Microelectronics Essentials

Philipp Häfliger

Spring 2024

Edition 0.3 (beta-edition)
Department of Informatics, University of Oslo, Oslo 22-Jan 2024
Edition 0.1 E-book ISBN: 978-82-7368-466-0

Change Log

edition	date	change
0.1 (beta)	22-Jan-2024	base beta edition
0.2 (beta)	22-Jan-2024	Index had not been compiled, so a small correction just hours after having registered 0.1
0.3 (beta)	25-Jan-2024	In large signal model: clarifying equation forms where voltages only appear as explicit differences (can be thought of as relative to -Vss or other global reference) and equation form without explicit differences where $V_B := 0$ is the reference.

Abstract

While research is on the hunt for an even smaller replacement, silicon as a base material and the microscopic field effect transistor (FET) is for now still the essential building block of microelectronics. A central- or a graphics processing unit (CPU/GPU) contains billions of the things, and sensors and actuators in our everyday commodity devices, industrial machines, medicine and space exploration, are read-out and controlled by transistor circuits.

This lecture script aims at enabling the reader to use FET models to construct some of the more common composite building blocks of both digital and analog microelectronics. It teaches how to manipulate design parameters at transistor level to optimize logic gates and simple amplifiers, and showcases the inner workings of those devices. The reader shall learn their non-idealities and pitfalls to better be able to use them with confidence for higher order digital-, analog-, and mixed-signal systems, and to tweak the elemental electronics building blocks for optimal performance.

IMOPRTANT: the e-book PDF you are looking at here is the beta edition 0 of Spring 2024 (starting from beta 0.1 with major edits marked as subsequent 0.x). It is mainly intended to be used by students of the lecture IN3170 at University of Oslo that Spring term. While it DOES contain all the parts and necessary descriptions of the models for the course curriculum, it will still be edited and updated during that first term, based on student feedback and feedback from colleagues. There is a first edition (1.1) planned in Fall 2024 after that process, which will have undergone much more quality control than this beta edition here.

Acknowledgment

This book results from a lecture script for a basic microelectronics course at the University of Oslo and was first used in lieu of pre-existing standard textbooks for the first time in the spring term of 2024. Consecutive editions have been improved based on the feedback of the students in the course and by colleagues at the department. I want to shout out to them and give them my heartfelt thanks!

Contents

List of Figures

List of Tables

Prologue

This is a book about CMOS integrated circuits. CMOS technology with the field effect transistor (FET) at its heart is (and will remain for a some time still) the predominant technology upon which all our electronics is built.

The cynical character Dogbert in a Dilbert cartoon once suggested that a standard component for any arbitrary presentation is a chart of Moore's law. I'll partially follow his advice here by at least remarking upon that law: it has quite precisely predicted the exponential growth in density of transistors from when they were first demonstrated as integrated circuits in 1958/1959 until today. Moore predicted in 1965 that the density would approximately double every year and revised his prediction in 1975 to a doubling every second year. Since then, his prediction has proven very apt. Some claim it has been a self-fulfilling prophecy as it has been adapted as a goal by the industry.

In recent years many experts have started to predict the end of that trend due to various physical limits to miniaturization. Interestingly, more recent voices attribute the coming end of Moore's law not so much to physical limits but rather to economical considerations: ever more companies that have been driving the miniaturization of integrated circuits have dropped out of the race, simply because their business model suggests they will never get their investment back. That is because not only the transistor density grows exponentially but also the development cost to achieve the next step in miniaturization. So only a handful of the biggest CMOS foundries remained and offer 7nm and smaller integrated circuit technology, while others do no longer drive towards more density and miniaturization, but offer more specialized properties of their integrated circuit devices and structures for specific niche markets.

In the meantime, completely different materials and computational principles start to enter into competition with (or supplement) semiconductor based integrated electronics, such as carbon nanotube transistors and quantum computers to name but two.

So while Moore's law will come to an end at some point in the near future, and FETs as the essential building block of computational devices are getting competition: for now and some time to come semiconductor electronics will still be the substrate on which most of our information processing will run, and the content of this book will remain useful. Philipp Häfliger

Oslo, 20-Dec-2023

Part I

Transistor Models

Chapter 1

Field Effect Transistor Cross Sections

Figures 1.1, 1.2 and 1.4 show cross sections of **complementary metal oxide semiconductor (CMOS) n-channel field effect transistors (nFETs)**. CMOS FETs are also referred to as **MOSFETs**. They are four-terminal devices where the terminals are denominated as **'bulk' (B), 'gate' (G), 'source' (S) and 'drain' (D)**. nFETs have a (P-)-doped bulk and (N+)-doped source/drain, and vice versa for **pFET** devices. The '+' stands for a relatively high doping concentration and the '-' for a relatively low doping concentration. Oftentimes the device's anatomy is symmetric with respect to source and drain. Which of the two terminals is called what is actually determined by the voltages applied: for nFETs the 'source' is the terminal with the lower voltage and vice versa for pFETs. One of the most common uses of FETs is to have a fixed voltage difference between drain and source and then to use the voltage of the gate to control the conductance and/or current between the two terminals. Note that the gate terminal is insulated, so while it can have different voltages no DC current flows in or out. Pure electrostatic properties let it modulate the conductive channel between source and drain by attracting or driving away free (i.e. mobile) electric carriers. In classical mode of operation the bulk is simply at a fixed voltage, -Vss/Gnd (i.e. the low supply voltage of an entire region of the integrated circuit) for P-bulks (i.e. nFETs) and Vdd (i.e. the high supply voltage for a region) for N-bulks (pFETs), but one actually *can* modulate its voltage and thus modulate the channel conductivity in the same manner as with the gate but with less efficiency. The bulk is then refered to as *back gate*. In **bulk CMOS** (figure 1.1) the P-bulk cannot be at a higher potential than the source/drain terminals, because then the PN-junction between bulk and the source/drain would be conducting and source/drain would be shorted with the bulk. So the range of 'allowed' voltages is more limited for the bulk than for the gate. This however, is not the case for a **silicon on insulator (SiO)** FET device (figure 1.2), where the back gate is more worry-free to use. Furthermore, SiO technologies have no PN-junction capacitance at their source/drain terminals and thus less 'parasitic' capacitance. Consequently, SiO technologies are popular for high speed and RF applications. The most modern CMOS technologies with gate lengths below 14nm use **FinFET** topologies to ensure better

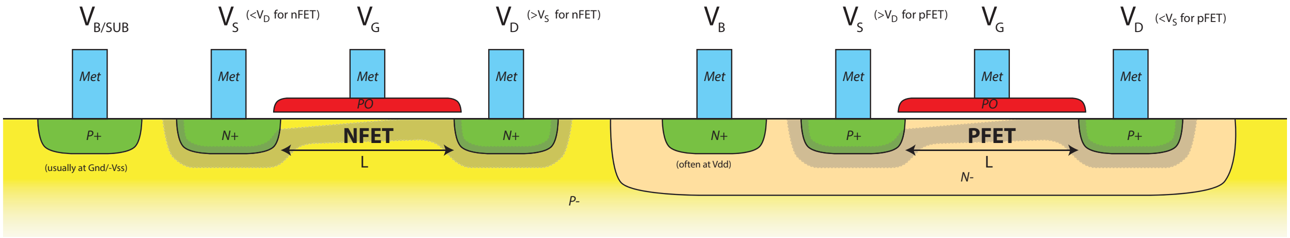


Figure 1.1: **Bulk CMOS** nFET (left) and pFET (right) cross section. Bulk CMOS is the original CMOS technology and still in use today, albeit not for the most downscaled CMOS processes. In normal biasing conditions, depletion regions (i.e. regions with no free/mobile carriers marked in grey) will insulate source and drain from the bulk, and separate a 'channel' of free carriers from the bulk as well. The fact that there is a depletion region in the PN-junctions is referred to as a 'reversed biased' PN-junction, i.e. with a lower voltage at the P-anode and a higher voltage at the N-cathode. The depletion region underneath the gate is due to the gate bulk voltage being biased in **inversion**, i.e. her for the P-bulk a higher voltage on the gate than on the bulk, driving the positively charged majority carriers away from the surface and even inverting the channel by attracting minority carriers, i.e. freeing electrons from their bonds rendering them mobile. The opposite effect is known as **accumulation**, i.e. a lower voltage on the gate than the bulk that increases the number of free majority carriers. However, this is not a state a FET is biased in when controlling current flow between drian and source.

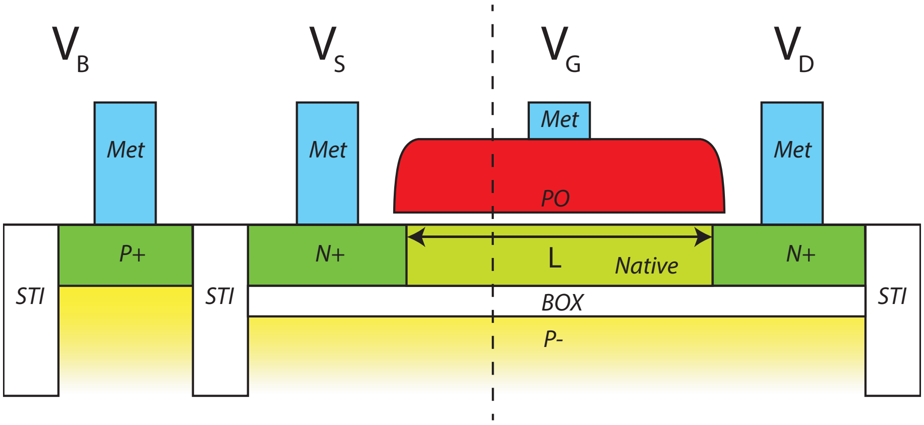


Figure 1.2: **Silicon on insulator (SOI)** nFET cross section for a variant with burried oxide (BOX). there are some other 'flavours' around, e.g. silicon on saphire. Advantages to bulk CMOS are reduced capacitance at the source and drain terminals that do not have junction capacitance, as well as a bigger voltage range when using the bulk as back gate (No need to be wary of forward biassed junction). In essence the cross section of a FinFET device looks the same along the channel, but is different in the cross section *across* the channel, i.e. along the dashed line. See figure 1.4

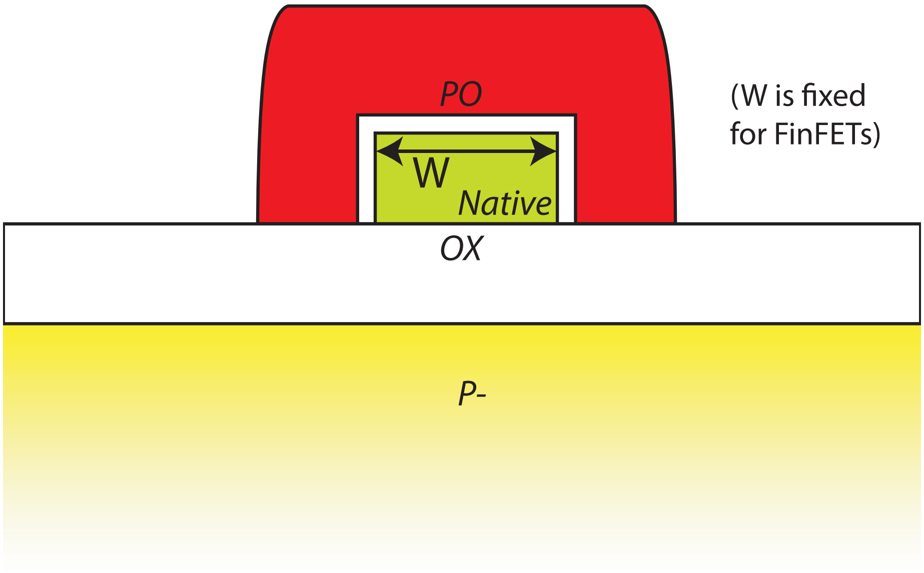


Figure 1.3: **FinFET technology** cross section *across* the channel. In contrast to Bulk and traditional SiO, here the gate 'folds' around the channel. FinFET technology is used for technologies of 14nm and smaller. A notable difference here is that the width of the channel is now fixed. So one has to use multiple transistors in parallel instead of just widening a channel. FinFETs can be produced both in SOI (shown here) and bulk CMOS (where there would be no burried oxide (BOX) and the channel/bulk material might be doped instead of native).

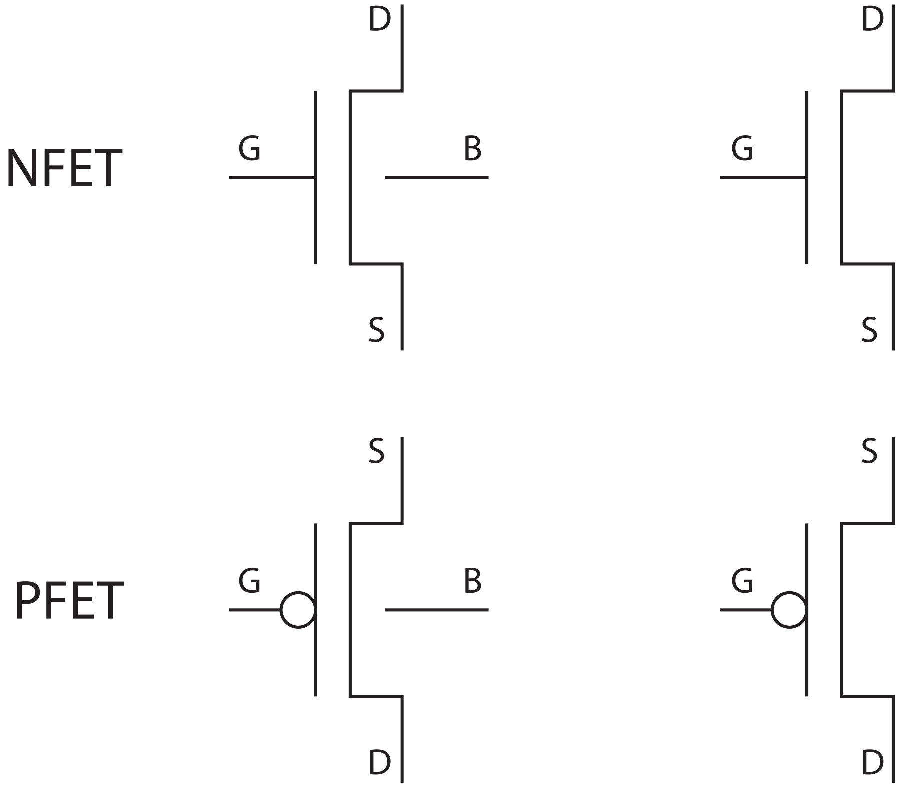


Figure 1.4: The symbols for NFETs and PFETs used in this book. If the bulk is ommitted it is implicitly connected to -Vss/Gnd for nFETs and to Vdd for pFETs.

control of channel conductivity at lower voltages for these maximally miniaturized devices. The gate is wrapped around the channel on three sides (see figure 1.4).

Chapter 2

FET Large Signal Models

2.1 I-V relationships

Large signal models are the most complete models discussed in this book and describe the transistor behaviour for any combination of a large range of voltages or currents applied to the four terminals of the MOSFET. The voltage range is limited by the **supply rails** given for a specific CMOS processing technology. Supply rails in this book shall be denominated as Vdd and $-Vss$. Generally the signal zero reference $Gnd := 0\text{V}$ for (analog) circuits is between those rails. However, sometimes it is defined as $Gnd = -Vss = 0\text{V}$, mostly for discussion of digital circuits.

For FETs the large signal model generally describes a function for the drain current I_D in dependency of the four terminal voltages V_S, V_G, V_D, V_B.

$$I_D = f(V_G, V_S, V_D, V_B) \tag{2.1}$$

The full model is often simplified some more, for instance by assuming that every voltage is referenced to $V_B := 0V$, oftentimes without explicitly mentioning this, so reducing the transistor to a three terminal device:

$$I_D = f(V_G - V_B, V_S - V_B, V_D - V_B) := f(V_{GB}, V_{SB}, V_{DB}) := f(V_G, V_S, V_D) \tag{2.2}$$

There are a few different schools of models. Here we shall base our discussion on the so called EKV model [3]:

$$I_D = I_F - I_R \tag{2.3}$$

Where the drain current I_D is the current *into* the drain node. The forward current I_F in the direction of I_D from drain to source and reverse current I_R in the opposite direction for nFETs are defined as:

$$I_{F(R)} = I_S \ln^2 \left[1 + e^{\frac{\left[V_G - V_{S(D)}\right] + (n-1)\left[V_B - V_{S(D)}\right] - V_t}{2nU_T}} \right] (1 + \lambda \left|V_D - V_S\right|) \tag{2.4}$$

Where

$$I_S = 2nkU_T^2 \tag{2.5}$$

and

$$k = \mu C_{ox}\frac{W}{L} \tag{2.6}$$

Most parameters in this equations belong into two categories, **process parameters** and **design parameters**.

