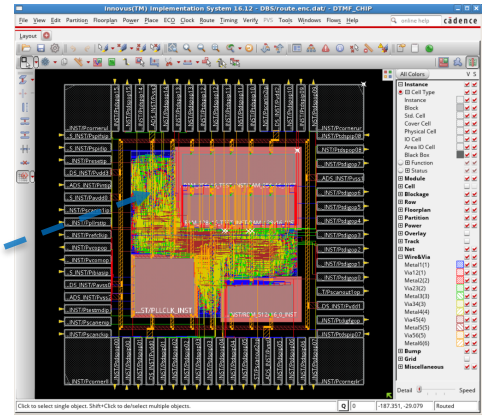
IEEE SSCS Standard-Cell Library

Tim Edwards

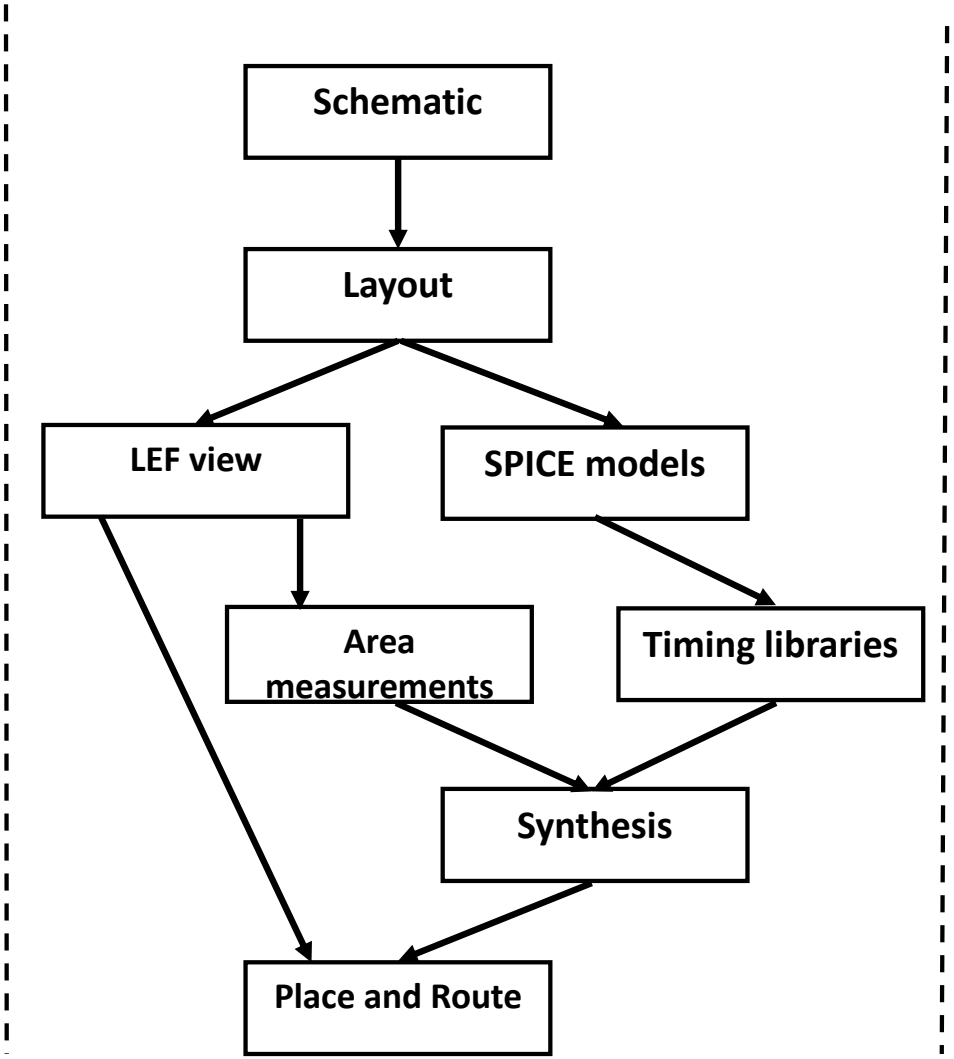
Standard-Cell or SoC Design

- Using pre-made layouts and stitching them together via software.
- Standard-cells are often the same height (although they do not have to be).
- This is often called Application-Specific Integrated Circuit (ASIC) or System on Chip (SoC) design as its application oriented.
- They are designed on a grid, so that all contacts (cuts) and routes (wires) can connect through these grids.
- HDL simulation is key and like the same concept in custom-based designs!
  - Verify everything!
  - Tools are only good as much as you can test things!

standard cells



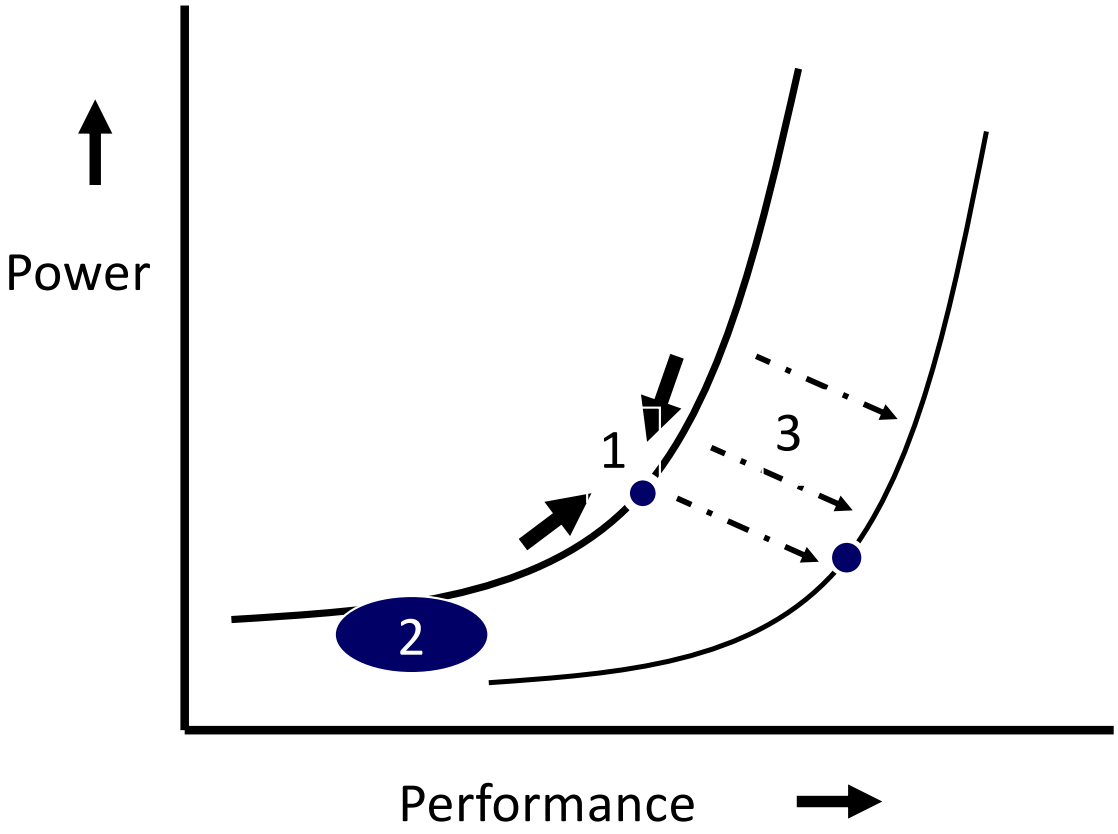
System on Chip (SoC) Design Flow



IC Chip Design Flow	Functions/Tasks
Analysis, Spec., Plan	Cross Functional Team: Marketing, Engineering, and Manufacturing
Signal/Behavioral Design	-Analog: Circuitry design and behaviorl modeling -Digital: Behavioral design with HDL and synthesis
Physical Design	-Custom layout: Polygon and Cell design, Place, Route, DRC -ASIC layout: Synthesis, Timing analysis, Floor planning, Optimization, Route -Block/Chip assembly -Chip finishing
Design Verification	-Physical design verification (DRC, ERC, LVS) -Signal design verification (RC Extraction, Re-simulation for signal integrity analysis.)
Pattern/Mask Generation	Generate masks for IC fabrication
IC Fabrication	Fabricate IC wafer in semiconductor manufacturing fab.
Test, Package, Test	Intial test, Chip Packaging, Chip test

Note: **IC**: Integrated Circuit, **HDL**: Hardware Description Language
DRC: Design Rule Check, **ASIC**: Application-Specific Integrated Circuit
ERC: Electrical Rule Check, **RC**: Resistance and Capacitance
LVS: Layout vs. Schematic

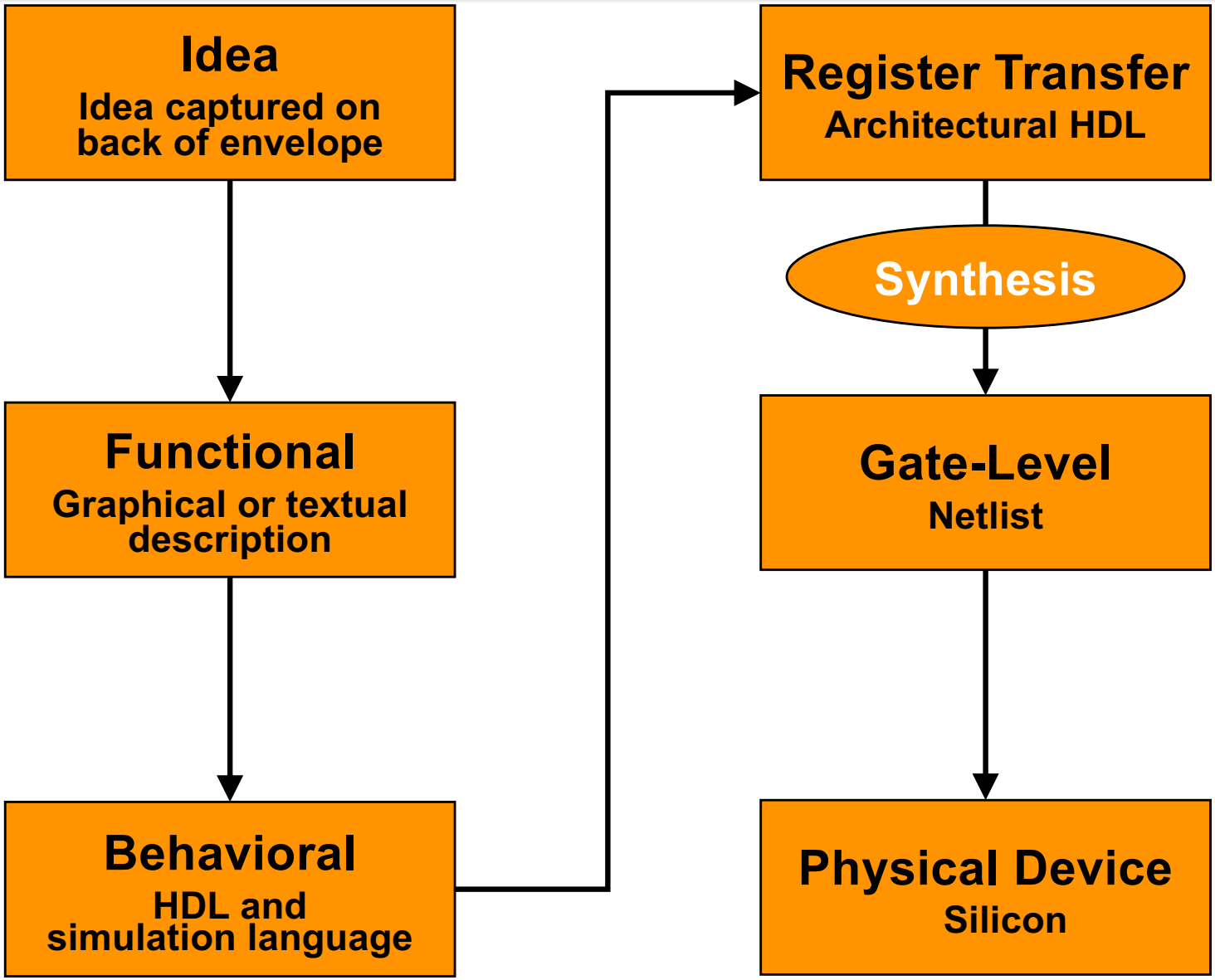
Design Tradeoffs: Power vs. Performance



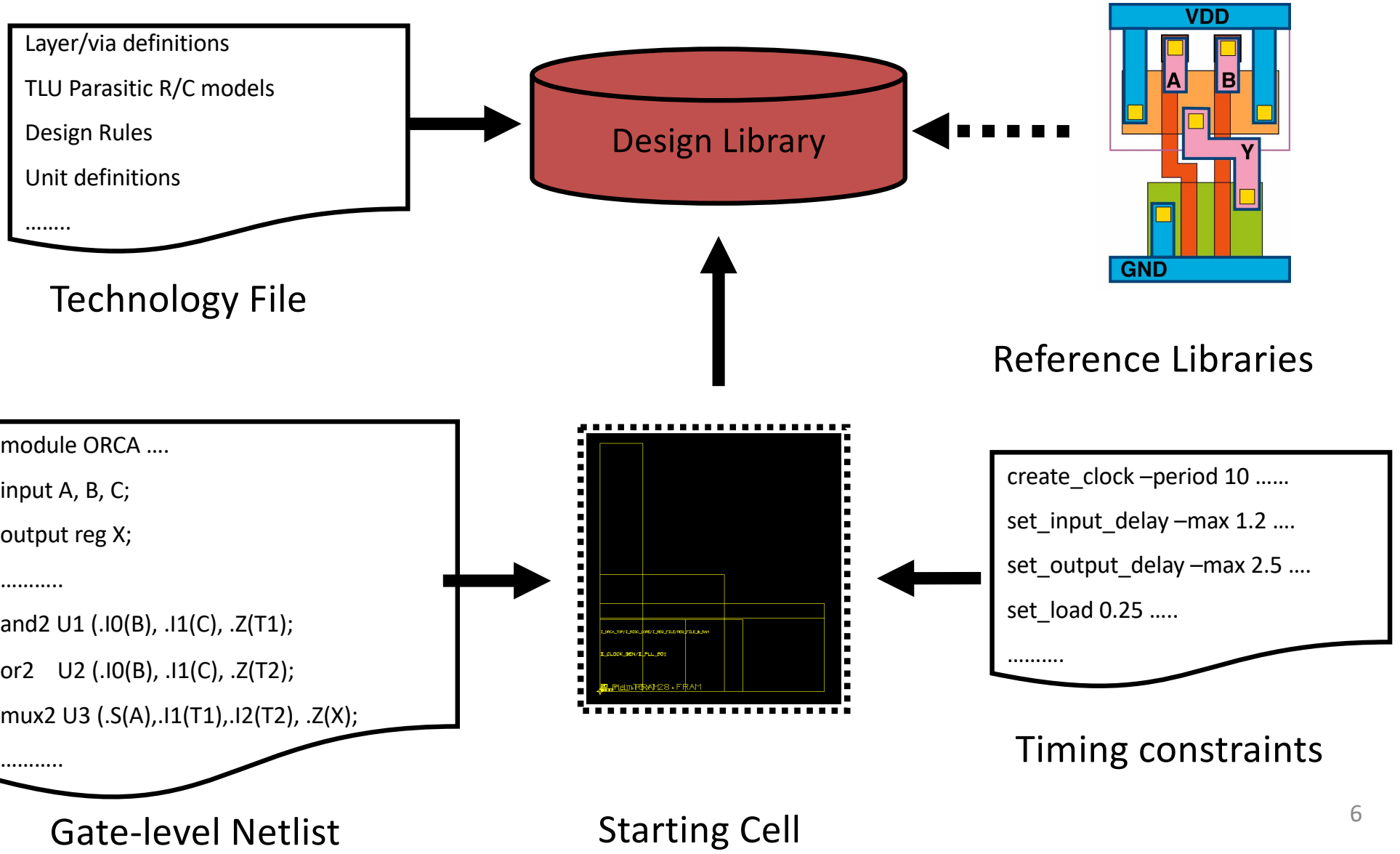
1. Move to More Energy Efficient Operating Point
   - More Energy Efficient w/ Custom
2. Trade Performance for Power
   - Larger Range w/ Custom
3. Move to Different Power vs. Performance Curve
   - More Architectural Choice with Custom

