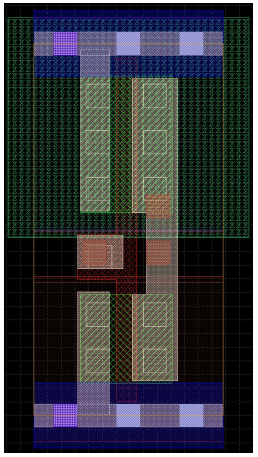
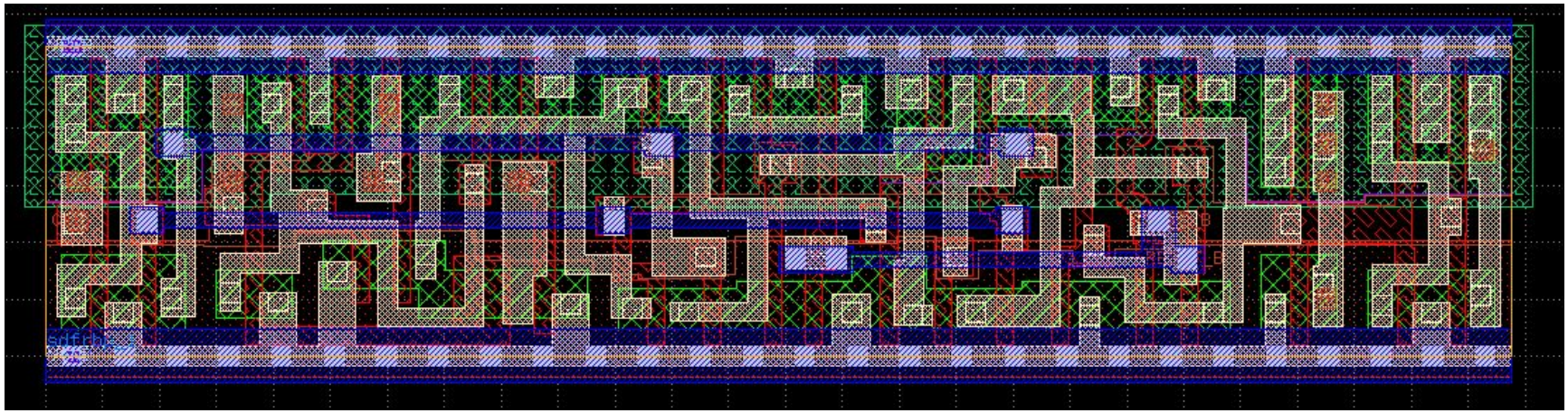
Lecture 08: Cell Libraries

Matthew Guthaus
Professor
UCSC, Computer Science & Engineering
http://vlsida.soe.ucsc.edu
mrg@ucsc.edu

Cell Heights

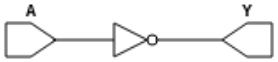
- Fixed height for placement in rows
  - Somes have libraries with 7-12 tracks tall
  - More tracks means less dense but easier to route