Process parameters are given by the choice of CMOS technology to produce an integrated circuit and are usually provided by a CMOS foundry to it's customers. They are the **threshold voltage $\boldsymbol{V_t}$**, the **carrier mobility $\boldsymbol{\mu}$**, the **gate oxide capacitance per area $\boldsymbol{C_{ox}}$**, the **slope factor $\boldsymbol{n}$**, as well as the parameter λ' to derive the **channel length modulation parameter $\boldsymbol{\lambda}$** that is $\lambda' = \lambda L = \frac{1}{V_A'}$ (see further along). Sometimes the foundry provides a small number of different device options in the same CMOS process, i.e. a set of device parameters that allows to produce transistors with different base properties in the same technology, e.g. very commonly there are different parameters for pFETs and nFETs (here we will distinguish these parameters with an extra index n or p, where relevant, such as V_{tn} or μ_p), but also different device options within these two major groups, such as low and high threshold devices.

Then the design parameters are parameters that the circuit designer can choose within a given CMOS technology, i.e. the device's **channel width $\boldsymbol{W}$** and **channel length $\boldsymbol{L}$**.

Finally the **thermal voltage $\boldsymbol{U_T}$** is a temperature dependent physical parameter independent of process technology or design choices. It is 26mV at room temperature and we'll treat it as constant in the rest of this text, as we'll not dive into temperature dependencies of CMOS devices. However, be aware that CMOS devices are *very much* dependent on temperature and that dependency needs to be considered when designing products that will be exposed to different temperatures (as most products will be!).

Note that in the form equation (2.7) is given, all voltages appear as differences between two voltages, so it does not matter what voltage is defined as a zero/Gnd reference. However, sometimes the equation is given in a form where the bulk voltage is used as reference, i.e. $V_B := 0$. Then the above equation can be written as:

$$I_{F(R)} = I_S \ln^2\left[1 + e^{\frac{V_G - nV_{S(D)} - V_t}{2nU_T}}\right](1 + \lambda\left|V_D - V_S\right|) \tag{2.7}$$

But for now we'll stick with the form (2.7) where we keep V_B in the equation and the zero reference may be -Vss, for example.

Equation (2.7) is the equation for *nFETs*! For pFETs on needs to inverse the differences in the equation, i.e. $V_{S(D)} - V_G$ instead of $V_G - V_{S(D)}$ and $V_{S(D)} - V_B$ instead of $V_B - V_{S(D)}$, and the sign of the expressions for I_F and I_R is inverted:

$$I_{F(R)} = -I_S \ln^2\left[1 + e^{\frac{\left[V_{S(D)} - V_G\right] + (n-1)\left[V_{S(D)} - V_B\right] - V_t}{2nU_T}}\right](1 + \lambda\left|V_D - V_S\right|) \tag{2.8}$$

Note that the naming of S and D is arbitrary for this symmetric device. However, by convention we shall assume in the following that for nFETs the

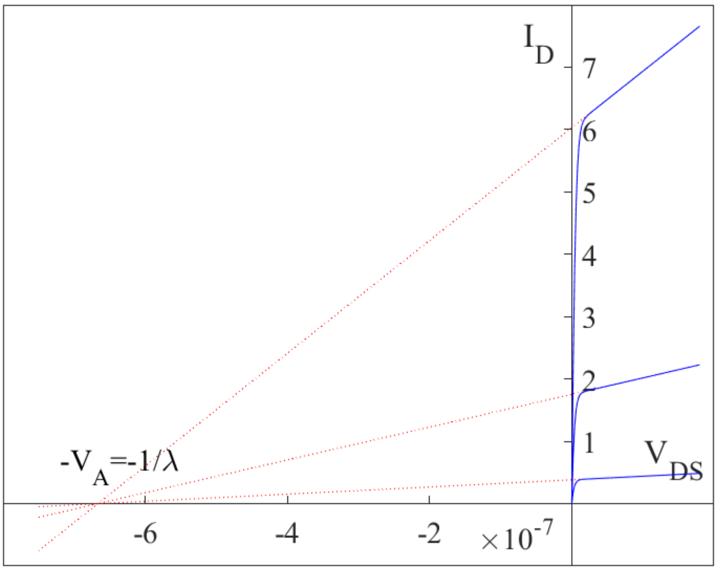


Figure 2.1: How to derive parameter λ graphically from a I_D vs V_{DS} plot: Extrapolating the straight line of the plot in the saturation region will cut the V_{DS} axis at $-V_A = \frac{1}{\lambda}$.

S-terminal is the one with the lower voltage than the D terminal, and vice versa for pFETs. This way we ensure that $I_F > I_R$, and consequently the drain current flows *into* a nFET's drain terminal, whereas it flows *out* from pFET's drain terminal (i.e. the result from equation (2.8) will be negative).

Oftentimes one abbreviates $V_G - V_S - V_t$ as the **overdrive voltage V_{OV}** (e.g. in [2], while others call it the **effective voltage V_{eff} [1]**, but we'll stick with V_{OV} in this compendium):

$$V_{OV} := V_G - V_{S(D)} - V_t \tag{2.9}$$

So one can rewrite (2.7) a bit more compactly:

$$I_{F(R)} = I_S \ln^2 \left[1 + e^{\frac{V_{OV}+(n-1)\left[V_B - V_{S(D)}\right]}{2nU_T}} \right] (1 + \lambda \left|V_D - V_S\right|) \tag{2.10}$$

Note that parameter λ is sometimes alternatively expressed as the Early Voltage $V_A = \frac{1}{\lambda}$ and V_A is proportional to the transistor length L and thus sometimes expressed as $V_A = V_A' L$, where $V_A' = \frac{1}{\lambda L}$ is a process parameter. There is a graphical way of determining V_A from a I_D vs V_{DS} plot (see figure 2.1): if one extends the slope of the curve in the saturation region all the way to where it intersects the V_{DS}-axis, the value of V_{DS} at that intersection is equal

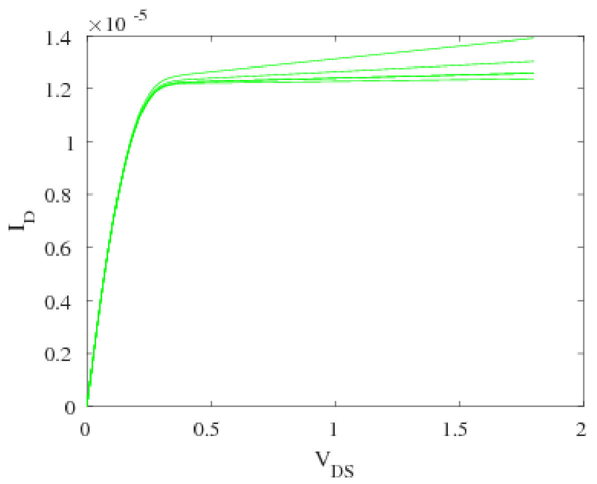


Figure 2.2: Typical I_D vs V_{DS} plots for varying λ. λ is the slope of I_D in the saturation region relative to the magnitude of I_D when $\lambda = 0$ (i.e. where the extension of the saturation region slope would hit the I_D-axis). The parameters used have been gleaned from a 180nm process and were $k = 270\frac{\mu\text{A}}{V^2}$, $\frac{W}{L} = 1$, $V_t = 0.45$V, $U_T = 26$mV, $n = 1$, $V_{GS} = 0.75$V (i.e. $V_{OV} = 0.3$V). λ was varied $\lambda = \{0.01, 0.02, 0.04, 0.08\}$

to $-V_A$.

Oftentimes one thinks of this formula as mostly determined by the difference of the gate voltage to the source voltage $V_{GS} := V_G - V_S$ and to a lesser degree by the difference of the drain to the source voltage $V_{DS} := |V_D - V_S|$, i.e. ignoring the dependency on V_{SB} which is completely appropriate if the so called slope factor $n = 1$.

$$I_{F(R)} = I_S \ln\left[1 + e^{\frac{V_{GS(GD)} - V_t}{2U_T}}\right]^2 (1 + \lambda(V_{DS})) \qquad (2.11)$$

However, if $1 < n \leq 2$ and $V_S > V_B$, the same V_{GS} will now result in a smaller I_F. Then one can think of the difference of the bulk voltage to the source voltage having the same effect as the gate voltage to the source voltage, but with reduced efficiency by a factor $n - 1$. This is sometimes referred to as the **back gate** effect

This EKV model has been introduced to unify in a single equation the earlier models that used 4 different formulas that are used dependent on regions of operation of the device. These regions depend on the voltages V_{GS} and V_{DS} as depicted in an example in figure 2.5. When clearly within one of those regions of operation, the EKV model converges to those formulas. They are still easier to use for hand calculations in this case. However, be careful when close to the borders of these regions. Then the simplified equations will be quite wrong and the full formula (2.7) has to be used. Figure 2.6 illustrates the danger of relying on these simplified models when being in moderate inversion, where especially the strong inversion approximation, can give very wrong results. Similarly when

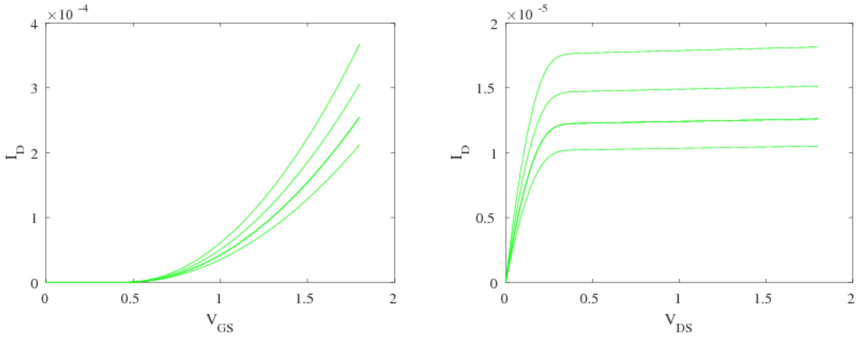


Figure 2.3: Typical I_D vs V_{GS} and V_{DS} plots for varying $I_S = 2nkU_T^2$. The parameters used have been gleaned from a 180nm process and were $V_t = 0.45$V, $U_T = 26$mV, $n = 1$. For the V_{GS}-plot $V_{DS} = 1.8$V. For the V_{DS}-plot, $V_{GS} = 0.75$V (i.e. $V_{OV} = 0.3$V). I_S was varied as $I_S = \{304.2, 365.0, 438.0, 525.7\}$nA. Note that in a linear V_{GS}-plot, one cannot really see the weak inversion currents, as they are smaller than I_S, i.e. orders of magnitude smaller than the highest strong inversion currents. A Logarithmic I_D-axis would reveal those currents (see figure 2.6) for an example.

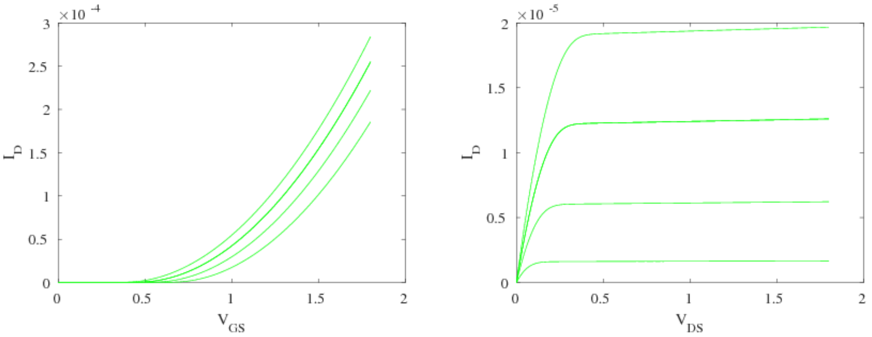


Figure 2.4: Typical I_D vs V_{GS} and V_{DS} plots for varying V_t. The parameters used have been gleaned from a 180nm process and were $k = 270\frac{\mu\text{A}}{V^2}$, $\frac{W}{L} = 1$, $U_T = 26$mV, $n = 1$, $V_{GS} = 0.75$V (i.e. $V_{OV} = 0.3$V). For the V_{GS}-plot $V_{DS} = 1.8$V. For the V_{DS}-plot, $V_{GS} = 0.75$V (i.e. $V_{OV} = 0.3$V). V_t has been varied as $V_t = \{0.375, 0.450, 0.540, 0.648\}$V.

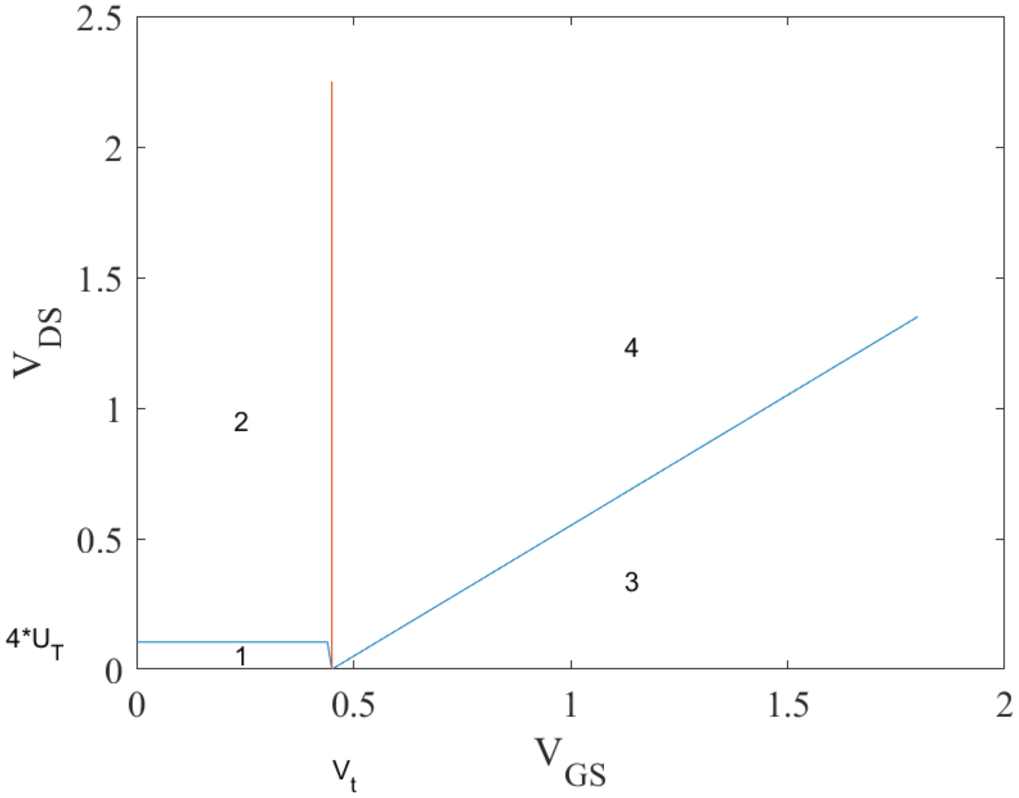


Figure 2.5: Four regions of operation where the simplified FET large models can be applied. However, the full model has to be used when near the transitions of these regions. 1) weak inversion and conduction; 2) weak inversion and saturation; 3) strong inversion and conduction; 4) strong inversion and saturation.

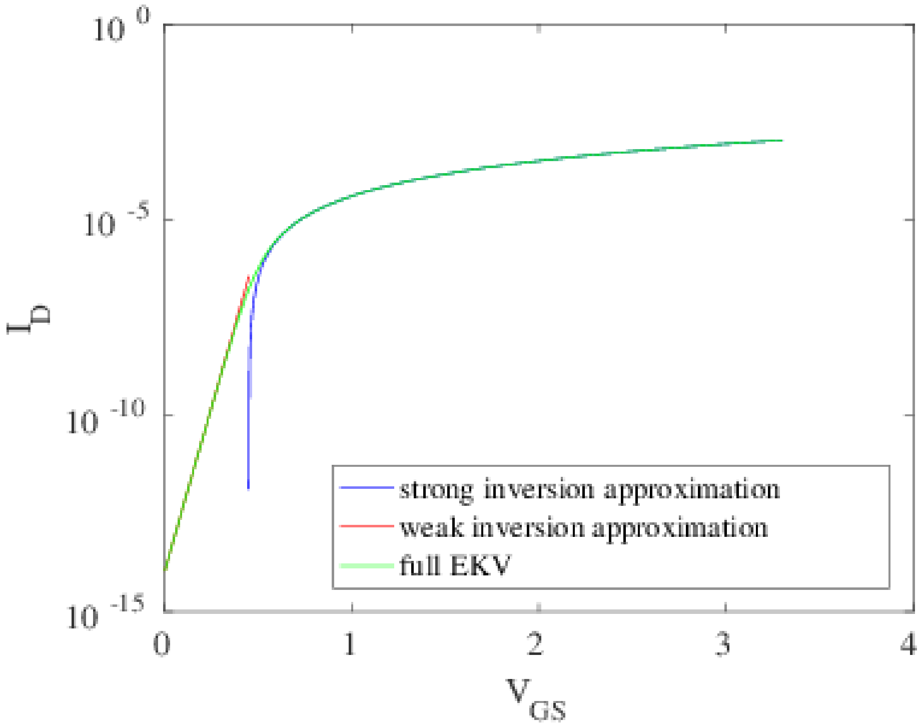


Figure 2.6: Illustrating the shortcomings of the simplified models in specific regions of operation. Here we show an I_D vs V_{GS} plot with a logarithmic I_D-axis, to show both the weak and strong inversion region clearly. The simplified models in red and blue, are discontinuous and non-differentiable at the point of separation V_t between weak- and strong inversion here. Only the full EKV in green offers a smooth transition and correct results in moderate inversion.

operating with V_{DS} close to the saturation voltage V_{sat}, the results will be inaccurate. So always verify that the simplified models can actually be applied, i.e. that you are a good way away from the separating voltages, i.e. from V_t for V_{GS} and from V_{sat} for V_{DS} when using those models!

