Insight EDA Inc
www.InsightEDA.com

Fanout / Loading Calculations & Checking Insight Analyzer

Contents

Overview

This paper provides a technical overview of the methods of calculating and checking fanout of logic circuits, using Insight Analyzer.

- Wire Gate Load
- Wire Parasitic Load
- Drive Strength
- Fanout Ratio
- Beta Ratio

Fanout

Insight Analyzer calculates fanout for individual transistors as a ratio of output capacitance to input capacitance. Input capacitance is proportional to the area of a driver's gate while load is the total capacitance being driven.

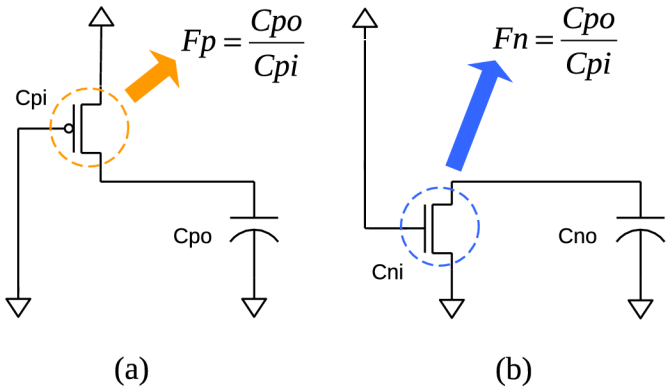


Figure 1: Fanout for a) a PFET driver, and b) an NFET driver

To be consistently meaningful, fanout is normalized to the standard reference inverter. This is an inverter with transistors sized for roughly equal current such

that the gate width of the PFET is twice that of the NFET (assuming both use minimum length.)

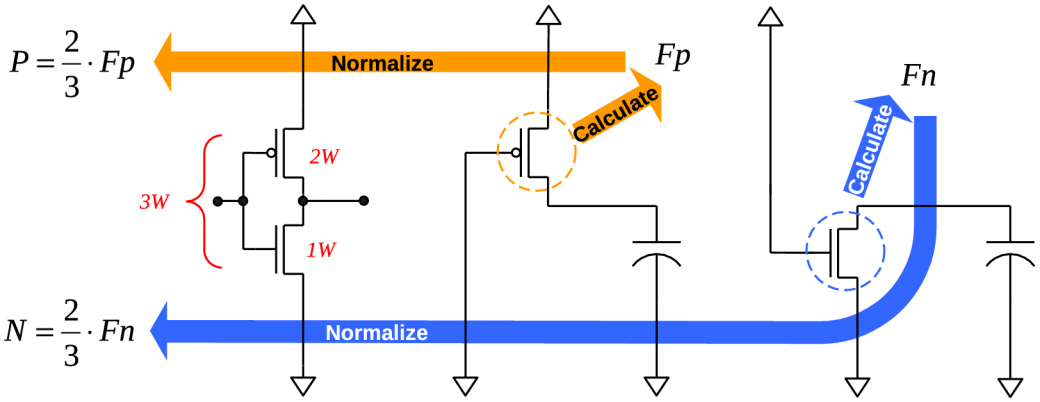


Figure 2: Fanout normalized to the standard reference inverter

Using standard reference inverters, a fanout of 4 is defined as 1 inverter driving 4 identical inverters.

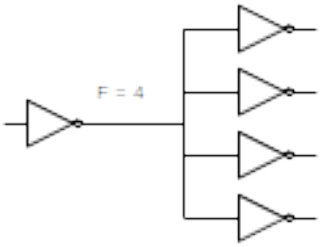


Figure 3: Fanout = 4 using standard reference inverters

Fanout calculations use device parameters defining gate and edge capacitance which are entered as a part of initial software setup.

Cg	Gate capacitance per unit width at *DefL*
Cedge	Edge capacitance per unit width
DefL	The length at which *Cg* is defined

Transistor gate dimensions are instance specific:

W	Width of gate
L	Length of gate

Methodology

The basic methodology for calculating fanout is:

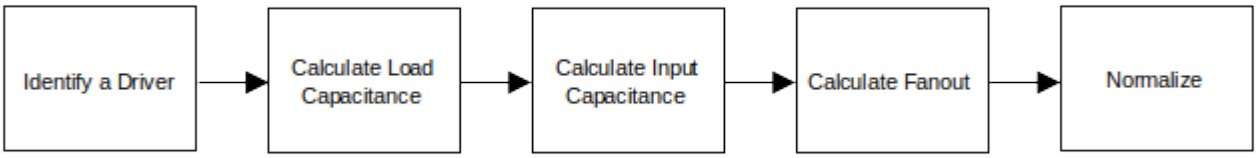


Figure 4: Basic methodology for calculating fanout

Drivers are individual transistors whose outputs are connected to logic nets and for which an input capacitance is calculated. Load is the total capacitance on the driver's output.