Power, Performance and Area (PPA)

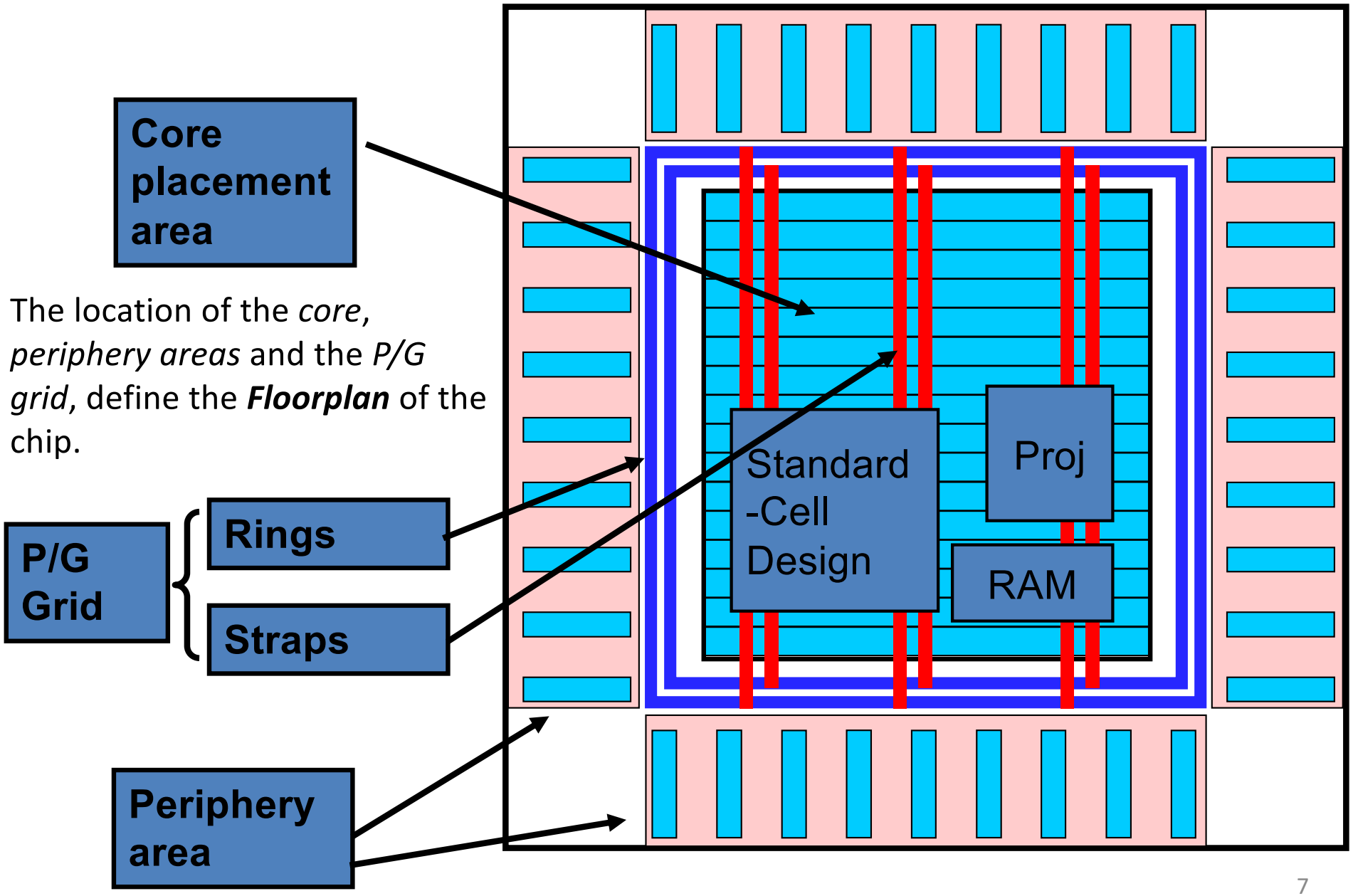
Levels of Abstraction



Design Library: Container of the design



System on Chip (SoC) Design



Floorplanning

- System on Chip (SoC) systems must use proper floorplanning of wires or interconnect to allow current to be divided correctly against multiple inputs.
- Several tools will have ways to measure or extract values for each wire and transistor.
  - This is called Parasitic Extraction or PeX.
- Much of this process is typically done through the use of Electronic Design Automation or EDA tools.
  - EDA tools are critical to get the design to meet its given constraints.

Stack Defines Metallization

- Contact – connection between metal and poly or diffusion
  - Defined in the simple sense of the word.
- Via – defines any connection between metal and metal
  - Metal1 to Metal2
  - Metal1 to Metal3
- Vias can be done different ways
  - Stacked – connection is over each other and provides elevator shaft down to correct level.
  - Non-Stacked – connections are done via connection that abuts to each other.
- AMI C5N has two different types depending on stack utilization.
  - As stated earlier, we currently utilize stacked vias.
- TSMC SCN4ME_SUBM uses stacked vias!

Ramp + Gate

- Gates have metrics of delay, area and energy/power.
- All of these items are a function of their drawn sizes as well as what they are connected to.
- A given orientation and implementation will contribute a set amount of gate delay.
- The ”shaping” of a circuit or the ramp function also contributes and is typically called the transition delay in System on Chip (SoC) domains.
  - Gates are typically characterized for both custom and standard-cells to give an idea what they would be in a given scenario.
  - For SoC designs, lots of characterization is performed, usually with scripts, to make sure gates have libraries that designers can utilize and assess of in terms of performance.

EDA Tool Usage

- For this competition, the Synopsys Tool sets are planned on being utilized.
- This includes the following for standard-cell creation.
- Standard-Cell Creation (Synopsys)
  - PrimeLib
  - NanoTime
  - Hspice
- Layout
  - Magic
- Schematic
  - Xschem
- Validation of Layout
  - Netgen
- Synthesis
  - Design Compiler (Synopsys)
- Place & Route
  - Fusion Compiler (Synopsys) and/or ICC2

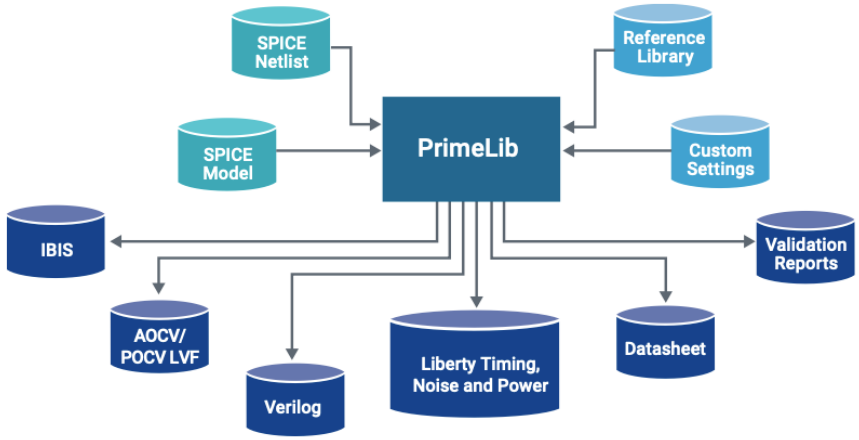


Figure 2: PrimeLib input and outputs

[Synopsys]

Standard Cell Design Philosophy

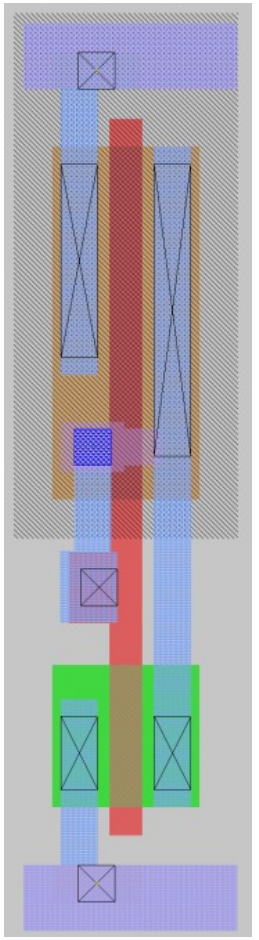
- Standard cells sit at the base of all good designs. Without good standard cells you cannot make a good chip!
- The main design goals in a standard cell kit are to minimize area and power consumption while maximizing speed.
  - Each of these comes at the cost of the others: you must sacrifice some to meet the others.

Standard Cell Design Philosophy (Continued)

- Modern standard cell kits are commonly provided with multiple versions of each cell, some optimized for power while others for speed.
  - Usually done by changing the threshold voltage of the cells
  - These can be intermixed within a single design
  - Makes it easy to trade power for speed within the scope of a single design
- The main goal is providing the ability to implement SoC designs as well as provide files that others can use and hopefully contribute back to the project.

Layout

- For standard cell layout, we use Magic, an open-source tool.
- It is preferred over commercial tools for our research, even involving nodes all the way down to 14nm.
- http://opencircuitdesign.com/magic/



Layout (Continued)

- Magic was originally designed with SCMOS in mind but can be used at any technology by editing a tech file.
- The Magic tech file for the 130nm Open PDK will be included as part of the general release, so anyone can start creating layouts from day one.

SPICE Models

- To obtain SPICE models we begin with our magic layout. Magic makes it to extract a baseline spice model, containing all devices and dimensions.
- We then take this baseline spice model along with the cell's GDS and perform parasitic extraction (PeX).
- PeX consists of accurate extraction of the resistors, capacitors, and parasitic devices present within a design.
- We currently use Calibre to perform our PeX.
- This gives us a much better model of how the cells will behave.

```
* File: INVX1.spice
* Created: Sun Sep 20 09:38:05 2020
* Program "Calibre xRC"
* Version "v2020.2_35.23"
*
.include "INVX1.pex.spice"
.subckt INVX1  GND VDD A Y
*
* Y Y
* A A
* VDD   VDD
* GND   GND
MM1000 N_Y_M1000_d N_A_M1000_g N_GND_M1000_s GND NSHORT L=0.15 W=1 AD=0.265
+ AS=0.265 PD=2.53 PS=2.53 NRD=0 NRS=0 M=1 R=6.66667 SA=75000.2 SB=75000.2
+ A=0.15 P=2.3 MULT=1
MM1001 N_Y_M1001_d N_A_M1001_g N_VDD_M1001_s N_VDD_M1001_b PSHORT L=0.15 W=3
+ AD=0.795 AS=0.795 PD=6.53 PS=6.53 NRD=0 NRS=0 M=1 R=20 SA=75000.2 SB=75000.2
+ A=0.45 P=6.3 MULT=1
*
.include "INVX1.pxi.spice"
*
.ends
*
```

SPICE Simulation

- SPICE can be used to accurately simulate our cells in a variety of condition.
- The SkyWater 130nm Open PDK has been hard at work over the last few weeks to release a full set of SPICE models for all devices that can be fabricated in this technology node.

LEF View

- LEF/DEF is a globally-used format for representing design rules, abstract information about standard cells, and circuit layout.
- The format is entirely text-based and used by most open-source EDA tools.
- LEF (library exchange format) provides abstract information about cells to be used during place and route, such as pin locations, routing obstructions, and antenna restrictions.
- One can extract LEF views from all any design for easy use.

```
MACRO ADDFX1
  CLASS CORE ;
  ORIGIN 0 0 ;
  FOREIGN ADDFX1 0 0 ;
  SIZE 7.04 BY 6.66 ;
  SYMMETRY X Y ;
  SITE 18T ;
  PIN A
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER met1 ;
        RECT 5.01 1.735 5.3 1.965 ;
        RECT 0.34 1.765 5.3 1.935 ;
        RECT 2.35 1.735 2.64 1.965 ;
        RECT 0.34 1.735 0.63 1.965 ;
    END
  END A
  PIN B
    DIRECTION INPUT ;
    USE SIGNAL ;
    PORT
      LAYER met1 ;
        RECT 4.12 2.475 4.41 2.705 ;
        RECT 0.34 2.51 4.41 2.675 ;
        RECT 4.06 2.505 4.41 2.675 ;
```

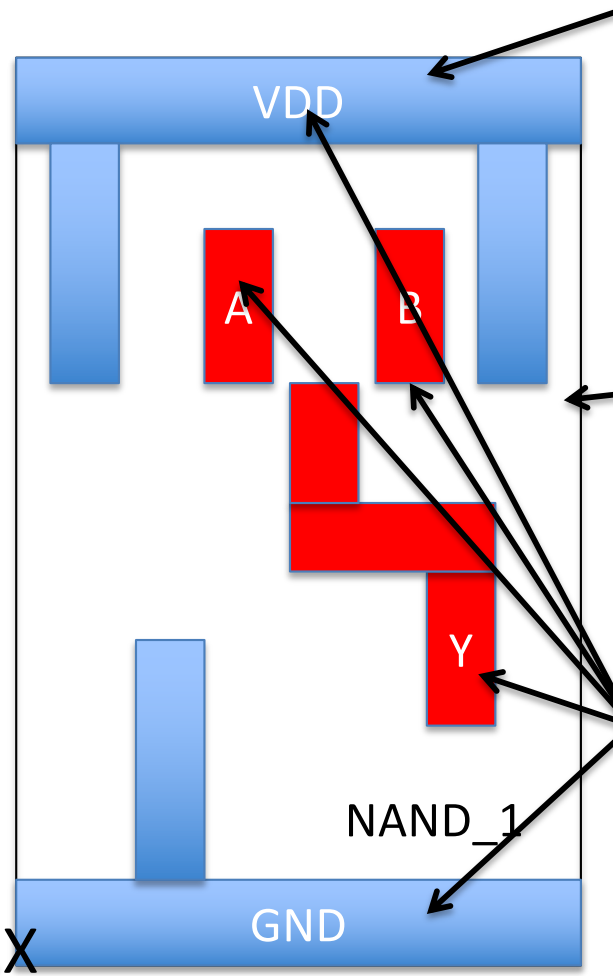
What is Layout Exchange Format (LEF) File?