As introduced before V_t is a constant and it is the point of separation for the gate to source voltage V_{GS} separating the **weak inversion** region from the **strong inversion** region:

$$V_{GS} \begin{cases} << V_t : \text{ weak inversion rules can be used} \\ \approx V_t : \text{ moderate inversion: full EKV needs to be used} \\ >> V_t : \text{ strong inversion rules can be used} \end{cases} \tag{2.12}$$

The saturation voltage V_{sat} is not always constant. It depends on whether the device is operated in strong or weak inversion:

$$V_{sat} = \begin{cases} 4U_T \text{ if } V_{GS} << V_t \\ 4U_T \text{ if } V_{GS} \approx V_t \text{ (i.e. use } 4U_T \text{ as a minimum until } \frac{V_{GS}-V_t}{n} > 4U_T) \\ \frac{V_{GS}-V_t}{n} \text{ if } V_{GS} >> V_t \end{cases} \tag{2.13}$$

Its the point of separation on the V_{DS} axis between the **conduction** region and the **saturation** region:

$$V_{DS} \begin{cases} << V_{sat} : \text{ conduction rules can be used} \\ \approx V_{sat} : \text{ full EKV needs to be used} \\ >> V_{sat} : \text{ saturation rules can be used} \end{cases} \tag{2.14}$$

So these two separators leave us with 4 distinct regions as depicted in figure 2.5. Her are the simplified equations one can employ if clearly within one of these regions (but not (!) if one is close to the transition between those regions!). Note that for these equations we write tehm in the form where all voltages are defined relative to $V_B := 0$ as zero reference potential.

strong inversion, saturation, nFET

Conditions for it to be applicable:

1. $V_{GS} >> V_t$ or equivalent $I_F >> I_S$ (condition for strong inversion)
2. $V_{DS} >> V_{sat} := \frac{V_{GS}-V_t}{n}$ (⚠: different for weak inversion) or equivalently $I_F >> I_R$ (condition for saturation)

Simplified equation for this region:

$$I_D = \frac{1}{2n} k_n \left(V_G - V_{tn} - nV_S\right)^2 \left(1 + \lambda V_{DS}\right) \tag{2.15}$$

I_D vs V_{GS} is a square function where:

$$I_D \propto \left(V_G - nV_S - V_t\right)^2 \tag{2.16}$$

I_D vs V_{DS} is a linear function where:

$$I_D \propto (1 + \lambda V_{DS}) \tag{2.17}$$

A square law governs the I_D vs V_{GS} behaviour. Many older books do in fact only discuss the strong inversion cases as weak inversion currents were considered negligibly small.

The dependency of I_D vs V_{DS} is linear: $I_D \propto (1 + \lambda V_{DS})$ (both strong and weak inversion). For $\lambda = 0$ the V_{DS} curve is completely flat, i.e. the FET is behaving like an ideal current source producing the same current irrespective of the voltage V_{DS}.

weak inversion, saturation, nFET

Conditions for it to be applicable:

1. $V_{GS} << V_t$ or equivalent $I_F << I_S$ and $I_R << I_S$ (condition for weak inversion)
2. $V_{DS} >> V_{sat} := 4U_T$ (⚠: different for strong inversion) or equivalently $I_F >> I_R$ (condition for saturation)

Simplified equation for this region:

$$I_D = I_S e^{\frac{V_G - V_{tn} - nV_S}{nU_T}} (1 + \lambda V_{DS}) \tag{2.18}$$

I_D vs V_{GS} is an exponential function where :

$$I_D \propto e^{\frac{V_G - nV_S - V_t}{nU_T}} \tag{2.19}$$

I_D vs V_{DS} is a linear function where:

$$I_D \propto (1 + \lambda V_{DS}) \tag{2.20}$$

So in weak inversion in saturation for I_D vs V_{GS} this exponential law applies ($I_D \propto e^{\frac{V_{GS} - V_t}{nU_T}}$).

And for I_D vs V_{DS} we get the same proportional linear relationship ($I_D \propto (1 + \lambda V_{DS})$) like in strong inversion, i.e. a current source-like behaviour for small λ.

strong inversion, conduction

Conditions for it to be applicable:

1. $V_{GS} >> V_t$ or equivalent $I_F >> I_S$ or $I_R >> I_S$ (condition for strong inversion)
2. $V_{DS} << V_{sat} := \frac{V_{GS} - V_t}{n}$ (⚠: different for weak inversion) or equivalently $I_F \approx I_R$ (condition for conduction)

Simplified equation for this region:

$$I_D = k_n V_{DS} \left[V_G - V_{tn} - \frac{n}{2}(V_D + V_S) \right] \tag{2.21}$$

I_D vs V_{GS} is a linear function.

I_D vs V_{DS} in deep conduction is a linear function where for $V_S = 0$ and $V_{DS} << V_{OV}$ the first order Taylor expansion in V_{DS} results in the behaviour of a resistor:

$$I_D = k_n V_{ov} V_{DS} \Rightarrow g_{DS} = k_n V_{ov} \tag{2.22}$$

So in the conduction region (both strong and weak inversion) the I_D vs V_{DS} behaves very much like a resistor, and one that can be tuned by V_{GS} at that.

weak inversion, conduction

Conditions for it to be applicable:

1. $V_{GS} << V_t$ or equivalent $I_F << I_S$ and $I_R << I_S$ (condition for weak inversion)

2. $V_{DS} << V_{sat} := 4U_T$ (⚠: different for strong inversion) or equivalently $I_F \approx I_R$ (condition for conduction)

Simplified equation for this region:

$$I_D = I_S e^{\frac{V_G - V_{tn}}{nU_T}} \left(e^{\frac{-V_S}{U_T}} - e^{\frac{-V_D}{U_T}} \right) \tag{2.23}$$

I_D vs V_{GS} is an exponential function.

I_D vs V_{DS} in deep conduction is a linear function, where for I_D vs V_{DS} and $V_S = 0$ close to the origin $V_{DS} << V_{ov}$ (1st order Taylor expansion around $V_{DS} = 0$):

$$I_D = I_S e^{\frac{V_{ov}}{nU_T}} \frac{V_D}{U_T} \Rightarrow g_{DS} = I_S e^{\frac{V_{OV}}{nU_T}} \frac{1}{U_T} \tag{2.24}$$

So a resistive behaviour $r_{DS} = \frac{1}{g_{DS}}$

Note that there are other names in use for those regions: Strong inversion is also called '**above threshold**' and weak inversion '**subthreshold**'. The saturation region is also referred to as 'active' region, and the conduction region as '**triode**' or '**linear**' region.

2.2 Capacitances

Looking at figure 1.1 and 1.2 in search of capacitances that add to the model one can find two types of structures that cause capacitors: Conductors separated by an insulator, and reversed biased PN-junctions (i.e. conductive regions separated by a depletion region). In figure 1.1 the depletion region for a transistor in saturation are sketched as grey areas. So it gives rise to a capacitance between source and bulk C_{SB}, and drain and bulk C_{DB}. The $\mathrm{SiO_2}$ insulating the gate from the channel is responsible for C_{GS}, as well as (generally smaller) C_{GD}. One gets the same capacitances also for a silicon on insulator FET (figure 1.2). However, it is the buried silicon dioxide (BOX) instead of a depletion region that governs C_{SB} and C_{DB}, which generally are smaller.

The plate capacitors' capacitance depends on a technology dependent capacitance per area, i.e. C_{OX} for the gate oxide and C_{BOX} for the buried oxide. Most straight forward is C_{GD} which is:

$$C_{GD} = L_{OV} W C_{OX} \tag{2.25}$$

where L_{OV} is the length of the overlap between the gate and the drain, also taking into account some 'length' accounting for the fringe capacitance along the edge of the gate towards the drain. This length is usually a constant value for a given technology, independent of the transistor dimensions. So note the take home message: C_{DB} increases with W.

A bit more tricky is C_{GS}. It is NOT actually constant and depends on all the transistor terminal voltages. This compendium, however, is not covering this and uses a common approximate voltage independent value:

$$C_{GS} \approx \left(\frac{2}{3} WL + WL_{OV}\right) C_{OX} \tag{2.26}$$

For SiO technologies (Fig. 1.2) the capacitances towards the bulk are functions of C_{BOX} and the source and drain areas respectively. For bulk technologies its the reversed biased PN-junctions as well as the depletion region beneath the channel. Junction capacitance in reversed biassed diodes is mostly dependent on the doping profile, but also not a constant for different voltages applied. This compendium is not dealing with these voltage dependencies either, but to give the students at least a qualitative understanding of the dependency on doping: higher doping differences lead to more narrow depletion regions and thus higher capacitance. Generally when doing circuit design this is not something the designer can choose, so we can only modify the source and drain areas to influence these capacitances. If we assume a fixed junction capacitance per area C_J at source and drain and a channel to bulk capacitance per area C_{ChB}, we get:

$$\begin{aligned} C_{DB} &= C_J A_D && (2.27) \\ C_{SB} &\approx C_J A_S + \frac{2}{3} WLC_{ChB} && (2.28) \end{aligned}$$

If the layout of the S/B terminals is rectangular with a width equal to the channel width W and minimal constant length $L_{S/D}$ then these capacitances are proportional with W as:

$$A_{S/D} = WL_{S/D} \tag{2.29}$$

Note that oftentimes it is important to also keep an eye out for other parasitic capacitances, e.g of interconnections between transistors, for overall circuit performance. Also metal wires, if not kept very short, will add capacitance to an electric node. If they are very long they need to be treated using cable equations, i.e. they are not a single electrical node but signals travelling along a cable can be changed along the way and reflected at the cable termination. This is again not something this compendium will cover, but can none the less be important when designing high performance/frequency (digital or analog) circuits.

Chapter 3

FET Small Signal Models

Small signal models are used in analogue circuit analysis. They are linearised models of the approximate behaviour of a circuit around a particular **point of operation (biasing point)** mainly used for analysis of analogue circuits. Linear circuit theory has over the decades accumulated a lot of methods and tools for circuit analysis. Unfortunately the model for FETs we have introduced so far is anything but linear. However, in the following we will introducer a linear approximation for the FET behaviour using 1^{st} order Taylor expansion around a **point of operation** . This linearised model can then only model the behaviour with any accuracy for **small signal** variations around this point of operation, in contrast to the full scale or **large signal** model we have introduced before.

3.1 FET behaviour linear approximation around a point of operation

A general expression for a first order Taylor expansion $i_D(v_G, v_S, v_D)$ of I_D in variables V_{GS} and V_{DS} of function (2.2) around a point of operation is:

$$i_{DS} = I_{DS}(V_{GS}, V_{DS}) + \delta v_{GS} * \frac{\delta I_{DS}}{\delta V_{GS}} + \delta v_{DS} * \frac{\delta I_{DS}}{\delta V_{DS}} \tag{3.1}$$

Look at figure 3.1 for an illustration. The blue lines are the large signal model and the red lines are the linear approximation around the DC point of operation/biasing point marked with the red circles expressed by equation 3.1.

Note here that we use upper case variable names and upper case indices for *biasing voltages/currents*, i.e. the values of the point of operation that are derived by large signal models and lower case variables with upper case indices for the linearized signals around that point of operation/biasing point. δ indicates an infinitesimally small deviation from a point of operation. As a further simplification of the notation we will also refer to the infinitesimally small variations indicated by the δ plus variable name with using lower case variables and index letters, so for example $v_{gs} := \delta v_{GS}$. These latter signals we refer to as the **small signal variables** or simply **small signals**. These small deviations from the point of operation are illustrated in the figure by the green

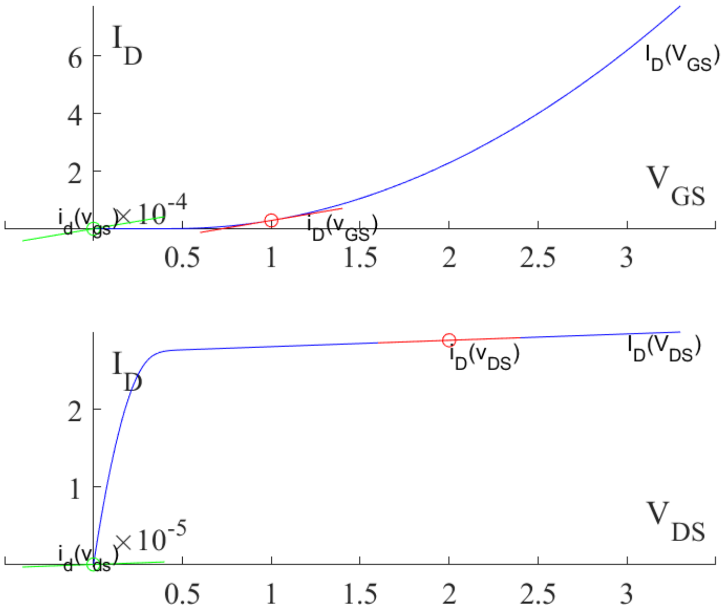


Figure 3.1: Illustrating the linear approximation of the nFET large signal model in a particular point of operation/biasing point. In blue there is the large signal behaviour, in red the linearized behaviour around an (arbitrarily chosen as $V_{GS} = 1$V and $V_{DS} = 2$V) point of operation/biasing point, and in green the small signal behaviour (i.e. the same as the linearized behaviour but 'ignoring' the biasing point/DC offset/point of operation). This is for an assumed 180nm CMOS technology with $\mu_n C_{ox} = 270\frac{\mu\text{A}}{\text{V}}$, Vdd=3.3V, $V_t = 0.45$V, $n = 1.5$, $\lambda = 0.03\frac{1}{\text{V}}$

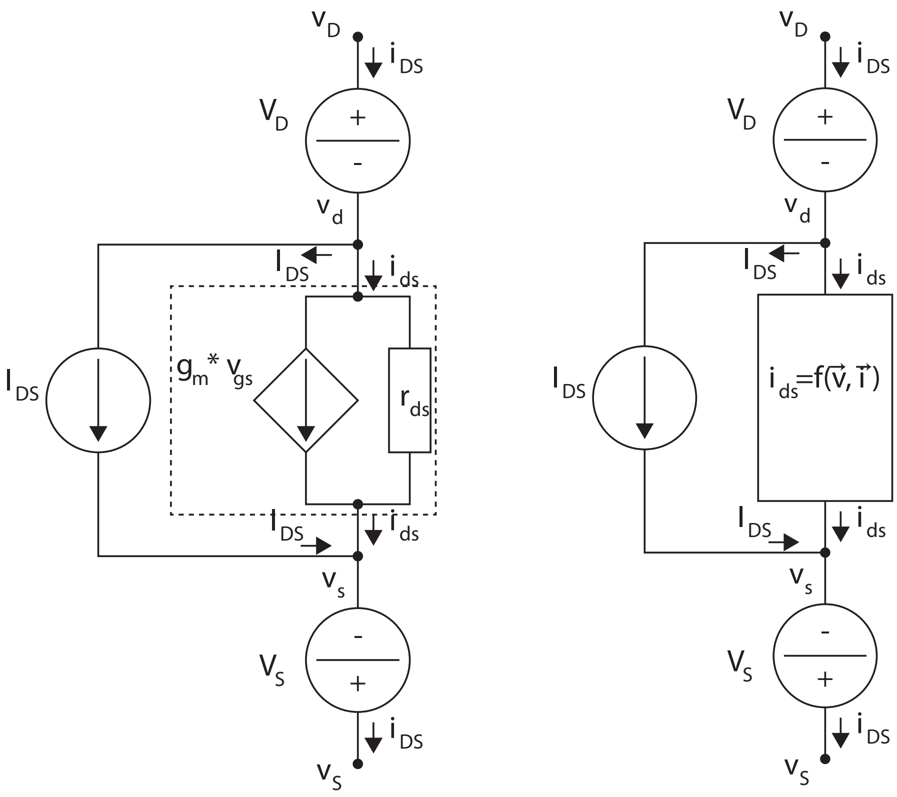


Figure 3.2: Left: Linear circuit approximation of a FET in a particular point of operation/biasing point. Right: General two terminal linear circuit model with explicit DC-offset, i.e. point of operation. In general, f would be a linear operator of all network voltage signals and current signals, but for now we shall think of it as a function.

lines,i.e. *only* the small variations, without the point of operation. So rewriting the same equation with these notation becomes:

$$i_{DS} = I_{DS} + v_{gs} * \frac{\delta I_{DS}}{\delta V_{GS}} + v_{ds} * \frac{\delta I_{DS}}{\delta V_{DS}} \tag{3.2}$$

Furthermore the two derivatives in the equation are now parameters of this new linearised function derived from the large scale model. For further simplification of the notation we do give those two parameters their own symbols, i.e. $g_m := \frac{\delta I_{DS}}{\delta V_{GS}}$ and $g_{ds} := \frac{1}{r_{ds}} := \frac{\delta I_{DS}}{\delta V_{DS}}$. So rewriting the same function once again:

$$i_{DS} = I_{DS} + v_{gs} * g_m + v_{ds} * \frac{1}{r_{ds}} \tag{3.3}$$

We can then draw a linear circuit that models the linear model current i_{DS} that can be used in a linear circuit model in lieu of a transistor between the terminals (also resulting from interaction in a linear version of the overall circuit this FET might be part of) v_D and v_S as drawn in the left of figure 3.2. Note that the circuit in the dashed box on it's own behaves just like the entire circuit does, but with only the small signals at its terminals. In other words it behaves like the complete circuit but its point of operation is defined by all bias voltages and currents being zero. The circuit in the dashed box is called the **small signal equivalent circuit element**.