Calculating Load Capacitance

Load is calculated as the sum of all capacitive loading on the driver. Sources of capacitive loading include:

- Gate capacitance from inputs of other devices (accounting for non-default *L)*

$$Cload = W \cdot Cg \cdot \frac{L}{DefL}$$

- Edge capacitance from outputs of inactive devices

$$Cload = W \cdot Cedge$$

- Wire load from either estimated or extracted capacitance

$$Cload = Cestimated \,||\, Cextracted$$

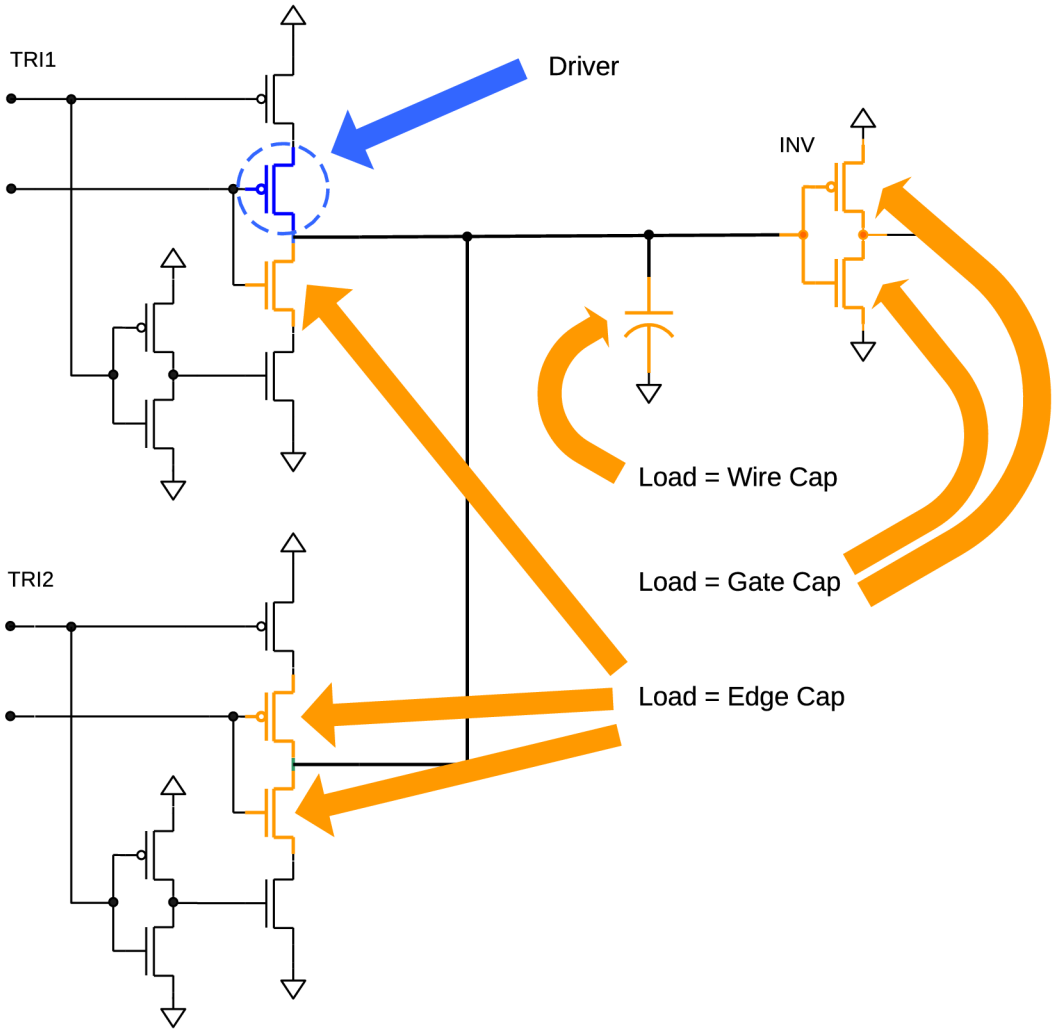


Figure 5: Example of various sources of capacitive loading on the PFET driver of TRI1

Calculating Input Capacitance

Input capacitance is calculated in units of equivalent gate width taking into account the following factors:

- Non-default gate length

$$Wdrv = W \cdot \frac{L}{DefL}$$

- Series devices

$$Wdrv = \sum Wdrv_x$$

- Parallel drivers

$$Wdrv = \frac{1}{\sum \frac{1}{Wdrv_x}}$$

Wdrv for series and parallel devices is calculated using Ohm's Law for conductance. Ohm's Law is applicable because for fanout, transistors are modeled as pure resistors such that an OFF transistor has no conductance while an ON transistor has conductance proportional to it's width.

Series devices (such as those in driver stacks or as part of a switched power rail) are those found between the driver and its path to power or ground. These act as series resistance reducing effective drive strength. A driver's effective width is calculated using Ohm's Law for series conductance.