Technology Section

```
LAYER m1
      TYPE ROUTING ;
      WIDTH 0.50 ;
END m1
LAYER via
      TYPE cut ;
END via
```

Definition of Std Cell/Macro

```
MACRO NAND_1
      FOREIGN NAND_1 0.00 0.00
      ORIGIN 0.00 by 0.00 ;
      SIZE 4.5 by 12.0 ;
      SITE core ;
      PIN A
            DIRECTION input ;
            PORT
            LAYER m1 ;
            RECT 6.4 10.0 6.8 10.4 ;
      END
      PIN Y
            …..
      OBS
            LAYER via ;
                  RECT …
                  RECT …
END NAND_1
```



Dimension or "bounding box"

Symmetry (X, Y, or 90 degrees)

Pins
- Direction
- Layer
- Form

Reference point (typical)

Technology and Std. Cell/Macro can be separate files or the same!

Timing Libraries

- The Liberty Timing Format (LIB) is a file format for providing timing models containing cell delays and transition times, setup and hold time requirements, logical cell function, and optionally area.
- The format is entirely text-based and used by most open-source EDA tools.
- All libraries are fully characterized and include scripts to help others in expanding the library, if needed
- You should provide both CCS and ECSM liberty files as well as back-annotated VHDL and Verilog files.

Delay Information

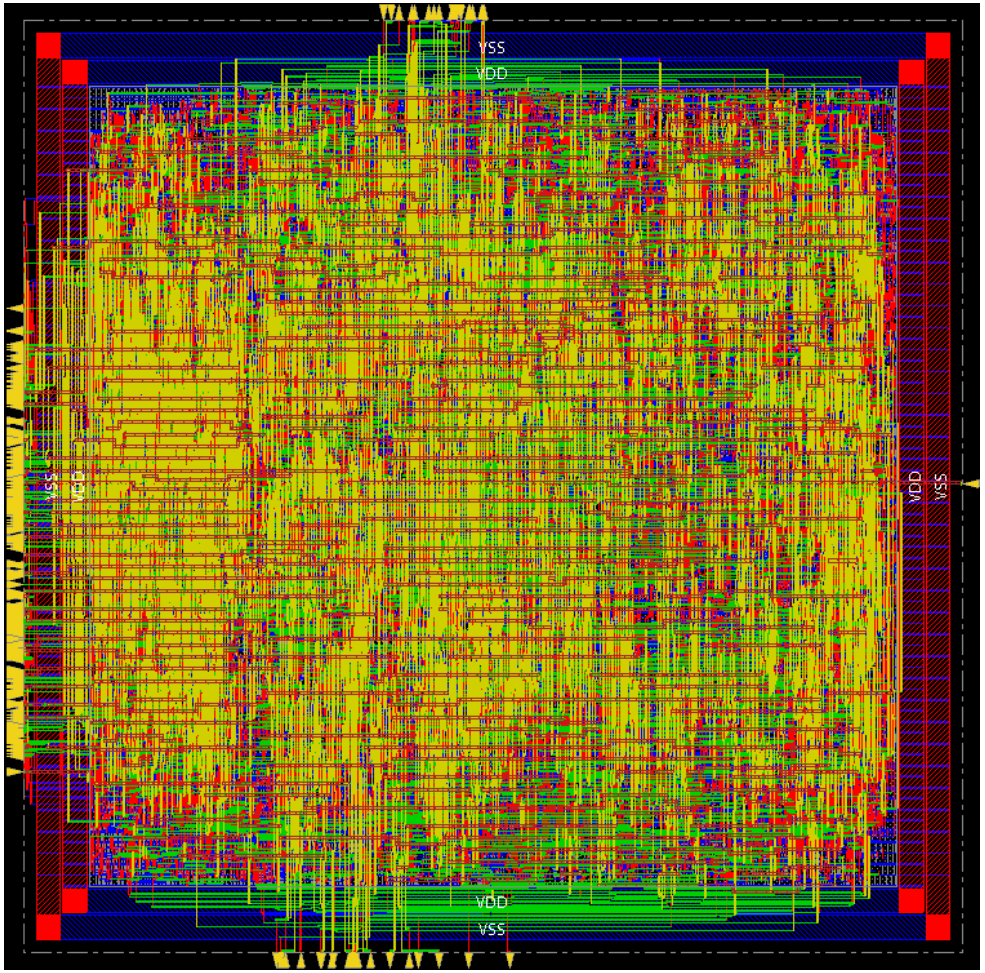
Delay(ns) to Y rising :

Cell Name	Timing Arc(Dir)	Delay(ns)		
		First	Mid	Last
AND2X1	A->Y (RR)	0.07489	0.20503	0.65648
	B->Y (RR)	0.07985	0.20601	0.65841
AND2X2	A->Y (RR)	0.08542	0.20967	0.70273
	B->Y (RR)	0.09034	0.20929	0.69907
AND2X4	A->Y (RR)	0.11595	0.24205	0.79319
	B->Y (RR)	0.12084	0.24136	0.78209
AND2X6	A->Y (RR)	0.14558	0.27412	0.86364
	B->Y (RR)	0.15042	0.27399	0.84680
AND2X8	A->Y (RR)	0.17518	0.30612	0.91873
	B->Y (RR)	0.18007	0.30631	0.89836
AND2XL	A->Y (RR)	0.08460	0.22592	0.66372
	B->Y (RR)	0.08984	0.22668	0.66538

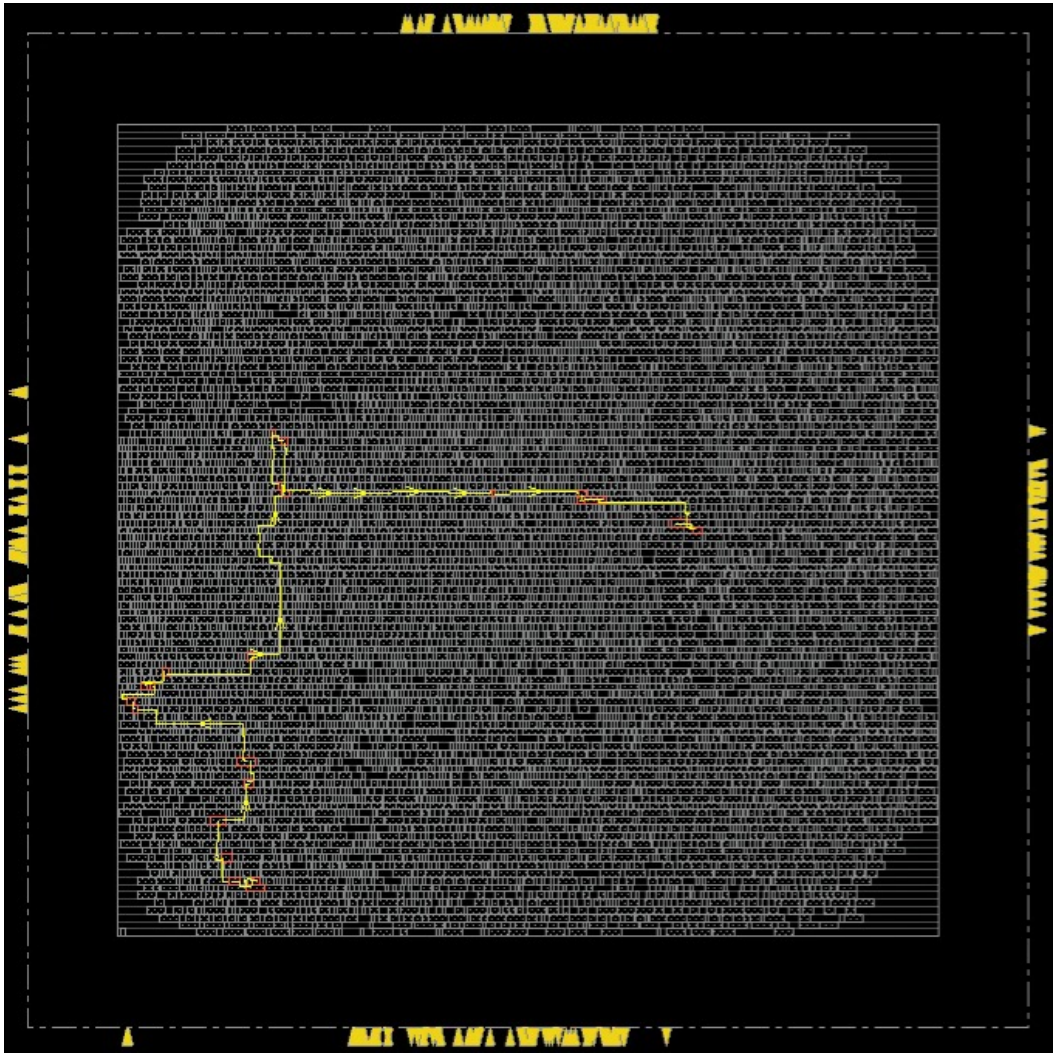
Synthesis and Place and Route

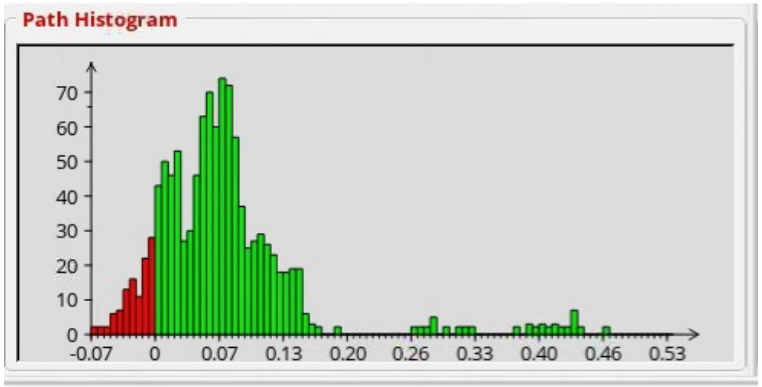
- Once cells are fully characterized, the best way to test them is by using them to implement an HDL design.
- Liberty timing files and LEF view provide only an abstract description of timing and physical layout, respectively.
  - Cells behavior may change once implemented
- Synthesis is commonly used in literature to compare results across different designs
- Place-and-route results depend on a physical layout, so they are inherently more accurate but can also show greater variance with regards to initial parameters.

Place and Route Output Example



Place and Route Output Example (Continued)



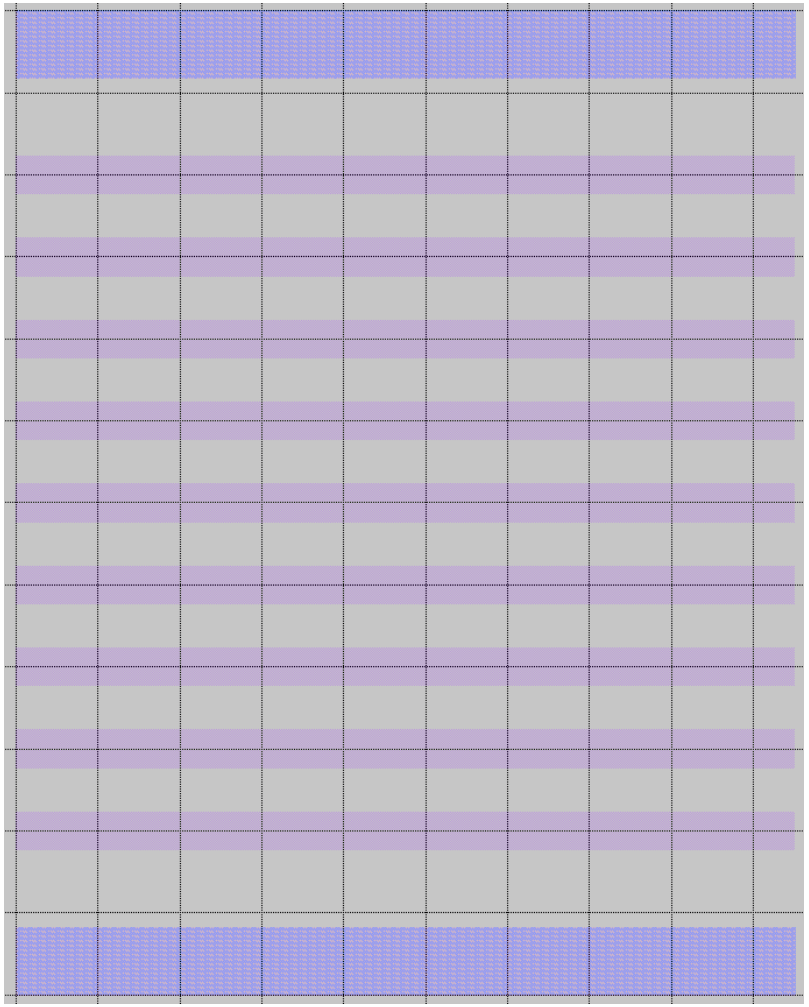


Standard Cell Tracks

- One key design choice for any library is the height of the cells.
  - Taller cells have transistors with larger widths, resulting in higher speeds.
  - Taller cells also have more room available for routing wires, meaning lowering congestion.
- Cell height is measured in terms of routing tracks of the lowest non-interconnect metal layer.
  - A track is the smallest section of space that can accommodate a drawn wire.
  - Typically defined in terms of the width and spacing design rules for the respective layer.
- Typically, 9-track (9T) cells are standard.
  - 12-track (12T) cells are often provided for high-speed designs.
  - Smaller cells, such as 7T are used for ultra-high density designs.

