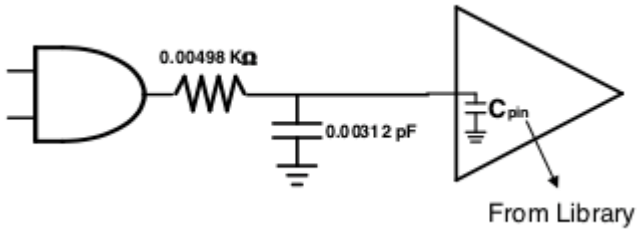


Courtesy: Synopsys

Liberty (.lib): Wire Load Models

- How do you estimate the parasitics (RC) of a net before placement and routing?
- Wire Load Models estimate the parasitics based on the *fanout* of a net.

Net Fanout	Resistance KΩ	Capacitance pF
1	0.00498	0.00312
2	0.01295	0.00812
3	0.02092	0.01312
4	0.02888	0.01811




```
library (myLib) {
  wire_load("WLM1")
    resistance: 0.0006 ; // R per unit length
    capacitance: 0.0001 ; // C per unit length
    area : 0.1 ; // Area per unit length
    slope : 1.5 ; // Used for linear extrapolation
    fanout_length(1, 0.002) ; // for fo=1, Lwire=0.002
    fanout_length(2, 0.006) ; // for fo=2, Lwire=0.006
    fanout_length(3, 0.009) ; // for fo=3, Lwire=0.009
    fanout_length(4, 0.015) ; // for fo=4, Lwire=0.015
    fanout_length(5, 0.020) ; // for fo=5, Lwire=0.020
    fanout_length(6, 0.028) ; // for fo=6, Lwire=0.028
  }
} /* end of library */
```

Instead, try topographical synthesis!

Other Contents of SC Library

Other contents of SC Library

- **Many other files and formats may be provided as part of a standard cell library:**
  - GDS
  - Verilog
  - ATPG
  - Power Grid Models
  - OA Databases
  - Spice Models
  - etc.
- **We will go over a real life example in the discussion.**

Documentation and Datasheets

- **So, are we just supposed to look through and see what the vendor decided to provide us with?**
  - Yes!
  - However they probably supplied some PDFs describing the library.
  - And usually there are data sheets with numbers for each corner.

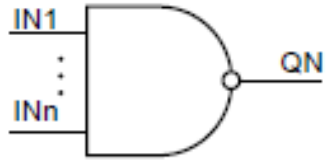


Figure 9.6. Logic Symbol of NAND

Table 9.11. NAND Truth Table (n=2,3,4)

IN1	IN2	. . .	INn	QN
0	X	. . .	X	1
X	0	. . .	X	1
. . .	. . .	. . .	. . .	1
X	X	. . .	0	1
1	1	1	1	0

Table 9.12. NAND Electrical Parameters and Areas

Cell Name	Operating Conditions: VDD=1.2 V DC, Temp=25 Deg.C, Operating Frequency: Freq=300 MHz, Capacitive Standard Load: Csl=13 fF				Area
	Cload	Prop Delay (Avg)	Power		
			Leakage (VDD=1.32 V DC, Temp=25 Dec.C)	Dynamic	
		ps	nW	nW/MHz	(um^2)
NAND2X1	1 x Csl	51	336	15	5.5296
NAND2X2	2 x Csl	51	673	28	9.2160
NAND3X1	1 x Csl	130	492	38	11.9808
NAND3X2	2 x Csl	142	770	59	12.9024
NAND4X0	0.5 x Csl	66	400	22	8.2944
NAND4X1	1 x Csl	127	716	57	12.9024

And what about other IPs?

- **All IPs will be provided as a library, including most of the views a standard cell library will have.**
- **These are required for integration of the hard macros in the standard design flow (simulation, synthesis, P&R, verification, etc.)**
- **Memories (SRAMs) are a special case, as they usually come with a *memory compiler* that generates the particular memory cut the designer requires.**

Main References

- IDESA
- Synopsys
- Cadence
- Mississippi State
- Arizona State