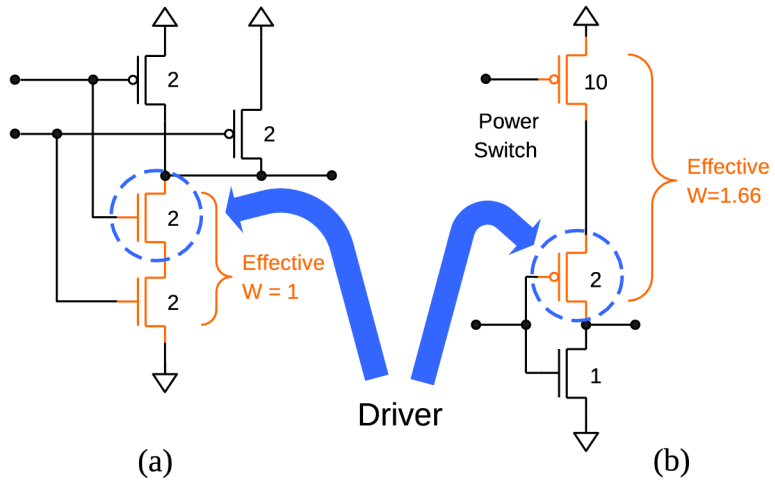
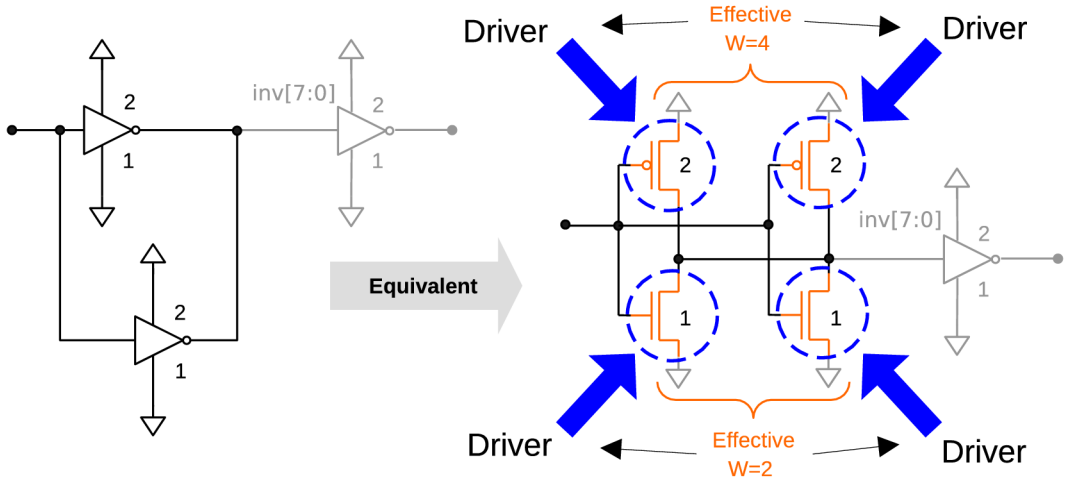


Figure 6: Examples of series devices: a) an NFET stack in a NAND gate, and b) a power switch in series with the PFET of an inverter.

Parallel drivers are those that share the same input and output. These act in concert reducing the current requirement for each individual driver. A driver's effective width is calculated using Ohm's Law for parallel conductance.



(a) (b)

Figure 7: Example of a) parallel inverters, and b) the equivalent circuit showing effective width is double for each of the parallel drivers

Calculating Fanout

Fanout for individual drivers is calculated as the ratio of equivalent widths at DefL of the driver corresponding to both input and load capacitance:

$$F = \frac{Wload}{Wdrv} = \frac{Cload}{Cg \cdot Wdrv}$$ *Fanout of an individual driver*

where:

$$Wload = \frac{Cload}{Cg}$$ *Equivalent width at driver's DefL*

If Fp and Fn are the fanout values calculated for PFET and NFET drivers, respectively, the normalized fanout values P and N are:[1]

$$P = \frac{2}{3} \cdot Fp$$ *Normalized fanout for PFETs*

$$N = \frac{1}{3} \cdot Fn$$ *Normalized fanout for NFETs*

1 Fanout for each device of a logical output pair may vary significantly due to edge loading of the complementary device and/or imbalances in drive strength.

Figure 8 shows an example of fanout calculations for an inverter driving a capacitive load Cout. The total load for each device is Cout plus the edge capacitance of the complementary device.

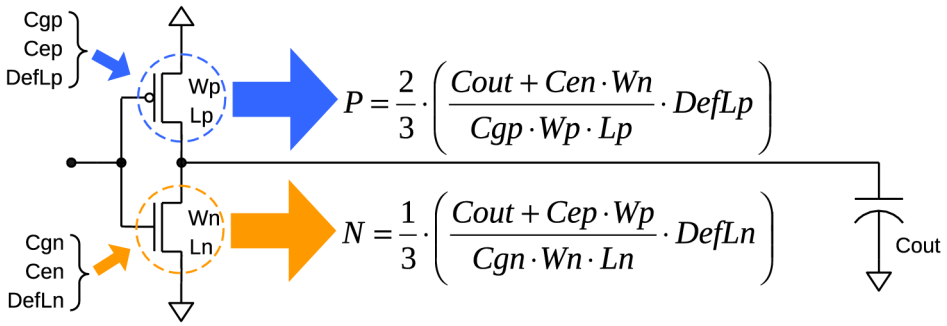


Figure 8: Example of an inverter's reported fanout for both P and N devices each with parameters as shown and driving a downstream load of Cout.