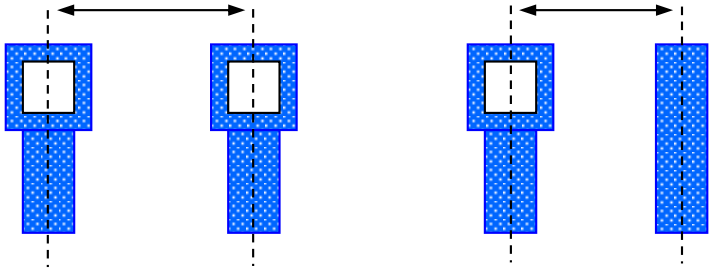
Standard Cell Tracks (Continued)

- An example of a 12-track setup can be seen to the right.
  - Note that although being 12 tracks tall, the cell can only accommodate 9 routed wires
  - Power rails take space on these tracks as well.
- Cells of different heights provide a simple solution for trading off speed vs area.
  - But they cannot typically be mixed within the same design!

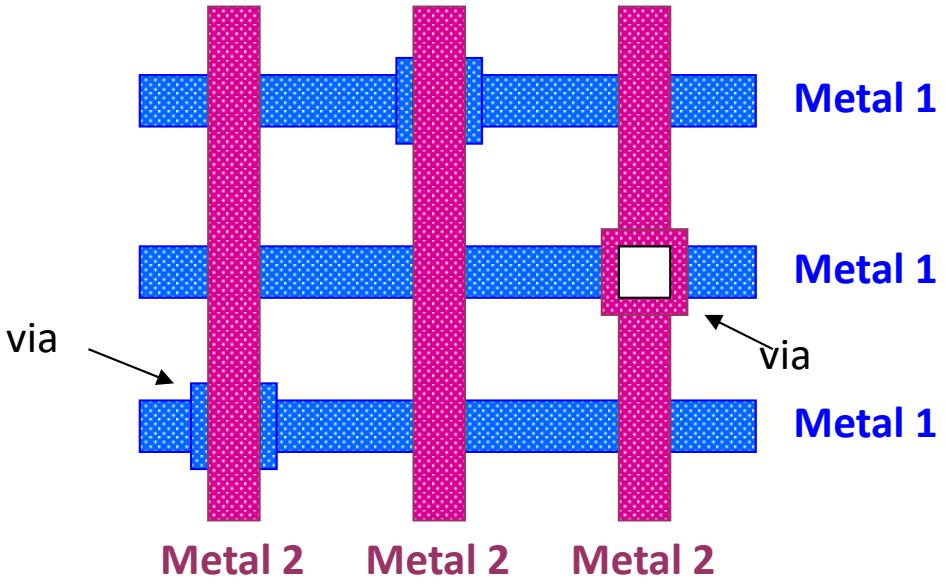


Track Pitch and Direction

- The pitch at which wires are spaced defines the size of a routing track.
- Can be defined either via-to-via or line-to-via.
- To simplify Place and Route, metal layers are used in alternating directions.
  - Odd layers for horizontal routing and even layers for vertical routing (HVH).
  - On more advanced tech nodes, it is common to see both of the first two metal layers run horizontally.

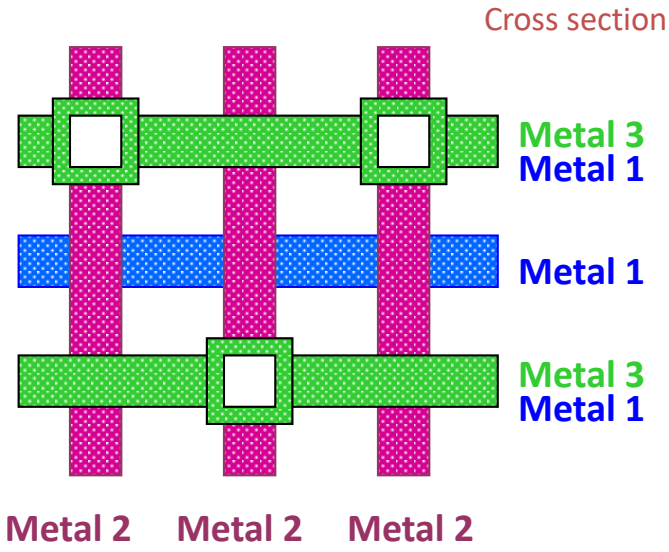


Line-to-via because it is the minimum pitch in which you can place a via next to a route track

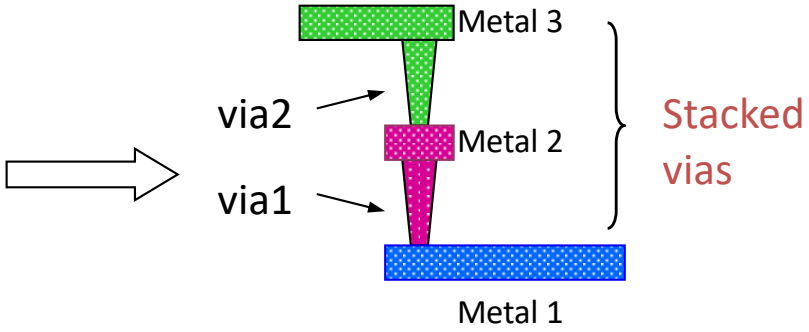


Routing Grid Definitions

- Different layers with the same routing direction will have different thickness and spacing: different pitch.
- If M3:M1 have a track pitch ratio such as 11:8 both horizontal tracks would be on top of each other seldomly making it difficult or even impossible to create an M3-to-M1 contact.
- Importance of using simple ratios like 1:1, 1:2, 2:3, 3:4 between adjacent same direction routing tracks.
- It is desirable for the tracks to be aligned to allow stacked vias.

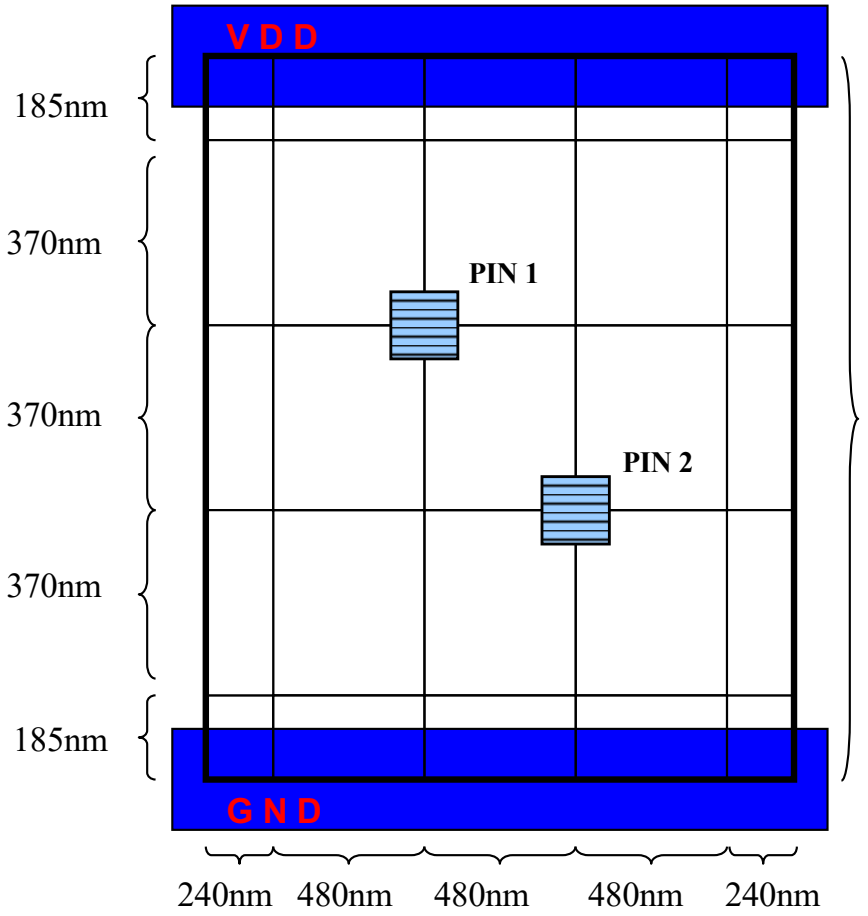


Example shows **1:2** M3-to-M1 ratio.



Layout Creation and Routing Grid

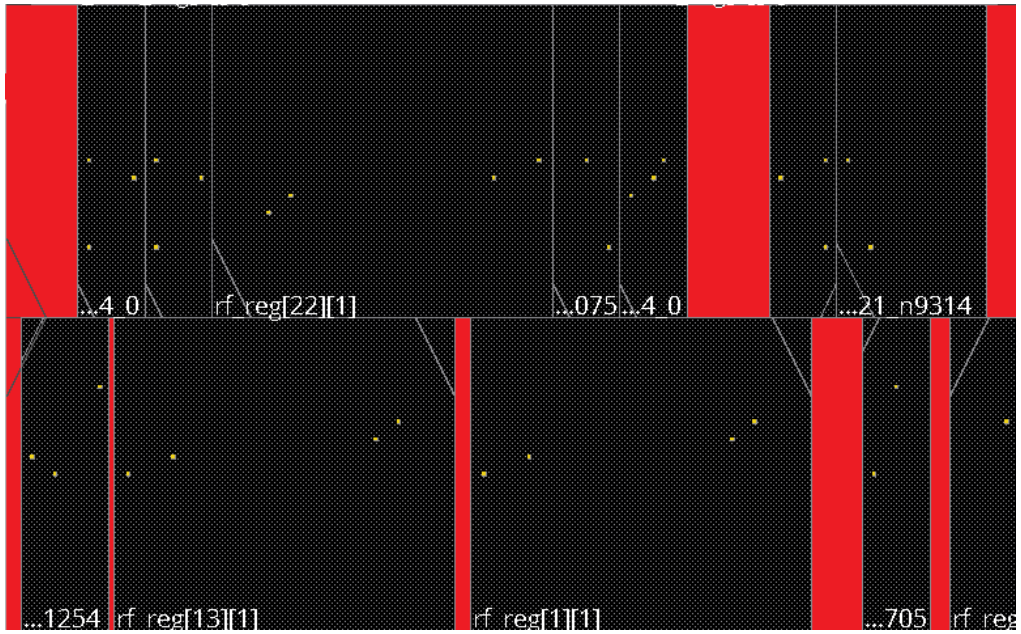
- Pins should placed on the grid (reduce routing efforts).
- Staggered if possible (reachable by both horizontal and vertical layers).
- Power signals routed through abutment (increase density).



Power rails designed to allow abutment.

Standard Cell Dimensions

- Standard cell heights must be the same across the entire library.
  - Gaps between cells are not allowed and must be filled with “filler” cells.
- All standard cells widths must be integer multiples of our smallest filler cells.
  - The smaller this common divisor is, the less wasted space.



Dealing with Local Interconnect

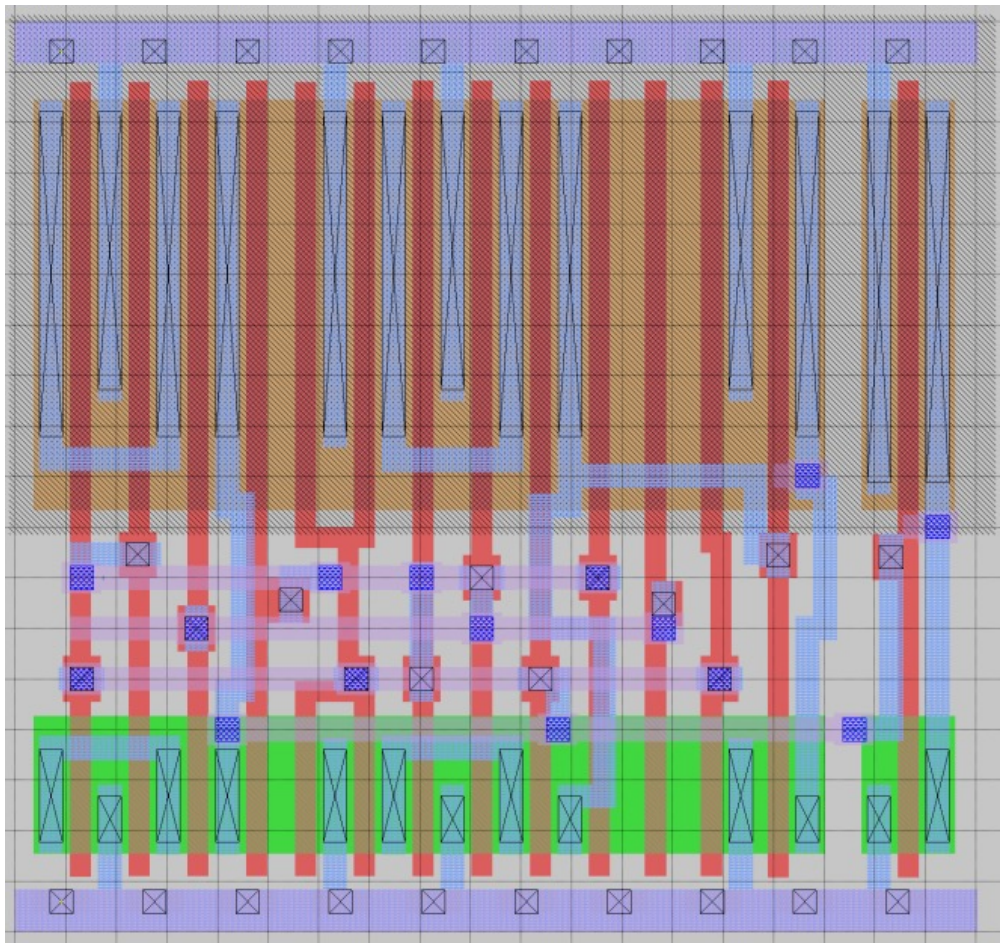
- The design rule manual suggest that local interconnect wires should be restricted to an aspect ratio of 10:1.
- As mentioned previously, this TiN layer has much more resistivity than the aluminum layers.
- The decision was made to provide all cell in/out pins on metal1 and prevent routing on li1.
- We must take care that our in/out pins do not block metal1 routing tracks

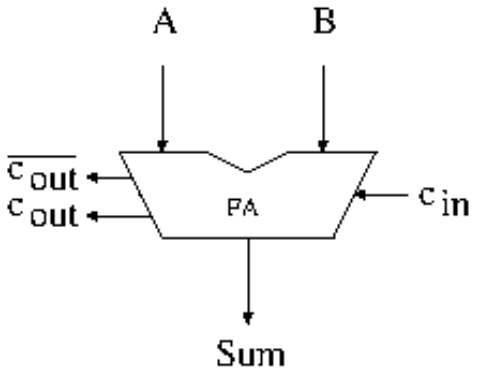
```
# NOTE:
# The use of li1 as a routing layer is commented
# out and replaced with a definition of li1 as a
# non-routing layer. This is done to ensure good
# results regardless of tool used.
#
# If li1 is enabled as a routing layer take note
# of its resistance compared to that of metal1.

LAYER li1
  TYPE MASTERSLICE ;
END li1

#LAYER li1
#  TYPE ROUTING ;
#  DIRECTION VERTICAL ;
```