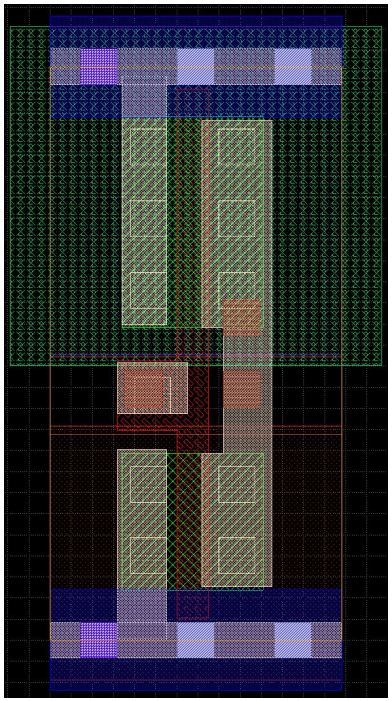




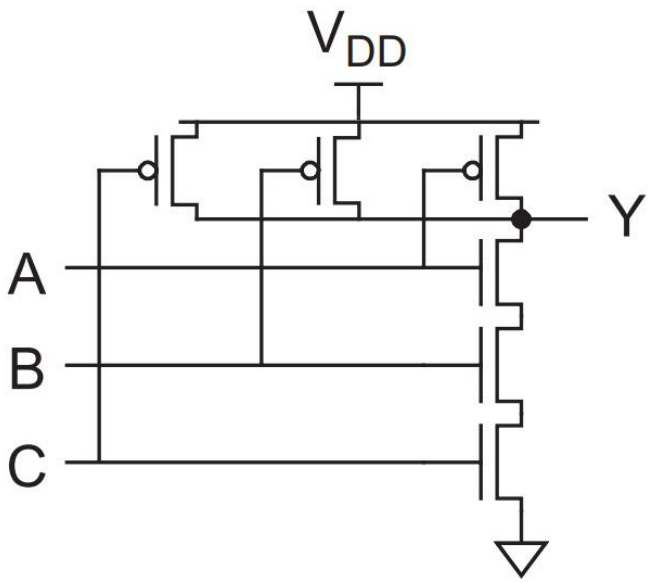
Logic Functions

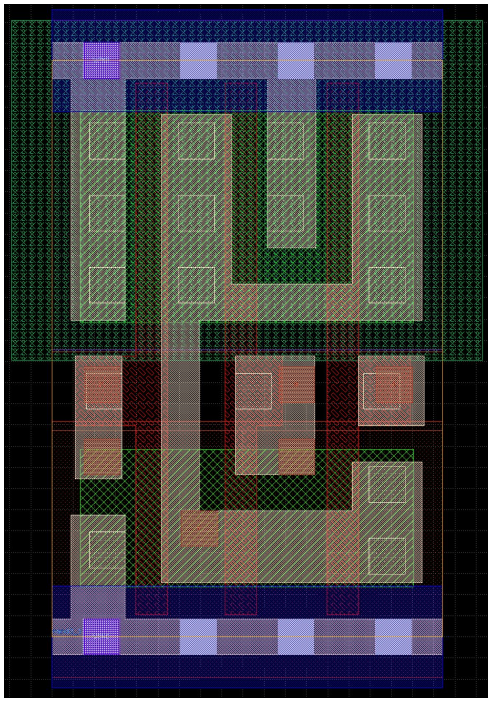
Single CMOS + input or output inverters



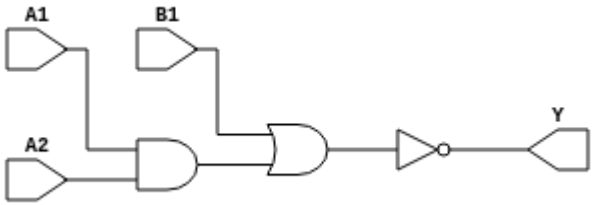


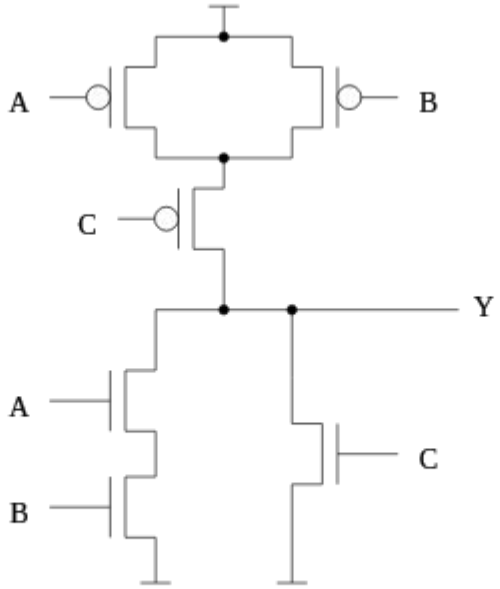
INV

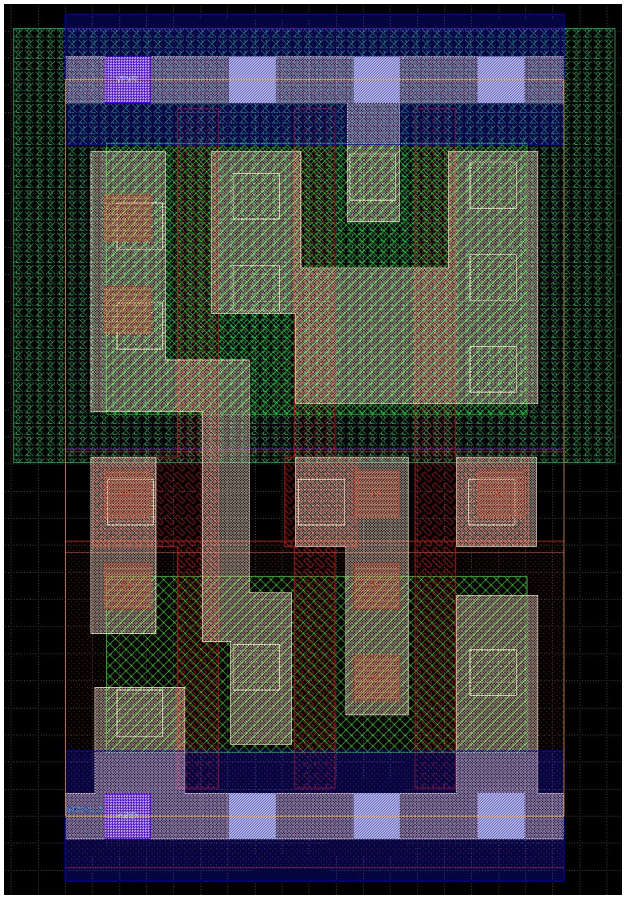




And2-Or1 Invert (a21oi)