Special Considerations

Length Variation

In most fanout calculations, transistor length is assumed to be fixed, with a standard value of "L" applied to all transistors. However, calculations must account for cases where length is changed. Insight Analyzer handles these as follows:

- Gate loads: As the length of a gate load device is increased, the gate loading capacitance increases accordingly (as described previously).
- Edge loads: As the length of a transmission gate or other edge loading device is increased, the loading capacitance remains unchanged.
- Driver strength: This is where traditional fanout calculations lead to a fundamental paradox. As the traditional fanout method is based on Cout/Cin, scaling driver length up would result in a lower fanout value. However, this is contrary to the core intent of fanout.

  - When all lengths are equal, fanout calculation is straightforward.
  - As driver length increases, traditional fanout values decrease, implying a stronger driver.
  - As driver length increases, the series resistance is increased, resulting in an actually weaker driver. This is contrary to the key concept of a fanout value, where lower means stronger, not weaker.
- An optional switch is provided, to set a policy on the handling of length variations:
  - Normally, increased driver length will result in lower fanout values, as described in the previous sections of this document. This is the “traditional” application of fanout.
  - If inverse scaling is selected, Insight Analyzer will scale driver strength down as length is increased. A driver with twice its normal length will have half its normal strength, and twice its normal fanout value.
  - The inverse scaling option is available in the Fanout check, and some API commands. Please see the Insight Analyzer documentation for details.

Size Ratio Adjustment

Traditional fanout calculations assume that an NFET is 2x stronger than a PFET of the same size. The PFET must be twice the size of the NFET, to obtain equivalent strength. This is represented as 1/3 and 2/3 scaling factors for NFET and PFET, respectively.

For MOSFETs that do not adhere to the typical strength ratio of 2:1 PFET to NFET, it is necessary to adjust these scaling factors of 1/3 and 2/3.

Insight Analyzer handles this with a dedicated property on a per-device basis. The property, or “device parameter”, is named PsizeRatio. If this value is not set (left empty in the GUI editing table), the typical ratio of 2:1 is assumed. If you are

using FinFETs, you will probably want to set this value, as FinFETs generally don't adhere to the typical 2:1 size ratio.

This parameter should not be inverted for the NFET member. If you set this parameter, the value must be the same for both the P and N parts of a corresponding pair. For example: If a PFET device “Pchx” is used with an NFET device “Nchx”, and their P:N size ratio is 5:4, you would set this value as follows:

- Pchx PsizeRatio = 1.25 (5/4)
- Nchx PsizeRatio = 1.25 (5/4)

Transmission Gates

When a transmission gate is used to isolate a driver from a load, the driver must still be capable of driving the load while the transmission gate must be capable of handling the driver’s current. To ensure both the driver and the transmission gate are sized properly, each are treated in isolation and fanout calculated for each.

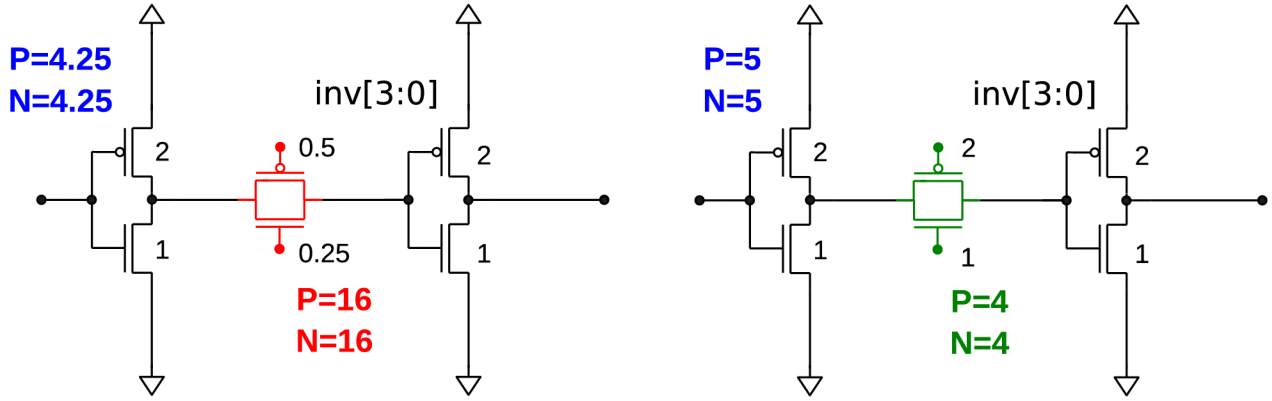


Figure 9: Example of isolated fanout calculations for both driver and transmission gate where a) the transmission gate is too small, and b) the transmission gate is adequate.