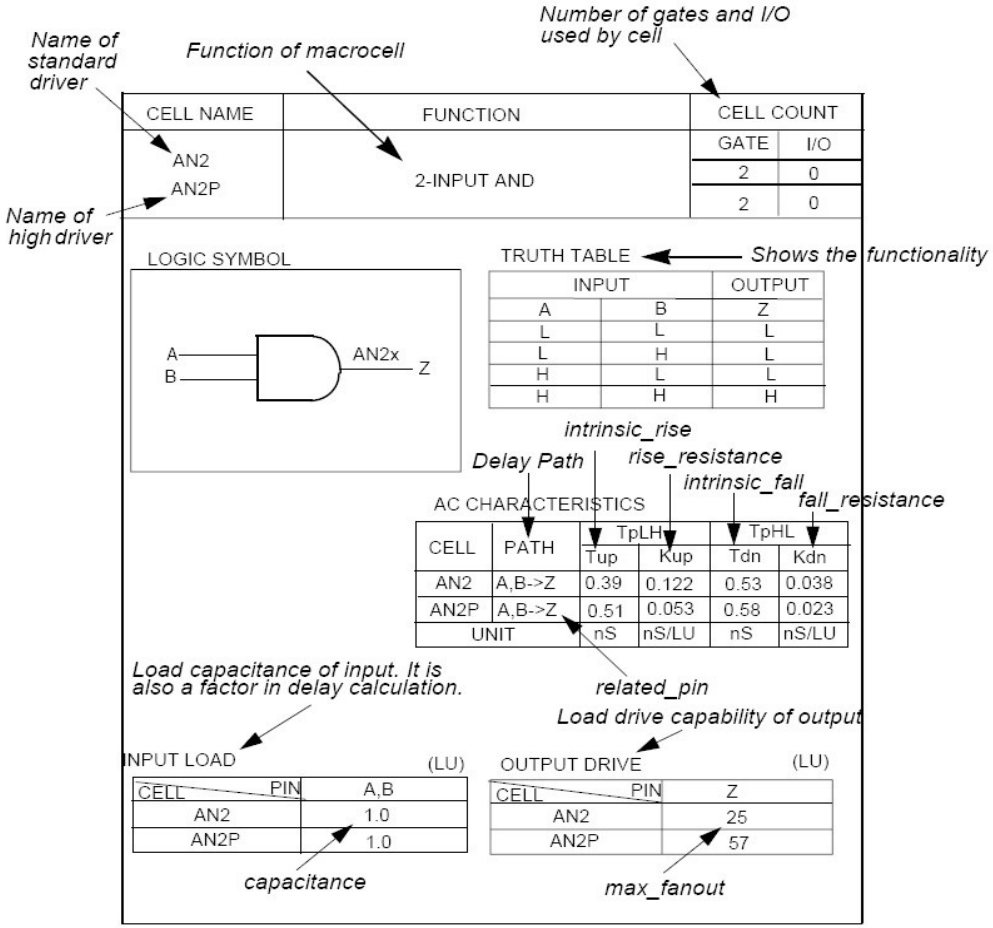
Pins Attached to metal1 Routing Tracks





- There are three inputs and three outputs.
- That means six total pins, each on a separate metal1 track.
- Next, we will have a clearer view!

Datasheet View of AND2

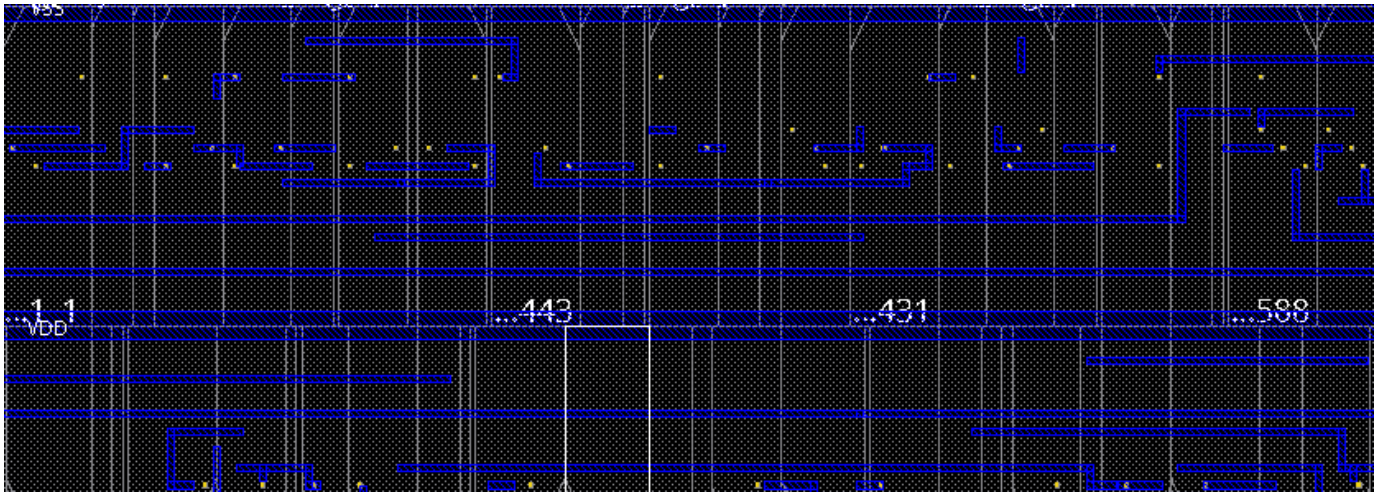


- Correlation between datasheet and .lib representation of a 2 input AND gate

```
pin(A,B) {
  direction : input ;
  capacitance : 1 ;
}
pin(Z) {
  direction : output ;
  function : "A * B" ;
  timing() {
    intrinsic_rise : 0.39 ;
    intrinsic_fall : 0.53 ;
    rise_resistance : 0.122 ;
    fall_resistance : 0.038 ;
    related_pin : "A B" ;
```

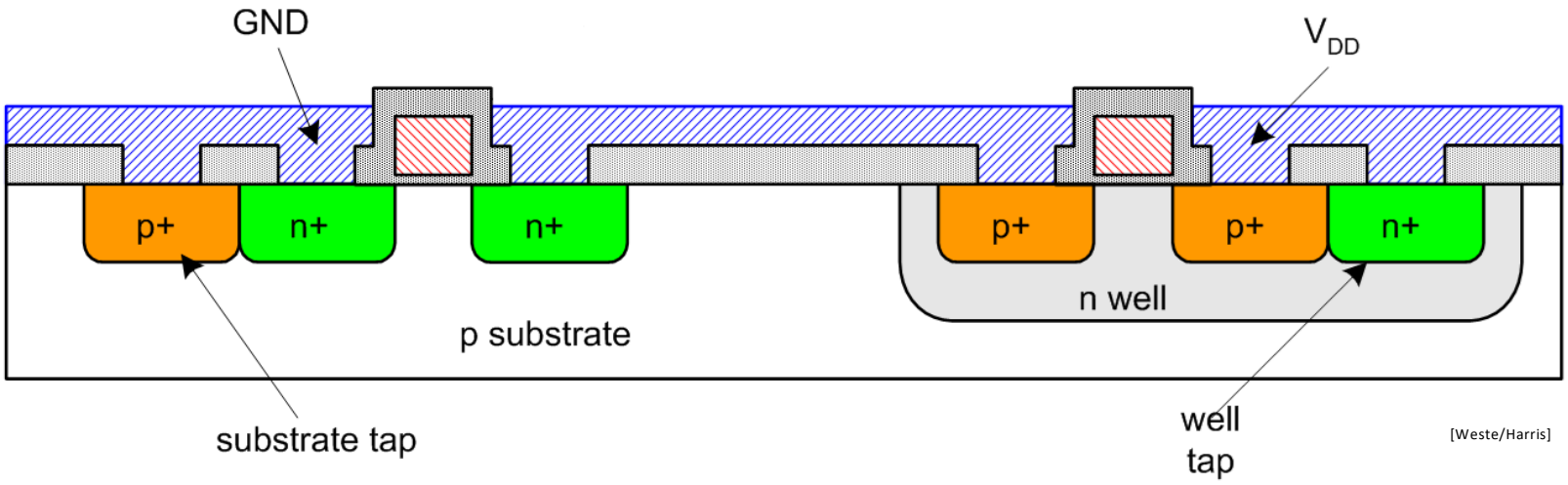
Metal1 Inter-Cell Routing

- To allow for efficient inter-cell routing, metal1 usage must be kept to a minimum within cells.
- We complement this effort in our cells by keeping all pins limited to the same six metal1 tracks throughout our library.
  - This guarantees that all other tracks are available for top-level routing.
  - Less routing congestion.
  - An example of the tools able to draw long, straight wires, shown here:



Well and Substrate Taps

- The pre-existing SkyWater standard cells do not have well and substrate taps built into their power rails.
- Instead, they offer specialized tap cells that are meant to be placed alongside the standard cells in a design.
- Our cells offer thicker power rails with built-in well and substrate taps.
- This provides better connectivity with the body of the transistors, minimizing the impact of the body effect.

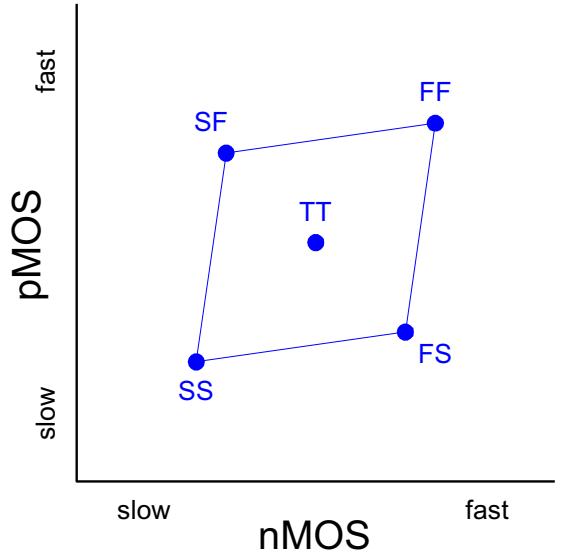


So What?

- So what if transistors are not ideal?
  - They still behave like switches.
- But these effects matter for…
  - Supply voltage choice
  - Logical effort
  - Quiescent power consumption
  - Pass transistors
  - Temperature of operation

Parameter Variation

- Transistors have uncertainty in parameters
  - Process: L_{eff}, V_t, t_{ox} of nMOS and pMOS
  - Vary around typical (T) values
- Fast (F)
  - L_{eff}: short
  - V_t: low
  - t_{ox}: thin
- Slow (S): opposite
- Not all parameters are independent for nMOS and pMOS



[Weste/Harris]

Environmental Variation

- V_{DD} and T also vary in time and space
- Fast:
  - V_{DD}: high
  - T: low

Corner	Voltage	Temperature
F	1.98	0 C
T	1.8	70 C
S	1.62	125 C

Process Corners

- Process corners describe worst case variations
  - If a design works in all corners, it will probably work for any variation.
- Describe corner with four letters (T, F, S)
  - nMOS speed
  - pMOS speed
  - Voltage
  - Temperature

Important Corners

- Some critical simulation corners include

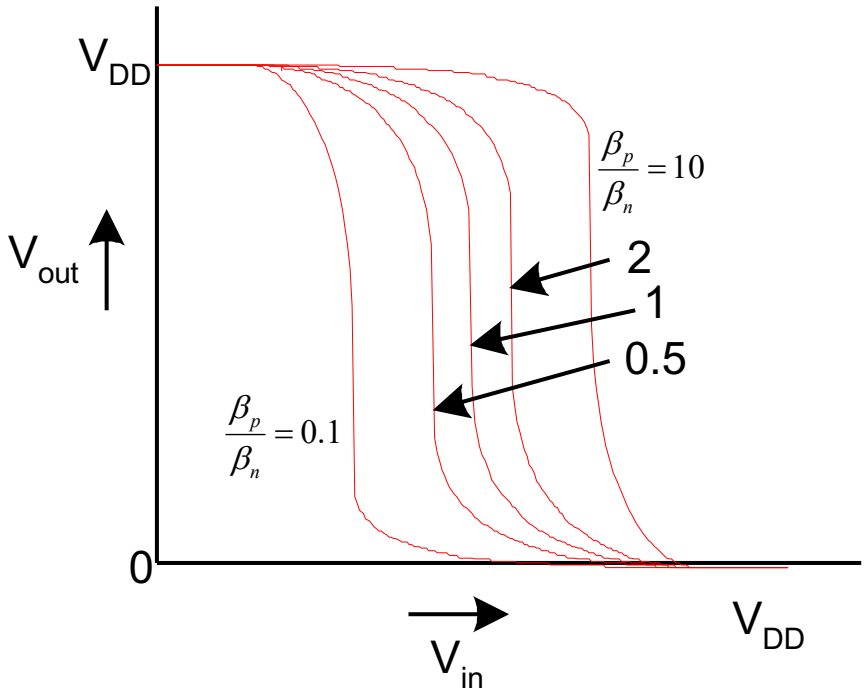
Purpose	nMOS	pMOS	V_{DD}	Temp
Cycle time				
Power				
Subthreshold leakage				

Sizing Considerations

- Not everything has to be symmetrical and the key to innovation is thinking outside the box.
- Never give up on any idea but be practical on ideas.
- Science always dictates what can and will not happen – remember this!
- Let's examine some impacts on the sizing of transistors and what this means for the overall Width and Length of a transistor.