3.2 Small signal equivalent circuit

The left of figure 3.3 shows an example of a small network of such circuit elements linearized around a biasing point. Note that all the bias currents are derived from solving the Kirchhoff and Ohm equations for the large signal models at that point, so $I_3 = I_1 + I_2$ and $I_3 = I_4$. Also, the small signal deviations from the biasing point need to fulfil the same requirement for the small signals $i_3 = i_1 + i_2$ and $i_3 = i_4$. In a network of general circuit elements that are linearised this way for a specific point of operation (see example in the left of figure 3.3) one can 'transpose' the point of operation defined by the two input voltages and the biasing current to all of those three parameters being zero: The bias voltage sources as well as the bias current sources cancel each other out in points v_3 and v_4. So one can remove the voltage sources and disconnect the current sources and the terminal currents and voltages for the small signal circuit element i_3 are still the same. One is left with the network on the right: it behaves exactly the same in all other nodes, but now for nodes v_3 and v_4 and the current between them i_3 only the small signals, i.e. the deviations from the biasing point, are modelled.

This brings us to a last simplification used in the so called small signal analysis of a circuit: when analysing the small signal behaviour of a circuit around a particular point of operation one does replace **ALL** the circuit elements with these small signal equivalent circuits (i.e. the 1^{st} order Taylor equivalent with all DC sources removed), i.e. one does set all bias/DC sources to zero (disconnecting DC currents and shorting DC voltages). The analysis of the resulting small signal equivalent circuit does only model the linearised *deviations* from the point of operation, i.e. if one wants the entirety of the signal back, one has to add the DC point of operation values to the small signal values again.

Note that the basic circuit elements, namely resistors, capacitors, and inductors, already *ARE* linear circuit elements (see definition of a linear system in section 5.1) and their small signal equivalent circuits simply are themselves. That is because for example the total current through a resistor can be simply linearly separated into a large signal DC component and a small signal offset.

$$I_R + i_r = \frac{V_R}{R} + \frac{v_r}{R} \tag{3.4}$$

$$I_C + i_c = \frac{\partial}{\partial t} V_C C + \frac{\partial}{\partial t} v_c C \tag{3.5}$$

$$V_L + v_l = \frac{\partial}{\partial t} I_L L + \frac{\partial}{\partial t} i_l L \tag{3.6}$$

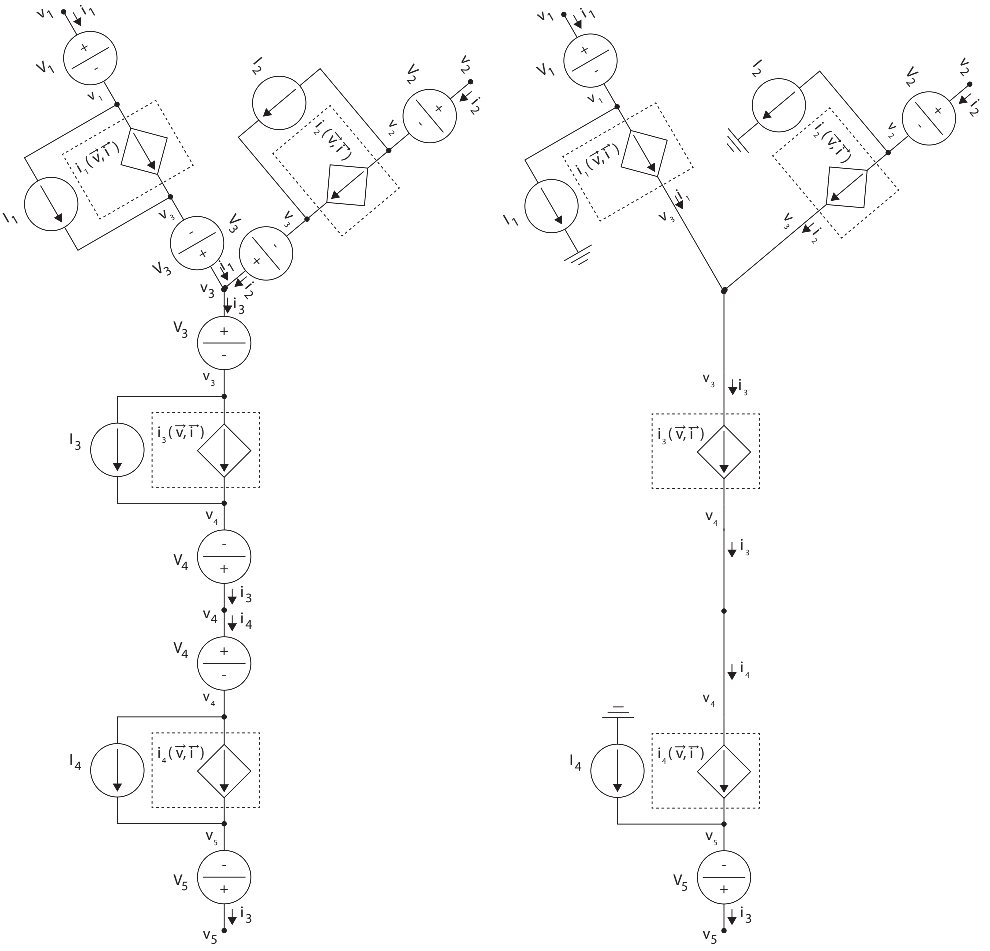


Figure 3.3: Left: A electronic circuit network of linear approximation elements Right: Replacing the element in the center with its small signal equivalent that neglects any DC offsets and does not affect the behaviour of the rest of the network. This is to illustrate that one can in fact replace any and all elements with the small signal equivalents, i.e. ignore any DC offsets/biasing points and still model the linear changes in the network correctly.

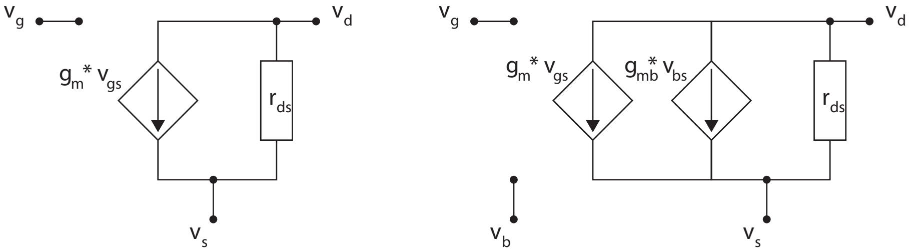


Figure 3.4: FET Small signal equivalent model for low frequencies. The right hand side version includes the so called **back gate effect**, i.e. the influence of varying the bulk to source voltage. g_{mb} is usually about 10 times smaller than g_m and often neglected.

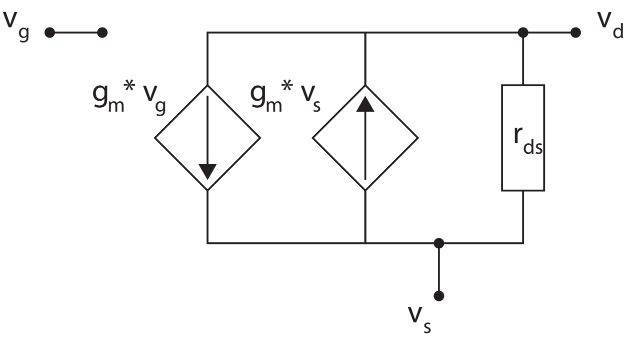


Figure 3.5: The same as the left hand side in figure 3.4, but explicitly separating the influence of respectively v_g and v_s. This is sometimes a bit less prone to making sign errors when deriving small signal equivalent circuits where one of the two is constant and can thus be removed.

3.3 FET low frequency small signal model

In a first instance, let us look at the **low frequency small signal equivalent** for FETs. Low frequency means that we'll for now neglect 'parasitic' capacitors in these models that would introduce dependencies on the *rate of changed* of dynamic signals. The appropriate model already mentioned in figure 3.2 is given without distraction in figures 3.4 and 3.5.

The model parameters g_m and r_{ds} need to be derived for a specific point of operation from the derivatives of the large signal model. The derivation of g_m in dependency of I_D does not depend on whether the device is in saturation or conduction. Only if it is in strong or weak inversion. (However, note that if you derive the dependency on V_{OV} instead of I_D THAT will depend on whether the device is in saturation or conduction!)

$\boldsymbol{g_m}$ **for strong inversion, saturation**:

$$g_m = \frac{dI_D}{dV_{GS}} = \sqrt{2k_n I_D} \tag{3.7}$$

$\boldsymbol{g_m}$ **for strong inversion, conduction**:

$$g_m = \frac{dI_D}{dV_{GS}} = k_n V_{DS} \tag{3.8}$$

$\boldsymbol{g_m}$ **for weak inversion**:

$$g_m = \frac{dI_D}{dV_{GS}} = \frac{I_D}{U_T} \tag{3.9}$$

r_{ds} in dependency of I_D in saturation does not depend on whether the FET is biased in weak or strong inversion and is:

$\boldsymbol{r_{ds}}$ **in saturation**:

$$r_{ds} = \frac{1}{\lambda I_D} = \frac{V_A}{I_D} \tag{3.10}$$

In deep conduction ($V_{DS} << V_{OV}$), it *does* depend on whether one is in weak or strong inversion, though:

$\boldsymbol{r_{ds}}$ **in deep conduction, strong inversion**:

$$r_{ds} = \frac{1}{kV_{OV}} \tag{3.11}$$

$\boldsymbol{r_{ds}}$ **in deep conduction, weak inversion**:

$$r_{ds} = \frac{1}{I_S U_T} e^{-\frac{V_{OV}}{nU_T}} \tag{3.12}$$

Note: since these equations only can be applied when clearly within a region of operation, you need to be careful when close to the point of transition between these regions. You then better apply a numerical approximation of the derivatives using the full EKV model, or rely on simulation tools entirely.

The bulk B exerts a similar effect on the channel as the gate, with the restriction that in bulk CMOS its V_B cannot move above V_S for nFETs and not below V_S for pFETs, to avoid a forward biased PN junction between them(!), so this effect can be used more freely in SOI technologies, where this problem does not occur. The bulk is capacitively coupled to the channel across the depletion region under the gate, just like the gate itself across the gate oxide. This is even more obvious in SOI transistors where th BOX is the dielectric separating the channel from the bulk and thus forming a plate capacitor. This is sometimes referred to as **back gate effect**. As the depletion region and BOX are not as thin as the gate oxide, the effect is minor by comparison. It is $(n-1)$ as effective in controlling the drain current according to (2.7), or as a rule of thumb $\frac{1}{10}$ as effective if n is not known. Oftentimes the bulk is connected to the source though, and g_{mb} does thus not have any effect. If the source is at a higher potential than the bulk however, one has to be aware that the transistor will conduct less current for the same V_{GS} in the large signal world and that v_b can be used to modify the drain current in the small signal world.

$$g_{mb} = (n-1)g_m \approx g_m \frac{1}{10} \tag{3.13}$$

3.4 Applying small signal analysis

So to summarize the recipe for replacing a circuit by it's small signal equivalent to determine its behaviour around a particular biasing point:

1. Fully determine the point of operation, i.e. all biasing parameters.
2. Derive small signal parameters (e.g. g_m and r_{ds}) based on that point of operation.
3. Replace all circuit elements with their small signal equivalent.
4. Remove all independent/constant current sources and replace all independent/constant voltage sources with Gnd (i.e. NOT the current sources that depend on a small signal input such as the $i_{ds} = g_m V_{gs}$).
5. perform a normal Kirchhoff's or Ohm's law analysis of the circuit.

Example

A simple first example of such a small signal analysis of the so called common source (CS) amplifier is shown in figure 3.6. An example of it's voltage I/O relationship is shown in figure 3.7. In the middle of its input range the output voltage falls steeply. If the circuit is used as an amplifier, this part is the interesting **input voltage range**. The best point of operation/biasing point is the red circle smack in the middle of the steep signal transition. If one follows the red linearised model, the output voltage v_O is given as $v_O = Av_I$ where $A < -1$ is a negative voltage amplification.

So for the small signal analysis we now have to first find all the correct biases to have the circuit operate in this point of operation. For that, we have to use the large signal model! Once we do have found all these biasing parameters to end up in the correct point of operation we can then find the correct small signal parameters for that point of operation and finally find the correct number for A.

Step 1: find biasing point, i.e. all biasing parameters.

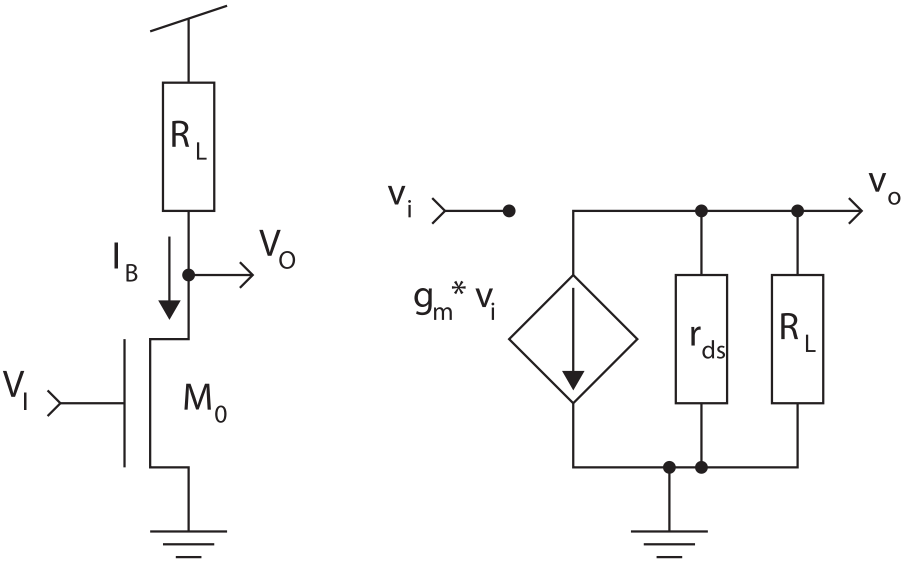


Figure 3.6: CS amplifier circuit and its small signal equivalent circuit.

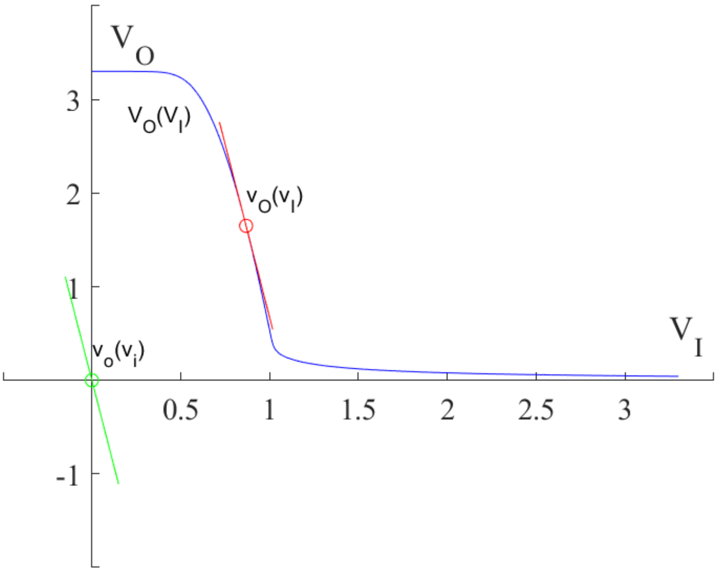


Figure 3.7: CS stage I/O relationship according to the large signal model (blue), the linearised model (red), and the small signal model (green).

To find/set the point of operation we move in the large signal model world. Some assumptions/parameters need to be defined/given in the task to find those biasing voltages and the current. As process parameters and design parameters the transistor has a characteristic current $I_S = 50\text{nA}$, a threshold voltage $V_{tn} = 0.5\text{V}$, a $\lambda = 0.05\frac{1}{\text{V}}$, and a negligible slope factor $n = 1$. At room temperature we use $U_T = 26\text{mV}$. The bulk is shorted to the source $V_S = V_B = 0\text{V}$.The supply voltage is 3.3V and the load resistance is given R_L =1MΩ. Furthermore the task is given the point of operation for the output voltage V_O to be at $\frac{\text{Vdd}}{2}$.

