SPICE Computation of MOS Capacitance

```
.
M1 4 3 5 0 NFET W=4U L=1U AS=15P AD=15P PS=11.5U PD=11.5U
.
.
.MODEL NFET NMOS
+ TOX=200E-8
+ CGBO=200P CGSO=600P CGDO=600P
+ CJ=200U CJSW=400P MJ=0.5 MJSW=0.3 PB=0.7
+ . . . . .
.
.
```

Definitions:

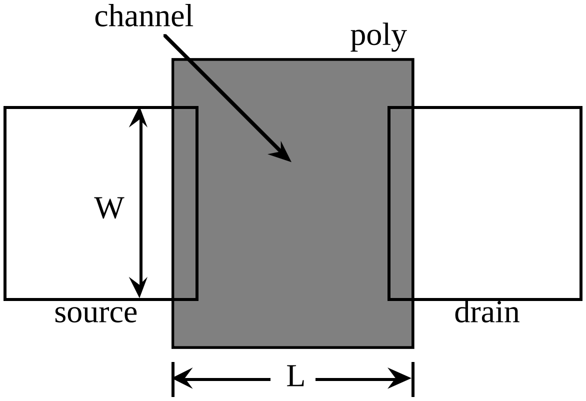
```
AS = area of Source
AD = area of Drain
PS = perimeter of Source
PD = perimeter of Drain
```

The `TOX` parameter allows computation of C_{ox}

$C_g = C_g$ (intrinsic) + C_g (extrinsic)

C_g (intrinsic) = $C_{ox} \times W \times L_{eff}$ ($\Leftarrow$ only $\frac{2}{3}$ if in saturation)

Extrinsic C_g caused by overlap of gate with source/drain and channel



$C_{gbo} \Rightarrow$ caused by poly extension past channel

C_{gso}, $C_{gdo} \Rightarrow$ caused by overlap of poly with source/drain

C_{gbo} multiplied by channel length; C_{gso}, C_{gdo} multiplied by channel width

Typically, gate capacitance will tend to dominate drain, source capacitance but can vary significantly with process.

Example from book:

$$\begin{aligned}C_{g(intrinsic)} &= W \times L \times C_{ox} = 4 \times 1 \times 17 \times 10^{-4} \text{ [pF]}\\ &= 0.0068 \text{ [pF]}\end{aligned}$$

In this example, the extrinsic gate capacitance for a typical MOS transistor is

$$\begin{aligned}C_{g(extrinsic)} &= (W \times C_{gso}) + (W \times C_{gdo}) + (2L \times C_{gbo})\\ &= (4 \times 6 \times 10^{-4}) + (4 \times 6 \times 10^{-4}) + 2 \times (1 \times 2 \times 10^{-4}) \text{ [pF]}\\ &= 0.0052 \text{ [pF]}\end{aligned}$$

In SPICE the capacitance of a source or drain diffusion is calculated as follows:

$$C_j = \left(\text{Area} \times CJ \times \left(1 + \frac{VJ}{PB}\right)^{-MJ}\right) + \left(\text{Periphery} \times CJSW \times \left(1 + \frac{VJ}{PB}\right)^{-MJSW}\right)$$

where

CJ = the zero-bias capacitance per junction area
CJSW = the zero-bias junction capacitance per junction periphery
MJ = the grading coefficient of the junction bottom
MJSW = the grading coefficient of the junction sidewall
VJ = the junction potential
PB = the built-in voltage (~ 0.4 to 0.8 [V])
Area = AS or AD, the area of the source or drain
Periphery = PS or PD, the periphery of the source or drain

PB, CJ, CJSW, MJ, and MJSW are specified in the model card. AS, AD, PS, and PD are specified by the element card. VJ depends on circuit conditions. At VJ = 2.5 [V] (half rail (V_{DD} = 5 [V])),