Beta Ratio (Review)

- If $\beta_p / \beta_n \neq 1$, switching point will move from $V_{DD}/2$
- Called *skewed* gate
- Other gates: collapse into equivalent inverter



V_M is never a good point to use for sizing → use SPICE

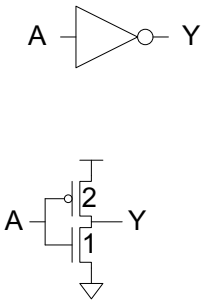
[Weste/Harris]

Compound Gates

- Logical Effort of compound gates
- Compound gates are typically given in SoC designs for multiple drive strengths or different Ws

unit inverter

$Y = \overline{A}$

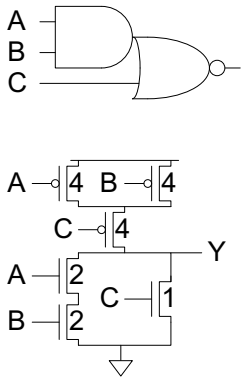


$g_A = 3/3$

$p = 3/3$

AOI21

$Y = \overline{A \bullet B + C}$



$g_A = 6/3$

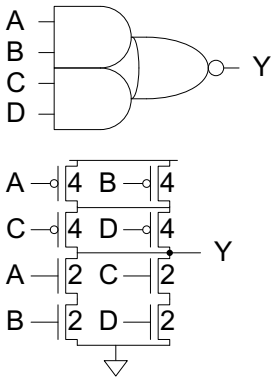
$g_B = 6/3$

$g_C = 5/3$

$p = 7/3$

AOI22

$Y = \overline{A \bullet B + C \bullet D}$



$g_A = 6/3$

$g_B = 6/3$

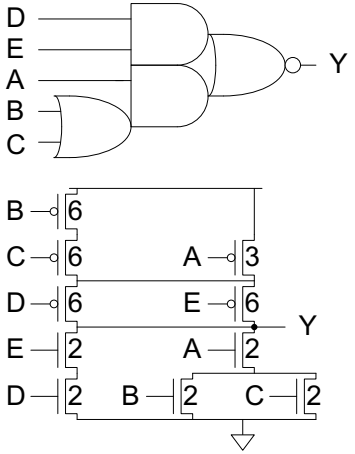
$g_C = 6/3$

$g_D = 6/3$

$p = 12/3$

Complex AOI

$Y = \overline{A \bullet (B + C) + D \bullet E}$



$g_A = 5/3$

$g_B = 8/3$

$g_C = 8/3$

$g_D = 8/3$

$g_E = 8/3$

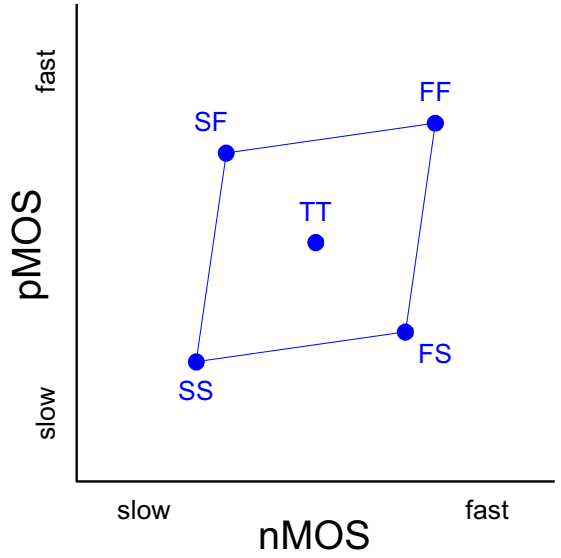
$p = 16/3$

So What?

- So what if transistors are not ideal?
  - They still behave like switches.
- But these effects matter for…
  - Supply voltage choice
  - Logical effort
  - Quiescent power consumption
  - Pass transistors
  - Temperature of operation

Parameter Variation

- Transistors have uncertainty in parameters
  - Process: L_{eff}, V_t, t_{ox} of nMOS and pMOS
  - Vary around typical (T) values
- Fast (F)
  - L_{eff}: short
  - V_t: low
  - t_{ox}: thin
- Slow (S): opposite
- Not all parameters are independent for nMOS and pMOS



[Weste/Harris]

Environmental Variation

- V_{DD} and T also vary in time and space
- Fast:
  - V_{DD}: high
  - T: low

Corner	Voltage	Temperature
F	1.98	0 C
T	1.8	70 C
S	1.62	125 C

Process Corners

- Process corners describe worst case variations
  - If a design works in all corners, it will probably work for any variation.
- Describe corner with four letters (T, F, S)
  - nMOS speed
  - pMOS speed
  - Voltage
  - Temperature

Important Corners

- Some critical simulation corners include

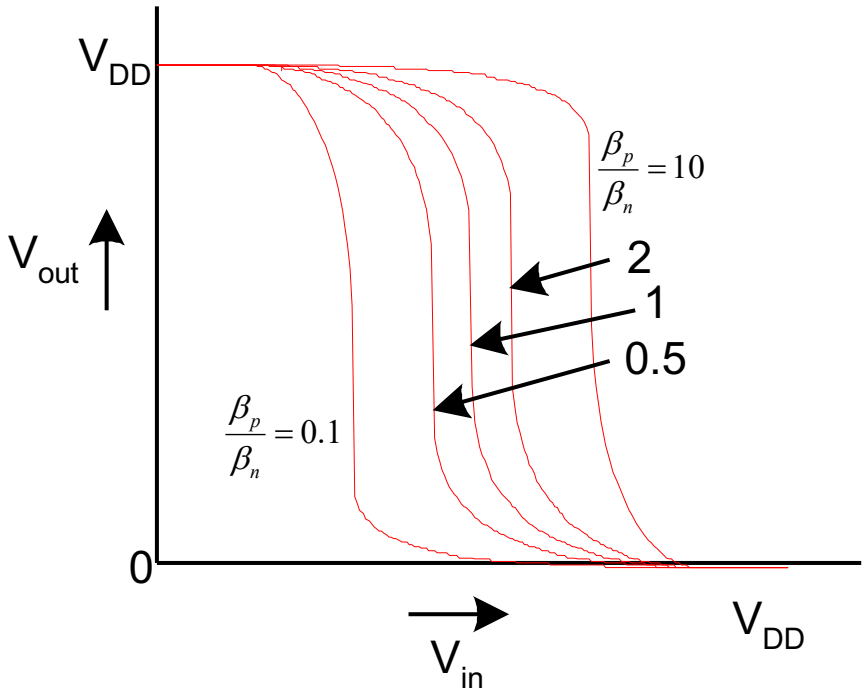
Purpose	nMOS	pMOS	V_{DD}	Temp
Cycle time				
Power				
Subthreshold leakage				

Sizing Considerations

- Not everything has to be symmetrical and the key to innovation is thinking outside the box.
- Never give up on any idea but be practical on ideas.
- Science always dictates what can and will not happen – remember this!
- Let's examine some impacts on the sizing of transistors and what this means for the overall Width and Length of a transistor.

Beta Ratio (Review)

- If $\beta_p / \beta_n \neq 1$, switching point will move from $V_{DD}/2$
- Called *skewed* gate
- Other gates: collapse into equivalent inverter



V_M is never a good point to use for sizing → use SPICE

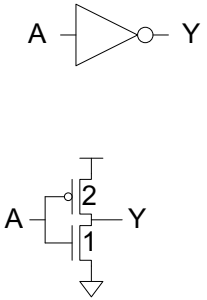
[Weste/Harris]

Compound Gates

- Logical Effort of compound gates
- Compound gates are typically given in SoC designs for multiple drive strengths or different Ws

unit inverter

$Y = \overline{A}$

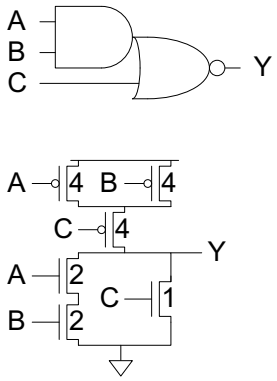


$g_A = 3/3$

$p = 3/3$

AOI21

$Y = \overline{A \bullet B + C}$



$g_A = 6/3$

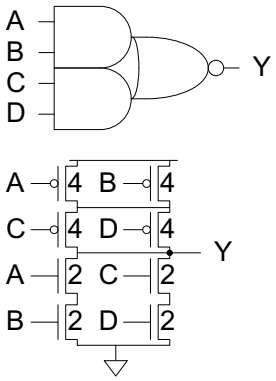
$g_B = 6/3$

$g_C = 5/3$

$p = 7/3$

AOI22

$Y = \overline{A \bullet B + C \bullet D}$



$g_A = 6/3$

$g_B = 6/3$

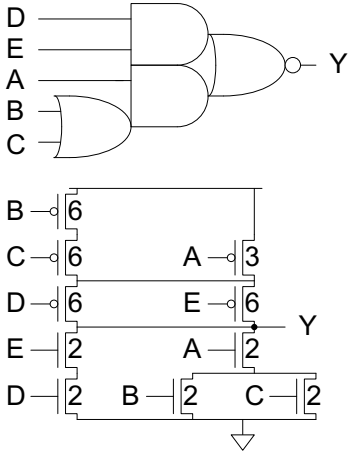
$g_C = 6/3$

$g_D = 6/3$

$p = 12/3$

Complex AOI

$Y = \overline{A \bullet (B + C) + D \bullet E}$



$g_A = 5/3$

$g_B = 8/3$

$g_C = 8/3$

$g_D = 8/3$

$g_E = 8/3$

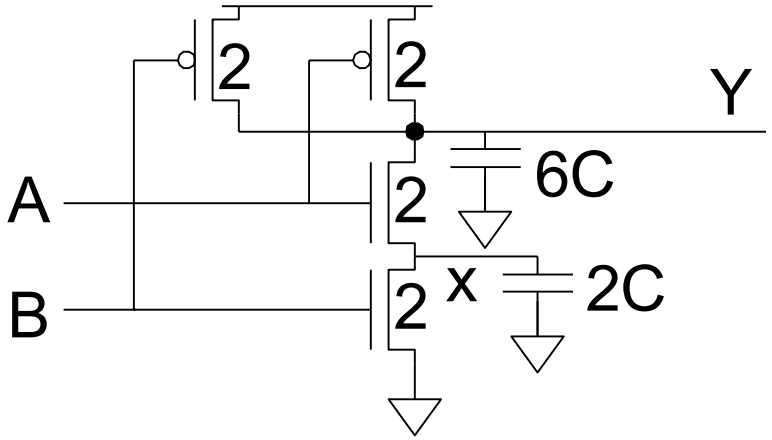
$p = 16/3$

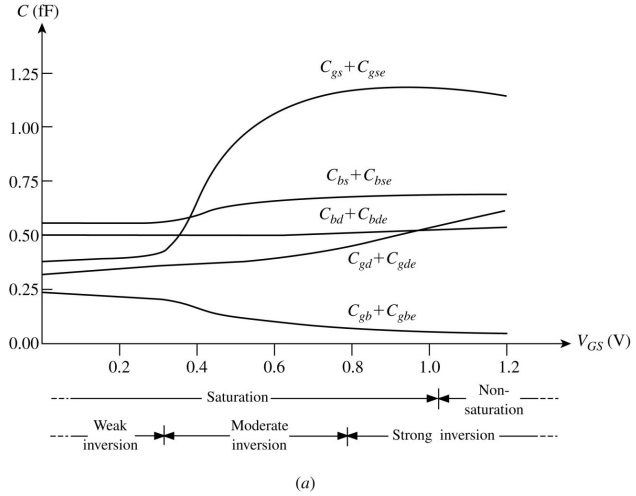
[Sutherland/Sproull/Harris]

Input Order

- Our parasitic delay (P) model was too simple
  - Calculate parasitic delay for Y falling
    - If A arrives latest? 2τ
    - If B arrives latest? 2.33τ

$$C = C_g = C_{ox} \cdot W \cdot L$$



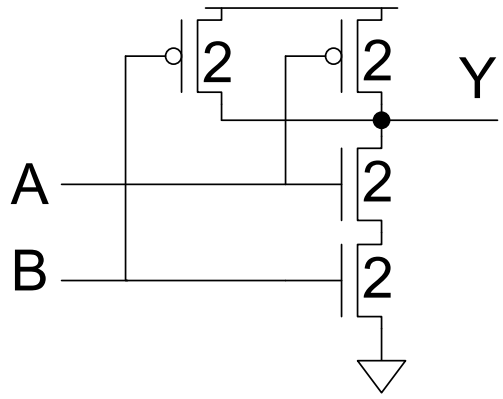


[Weste/Harris]

Inner & Outer Inputs

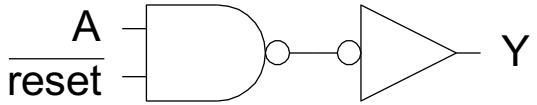
- *Inner* input is closest to output (A)
- *Outer* input is closest to rail (B)

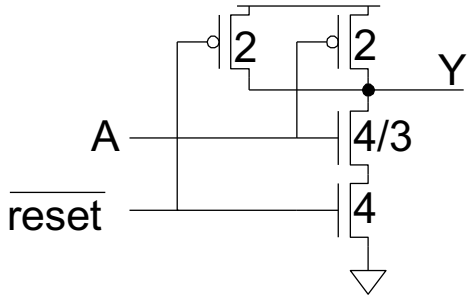
- If input arrival time is known
  - Connect latest input to inner terminal



Asymmetric Gates

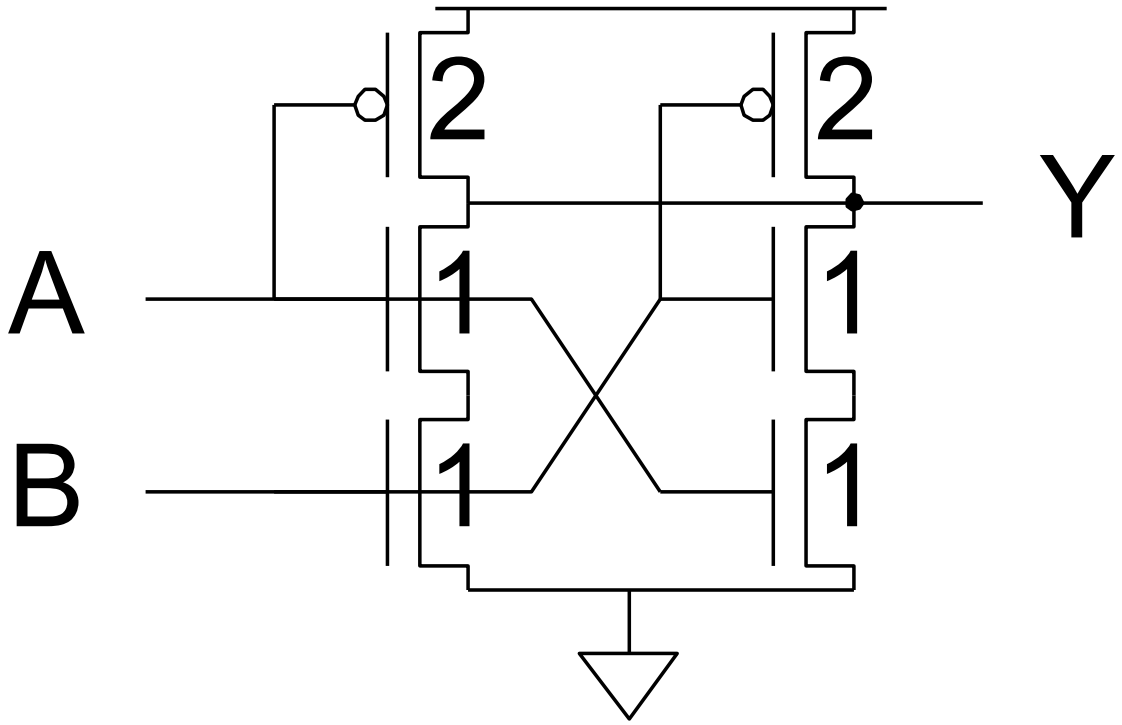
- Asymmetric gates favor one input over another
- Ex: suppose input A of a NAND gate is most critical
  - Use smaller transistor on A (less capacitance)
  - Boost size of noncritical input
  - So total resistance is same
- $g_A = 10/9$
- $g_B = 2$
- $g_{total} = g_A + g_B = 28/9$
- Asymmetric gate approaches g = 1 on critical input
- But total logical effort goes up.