When calculating fanout for a driver, it is assumed that the transmission gate of the worst case load is ON, while all others are OFF. The ON transmission gate is

modeled as a short circuit with a small parasitic capacitance while the ON transmission gates are modeled as edge capacitance.

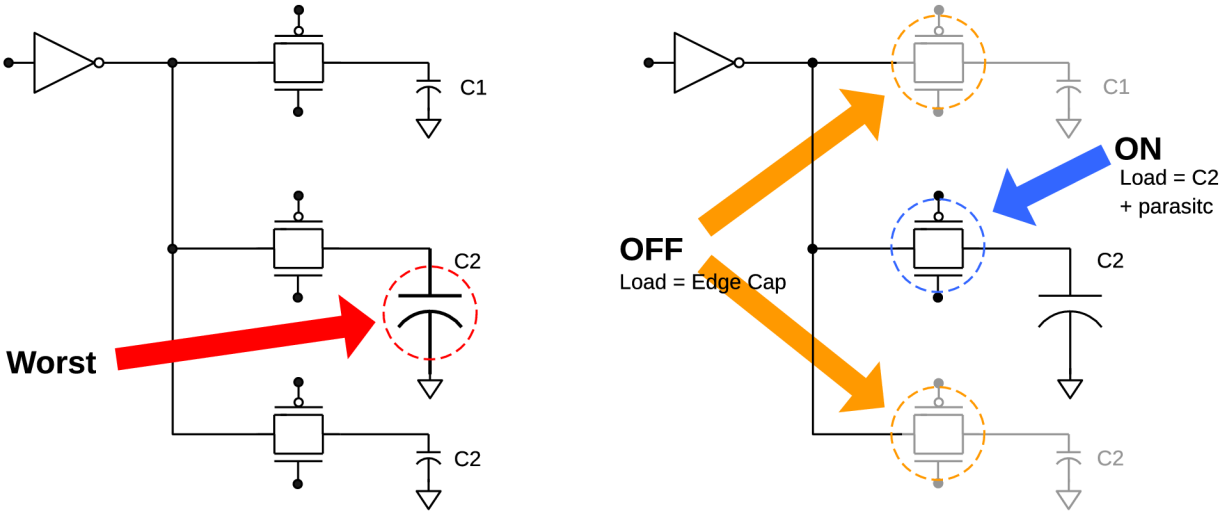


Figure 10: Fanout calculations for a driver of multiple transmission gates will use the worst case load. Total load on the driver also includes edge capacitance of the OFF transmission gates and parasitic capacitance of the ON transmission gate.

When calculating fanout for a transmission gate, it is modeled assuming an ideal connection to power or ground.

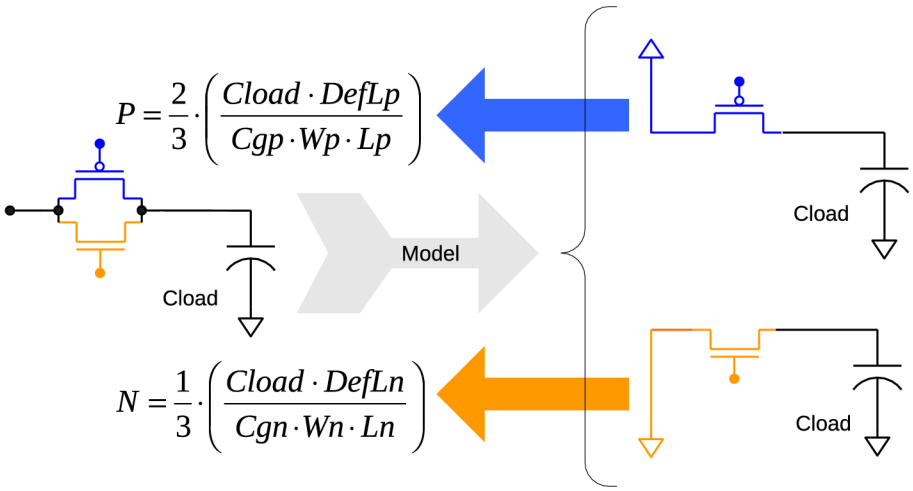


Figure 11: The equivalent circuits for calculating fanout of a transmission gate.

Header & Footer Switches

Some designs include power switches above or below standard cells, connecting them to power or ground. If these power switches are used in the calculation, they can have an impact on the final result. But, they often should be excluded from the calculation because they do not have a tangible effect on fanout. For example, if a power switch is assumed to be always on while the circuit is operating, it will not affect delay times.

The fanout and beta ratio calculations in Insight Analyzer use an option to exclude or include these "header" and "footer" power switches. It is an optional argument, "excludeStdcellHeaderFooter" which the user can add to a command to exclude them.