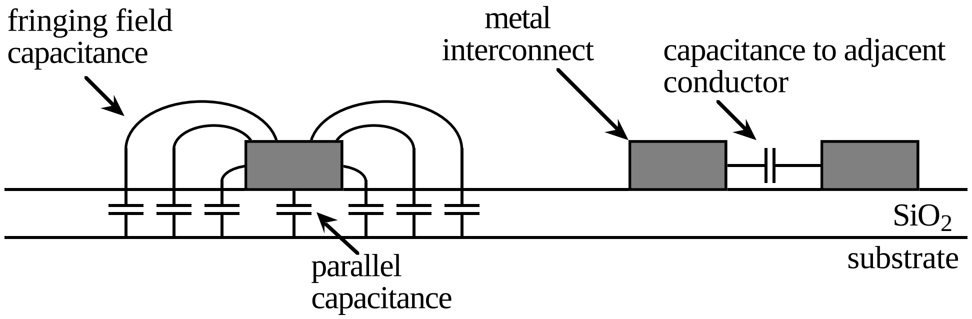
$$\begin{aligned}C_{jdrain} &= \left(15 \times 10^{-12} \times 2 \times 10^{-4}(1 + 2.5/0.7)^{-0.5}\right)\\ &\quad + \left(11.5 \times 10^{-6} \times 4 \times 10^{-4}(1 + 2.5/0.7)^{-0.3}\right) \text{ [pF]}\\ &= (15 \times 2 \times 10^{-4} \times 0.47) + (11.5 \times 4 \times 10^{-4} \times 0.63) \text{ [pF]}\\ &= 0.0014 + 0.0029 \text{ [pF]}\\ &= 0.0043 \text{ [pF]}\\ &= 4.3 \text{ [fF]}\end{aligned}$$

Summarizing these capacitances then,

$$C_{gtotal} = 0.0068 + 0.0052 \text{ [pF]} = 12 \text{ [fF]}$$

$$C_{drain} = C_{source} = 0.0043 \text{ [pF] (@ 2.5 [V])}.$$

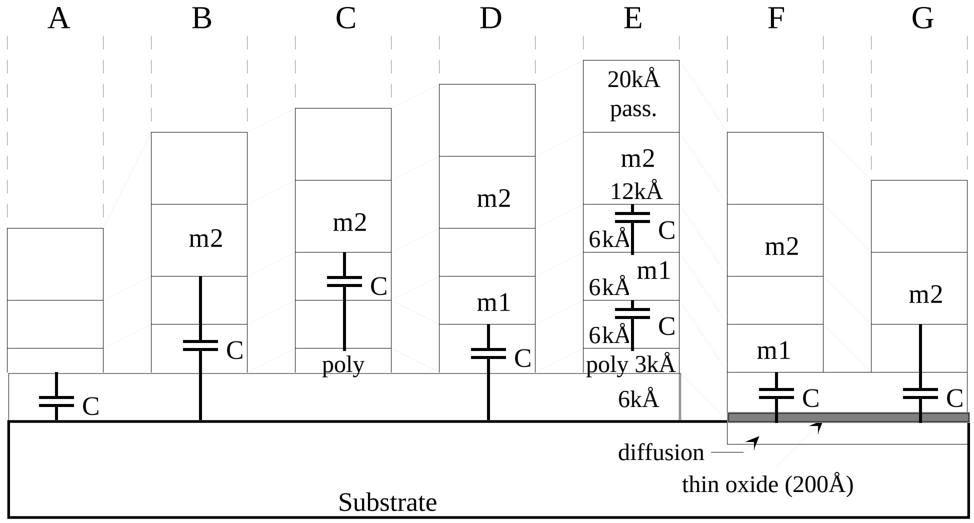
Routing Capacitance



Fringing Field Capacitance occurs at edge of the conductor and is due to the conductor's finite thickness.

Fringing Field Capacitance will cause effective capacitance to increase.
⇒ Use empirical formulas to estimate.

Also have inter-layer capacitances (from p. 196 of text):



Very computationally intensive to calculate

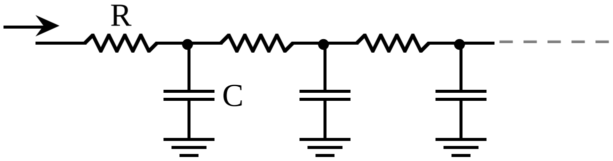
Requires 3-D CAD

Typically, just use substrate capacitance multiplied by a "fudge" factor of ~1.1, ~1.3, or even ~2.0

CONDITION	LAYER	LINE-to-GROUND EQUATION # (see text)	LINE-to-LINE EQUATION # (see text)
A	Poly-substrate	4.19	4.21
B	Metal2-substrate	4.19	4.21
C	Poly-metal2	4.20	4.22
D	Metal1-substrate	4.20	4.22
E	Metal1-poly	4.20	4.22
E	Metal1-metal2	4.20	4.22
F	Metal1-diffusion	4.20	4.22
G	Metal2-diffusion	4.19	4.21

Delay

Long wire ⇒ distributed RC line



First-order approximation:

$$\text{delay} = \frac{r \cdot c \cdot l^2}{2}$$

where

r = resistance per unit length
c = capacitance per unit length
l = length of the wire

Important fact ⇒ interconnect delay does not scale with lambda, it is constant. When lambda decreases, R increases and C decreases, resulting in delay constant

Inserting a buffer in a long resistance line can be advantageous.