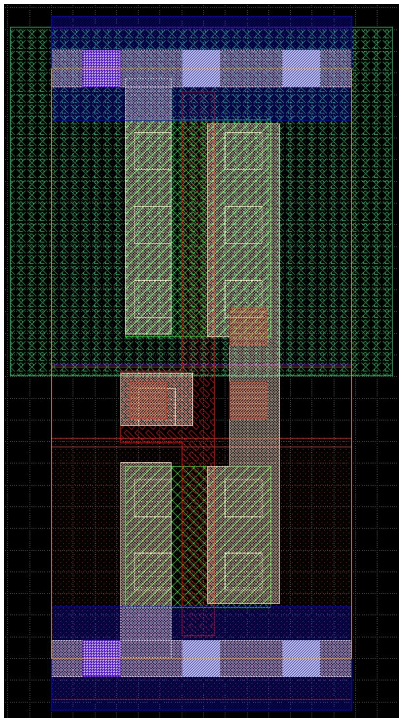




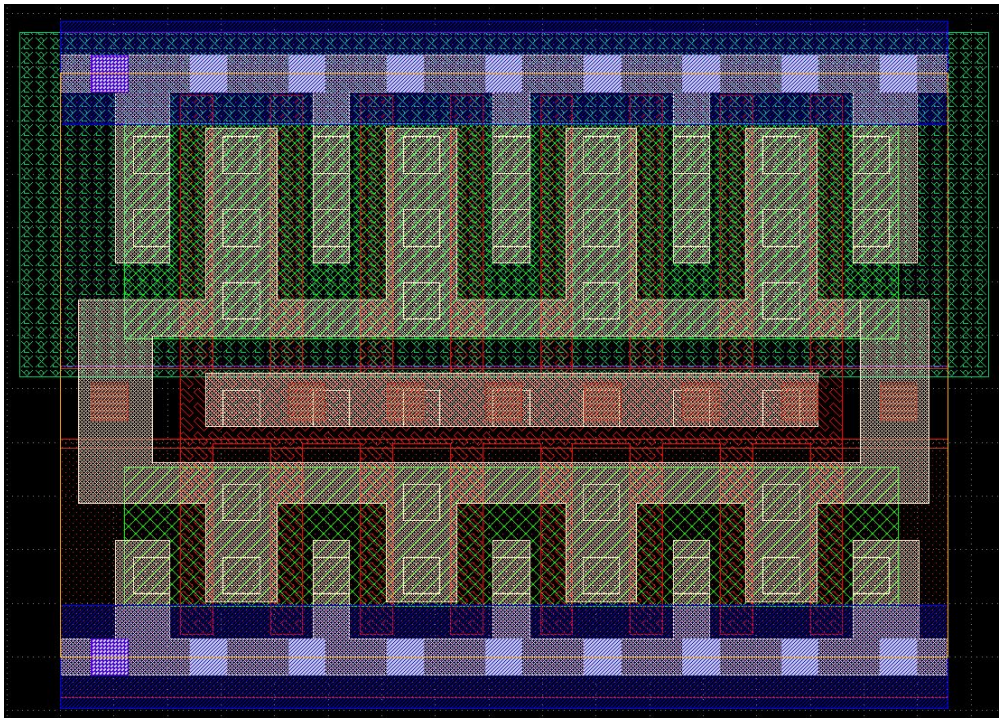


Drive Strengths (Sizes)

Adjusts sizes and number of parallel transistors to drive higher fanout and more wires