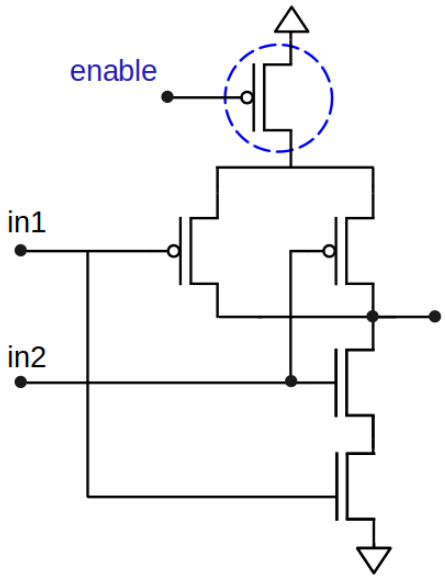


Figure 12: a simple NAND gate with a "header" power switch connecting it to the power rail. This power switch can be included or excluded from fanout and beta ratio calculations.

Example

The following simple example illustrates basic fanout calculations. All inverters use the same PFET and NFET devices with the following parameters:

Device	Cg (fF/µm)	Cedge (fF/µm)	DefL (fF/µm)
PFET	0.639	0.571	0.04
NFET	0.708	0.498	0.04

Gate dimensions W and L are instance specific and shown as a pair of values for each device.

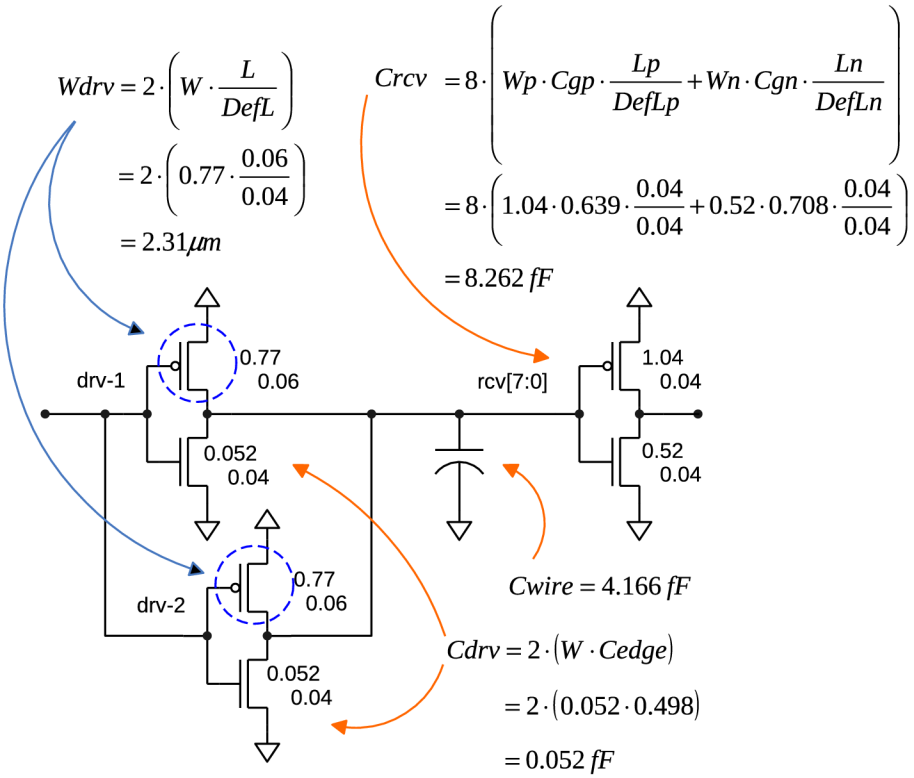


Figure 13: Fanout calculations for the PFETs of two parallel inverters (w/ a purposely undersized NFET) driving an array of 8 similar inverters.

$$Fanout\ for\ the\ PFET = \frac{2}{3} \cdot \left(\frac{Crcv + Cwire + Cdrv}{Wdrv \cdot Cg} \right) = \frac{2}{3} \cdot \left(\frac{12.43}{2.31 \cdot 0.639} \right) = 5.614$$