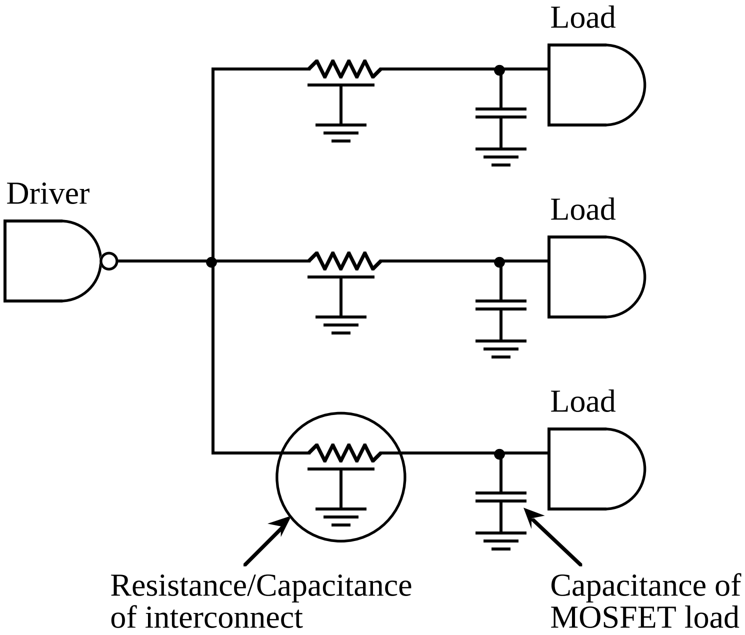
For a poly run = 2mm length,

$r = 20\ \Omega/\mu m$
$c = 4 \times 10^{-4}\ pF/\mu m$

$$\text{2mm delay} = \frac{20 \times 4 \times 10^{-4} \times (2)^2}{2} = 16\ \text{ns}$$

If broken into two 1mm sections, then delay of each section = 4ns. Add a buffer with delay = 1ns and total delay becomes 4 + 1 + 4 = 9ns.

Typically, resistive effects of interconnect much more important than capacitive effects since capacitance tends to be dominated by the gate capacitances.


Load
Driver
Load
Load
Resistance/Capacitance of interconnect
Capacitance of MOSFET load

MOSFET load capacitance >> wire capacitance [unless DSM (deep submicron (≤ 0.25μm) CMOS technology]

So, if we decrease interconnect resistance, then we reduce overall propagation delay between driver and load.

Reduce interconnect resistance by using metal, increasing the width of the interconnect.

Usually just want delay (RC), where R is the resistance of the interconnect and C is the total of all the capacitive loads.

Example (from text)

A register that fits in data-path is 25µm tall (the direction of repetition). A metal2 clock line runs vertically to link all registers in an n-bit register. The register has 30µm of 1µm metal1, 20µm of 1µm poly (over field oxide), and 16µm of 1µm gate capacitance.

1. Calculate the per-bit clock load and the load for a 16-bit register.

2. What would be the *RC* delay of the register from a clock buffer using 5mm of 1µm metal2 (0.05Ω/sq.)?

3. How wide would the clock line have to be to keep the skew below 0.5ns if a register file containing 32 16-bit registers was fed with the same 5mm metal2 wire?

Solution:
[Capacitance values found in Table 4.6, page 202 of text.]