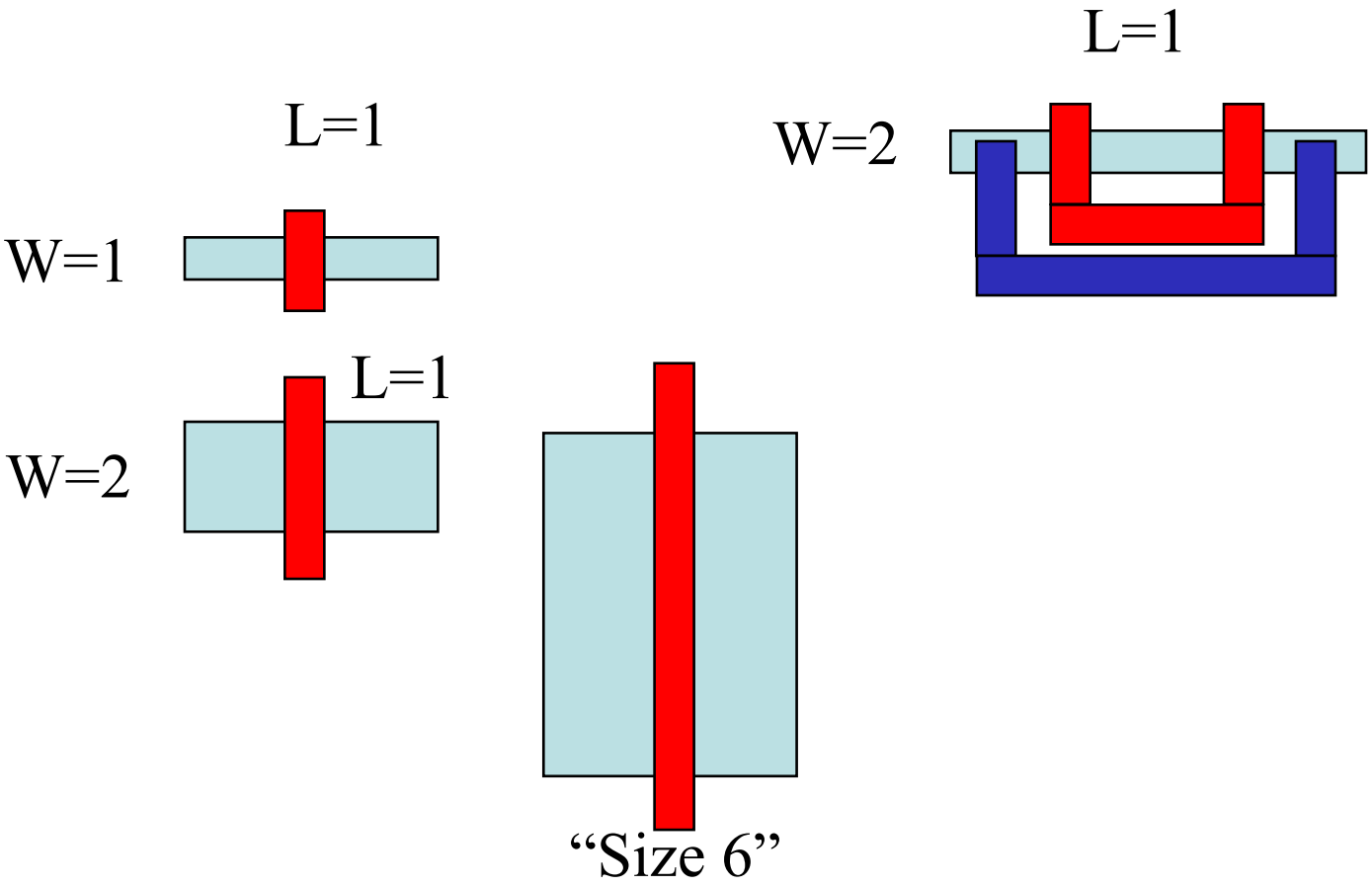


INV 1x



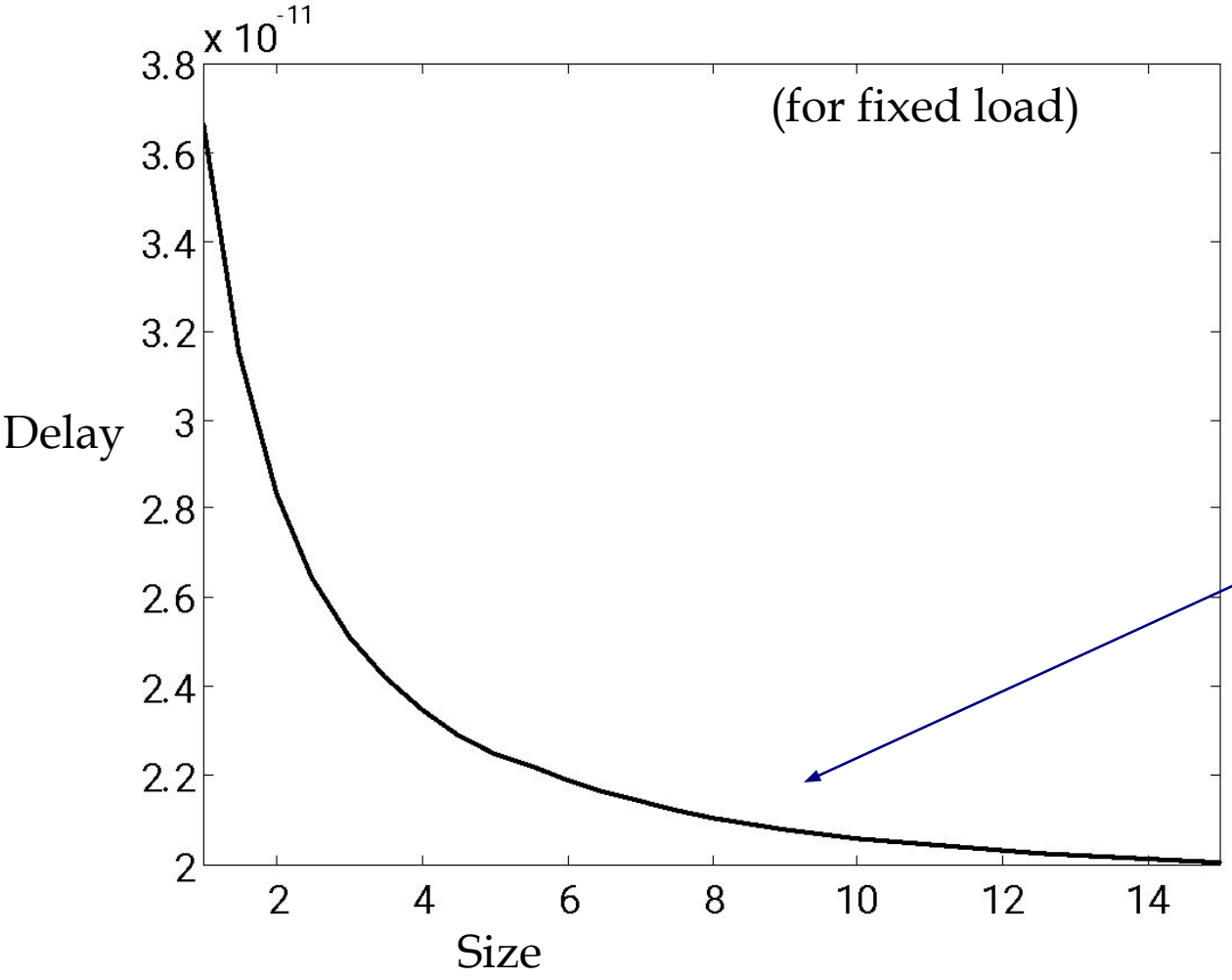
INV 8x

Transistor Sizes



Size is often compared to minimum.

Gate Sizing



Self-loading effect:
Intrinsic capacitances dominate

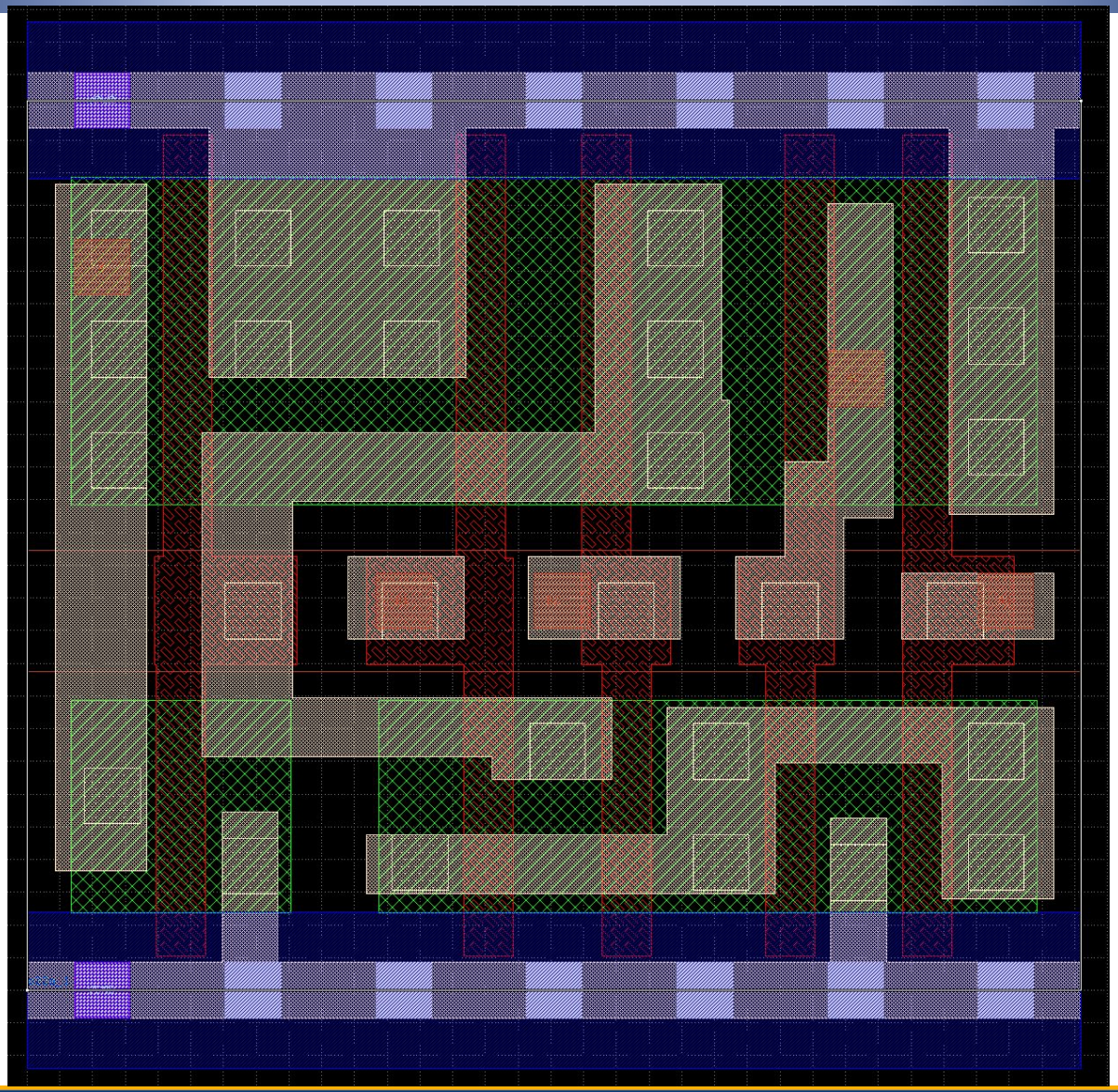
Short Fat vs Tall Skinny Libraries

- More logic types = “fat”
  - Can easily represent more complex functions, but stacked transistors are slower.
- More sizes = “tall”
  - Can find exact size to give the best area vs delay
- Common trade-off
  - Usually limited to 3-4 input CMOS gates
  - Usually inv/buf have 12+ sizes while other gates have 3-4 sizes