In summary:
Given: the process and design parameters (I_S, V_{tn}, λ, n, U_T, Vdd, V_B), V_O and R_L
Asked: I_B, V_I, and A

So the voltage fall across the resistor has to be 1.65V. From that it follows directly that the bias current I_B needs to be $I_B = 1.65\mu\text{A}$. This will put the output voltage V_O neatly between Gnd and Vdd at 1.65V. The last parameter that is not yet defined is the point of operation for the input voltage V_I. It is fully dependent on I_B and V_O and can be derived from them. First let's make the observation that $I_B >> I_S$. That is one way of determining that the strong inversion model simplification is appropriate in this case. We also suspect and assume (to be verified) that the transistor is in saturation. So we shall use equations (2.15) and (2.5) to derive V_I.

$$V_G = 2nU_T\sqrt{\frac{I_D}{I_S\left(1+\lambda V_{DS}\right)}} + V_{tn} + nV_S \tag{3.14}$$

Entering the already known parameters we get:

$$V_I = 2 * 26\text{mV}\sqrt{\frac{1.65\mu\text{A}}{50\text{nA}\left(1+0.05\frac{1}{\text{V}} * 1.65\text{V}\right)}} + 0.5\text{V} \tag{3.15}$$

One should always be **very careful with the units!!!** Do not mix 'micro-', 'milli-', 'nano-', 'mega-' and the like. I highly recommend to always write the units into the equation in order to be always aware of what unit is being used. Also, at some point one has to make sure to use consistent units everywhere in the equation. For example by getting rid of all the prefixes, preferably right from the start:

$$V_I = 2 * 26 * 10^{-3}\text{V}\sqrt{\frac{1.65 * 10^{-6}\text{A}}{50 * 10^{-9}\text{A}\left(1+0.05\frac{1}{\text{V}} * 1.65\text{V}\right)}} + 0.5\text{V} \tag{3.16}$$

One other advantage of writing the units into the equation is that one can verify that the end result actually gets the correct unit, [V] in this case. So that's a good control

$$V_I = 0.79\text{V} \tag{3.17}$$

Let us **not forget** to verify that we are in saturation here and have thus used the right equation. The overdrive voltage $V_{OV} = V_{GS} - V_{tn} = 0.29\text{V}$ and is much smaller than $V_DS = 1.65$V, so we are in saturation, and thus we are fine.

Step 2: find small signal parameters.

Let's move on to determine the small signal parameters g_m and r_{ds} at that point of operation. We use equations (3.7) and (3.10):

$$g_m = \sqrt{2k_nI_D} = \sqrt{2\frac{I_SI_D}{2nU_T^2}} = \frac{\sqrt{50*10^{-9}\text{A}*1.65*10^{-6}\text{A}}}{26*10^{-3}\text{V}} = 11.05\frac{\mu\text{A}}{\text{V}} \tag{3.18}$$

$$r_{ds} = \frac{1}{\lambda I_D} = \frac{1}{0.05\frac{1}{\text{V}}1.65*10^{-6}\text{A}} = 12.12\text{M}\Omega \tag{3.19}$$

Step 3 and 4: derive small signal equivalent circuit (replace elements and remove DC sources)

The small signal equivalent circuit is shown in figure 3.6 to the right. Note that there actually are *three* circuit elements to start with: the FET, the resistor *and* the voltage supply (i.e. an implicit DC voltage source for Vdd). Vdd is a DC voltage source that becomes Gnd in the small signal world. Thus r_{ds} and R_L actually end up in parallel between the transistor drain and Gnd. Also the DC offset at the input V_I and V_O at the output ar no longer represented in this small signal model, nor is the bias current I_B that flows in the large signal representation only. What it models is the green line in figure 3.7, moving around the origin (the green circle) if any of the small signal variables is moving away from zero.

Step 5: Kirchhoff's and/or Ohm's rule to analyse

If you move v_i you'll create a current $g_m * v_i$ flowing from Gnd through the parallel resistors *towards* v_o, i.e. v_o has the inverted sign compared to v_i. More precisely:

$$v_o = -g_m * v_i\,(r_{ds}||R_L) \tag{3.20}$$

and

$$A = \frac{v_o}{v_i} = -g_m\,(r_{ds}||R_L) = -g_m\frac{r_{ds}R_L}{r_{ds}+R_L} \tag{3.21}$$

Since $r_{ds} >> R_L$ it follows that $(r_{ds}||R_L) \approx R_L$, but let's compute the exact value:

$$(r_{ds}||R_L) = 9.23\text{M}\Omega \tag{3.22}$$

and thus:

$$A = 10.2 \tag{3.23}$$

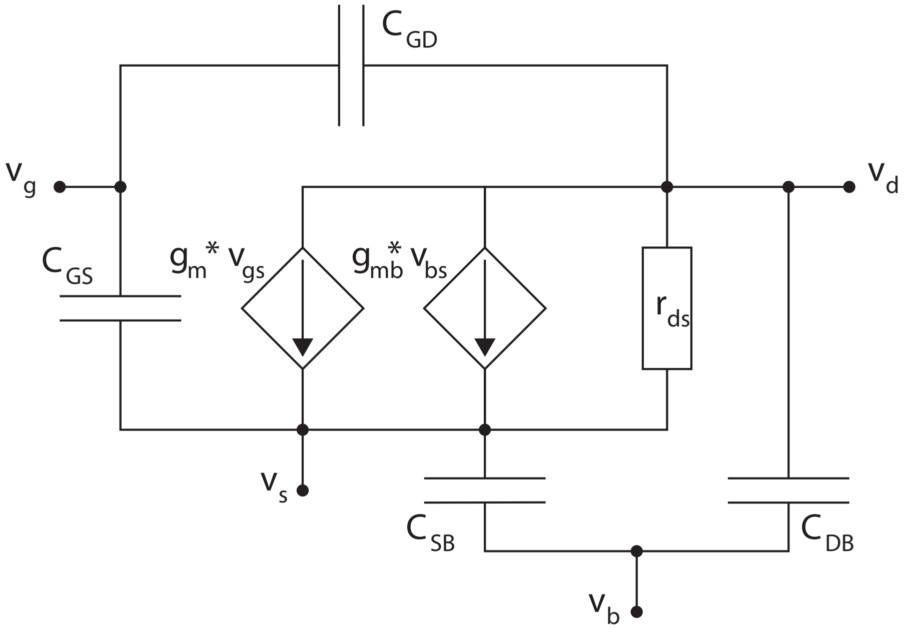


Figure 3.8: FET small signal equivalent for high frequency analysis.

3.5 High Frequency Models

The capacitors inherent in the FET layout can be glimpsed in the cross sections of figures 1.1, 1.2, and 1.4. Whenever a conductor is either separated by silicon-oxide or a depletion region of a reverse-biased PN junction, we get a capacitor. In the small signal model, capacitors represent themselves and can simply be 'imported' from the large signal representation. These are $C_{GS}, C_{GD}, C_{DB}, C_{SB}$ (see figure 3.8). The value of the capacitors C_{GS} and C_GD between the gate and the underlying conductor(s) depends on the oxide capacitance per area C_{OX}. C_{GS} scales with C_{OX} and the gate *area*, since it includes (most of) the channel underneath the gate. C_{GD} scales only with C_{OX} and the transistor *width* W, as it is caused by a fixed length of overlap of the gate onto the drain area. It's generally smaller than the other capacitors mentioned here, but often has an over-proportional influence (see section 5.2.7, buzzword 'Miller effect'). Capacitances of PN junctions depend on the doping profile and bias voltage, but are often considered approximately constant when the junction is reverse-biased, i.e. voltage independent. Then they will only depend on relative doping difference and are generally bigger for higher doping concentration differences so we can give a characteristic capacitance per area for the drain and source regions C_{PN}. In bulk CMOS these are capacitances C_{SB} and C_{DB}. In SOI CMOS, much the same way, we can also give a capacitance per area for source and drain C_{BOX}. Thus, source and Drain capacitors C_{SB} and C_{DB} scale with the source and drain area and either C_{BOX} or C_{PN}. Generally the source/drain lengths are left constant, so practically speaking they often only scale with transistor width W as well.

The capacitors play a role in the frequency dependent behaviour of analog circuits introduced in section 9.4 and speed and gate delays of digital circuits as discussed in section 4.2.

Chapter 4

FET Switch models

In binary digital electronics, Boolean variables are represented by binary voltages[1], i.e. ideally the voltage signals will only have two discrete levels and either be at Gnd or Vdd. More realistically voltages are considered 'high', i.e. corresponding to logic 'true' or '1', when they are above a certain minimum threshold V_{Hmin}, and low (i.e. logic 'false' or '0') when they are below another threshold V_{Lmax}. The two thresholds are usually distinct, leaving a middle range where a signal cannot be guaranteed to be correctly identified as either 'high' or 'low'. Under nominal operating conditions, signals in a digital circuit should never settle in this 'undefined' state. However, while in a transitional state there are always time periods where one cannot rely on signal states.

Digital transistor models are voltage controlled switches, i.e circuits with two states determined by a switching threshold of the input control signal(s) (compare figures 4.1 and 4.2).

Their most common use in digital circuits is as switches that either pull down a binary signal from Vdd to Gnd and keep it locked at Gnd thereafter, or vice versa: pull up a signal from Gnd to Vdd and lock it there. The formar is implemented efficiently by nFETs (Fig. 4.1 B)) and the latter by pFETs (Fig. 4.1 C)). We'll discuss later why each of them is best suited for one of the two scenarios. Sometimes, if the switch is used for both, pull up *and* pull down, one can use a nFET and a pFET in parallel with inverted control signals (Fig. 4.1 D), aka **transmission gate**), for good switching performance. This is in particular used for switches between analogue signals, i.e. signals that are not either Vdd or Gnd, but anything inbetween.

But for now let us discuss nFETs for pull down and pFETs for pull up actions. If we consider the limited conductance and generally parasitic capacitances of transistors and interconnect lines, the transition of such (digital) signals being pulled is not instantaneous and can be approximated with an RC-low pass filter or a constant current charging a load capacitance. These models are adapted to model the **propagation delay** t_p that causes delay in digital processing hardware and limits the speed of digital electronics. Assuming ideal switches with zero resistance or infinite current and/or no capacitance one can also disregard the delay and use the models to simply verify the correct implementation of logical functions.

[1] and less commonly currents

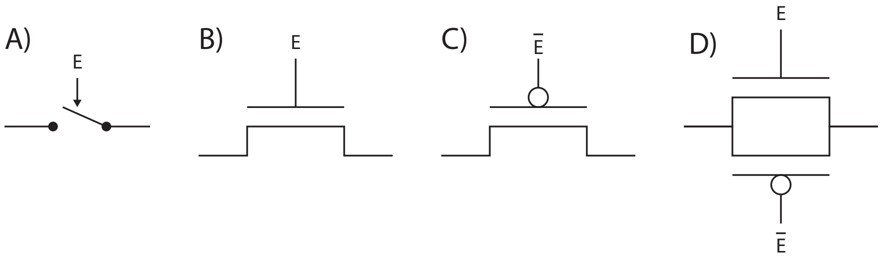


Figure 4.1: A) symbol for a switch. B) and C) single transistor implementations with severe non-ideality that a pFET is good for pulling up and an nFET for pulling down, but not vice versa. So if an nFET is used for pulling up, it will be much slower than pulling down. If this is not acceptable, one can use a **transmission gate** as depicted in D), i.e. both a pFET *and* and nFET in parallel with inverted control signals at their gate terminals. So at least one of them will always be a good conductor, irrespective of the voltages at the S/D terminals.

The two states are 'switched' between dependent on the gate voltage V_G. Again, ideally this input/gate voltage would also be perfectly discontinuous (i.e. instant transition) and binary (always precisely at either Gnd or Vdd). However, as V_G itself may undergo a non-instantaneous transition, the idealized binary state of the switch is modelled as being governed by a threshold V_{sw}: an nFET is considered non-conducting (or having a large resistance R_{OFF} or a very small current I_{leak}) if $V_G < V_{sw}$ and conducting (with resistance R_{ON} that can be considerd being zero or very small, or sourcing I_{ON}) if $V_G > V_{sw}$. See figure 4.2! Be aware that the switching threshold V_{sw} is normally NOT defined to be the same as the large signal threshold V_t! It's often the voltage at which the balance of currents between a pFET and an nFET that form an inverter (see chapter 7) is shifting to either pull an output towards Vdd/high or Gnd/low. More simply, it is sometimes approximated as being exactly in the middle of the supply rails Vdd and Gnd:=0V, i.e. $V_{sw} = \frac{\text{Vdd}}{2}$. In the following we will adapt the latter, i.e. $V_{sw} = \frac{\text{Vdd}}{2}$.

The load capacitance C_L will include the same relevant parasitic capacitances as the small signal model presented in figure 3.8 (i.e. capacitors connected to the output/drain C_{GD}, C_{DB} of the stage under consideration and capacitors connected to the gate C_{GS}, C_{GD} of the next stage it is connected to) that are not explicitly shown in figure 4.2, as well as other 'parasitic' capacitances, e.g. of interconnecting metal traces. This modelling of the output allows to approximately determine signal propagation delays, i.e. the time the output needs to reach the switching threshold of the next stage in a digital signal processing chain.

The magnitudes of R_{ON} or I_{ON} are derived from a more complete transistor model or otherwise approximated quite crudely. The predictions for absolute propagation delay t_p will thus also be very crude. They shall, however, be very useful to predict *relative* propagation delays, i.e. in predicting the effect of changing circuit parameters such as $\frac{W}{L}$ in the design or choosing different CMOS technologies with different parameters μC_{OX}, or of reducing or increasing the

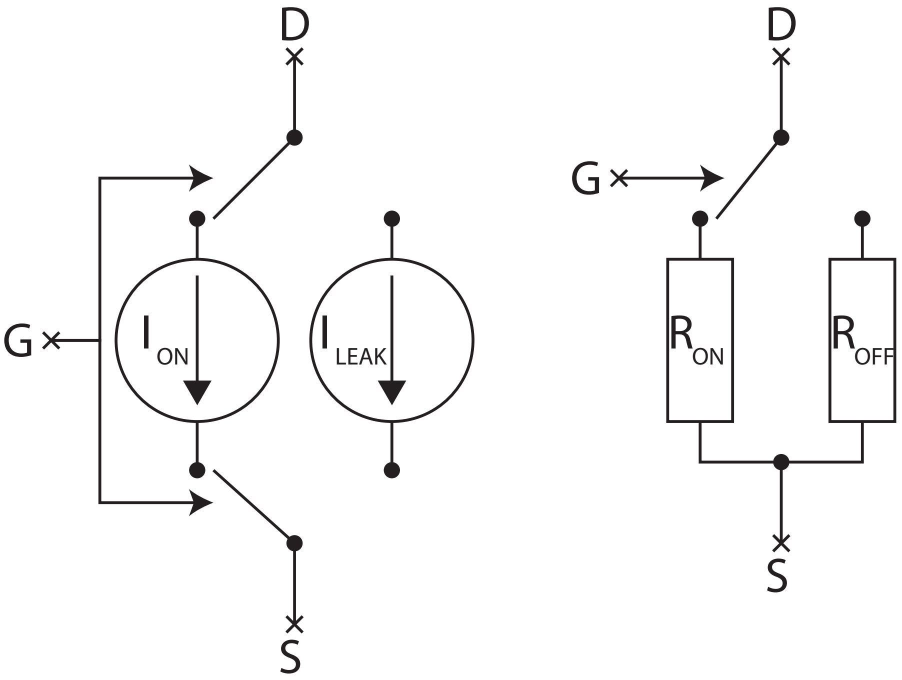


Figure 4.2: FET switch models. For an nFET the control terminal voltage $V_G > V_{sw}$ ($V_G < V_{sw}$ for pFETs) shorts the switch to R_{ON}/I_{ON} and $V_G < V_{sw}$ ($V_G > V_{sw}$ for pFETs) to R_{OFF}/I_{LEAK}. Often R_{OFF} and I_{LEAK} are neglected, i.e. considered to be $R_{OFF} = \infty\Omega$ and $I_{LEAK} = 0\text{A}$. For logic gates where the input is shorted to gate terminals of both nFETs and pFETs, V_{sw} is often defined as exactly in the middle of the supplies, i.e. when the supplies are Vdd and Gnd:=0V as $V_{sw} := \frac{\text{Vdd}}{2}$. However, realistically the input of that logic gate at which the output switches will always have some offset from $\frac{\text{Vdd}}{2}$.