1. The parasitics are as follows:

$C_{\text{m1}} = 30 \times 30$ [aF] = 900aF

$C_{\text{poly}} = 20 \times 50$ [aF] = 1000aF = 1fF

$C_{\text{gs}} = 16 \times 1800$ [aF] = 28,800aF

$C_{\text{reg1}} = 900 + 1000 + 28{,}800$ [aF] = 30fF

$C_{\text{reg16}} = 16 \times C_{\text{reg1}}$ = 480fF

2. $R_{\text{metal2}} = 5000 \times 0.05$ [Ω/sq.] = 250Ω

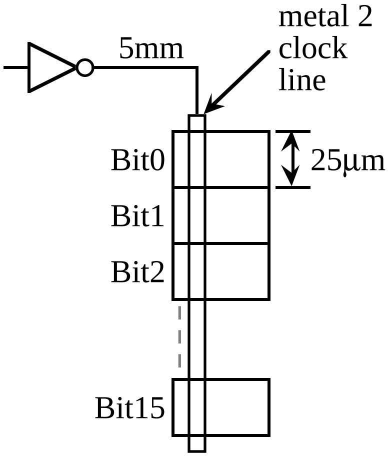
Because the capacitance load is at the end of the wire, we approximate the *RC* delay by adding the metal2 track capacitance to the load capactiance and performing a simpe *RC* calculation.

C_{total} = 0.48 + C_{metal2} [pF]
= 0.48 + $(5000 \times 20 \times 10^{-6})$ [pF]
= 0.58pF

RC = $250 \times 0.58 \times 10^{-12}$ seconds
= 0.145ns

3. We now have 32 registers, so the load capacitance of the registers is

$C_{\text{regfile}} = 32 \times C_{\text{reg16}}$ = 15.36pF.



The *RC* for a 1µm-wide clock feed is
250Ω × 15.36pF = 3.84ns.

Delay of 3.84ns too big, <u>widen</u> the wire to reduce <u>R</u>; will increase C somewhat but capacitance is dominated by cell capacitance.

The clock line has to be widened by 3.84/0.5 or 7.68. To be conservative, one might choose a 10µm wire.

Now

C_{total} = 15.36 + C_{metal2} [pF]
= 15.36 + $(5000 \times 10 \times 20 \times 10^{-6})$ [pF]
= 16.36pF

Note: R reduced by 10x, C_{total} slightly increased

RC = $25 \times 16.36 \times 10^{-12}$ seconds
= 0.41ns

Overall delay went down!

For short and lightly loaded wire lengths, can ignore the R and just model wires as lumped capacitances.

How short?

$$\tau_w << \tau_g$$

$$\tau_w = \frac{r \cdot c \cdot l^2}{2}$$

$$l << \sqrt{\frac{2\tau_g}{rc}}$$

Minimum width (1µm) Aluminum wire, gate delay = 200ps

$$l << \sqrt{\frac{2 \times 0.2 \times 10^{-9}(\tau_g)\ \lambda^2}{0.05[r] \times 30 \times 10^{-18}[c]}}$$
$$\approx 16000\lambda$$

So conservatively,

$$l < 5000\lambda.$$

Guidelines for ignoring RC wire delays:

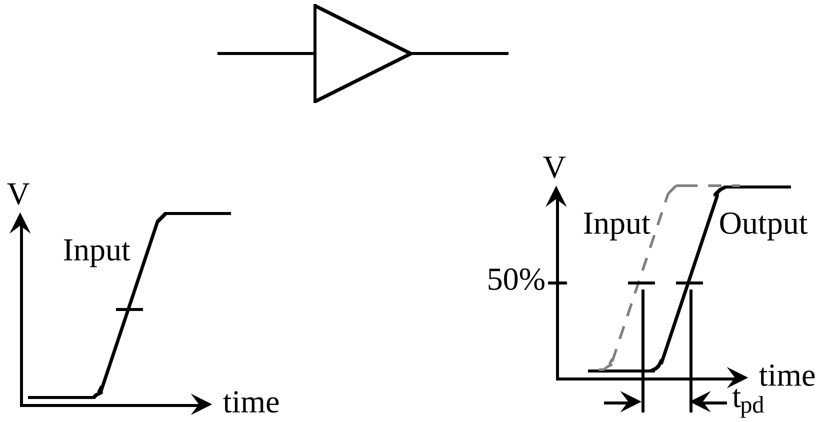
LAYER	MAXIMUM LENGTH (λ)
Metal3	10000
Metal2	8000
Metal1	5000
Silicide	600
Polysilicon	200
Diffusion	60

If lambda = 0.5µm, ignore RC delay for < 2.5mm metal runs.