Symmetric Gates

- Inputs can be made perfectly symmetric

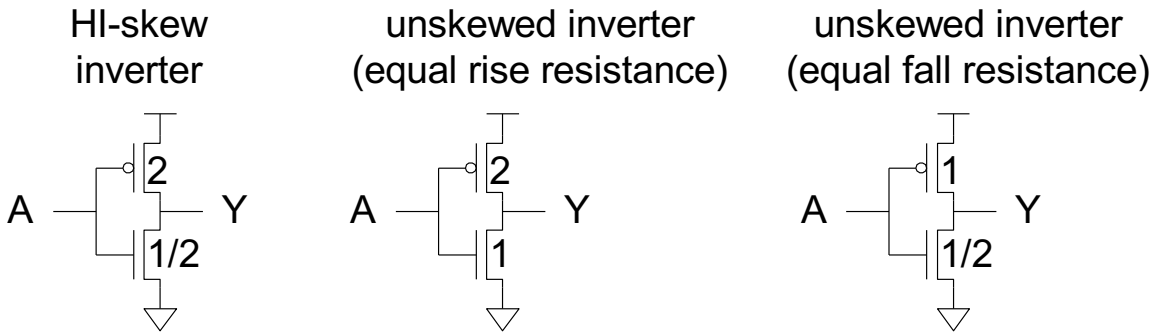


[Sutherland/Sproull/Harris]

Skewed Gates

- Skewed gates favor one edge over another
- Ex: suppose rising output of inverter is most critical
  - Downsize noncritical NMOS transistor

[Sutherland/Sproull/Harris]



- Calculate logical effort by comparing to unskewed inverter with same effective resistance on that edge.
  - $g_u = 2.5 / 3 = 5/6$
  - $g_d = 2.5 / 1.5 = 5/3$

HI- and LO-Skew

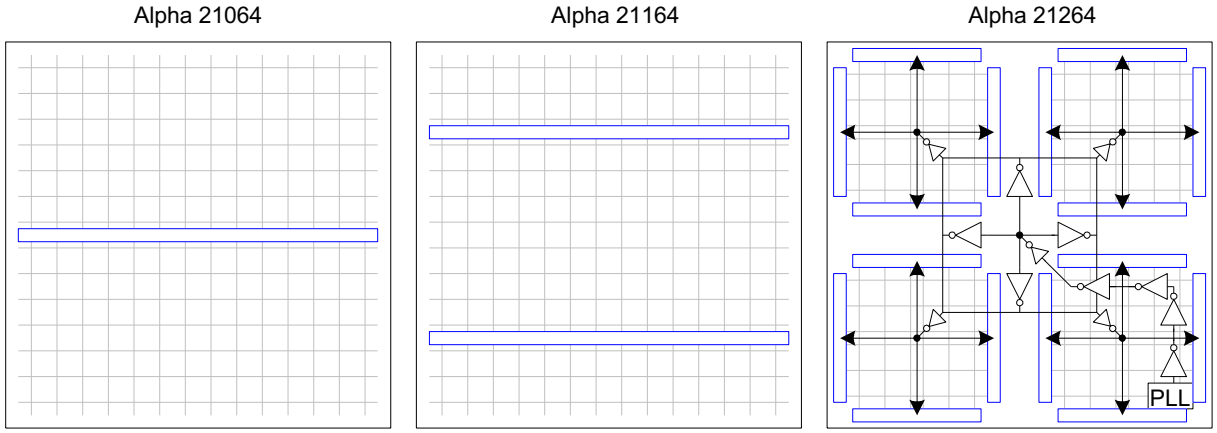
- Def: Logical effort of a skewed gate for a particular transition is the ratio of the input capacitance of that gate to the input capacitance of an unskewed inverter delivering the same output current for the same transition.

- Skewed gates reduce size of noncritical transistors
  - HI-skew gates favor rising output (small nMOS)
  - LO-skew gates favor falling output (small pMOS)
- Logical effort is smaller for favored direction
- But larger for the other direction

$$t_p = \frac{t_{plh} + t_{phl}}{2} = \frac{t_{pdr} + t_{pdf}}{2}$$

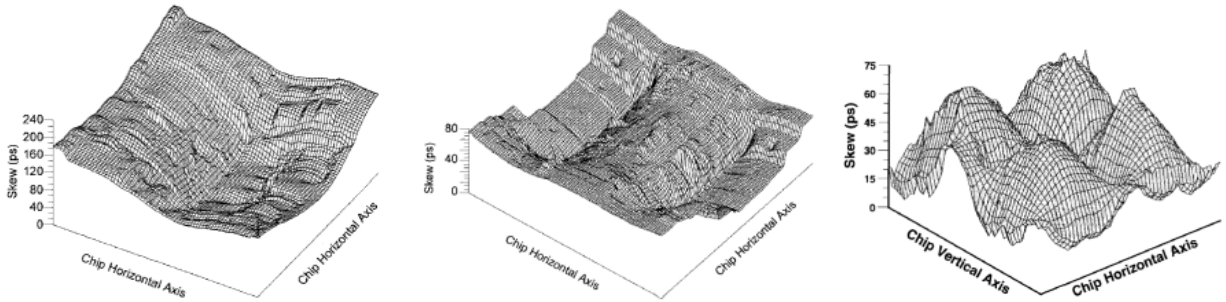
Symmetrical gates are gates which have equal t_{plh} and t_{phl} – where would these be useful?

Alpha Clock Grids



Clock Trees

CTS : clock tree synthesis

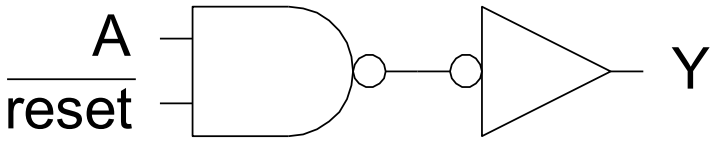


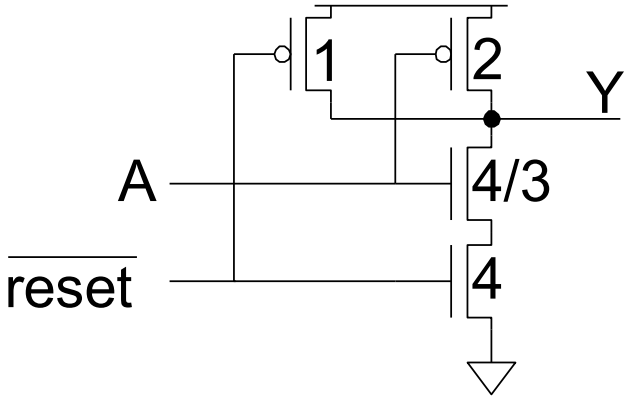
Alpha 21064 Alpha 21164 Alpha 21264

[Weste/Harris]

Asymmetric Skew

- Combine asymmetric and skewed gates
  - Downsize noncritical transistor on unimportant input
  - Reduces parasitic delay for critical input

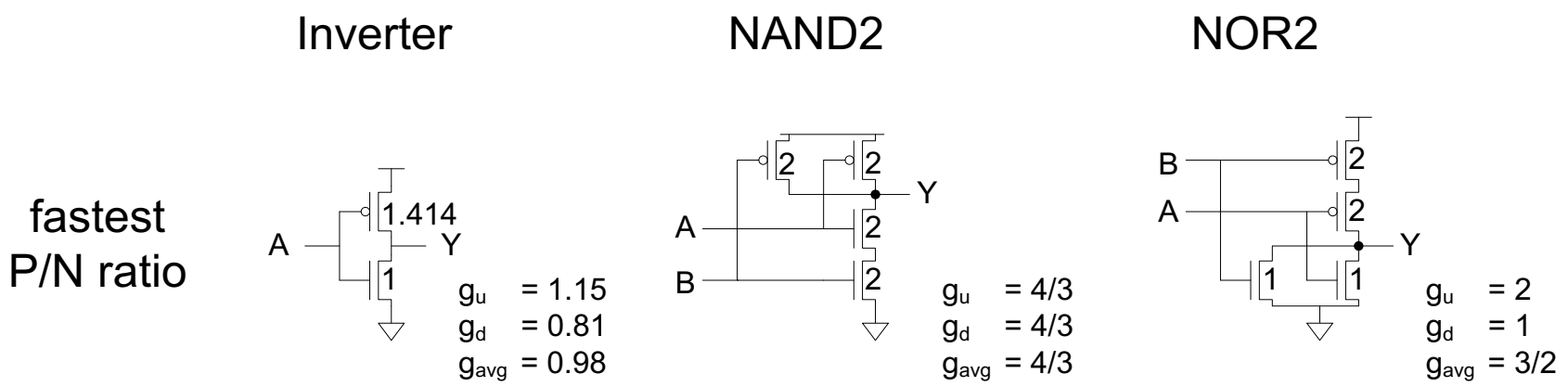




[Sutherland/Sproull/Harris]

P/N Ratios

- In general, best P/N ratio is sqrt of equal delay ratio.
  - Only improves average delay slightly for inverters
  - But significantly decreases area and power



[Sutherland/Sproull/Harris]

Observations

- For speed:
  - NAND vs. NOR
  - Many simple stages vs. fewer high fan-in stages
  - Latest-arriving input
- For area and power:
  - Many simple stages vs. fewer high fan-in stages

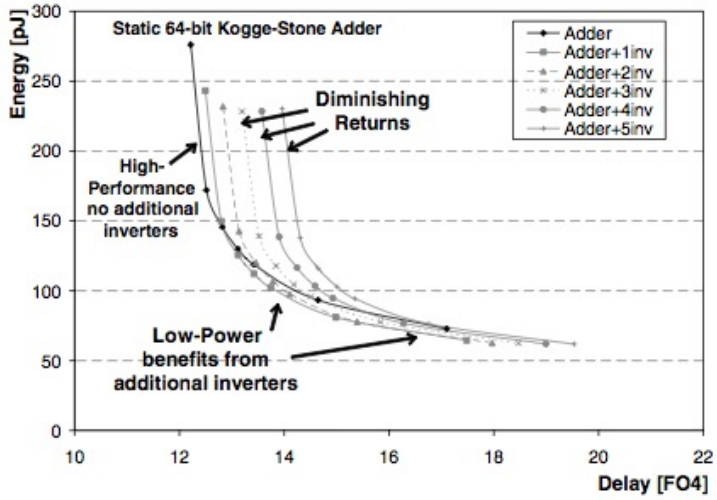


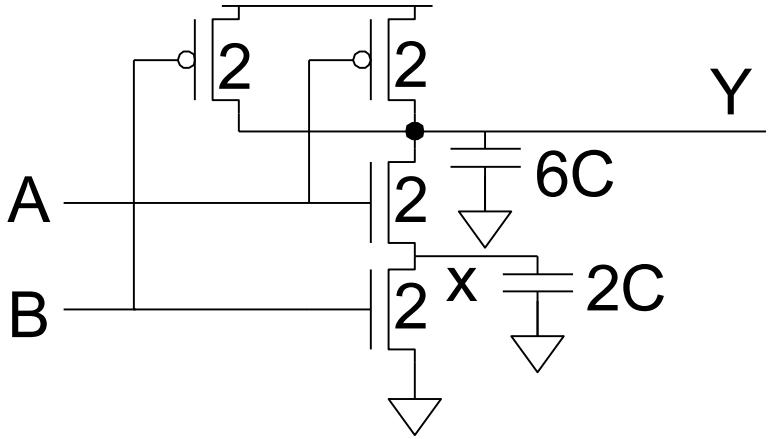
Figure 21. Impact of Buffers Insertion at the Output of a 64-bit KS Adder.

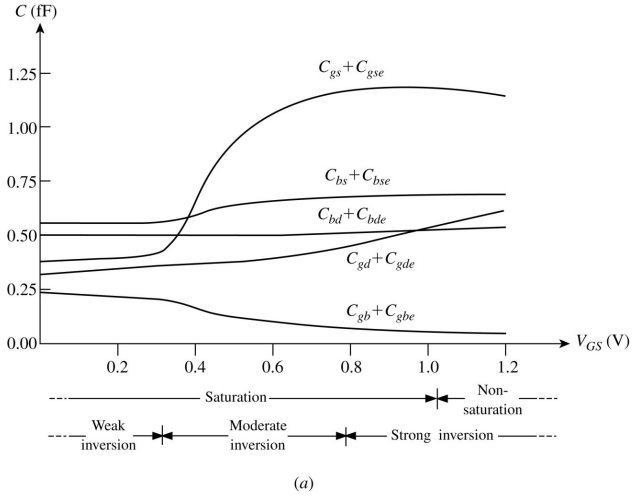
[B. R. Zeydel and V. Oklobdzija, "Design of Energy Efficient Digital Circuits," High Performance Energy-Efficient Microprocessor Design, pp. 31-36, 2006.]