Cell Abutment

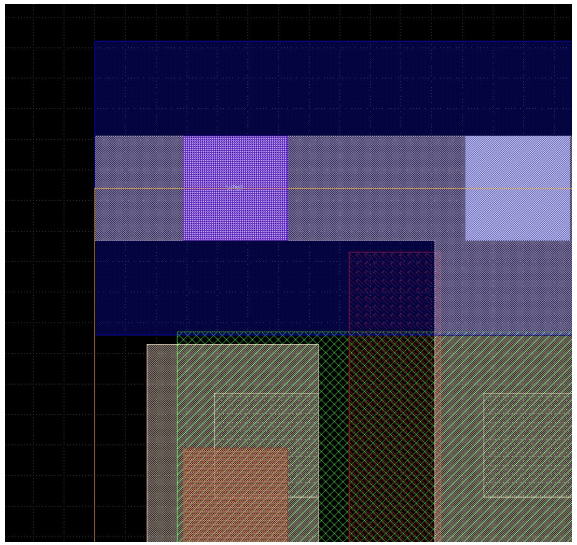
- Cells are designed with a “placement boundary”
  - ½ DRC rule to adjacent cells
- Cells are usually a multiple of a fixed width too
- Power rails connect automatically to adjacent cells (goes up to boundary)
- Nwell and psdm/nsdm may extend beyond boundary

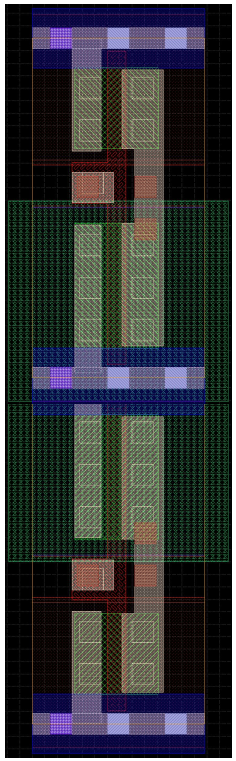
Placement Boundary



Alternating Rows

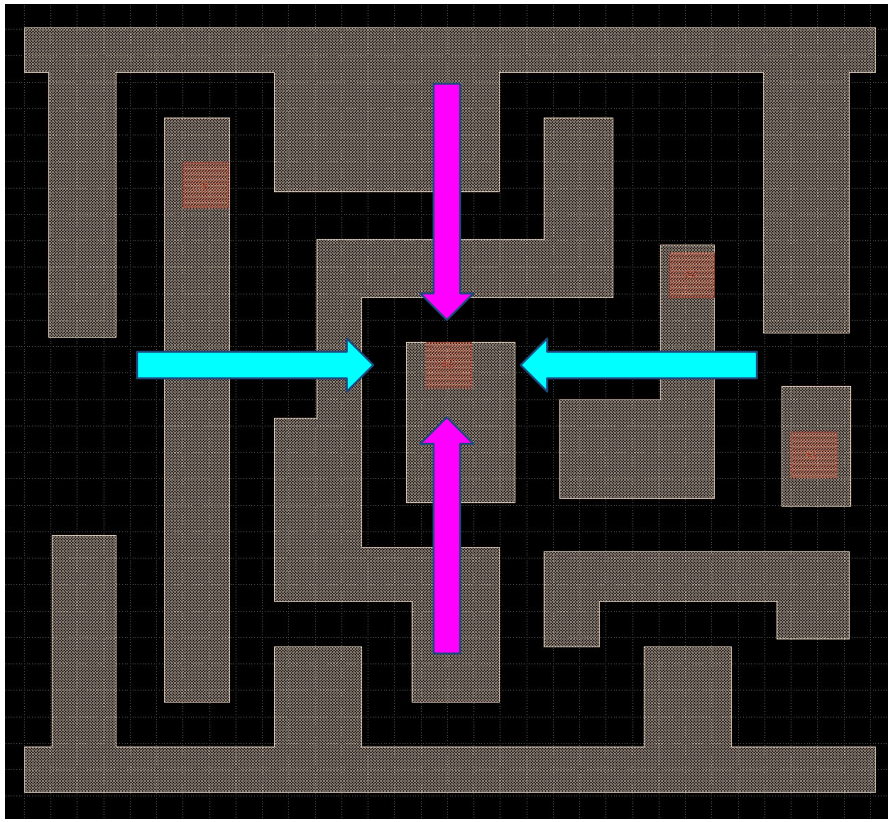
- Usually rows alternate flipping vertically to share the ground and power rails
- Placement boundary “splits” the rail
- Nwell is shared between rows