Do NOT ignore for heavily loaded lines like clock lines!

Gate Delay Models for Rise/Fall Time

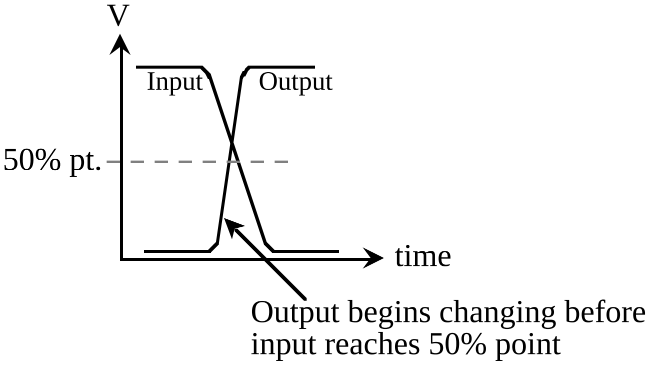
Definition of Rise/Fall Delay Times:



$t_{delay,50\text{-}50}$ (or t_{pd}) = time between input reaching 50% point and output reaching 50% point

One advantage of using 50% points for measurement is that it does not matter if output is rising or falling (gate inverting or non-inverting).

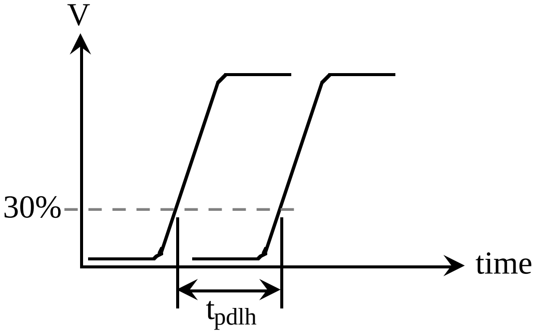
One problem with 50% propagation delays is that you can end up with a negative propagation delay for slowly rising/falling inputs.



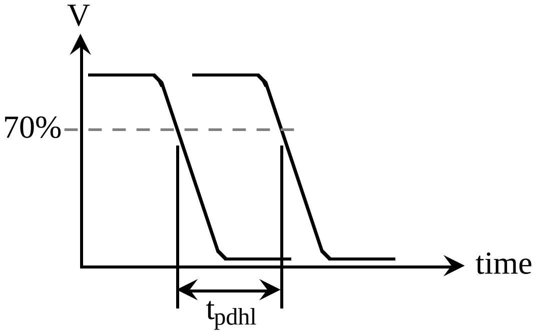
Can also define delay at 30% - 70% points, 10% - 90% points, etc.

For non-inverting gates, if we use 30% - 70% points:

t_{pdlh} - prop delay low to high (measure between 30% input, 30% output)

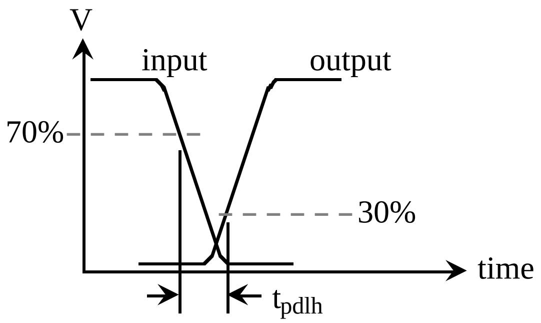


t_{pdhl} - prop delay high to low (measure between 70% input, 70% output)

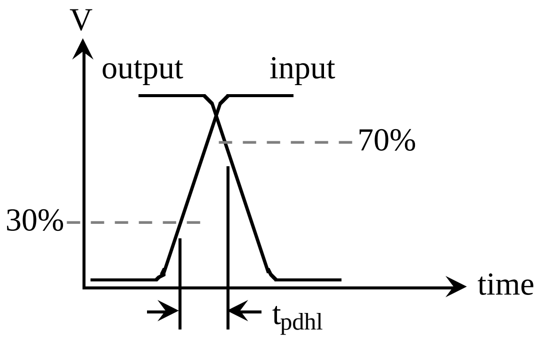


For inverting gates, if we use 30% - 70% points:

t_{pdlh} - measure 70% input to 30% output



t_{pdhl} - measure 30% input to 70% output


V
output
input
70%
30%
time
t_{pdhl}