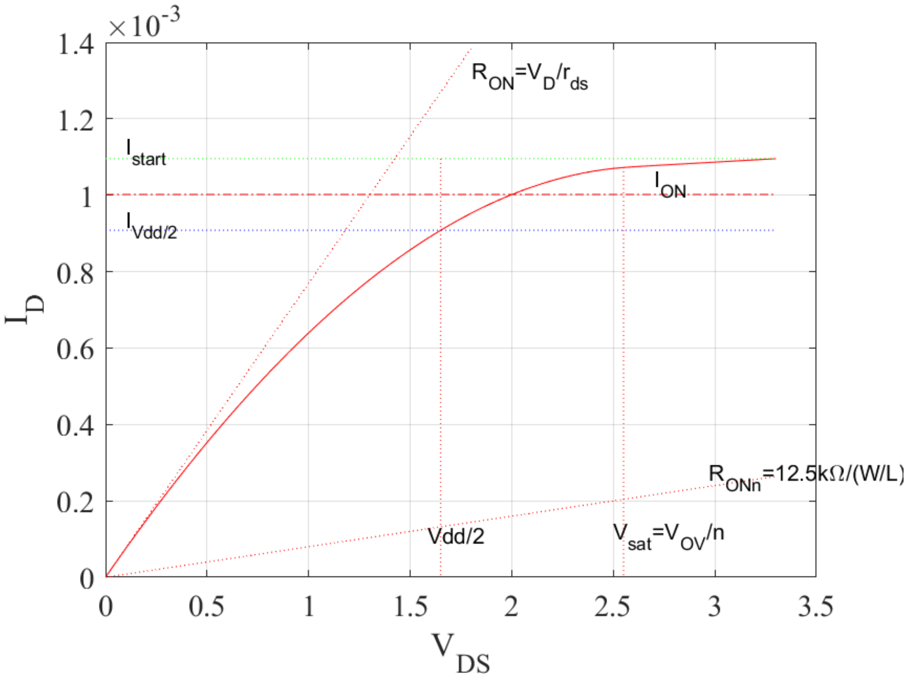


Figure 4.3: Comparing the large signal FET model with switch models. The drain current of an nFET according to the large signal model when its gate is at Vdd and its source at Gnd for a 180nm CMOS process is shown as drawn out line. $R_{ON} = \frac{V_D}{r_{ds}}$ marks the approximation for R_{ON} appropriate to model the output resistance keeping the drain voltage V_D low, i.e. *after* a transition. If used to model the transition from high to low, the current is substantially overestimated, especially at the start of a transition. I_{ON} shows the constant current approximation where I_{ON} is the average current of the starting current and the current when the drain reaches $\frac{\text{Vdd}}{2}$. **** Figure needs update to reflect (4.13). ****

power supply voltage Vdd.

In the following we will derive approximations of R_{ON} and I_{ON} under the assumption that we us nFETs as switches pulling a signal down towards Gnd, and pFETs as switches pulling a signal up towards Vdd. This assumption assures a good performance as the gate to source voltage difference will be constant and equal to a full Vdd for the entire transition, and thus the transistor's current in the large signal model will follow the I_D vs V_{DS} curve for a constant V_{GS} and be much larger than it would be if not only V_{DS} but also V_{GS} would be shrinking under the transition.

4.1 Static output resistance R_{ON}

If one considers that generally the threshold voltage is a fraction smaller than $\frac{1}{2}$ of the supply voltage Vdd, one notices that an nFET with V_S at Gnd and

Vdd as its gate voltage V_G is in conduction for most of the range of drain voltages $V_D \in [0, \text{Vdd}]$ (see figure 4.3). So one might approximate R_{ON} with the equation valid for deep conduction where the transistor actually DOES behave like a resistors, i.e. equations (2.22) and (2.24), for respectively strong- and weak inversion. The curve for $R_{ON} = r_{ds}$ is also plotted for comparison. This is a good approximation for R_{ON} at the end of a transition when the transitor is *maintaining* a binary state, i.e. a measure of how much resistive load can be overcome to reach and maintain a voltage close to the supply rails. At this instance, please note that this is NOT the case if you use a pFET as a pull down device or an nFET as a pull up device, because in THAT case you will attempt to pull the source terminal voltage V_S and NOT the drain terminal voltage V_D. In THAT case, at the end of a transition you will not only have a $V_{DS} = 0$ but also $V_{GS} = 0$ and thus an extremely high resistance/low conductance trying to tie that V_S to respectively Gnd and Vdd, and extremely low capability of overcoming any kind of resistive load at that terminal pulling the voltage elsewhere!

So just to repeat: equations (2.22) and (2.24) provide good approximations for R_{ON} to gauge the pull down capability of nFETs and the pull up capability of pFETs to *maintain* a digital state.

4.2 Dynamic transitions and propagation delay

4.2.1 Approximating I_{ON} for signal transition

For modelling an entire transition of V_D from one digital/logic state to the switching point, however, (2.22) and (2.24) do overestimate the actual current, especially if the threshold voltage is relatively high compared to the supply voltage Vdd (see figure 4.3). In that case the device will be in saturation rather than conduction for a larger part of the transition and will 'cap' the current at a maximum level for large V_{DS}.

[2] suggests to derive I_{ON} as probably the best approximative value so far from the large signal model as the average of the drain currents at the start of the transition I_{start}, i.e. from an nFET pulling down when $V_D = \text{Vdd}$, and the drain current $I_{\frac{\text{Vdd}}{2}}$ when the switching threshold is reached, i.e. $V_D = \frac{\text{Vdd}}{2}$, thus reaching the switching point and flipping a next switch connected to that drain.

$$I_{ON} \quad = \quad \frac{I_{\text{start}} + I_{\frac{\text{Vdd}}{2}}}{2} \tag{4.1}$$

For the best result one uses the full EKV model to compute those two currents. This is also indicated in figure 4.3. Note that in any case this current will scale with $k = \mu C_{OX} \frac{W}{L}$, so the primary weapon of the designer to change propagation delay in a given CMOS technology node is to adjust $\frac{W}{L}$.

The book [2] derived the two currents I_{start} and $I_{\frac{\text{Vdd}}{2}}$ for the very specific (and traditionally the most common) case of strong inversion and with the region operation approximations, yielding a somewhat less exact result. However, beyond the influence of parameter k it will allow us to express an important relationship of t_p and Vdd as well as the threshold voltage V_t relative to Vdd.

So for I_{start} one can use the appropriate saturation region equation. If one uses the strong inversion approximation (assuming $n = 1$, $\lambda = 0$ and $V_S = 0\text{V}$) one gets:

$$I_{\text{start}} = \frac{1}{2} k \left(\text{Vdd} - V_t\right)^2 \tag{4.2}$$

For $I_{\frac{\text{Vdd}}{2}}$ one uses the appropriate conduction expression, e.g. for strong inversion, $n = 1$ and $V_S = 0\text{V}$, and on the condition that we are in conduction, i.e. $\frac{\text{Vdd}}{2} = V_D < V_{OV} = \text{Vdd} - V_t \Rightarrow V_t < \frac{\text{Vdd}}{2}$:

$$I_{\frac{\text{Vdd}}{2}} = k \frac{1}{2} \text{Vdd} \left[\frac{3}{4}\text{Vdd} - V_t\right] \tag{4.3}$$

(Once more: Note that if you happen to be in weak inversion, that you have to compute the two currents appropriately with those equations, i.e not the ones used here for strong inversion!)

Thus, for strong inversion we can derive a more direct solution for the average of the two:

$$\begin{aligned}
I_{ON} &= \frac{1}{4} k \left((\text{Vdd} - V_t)^2 + \frac{3}{4}\text{Vdd}^2 - \text{Vdd}V_t \right) && (4.4)\\
&= \frac{1}{4} k \left(\text{Vdd}^2 - 2\text{Vdd}V_t + V_t^2 + \frac{3}{4}\text{Vdd}^2 - \text{Vdd}V_t \right) && (4.5)\\
&= \frac{1}{4} k \left(\frac{7}{4}\text{Vdd}^2 - 3\text{Vdd}V_t + V_t^2 \right) && (4.6)\\
&= \frac{1}{4} k \text{Vdd}^2 \left(\frac{7}{4} - 3\frac{V_t}{\text{Vdd}} + \frac{V_t^2}{\text{Vdd}^2} \right) && (4.7)\\
&= \frac{1}{4} k \text{Vdd}^2 \left[\left(\frac{3}{2} - \frac{V_t}{\text{Vdd}} \right)^2 - \frac{1}{2} \right] && (4.8)
\end{aligned}$$

With the condition from above that $V_t < \frac{\text{Vdd}}{2}$, we get an increase of this current with decreasing relative threshold $\frac{V_t}{\text{Vdd}}$, i.e. a faster switching at the cost of more current and thus power. Consequently, when lowering Vdd, this current is reduced quadratically with the term Vdd^2 as well as with the slope of the term $\left(\frac{3}{2} - \frac{V_t}{\text{Vdd}}\right)^2$.

Also note that the mobility (part of parameter $k = \mu C_{OX}\frac{W}{L}$) for pFETs is generally lower in P-bulk processes (which are the most common kind). So for same size transistors one usually gets a bigger current and faster switching for nFETs than for pFETs. If symmetry is important, one can compensate for that by increasing $\frac{W_P}{L_P}$ for pFETs in a logic circuit. More of that later in section 7.1!

So once a switch input at the gate G reaches V_{sw} and is shorted, one measures the propagation delay t_p as the time it takes the output to move from its resting potential V_{start} at one of the two supply rails to the switching threshold V_{sw} of the next stage. This is illustrated in figure 4.4 for an nFET pulling its drain voltage low. In case of a current source the voltage drops linearly and the time to reach $V_{sw} := \frac{\text{Vdd}}{2}$ starting from Vdd is then:

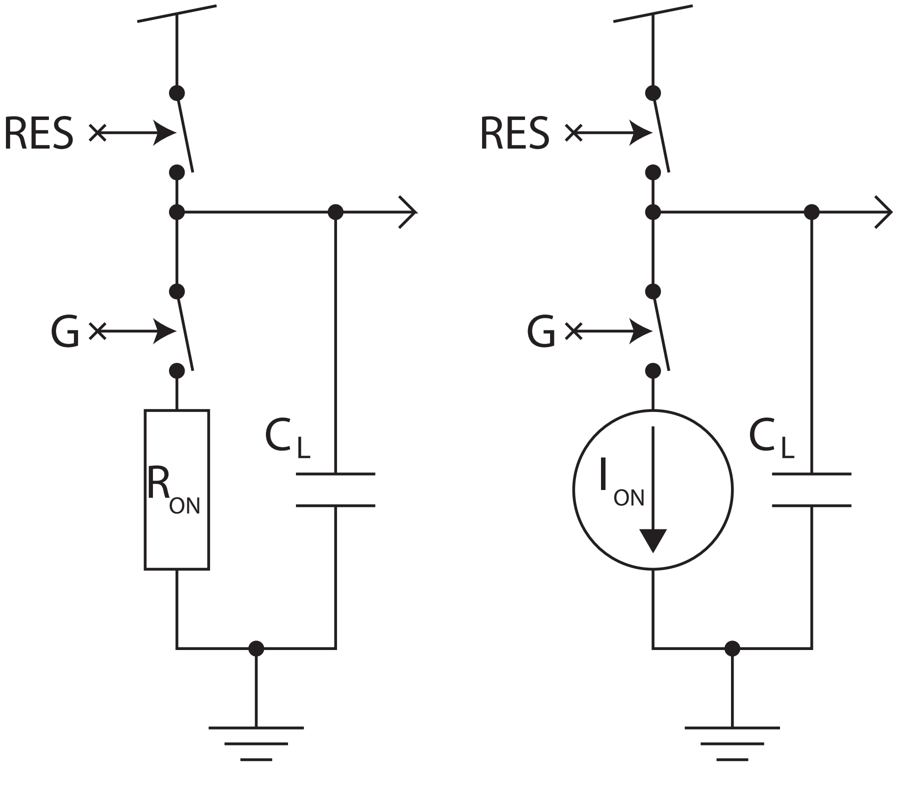


Figure 4.4: Illustrating the scenario to measure propagation delay in case an nFET is used as a switch to pull down its drain voltage from Vdd to $\frac{\text{Vdd}}{2}$. The reset switch RES is added here to reset the output to its starting voltage Vdd and establish the right starting point.

$$\begin{aligned} t_p &= \frac{\text{Vdd}C_L}{2I_{ON}} && (4.9)\\ &= \frac{\text{Vdd}C_L}{2\frac{1}{4}k\text{Vdd}^2\left[\left(\frac{3}{2}-\frac{V_t}{\text{Vdd}}\right)^2-\frac{1}{2}\right]} && (4.10)\\ &= \frac{2C_L}{k\text{Vdd}\left[\left(\frac{3}{2}-\frac{V_t}{\text{Vdd}}\right)^2-\frac{1}{2}\right]} && (4.11)\end{aligned}$$

4.2.2 R_{ON} for signal transition

A resistive model instead of a current source model can be preferable (i.e. is easier to use), when assembling more complicated digital gates that will require transistors in series (see section 7.1). The book [2] suggests the crudest approximation yet based on analysis of CMOS technology nodes from 0.25μm down to 180nm, where (very!) approximately all had the same resistance per $\frac{W}{L}$ ratio for nFETs and pFETs respectively:

$$\begin{aligned} R_{ON_n} &= \frac{12.5}{\frac{W_n}{L_n}}\text{k}\Omega \\ R_{ON_p} &= \frac{30}{\frac{W_p}{L_p}}\text{k}\Omega \end{aligned} \qquad (4.12)$$

The use of this model is purely to gauge the effects of changing $\frac{W}{L}$ and has no predictive value with respect to changing Vdd or the CMOS technology, nor for the absolute value of the propagation delay t_p. In figure 4.3 this approximation is indicated near the real curve, and would actually substantially underestimate the real drain current for this particular CMOS process.

A more accurate approximation in absolute terms one can use the current model for I_{ON} from equation (4.1) and derive a resistance that would result in the same average current:

$$\begin{aligned} \frac{\frac{\text{Vdd}}{R_{ON}}+\frac{\frac{\text{Vdd}}{2}}{R_{ON}}}{2} &= I_{ON} \\ \frac{3}{4}\frac{\text{Vdd}}{R_{ON}} &= I_{ON} \\ R_{ON} &= \frac{3}{4}\frac{\text{Vdd}}{I_{ON}} \end{aligned} \qquad (4.13)$$

And in case of a resistive model with a load capacitance on the drain on gets exponential decay of the output voltage and the resulting expression:

$$\begin{aligned} e^{\frac{-t_p}{R_{ON}C_L}} &= \frac{1}{2} \\ t_p &= -R_{ON}C_L\ln(\frac{1}{2}) \\ t_p &= 0.69R_{ON}C_L \end{aligned} \qquad (4.14)$$

4.2.3 What's next

In part III of this book, these switches are combined to **combinational logic circuits** implementing **Boolean functions** and **sequential logic circuits** implementing **finite state machines (FSM)**.

Part II

Linear Circuit Analysis Basics

Chapter 5

Linear circuit filters

5.1 Linear filter and transfer function

Definition: A **linear system** is an operator h that transforms an input signal (i.e. a function $x(t)$ of time t) into an output signal $h(x(t))$ such that $h(x(t)) + h(y(t)) = h(x(t) + y(t))$ and $sh(x(t)) = h(sx(t))$, for any scalar value s.

We also call h a **linear filter**. Incidently any electronic circuit composed of linear circuit elements fulfils the requirement of a linear system, i.e. is a linear filter. (No proof is presented here, but believe you me!) Low level linear circuit elements are real electronic passive components, i.e. resistors, capacitors and inductors, and more artificial active elements such as ideal current and voltage sources that are either constant or linearly dependent on a signal in the circuit (e.g. transconductances, transresistances, voltage- and current-amplifiers). So when replacing transistors with their small signal equivalent, linear circuit analysis then becomes a major tool in analysing transistor circuits. That is the reason why this topic occurs here.

In the following it is assumed that the reader is familiar with the Kirchhoff's law and the Ohm's law. They are just briefly repeated here without further explanations or illustrations. Some examples of how to do circuit analysis with them will follow.

Kirchhoff's current law: the sum of all currents into an electric node is zero.
Kirchhoff's voltage law: the sum of all voltages along a electric circuit loop is zero.
Ohm's law:

$$V = I * R \text{ or } V = I * Z \tag{5.1}$$

Linear circuit analysis uses Ohm's and Kirchhoff's laws to find the currents and voltages in a network of interconnected circuit elements. This is straight forward for operations that only use momentary signal values as inputs, e.g. operations introduced by using resistors and linearly dependent sources. Obviously any relationship between momentary (i.e. no dependence on history) current- and voltage values in a network consisting of only such elements can be expressed as a linear combination of other momentary signal values in the network. In other words $h(x(t))$ will the simply be a linear *function*, mapping scalar x to a scalar $h(x)$ (as opposed to an *operator*, mapping function $x(t)$ of time to a function $h(x(t))$ of time) where the history of the input signal $(x(t))$ does not affect the result in any way.

However, inductances L and capacitances C introduce operations using the derivative or integral of some signals (in other words the results will depend on signal history/dynamic) (also compare figure 5.1):

$$I = CV\frac{\partial}{\partial t} \tag{5.2}$$

$$V = LI\frac{\partial}{\partial t} \tag{5.3}$$

Three principles are used to tackle such differential operators in linear circuit analysis without actually having to resort to solving differential equations:

1. Any signal can be decomposed into its frequency components, i.e. into a sum/integral of sine waves at different phases and amplitudes. This decomposition is known as the **Fourrier Transform**.
2. The defining property of linear systems allows for an input signal to be decomposed into sine waves that can be sent through the system individually and reassembled/summed up at the output.
3. Linear combinations of different sine waves of the same frequency but different phases will always result in a sine wave of the same frequency.