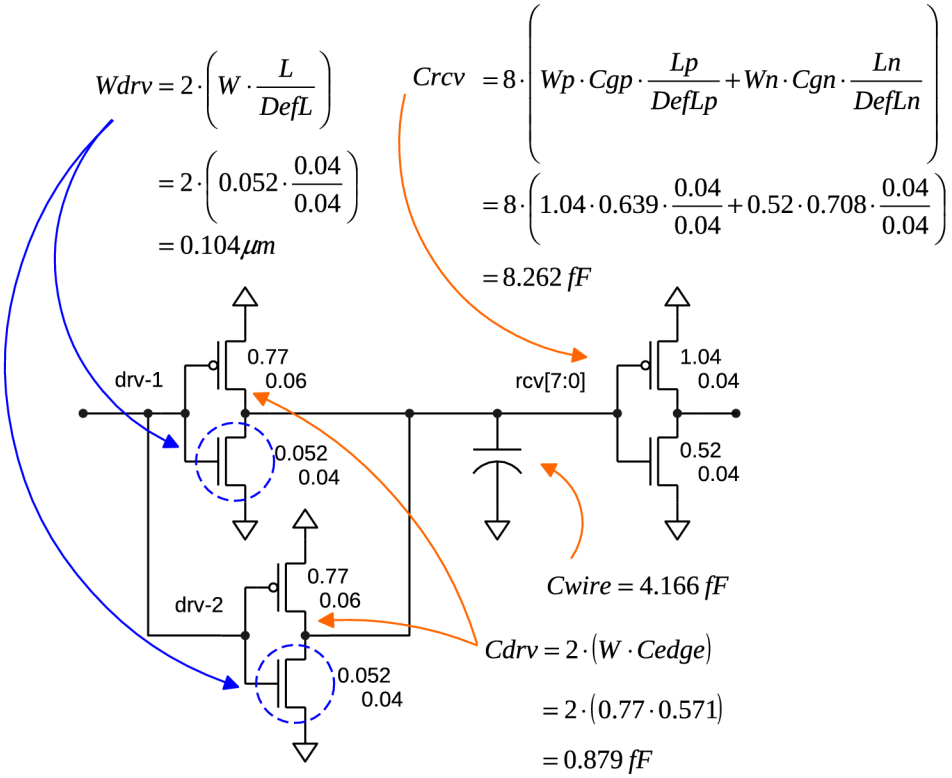


Figure 14: Fanout calculations for the two parallel undersized NFETs. Crcv and Cwire are the same as for the PFET calculations.

$$Fanout\ for\ the\ NFET = \frac{1}{3} \cdot \left(\frac{Crcv + Cwire + Cdrv}{Wdrv \cdot Cg} \right) = \frac{1}{3} \cdot \left(\frac{13.26}{0.104 \cdot 0.708} \right) = 60.03$$

Changing the undersized NFET driver from W = 0.052 to 0.52 gives a more reasonable fanout:[2]

$$Fanout\ for\ the\ NFET = \frac{1}{3} \cdot \left(\frac{Crcv + Cwire + Cdrv}{Wdrv \cdot Cg} \right) = \frac{1}{3} \cdot \left(\frac{13.26}{1.04 \cdot 0.708} \right) = 6.003$$

2 This change will also have a slight impact on the fanout of the PFET as the edge capacitance of the NFET with a wider gate is now greater.

Beta Ratio

Insight Analyzer addresses beta ratio in the following ways:

1. A plugin check, dedicated to checking beta ratio for within allowed range (please see scan reference).
2. API commands, for building a custom application.

Principles

Beta Ratio is normally thought of as a ratio of geometry, PFETs to NFETs. However, due to the various topology combinations with series / parallel stacking, etc, a good beta ratio calculation is more complex than simply "P divided by N" and must consider real world factors. More FETs in series scale drive strength down. FETs in parallel, which can manifest in many different ways, scale it up. Other circuit elements such as power switches and passgates add complications.

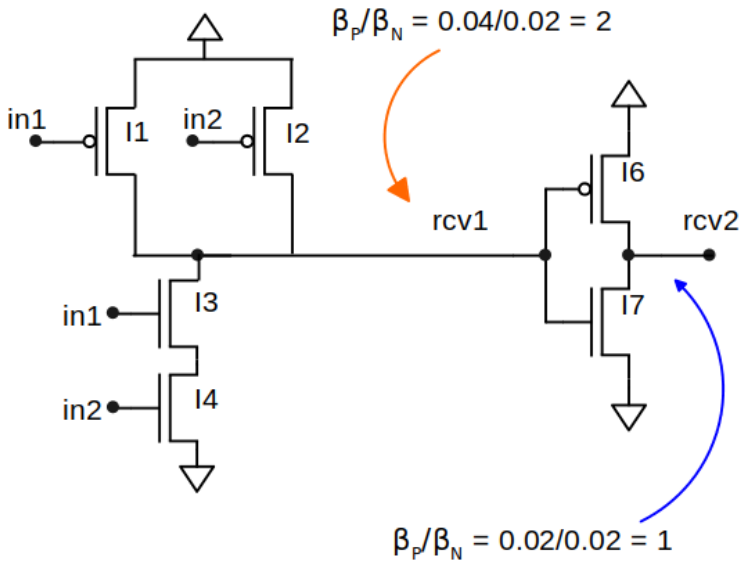


Figure 15: Two examples of beta ratio (assume all FETs have identical W and L). Net "rcv1" has a high beta ratio, due to the parallel PFETs in this NAND structure. The standard inverter on the right gives net "rcv2" a beta ratio of 1.

Calculation

Insight Analyzer makes this calculation by comparing the true drive strength of the PFET vs NFET parts of the stacks, on each CMOS net. The Beta Ratio plugin check works by scanning every net in the circuit. Some nets are excluded from the check, such as power rails and analog nets with no CMOS. For each valid net, it checks all the source and drain pins connected to it, then uses the "getStrength" API command (details below) to get the drive strength for that FET. Finally, it adds up the beta ratio for the net, and if it is outside the user-defined range, it is reported as a violation.