Input Order

- Our parasitic delay (P) model was too simple
  - Calculate parasitic delay for Y falling
    - If A arrives latest? 2τ
    - If B arrives latest? 2.33τ

$$C = C_g = C_{ox} \cdot W \cdot L$$



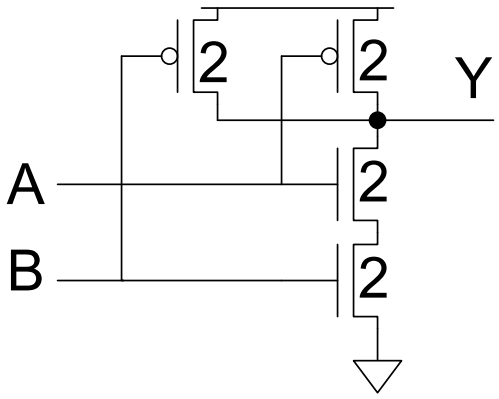


[Weste/Harris]

Inner & Outer Inputs

- *Inner* input is closest to output (A)
- *Outer* input is closest to rail (B)

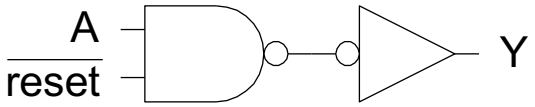
- If input arrival time is known
  - Connect latest input to inner terminal

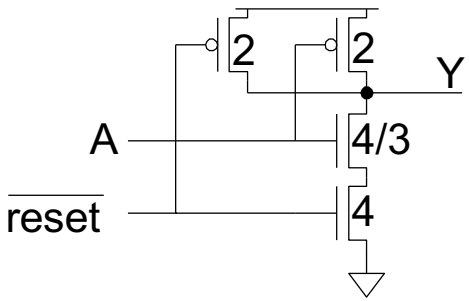


[Sutherland/Sproull/Harris]

Asymmetric Gates

- Asymmetric gates favor one input over another
- Ex: suppose input A of a NAND gate is most critical
  - Use smaller transistor on A (less capacitance)
  - Boost size of noncritical input
  - So total resistance is same
- $g_A = 10/9$
- $g_B = 2$
- $g_{total} = g_A + g_B = 28/9$
- Asymmetric gate approaches g = 1 on critical input
- But total logical effort goes up

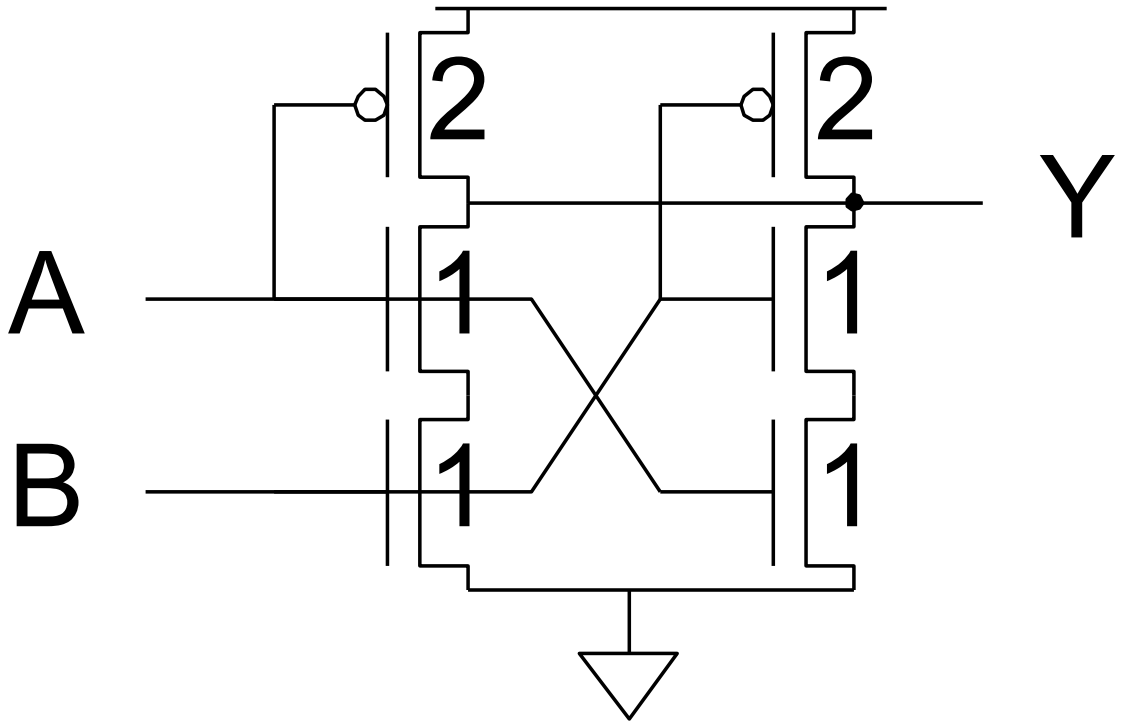




[Sutherland/Sproull/Harris]

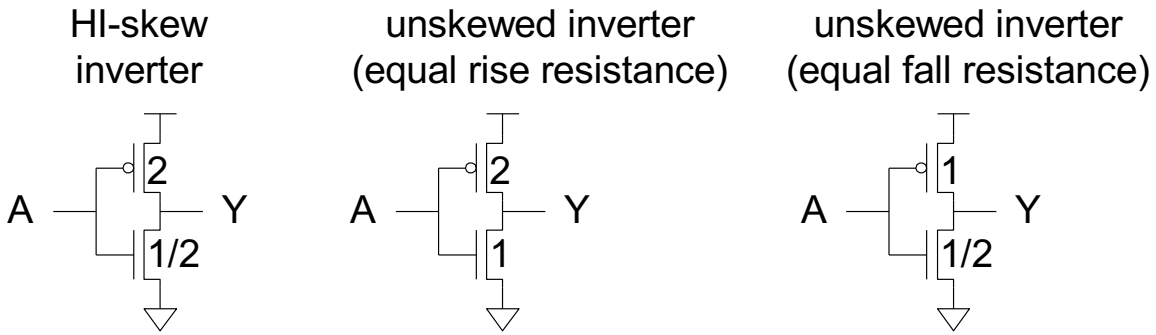
Symmetric Gates

- Inputs can be made perfectly symmetric



Skewed Gates

- Skewed gates favor one edge over another
- Ex: suppose rising output of inverter is most critical
  - Downsize noncritical NMOS transistor



- Calculate logical effort by comparing to unskewed inverter with same effective resistance on that edge.
  - $g_u = 2.5 / 3 = 5/6$
  - $g_d = 2.5 / 1.5 = 5/3$

HI- and LO-Skew

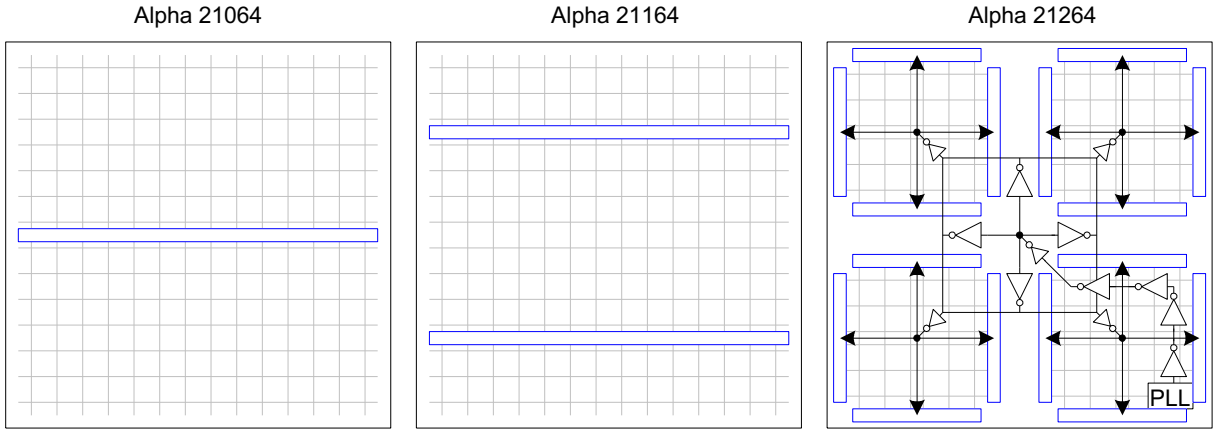
- Def: Logical effort of a skewed gate for a particular transition is the ratio of the input capacitance of that gate to the input capacitance of an unskewed inverter delivering the same output current for the same transition.

- Skewed gates reduce size of noncritical transistors
  - HI-skew gates favor rising output (small nMOS)
  - LO-skew gates favor falling output (small pMOS)
- Logical effort is smaller for favored direction
- But larger for the other direction

$$t_p = \frac{t_{plh} + t_{phl}}{2} = \frac{t_{pdr} + t_{pdf}}{2}$$

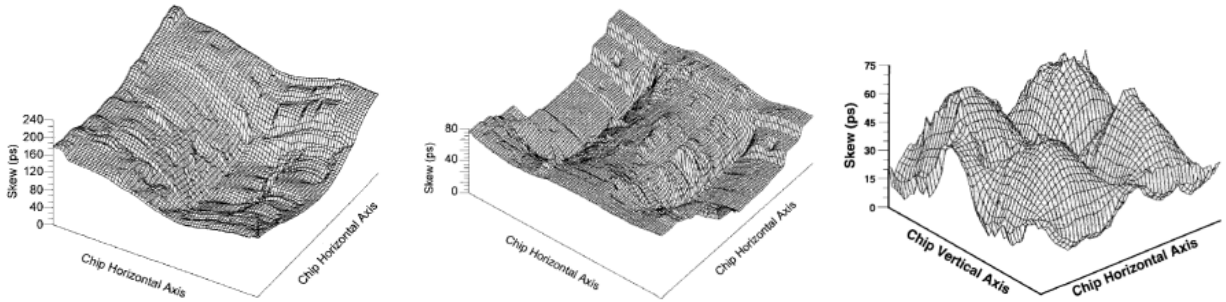
Symmetrical gates are gates which have equal t_{plh} and t_{phl} – where would these be useful?

Alpha Clock Grids



Clock Trees

CTS : clock tree synthesis

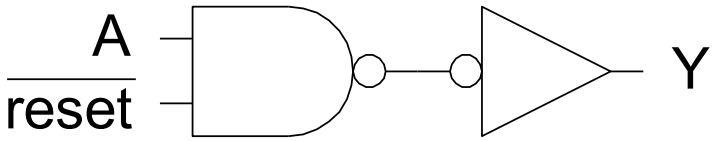


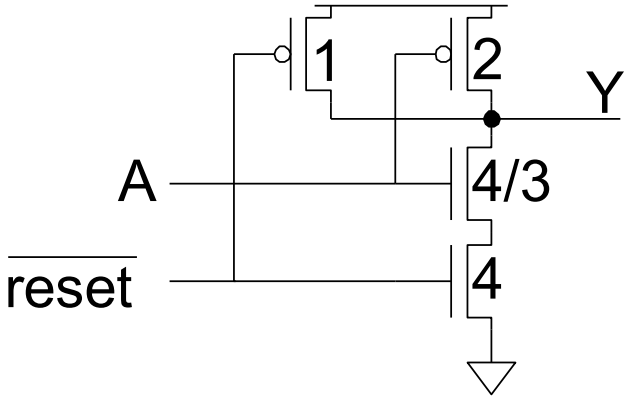
Alpha 21064 Alpha 21164 Alpha 21264

[Weste/Harris]

Asymmetric Skew

- Combine asymmetric and skewed gates
  - Downsize noncritical transistor on unimportant input
  - Reduces parasitic delay for critical input

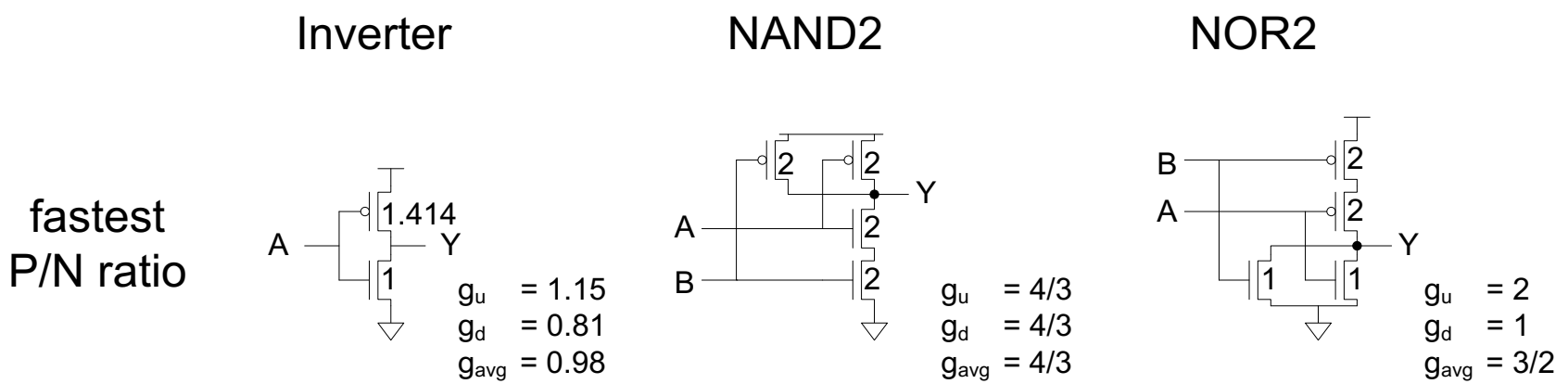




[Weste/Harris]

P/N Ratios

- In general, best P/N ratio is sqrt of equal delay ratio.
  - Only improves average delay slightly for inverters
  - But significantly decreases area and power



[Weste/Harris]

Observations

- For speed:
  - NAND vs. NOR
  - Many simple stages vs. fewer high fan-in stages
  - Latest-arriving input
- For area and power:
  - Many simple stages vs. fewer high fan-in stages

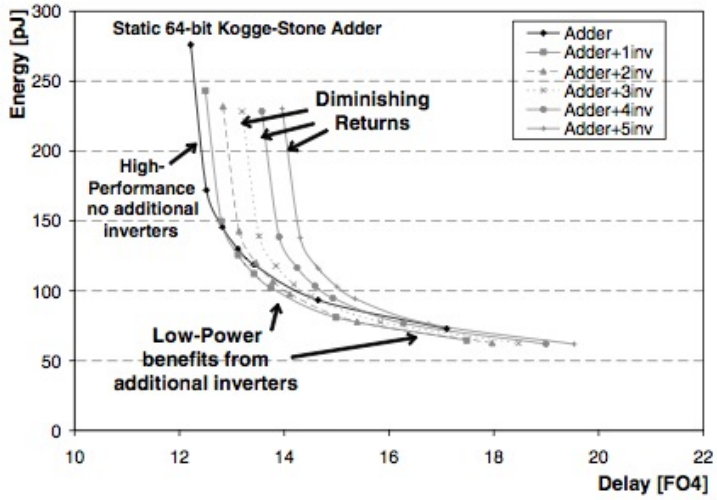


Figure 21. Impact of Buffers Insertion at the Output of a 64-bit KS Adder.

[B. R. Zeydel and V. Oklobdzija, "Design of Energy Efficient Digital Circuits," High Performance Energy-Efficient Microprocessor Design, pp. 31-36, 2006.]