So if we do know how a linear filter h treats a *sine wave* for any particular frequency and phase, we have completely described that filter and know what it does to any signal $x(t)$. Furthermore what it DOES do to a sine wave input is change its amplitude and shift its phase, but leaves the frequency untouched. We can then derive a **transfer function** $H(\omega)$ of frequency (instead of of time like the filter $h(t)$) that describes the change of amplitude and phase for each frequency given an input signal frequency spectrum $X(\omega)$ (the Fourrier transform of $x(t)$).

So if in an electronic circuit there is a single driving input node receiving a pure sine wave of frequency ω and phase φ, then any other signals in the network should be sine waves of the same frequency. Let us verify this also for those circuit elements introducing differential operators. For instance the current I through a capacitor of capacitance C when the voltage V across it is defined as

$$V := \sin(\omega t + \varphi) \tag{5.4}$$

is given by

$$I = CV\frac{\partial}{\partial t} = C\omega\cos(\omega t + \varphi) = C\omega\sin(\omega t + \varphi + \frac{\pi}{2}) \tag{5.5}$$

So a capacitor will cause a current through it to be a phase shifted copy with different magnitude of a sine wave voltage across it, where the phase $\angle I$ is given by the phase $\angle V$ of the voltage and a $\frac{\pi}{2}$ phase shift.

$$\angle I = \angle V + \frac{\pi}{2} \tag{5.6}$$

and the magnitude $|I|$ increases with frequency (i.e. a capacitor acts more and more like a short at higher frequencies).

$$|I| = C\omega|V| \tag{5.7}$$

Other nodes in our network h of circuit elements with a single sine wave input may now be linear combinations of phase shifted signals and/or may introduce more phase shift. So it is necessary to note that any linear combination of sine waves of the same frequency ω but different magnitude and phase, will again result in a sine wave of the **same frequency** but different magnitude and phase. Figure 5.1 illustrates that geometrically, using the fact that the projection of a vector onto the y-axis is the sine of the angle of the vector's polar representation multiplied with the vector length. So we assume two sine waves $a\sin(\omega t)$ and $bsin(\omega t + \varphi)$ with a relative phase shift φ. If you want to compute the resulting sine wave $c\sin(\omega t + \gamma)$ when summing those two sine waves you can do a vector summation of the polar representation vectors with their angle being the argument of the sine function and their length being the sine waves' amplitudes. The resulting vector's length is c and its polar form angle is γ. From figure 5.2 showing the situation at $t = 0$ we can glean the geometrical relationship:

$$\begin{aligned} a\sin(\omega t) + bsin(\omega t + \varphi) &= c\sin(\omega t + \gamma) \\ c &= \sqrt{(b\sin(\varphi))^2 + (b\cos(\varphi) + a)^2} \\ \gamma &= \tan^{-1}\left(\frac{b\sin(\varphi)}{b\cos(\varphi) + a}\right) \end{aligned} \tag{5.8}$$

Coincidentally (or not so coincidentally) this geometrical analysis can nicely be mapped to complex number representation in the (Re,Im) plain using polar form. So it happens to be the case that one can actually use complex number arithmetic to express the exact same relation: let A and B be complex numbers which are expressed in polar form $A = ae^{j\omega t}$ and $B = be^{j\omega t + \varphi}$. Then it so happens that

$$A + B = ae^{j\omega t} + be^{j\omega t + \varphi} = ce^{j\omega t + \gamma} \tag{5.9}$$

Where c and γ comply with (5.8).

Also, the relation (5.5) can actually be achieved when sine waves I and V are represented as complex numbers where $I = |I|e^{j\omega t + \angle I}$ and . Then the relation (5.5) can simply be expressed as:

$$I = e^{j\frac{\pi}{2}}\omega CV = j\omega CV \tag{5.10}$$

Thus, to deduce currents and voltages in a linear circuit network in frequency space one can use Kirchhoff's laws and Ohm's law just like with only resistors/conductors, but now using complex number arithmetic and any capacitor or inductor enters the equations as a frequency dependent imaginary

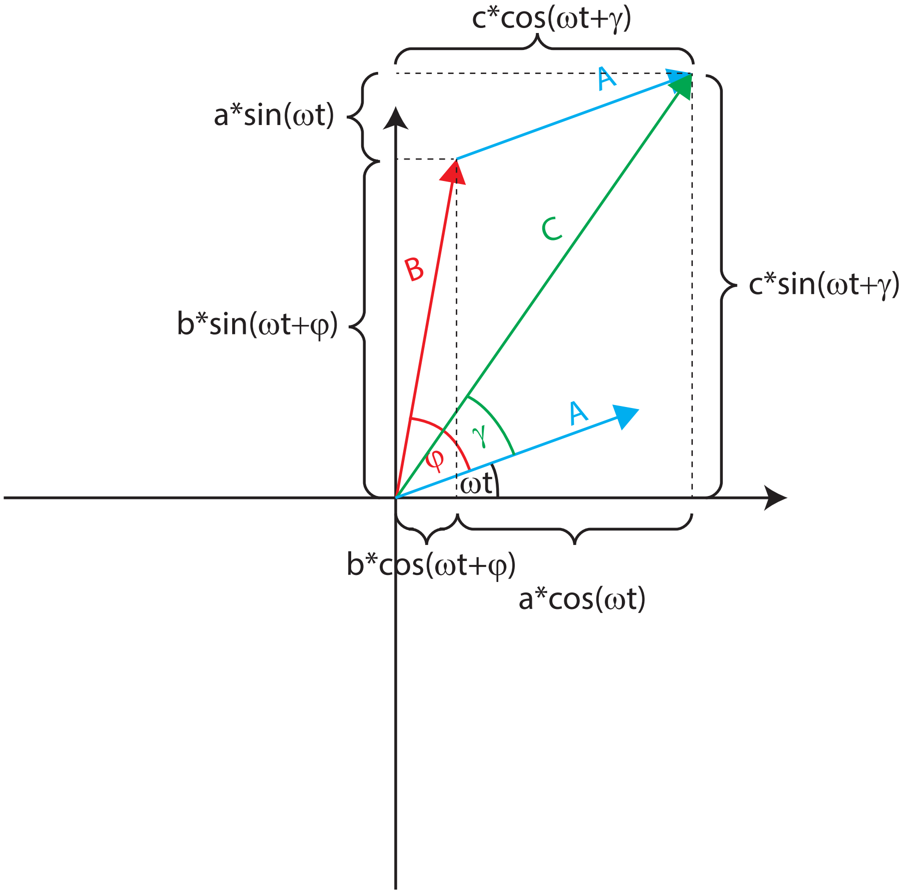


Figure 5.1: Geometrical illustration of the fact that a linear combination of two sine waves of the same frequency will result in a sine wave of the same frequency at different phase and magnitude. (This is ultimately the explanation why complex number arithmetic is so convenient to describe transfer functions, as will be shown later.)

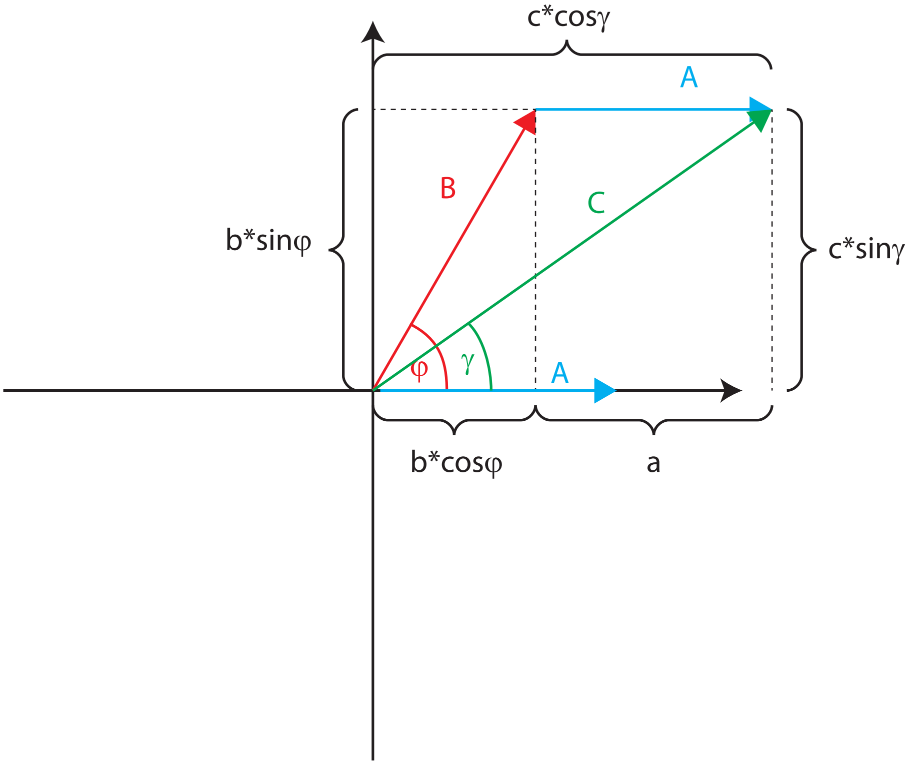


Figure 5.2: The same illustration as in figure 5.1 at $t = 0$. It can conveniently be used to derive c and γ as given in equation (5.8).

	time-space	frequency-space ($s := j\omega$)	
		impedance	admittance
resistor	$V = IR$	R	$\frac{1}{R}$
capacitor	$I = CV\frac{\partial}{\partial t}$	$\frac{1}{sC}$	sC
inductor	$V = LI\frac{\partial}{\partial t}$	sL	$\frac{1}{sL}$
transconductance	$I_O = G_M V_I$		
transresistance	$V_O = R_M I_I$		
voltage amp	$V_O = A_V V_I$		
current amp	$I_O = A_I I_I$		

Table 5.1: A table of circuit elements in linear electronic systems.

component as indicated in table 5.1. Note that one refers to a resistance (generally using the letter R) as the constant that turns a current into a voltage if that constant has no imaginary part. If that constant may have an imaginary component one refers to it more generally as **impedance**, generally using the letter Z. Correspondingly a conductance (the inverse of a resistance, i.e the constant applied to a voltage to result in a current not containing an imaginary component, generally represented by the letter G) that may have an imaginary component is more generally referred to as **admittance** , generally represented by letter Y. (Do not get confused in the following where Y is often used for the output of a filter!)

So if a known linear electronic circuit is used as a linear filter $y(t) = h(x(t))$, i.e. one (or some) element terminal's current or voltage is considered the input of the filter and another (or others) element's terminal current or voltage the output, one can express the input-output relationship as the output equal to the input multiplied with a linear combination of all the impedances/admittances of the network. So the relationship $y(t) = h(x(t))$ expressed as differential equation in the time space becomes a simple multiplication in frequency space $Y(\omega) = H(\omega) * X(\omega)$. For each frequency ω, $H(\omega)$ will be a complex number $H(\omega) = |H(\omega)| * e^{j\angle H}$ and the multiplication with it will cause a multiplication of the amplitude of the input with $|H|$ and a phase shift $\angle H$.

$$|Y| = |H||X| \tag{5.11}$$

$$\angle Y = \angle H + \angle X \tag{5.12}$$

5.2 Bode plot, poles and zeros

H can conveniently be described with a **Bode plot** where its magnitude and phase shift are plotted in two graphs vs frequency ω. See the example in figure 5.3! It shows the Bode plot of a so called low pass filter, where DC voltages and low frequency AC voltages 'pass' through with a constant amplification $A_{DC} := |H(0)| = 100$, but with $|H(\omega)|$ starting to decline at about $\omega = 10^5$ and decline more steeply at $\omega = 10^7$. This is typical for what is called a second order low pass filter. Also note (for later reference) that the phase shift is in the middle of a smooth step of a total of $-\frac{\pi}{2} = -90^o$ at those two frequencies. Note here that the ω axis and the $|H|$ axis usually are given in logarithmic scale

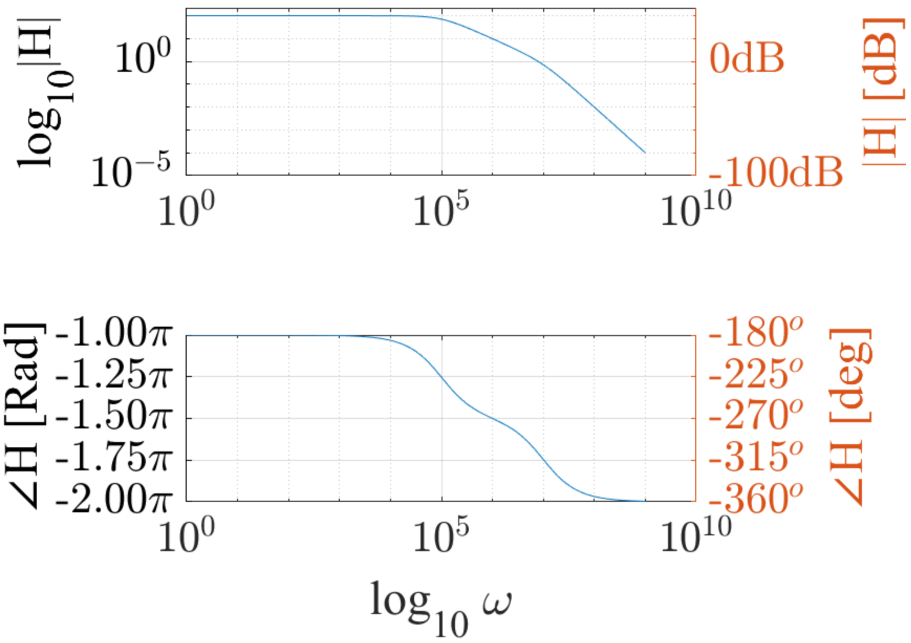


Figure 5.3: An example of a Bode plot of a second order low pass filter, with real poles at frequencies $\omega = 10^5$ and $\omega = 10^7$

to better represent $\frac{1}{\omega^i}$ hyperbolic decay. That's for low pass filters or also ω^i hyperbolic growth for high pass or band pass filters, or in general ω^i where i can be positive or negative: a factor ω^i will result in a straight line with inclination i. In particular the logarithmic $|H|$ axis is usually drawn in Decibel (dB), for historical reasons and to annoy students (as well as this author), where 20dB correspond to a factor 10 in amplitude:

$$\begin{aligned} A_{\mathrm{dB}} &= 20 \log_{10} A_{\mathrm{lin}} \\ A_{\mathrm{lin}} &= 10^{\frac{A_{\mathrm{dB}}}{20}} \end{aligned} \tag{5.13}$$

In this example in figure 5.3 H is:

$$H = A_{DC} \frac{1}{(1 + \frac{s}{\omega_{p_1}}) + (1 + \frac{s}{\omega_{p_2}})} \tag{5.14}$$

Where $A_{DC} = -100$, $\omega_{p_1} = 10^5$, and $\omega_{p_2} = 10^7$. The phase shift here starts at $-\pi\mathrm{Rad} = -180^o$, because A_{DC} is negative. So already at low frequencies the output has a -180^o phase shift to begin with.

Important notice: all other literature out there chooses to start the Bode plot of both inverting and non-inverting low pass transfer functions with a **zero degrees (0^o)** phase shift by convention and by long standing tradition. The author of this compendium chooses to break with this tradition since some of the subsequent explanations (in particular of phase margin in section 5.3.2) become easier (i.e. less confusing) that way, in his opinion. But be aware that most other texts will show Bode plots of inverting low pass filters having a 0^o phase shift at DC!!!