Pin Access

- Pins are staggered vertically and horizontally
  - Alternating metal directions M1 horizontal, M2 vertical, M3 horizontal…
  - Provides more access options
- Sometimes multiple pins are offered



Sequential Cells

- Latches
- Flip-Flops
  - Sync vs. Async reset
  - Enable signal
  - Test scan or not
  - Inverted and non-inverted output
  - Inverted clock

Arithmetic/Other Cells

- Half- and Full-Adder
- Full-adder with inverted carry
- Mux 2:1, 4:1
- Tristate inverter/buffer
  - Can “not drive” (output Z in Verilog)

Clock Buffers

Balanced rise/fall delays

Bigger sizes than regular buffers

Special Cells

Tap Cells

Filler Cells

Antenna Diodes

Layout File Formats

- GDSII = full layout
  - Includes FEOL (i.e. transistors)
- LEF = abstract layout
  - Only includes BEOL (i.e. interconnect)
    - Only pins, metal blockage, power rails
  - Used in placement and routing tools

Timing and Power File Formats

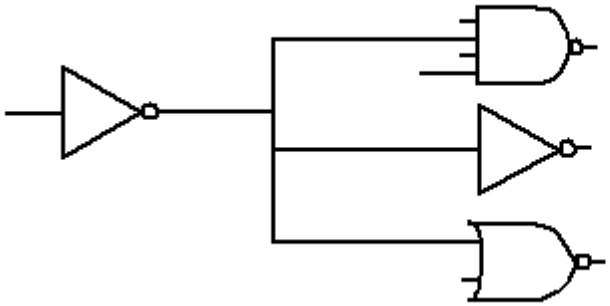
- Liberty (.lib) has the timing and power of each gate for different conditions
  - Used in Static Timing Analysis (later)
  - sky130_fd_sc_hd__ff_100C_1v65.lib
- Process
  - Typical, fast, or slow NMOS or PMOS
- Voltage
  - 1.28V to 1.8V
- Temperature
  - 40C, 80C, 100C

Liberty (.lib) Models

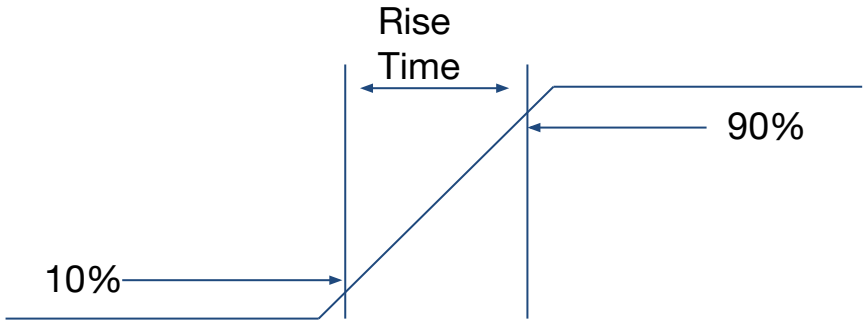
- Nonlinear Delay Models (NLDM)
  - Generated from many spice simulations
    - Input slew
    - Output load capacitance
  - Accuracy limited by range and spacing of table rows/columns
- Current Source Models (CSM or CCS)
  - Models delay as voltage-controlled current source (VCCS)
    - Input voltage
    - Output voltage
  - Two tables: output current and charge on output self-capacitance
    - Current is from DC simulations
    - Charge is from set of transient simulations
- Also includes cell area, cell function, sequential cells, and other stuff

Capacitive Load and Signal Slew

More later, but…

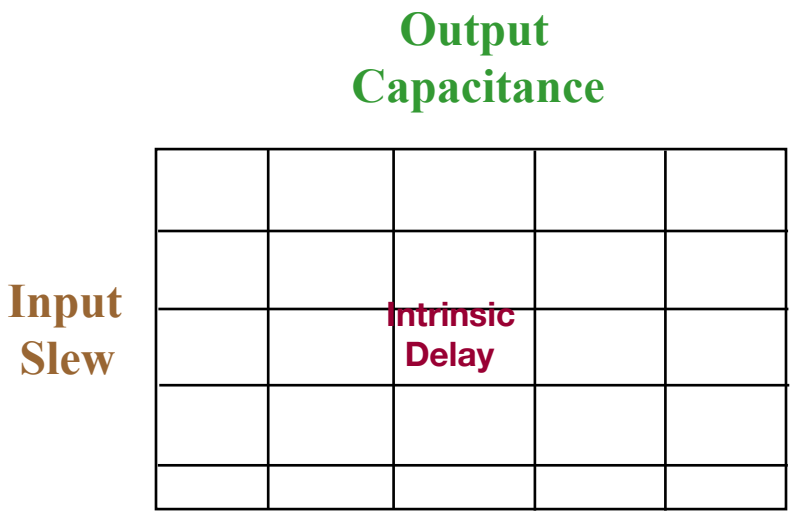


- Capacitive Load
  - Fanout gates add load
  - Longer wires add load
- Signal Slew (actually, measure of time)
  - Signals aren't perfect digital, so measure of time to rise or time to fall
  - Usually 10% to 90% of supply voltage