The "getStrength" API command can be used individually in a Tcl script or in the script console to investigate a pin:

cdb pin <id> getStrength ?-excludeStdcellHeaderFooter? ?-expanded?

The code below was run on the example in Figure 15.

```
set pin "/MI1-D"
puts $fp "$pin: [ cdb pin $pin getStrength ]"
set pin "/MI6-D"
puts $fp "$pin: [ cdb pin $pin getStrength ]"
    /MI1-D: 0.06
    /MI6-D: 0.02
```

In the getStrength command, the optional argument "excludeStdcell..." will exclude power switches at the top and bottom of the circuit. See the section "Header & Footer Switches" above for more information. The optional argument "expanded" will show extra information about the full stack and the calculation.

The device parameter "PsizeRatio", indicates the relative strength of PMOS to NMOS transistors. This parameter does not impact the numeric result of calculation, only the threshold of what is reported as a failure. For more information, see the section "Size Ratio Adjustment" above.

For more information on the getStrength command, see "Utilities" below.

Utilities

Insight Analyzer provides related utility functions that can be used in your own custom script applications. Full documentation is available through the Help system. Following is a summary list.

cdb pin <id> getStrength ?-excludeStdcellHeaderFooter? ?-expanded?

Full calculation of the total effective drive strength of the given device, through the given pin. This means, the pin used shall be the actual output pin of a CMOS driver, etc. Calculation will then follow the topology on the other side of the primitive device. The value of strength is expressed in terms of width. The value is increased by transistors connected in parallel, and decreased by other transistors in the stack. The value is also increased appropriately by parallelism further up in hierarchy, such as an array of INV drivers.

db inst <id> getGateCap ?-effective? ?-type <type>?

Find gate capacitance of the given FET instance.

cdb inst <id> getEdgeCap ?-effective?

Find body capacitance of the given FET instance.

cdb inst <id> getWidth ?-effective?

Find width of the given FET instance. Width can include effects of paralleclism,SPICE M, arrayed instance names, fingers, and fins where appropriate.

cdb inst <id> isLogic

Logic functions (CMOS and dynamic) are recognized as part of netlist loading. Logic may also be user-defined with subgraph templates. Once identified, the logic can be reported by this command.

cdb inst <id> isInverter ?-strictOnPower | -strictOnSupply | -strictOnReturn?

This command identifies the most common forms of inversion, including plain P:N inverter, NAND or NOR with all pins tied together, and similar CMOS structures with tied pins.

cdb inst <id> isTristate ?-enable|-data? ?-functional?

Detects tri-state buffers.

cdb inst <id> isKeeper

Keepers are recognized as part of netlist loading. A keeper is an unconditional,constant feedback path on a digital node.

cdb net <id> isStaticLogic

Determines whether the net has output driver stacks that behave in purely static mode.

cdb net <id> isDynamicLogic

Indicates that the net is a dynamic logic storage node, either by topological recognition, or user definition.

cdb net <id> isLatchNode ?-rawStorage | -buffered?

cdb net <id> latchNodeType

cdb net <id> getCMOSoutputCount

Determines the presence of multiple CMOS outputs on the same given net.

cdb net <id> getStackInstances ?-includeFeedback? ?-includeFullStack?

From the given net, searches outward to find all device instances in pullup / pulldown stacks.

cdb net <id> getWireLoad ?-groundOnly?

Calculates the total wire loading value, given a number of C values from extraction. This is an extracted value, only available after SPF back-annotation, and does not include model capacitors that might exist in schematic.

cdb net <id> getDeviceLoads

Calculates loading on the net due to primitive device connections.

cdb net <id> getTotalLoad ?-includePassgateBranch?

Find the total capacitance on the given net. Includes ideal C wire models,parasitic (extracted) loads, Gate and Edge capacitance of MOS transistors.

cdb net <id> getP2PResistance

Given two points on the same net, calculates the total effective resistance through parasitic R values. This assumes parasitic extraction has been loaded.

cdb net <id> getFanoutRCatPoint <point> ?-Tfactor <val>? ?-noDevCap?

Calculates an effective R and C for the net, to be used in conjunction with fanout calculations that are derated by extracted parasitics.

cdb net <id> getRCbetweenPoints

Given two points on the same net, finds the total resistance between them, and the capacitance from point B. This assumes parasitic extraction has been loaded.

cdb net <id> getCrossCoupling

Finds other nets that have cross-coupling values to the given net, and reads them back with C values corresponding to each. This assumes extracted parasitics have already been loaded.

cdb net <id> getFanout ?-RCreduced? ?-noEdgeLoading? ?-proportional? ?-excludeStdcellHeaderFooter?

Calculates the logical Fanout values for drivers and receivers on the net. Transistors are considered to be possible drivers in an optimistic way, to ensure that values are returned even where logic subgraphs might not be detected or defined. The only transistors that will be ignored (not drivers) are those wired as capacitors or diodes.

Reference Examples

Following this page are reference examples for the topics in this document. These examples are excerpts from the main example set, normally installed with Insight Analyzer under the root ‘examples’ path.