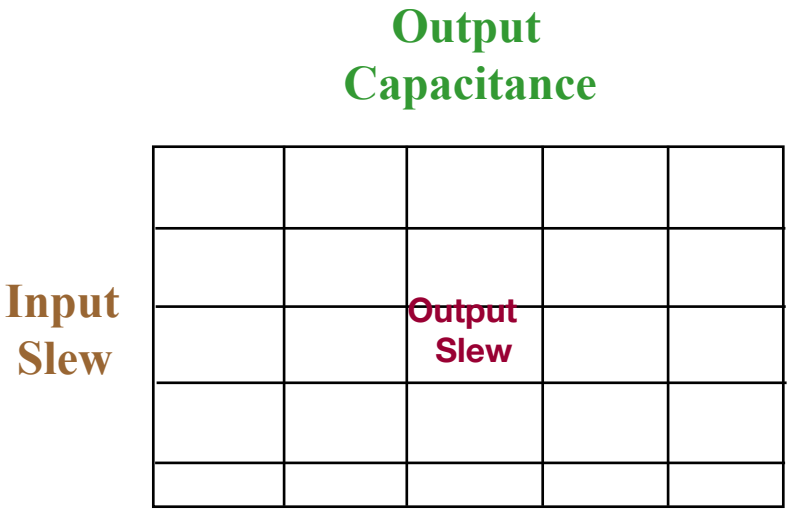


Non-Linear Delay Models (NLDM)

- $D_G = f(C_L, S_{in})$ and $S_{out} = f(C_L, S_{in})$
- Interpolate between table entries
- Interpolation error is usually below 10% of SPICE



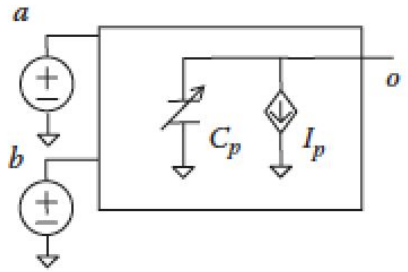
Delay at the gate output



10-90% slew rate

Current Source Models (CSM)

- $I_P = f(V_{in}, V_{out})$ and $Q_P = f(V_{in}, V_{out})$
- Interpolate between table entries
- Independent of input slew and output load



Vout

Vin

Current Drive

Vout

Vin

Output Charge

Liberty (.lib) file (without timing)

```
cell (nand2_4) {
area : 359.1;

pin(A1) {
direction : input;
capacitance : 12.547;
}

pin(B1) {
direction : input;
capacitance : 12.259;
}

pin(O) {
direction : output;
function : "(A1 * B1)'";
}
}
```

```
cell (dfr) {
area : 4819.5;

ff(IQ,IQN) {
next_state : "DATA1";
clocked_on : "CLK2'";
clear : "RST3'";
}

pin(DATA1) {
direction : input;
capacitance : 51.289;
}

pin(CLK2) {
direction : input;
capacitance : 52.305;
}

pin(RST3) {
direction : input;
capacitance : 28.602;
}

pin(Q) {
direction : output;
…
```

Liberty file timing

```
lu_table_template(t4x3) {
variable_1: total_output_net_capacitance ;
variable_2: input_net_transistion ;
index_1 {“5,  20, 60, 200”} ;
index_2 {“0.01,  0.1,  2.0”} ;
}
```

```
pin(O) {
direction : output;
function : "A1 * B1”;
timing() {
cell_rise(t4x3){
values("0.740,0.755,0.768”,
"0.803,0.823,0.838”,
"0.918,0.939,0.954,0.967”,
“1.439,1.497,1.554,1.610”);
}

cell_fall(t4x3) {values(....)};
rise_transistion(t4x3) {values(....)};
fall_transistion(t4x3) {values(....)};
related_pin: “A1” ;

}

timing () {  cell_rise.... cell_fall..
rise_transistion... fall_transition…
related_pin: “B1” ;
}
```

Liberty file sequential timing

```
pin(DATA1) {
direction : input;
capacitance : 51.289;
timing () {
related_pin : “CLK2”;
timing_type: setup_falling;
rise_constraint(dff3x3) {
values("0.199,0.229,0.242”,“0.252,0.261,0.268”,“0.275,0.282,0.289”);
}

fall_constraint(dff3x3) {
values("0.120,0.183,0.212”,“0.233,0.249,0.262”,“0.274,0.284,0.294”);
}
}

timing () {
related_pin : “CLK2”;
timing_type: setup_falling;
rise_constraint(dff3x3) { values(...); }
fall_constraint(dff3x3) { values(...); }
}
}
```

Next Lecture

- See Static Timing Analysis