5.2.1 Root form of the transfer function

In general, when solving a linear electronic network's I/O relationship, e.g. $v_o = Hv_i$, H can be written as the ratio of two polygons in s, i.e.:

$$H = \frac{A}{B} = \frac{a_0 + a_1 s + a_2 s^2 ... a_n s^n}{b_0 + b_1 s + b_2 s^2 ... b_m s^m} \tag{5.15}$$

And for convenience this expression is usually manipulated into **root form** by bringing $\frac{a_0}{b_0}$ as a constant in front of the division, and then finding the solutions $\{-\omega_{z_1}, ..., -\omega_{z_n}\}$ for $\frac{A}{a_o} = 0$, and $\{-\omega_{p_1}, ..., -\omega_{p_m}\}$ for $\frac{B}{b_o} = 0$. Note that in this text we sometimes refer to these solutions as **poles** using only the letter $p_i = -\omega_{p_i}$ and **zeros** using only the letter $z_i = -\omega_{z_i}$. Correctly the inverse signed poles and zeros, i.e. ω_{p_i} and ω_{z_i} should be referred to as the **pole frequencies** and the **zero frequencies**. One can rewrite (5.15) as a product of first order terms:

$$H = \frac{a_0}{b_0} \frac{(1 + \frac{s}{\omega_{z_1}})...(1 + \frac{s}{\omega_{z_n}})}{(1 + \frac{s}{\omega_{p_1}})...(1 + \frac{s}{\omega_{p_m}})} \tag{5.16}$$

Note that if one solution for $\frac{A}{a_o} = 0$ (or correspondingly $\frac{B}{b_o} = 0$) is $s = 0$, then there is one first order term in the product that is simply s. Also for convenience, let us choose to order these solutions by their magnitude, i.e. $|\omega_{z_1}| \leq |\omega_{z_2}| \leq$

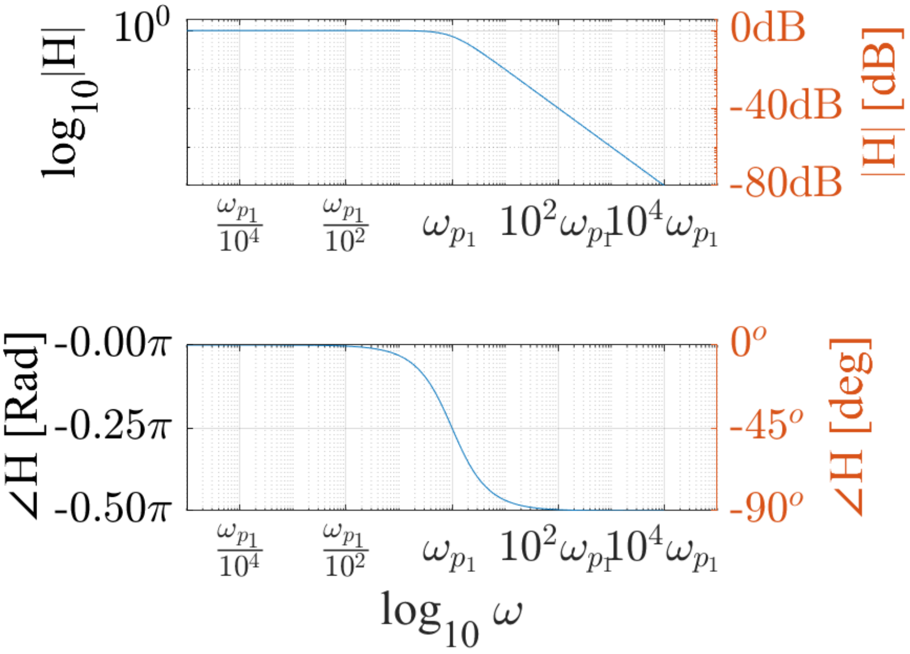


Figure 5.4: Contribution of a single pole to the Bode plot. Compare equation (5.17).

$... \leq |\omega_{z_n}|$ and $|\omega_{p_1}| \leq |\omega_{p_2}| \leq ... \leq |\omega_{p_m}|$. So if $s \in \{z_1, ..., z_n\}$, H becomes zero ($H = 0$), and if $s \in \{p_1, ..., p_m\}$ H becomes infinite ($H = \infty$), i.e. has a pole. Thus one calls $\{\omega_{z_1}, ..., \omega_{z_n}\}$ **zero frequencies** and $\{\omega_{p_1}, ..., \omega_{p_m}\}$ **pole frequencies** .

Let's look at the first pole and for now assume it is a negative real number. Consequently the pole frequency $\omega_{p_1} = -p_1$ is positive. When evaluating the function H for $s = j\omega$, $s = j\omega_{p_1}$ is not the pole that would make $|H|$ grow to infinity. Something else of interest is happening at that point: as long as $|s| = \omega << \omega_{p_1}$ the term $\frac{1}{(1+\frac{s}{\omega_{p_1}})} \approx 1$, so it does not affect the transfer function much. But as $|s| >> \omega_{p_1}$ the term becomes $\frac{1}{(1+\frac{s}{\omega_{p_1}})} \approx \frac{\omega_{p_1}}{s} = \frac{\omega_{p_1}}{j\omega} = -j\frac{\omega_{p_1}}{\omega}$. So this term adds hyperbolic decay to the magnitude $|H|$ (i.e. diminishing the response of H proportionally with $\frac{1}{\omega}$) and also adds a $-\frac{\pi}{2}$Rad $= -90^o$ phase shift to H. When $\omega = \omega_{p_1}$ exactly, so between those two effects, the the term is $\frac{1}{(1+\frac{s}{\omega_{p_1}})} = \frac{1}{(1+j)} = \frac{1}{\sqrt{2}e^{j\frac{\pi}{4}}} = \frac{1}{\sqrt{2}}e^{-j\frac{\pi}{4}}$, i.e. diminishing the magnitude by $\frac{1}{\sqrt{2}}$ and adding a phase shift of $-\frac{\pi}{4}$Rad $= 45^o$. So in summary for each pole frequency p_i it's corresponding contribution to the transfer function is:

$$\frac{1}{1+\frac{s}{\omega_{p_i}}} \begin{cases} \approx 1 & \text{if } |s| << \omega_{p_i} \\ = \frac{1}{\sqrt{2}}e^{-j\frac{\pi}{4}} & \text{if } |s| = \omega_{p_i} \\ \approx -j\frac{\omega_{p_i}}{\omega} & \text{if } |s| >> \omega_{p_i} \end{cases} \tag{5.17}$$

And the angle it is contributing is:

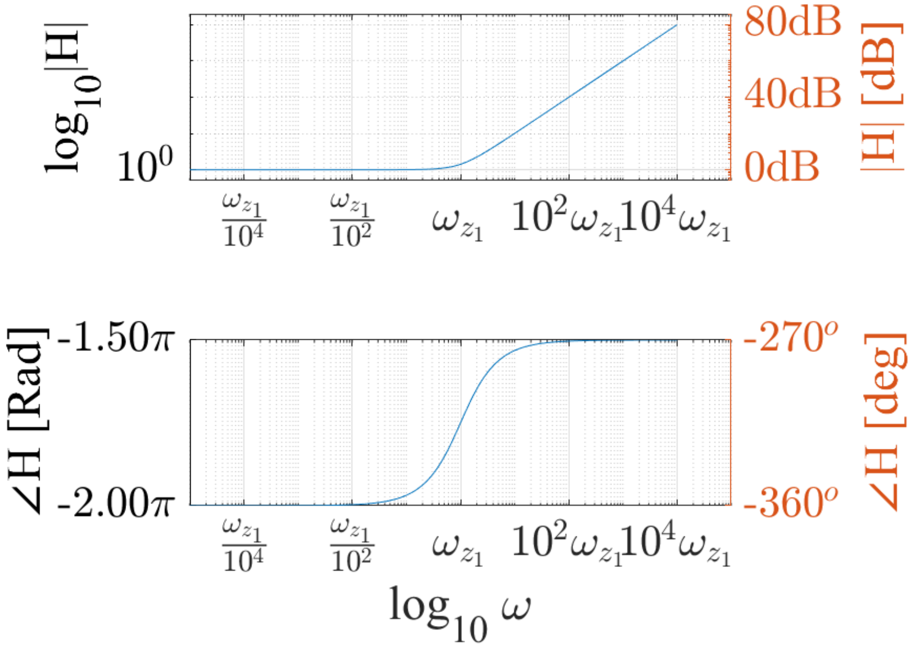


Figure 5.5: Contribution of a single *negative* zero to the Bode plot. Compare equation (5.19).

$$\angle\left(\frac{1}{1+\frac{s}{\omega_{p_i}}}\right) = -\arctan\frac{\omega}{\omega_{p_i}} \tag{5.18}$$

So a Bode plot of a single pole transfer function with no gain looks like figure 5.4. Individual contributions can be ’added’ on the loglog magnitude plot, i.e. in fact they are multiplied. And also the contributions to the phase shift can be added, in this instance it’s an addition for real.

Doing the same consideration for zeros on gets (also see figure 5.5):

$$1+\frac{s}{\omega_{z_i}}\begin{cases}\approx 1 & \text{if } |s| << \omega_{z_i}\\ = \sqrt{2}e^{j\frac{\pi}{4}} & \text{if } |s| = \omega_{z_i}\\ \approx j\frac{\omega}{\omega_{z_i}} & \text{if } |s| >> \omega_{z_i}\end{cases} \tag{5.19}$$

and

$$\angle\left(1+\frac{s}{\omega_{z_i}}\right) = \arctan\frac{\omega}{\omega_{z_i}} \tag{5.20}$$

So the zeros (if they are real and negative) contribute a positive $\frac{\pi}{2}$ phase shift for frequencies higher than the pole frequency and hyperbolic growth proportional to ω.

Interestingly, *positive* real zeros do also commonly occur. In that case they also contribute hyperbolic growth, but now *negative* phase shift (like the negative real poles, also compare figure 5.6):

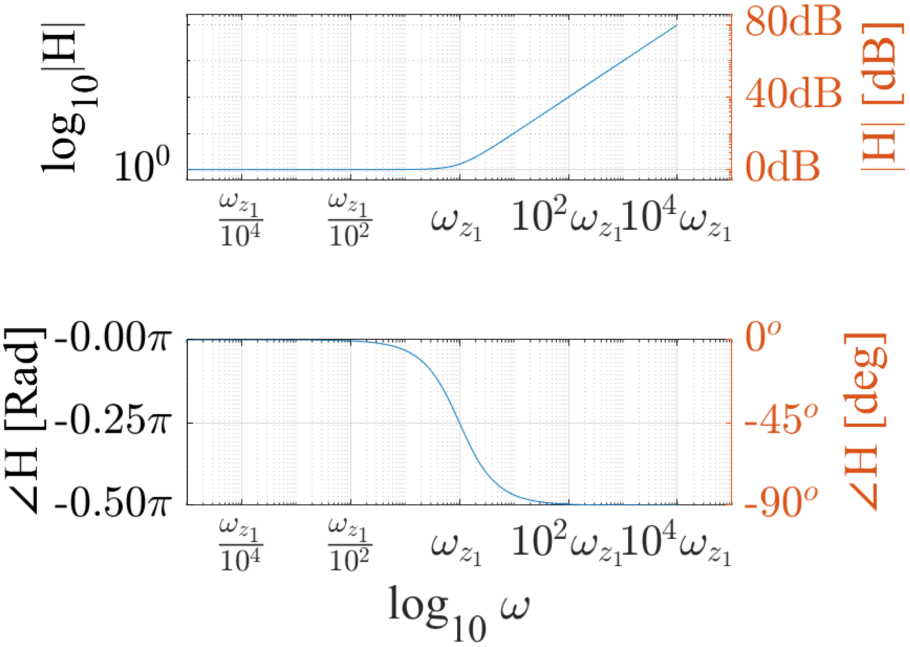


Figure 5.6: Contribution of a single *positive* zero to the Bode plot. Compare equation (5.21).

$$1 - \frac{s}{\omega_{z_i}} \begin{cases} \approx 1 & \text{if } |s| << \omega_{z_i} \\ = \sqrt{2}e^{-j\frac{\pi}{4}} & \text{if } |s| = \omega_{z_i} \\ \approx -j\frac{\omega}{\omega_{z_i}} & \text{if } |s| >> \omega_{z_i} \end{cases} \tag{5.21}$$

and

$$\angle\left(1 - \frac{s}{\omega_{z_i}}\right) = -\arctan\frac{\omega}{\omega_{z_i}} \tag{5.22}$$

5.2.2 How to draw Bode plots from the real poles and zeros of a transfer function

So lets go back to the Bode plot example in figure 5.3 and the corresponding transfer function already given in root form in (5.14). The equation has only two non-zero poles, and no zeros. So that makes the curve flat to start with. At the first pole frequency the decay kicks in. Note that at the pole frequency the $\frac{1}{\sqrt{2}}$ reduction corresponds to the often used **-3dB** to refer to the **cut off frequency** (here we shall refer to it as $\boldsymbol{\omega_H}$) of a filter, i.e. the frequency at which the decay or **roll off** sets in and the power (proportional to the square of the amplitude) has been reduced to $\frac{1}{2}$. Note that the slope settles onto a straight line in the log-log plot of a 1:1 slope (i.e. x-times bigger frequency, means x-times smaller amplitude), or in dB the slope is 20dB per decade. Also on the phase plot it's the middle of a $-\frac{\pi}{2}$Rad smooth step transition. At the

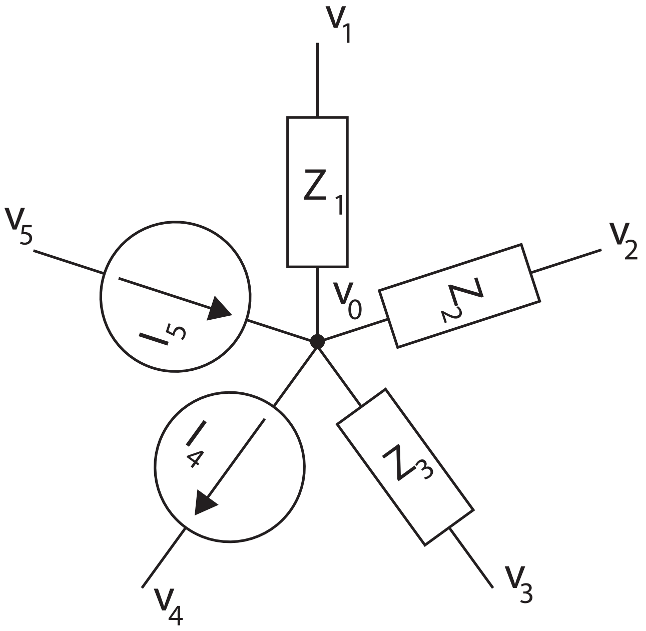


Figure 5.7: Example of a node in a circuit to illustrate a method to quickly write down the equation resulting from Kirchhoff's current law for node v_0.

next pole, the slope becomes steeper, since now $|H|$ diminishes with $\frac{1}{\omega^2}$. So for every multiplication of the frequency with x the magnitude shrinks by x^2, which again is reflected in the log-log plot with a straight line but now falling 2:1 or 40dB per decade.

From those observations we can deduce a general set of rules how to draw Bode plots knowing the poles and zeros of a transfer function, provided all poles and zeros are real numbers. It is simply the superposition from all contributions of the individual poles and zeros:

1. In the magnitude plot there is a convex knee at each real pole frequency changing the slope by -20dB/decade, and concave knee for each real zero changing the slope by +20dB/decade.

2. In the phase plot there is a step down of $-\frac{\pi}{2}$ for each negative real pole and each positive real zero and a step up by $\frac{\pi}{2}$ for each negative real zero.

3. The curve is flat if it has an equal number of poles and zeros at lower frequencies (so for example with no zero or pole at point 0 the curve starts flat.)

5.2.3 Example: General Transconductance Amplifier

So in the following let us look at the general transconductance amplifier model of figure 9.4 (redrawn with some colour-coding in figures 5.8 and 5.9) as an example that will be very useful later on and try to figure out its transfer function. Let

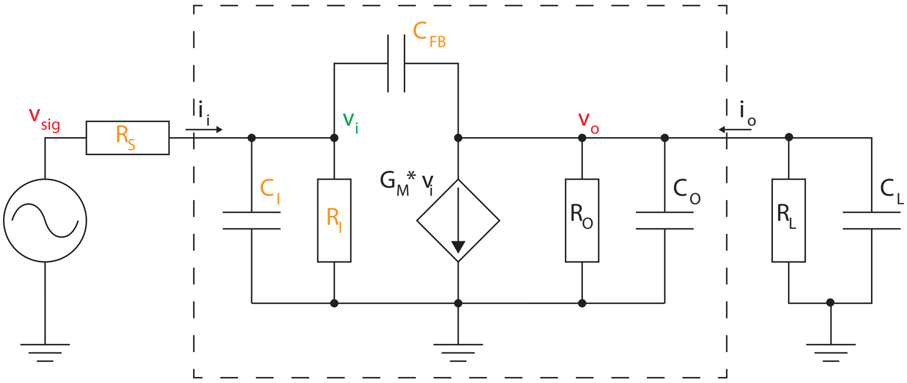


Figure 5.8: Color marked general transconductance amplifier illustrating derivation of Kirchhoff's current law equation for node v_i.

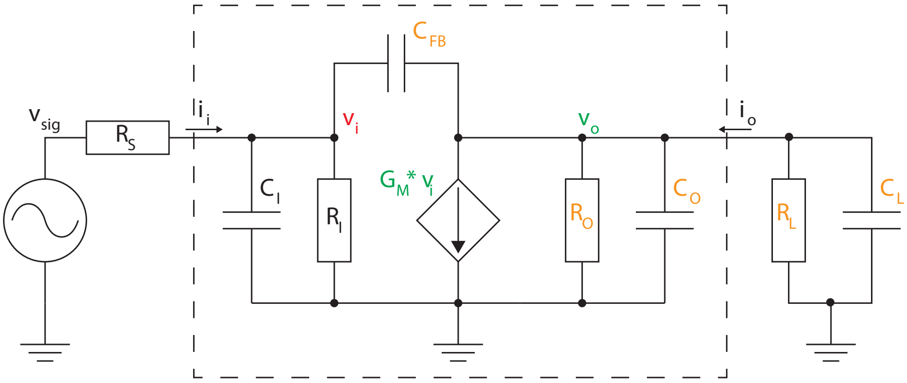


Figure 5.9: Color marked general transconductance amplifier illustrating derivation of Kirchhoff's current law equation for node v_o.

us first introduce a systematic method for deriving the Kirchhoff's current law equations for each node with unknown voltage in a electric circuit/network. Have a look at the example in figure 5.7. The method is to write down the equation for each node with unknown voltage, by adding on the left hand side of the equation the voltage of the node in question, here it's v_0, multiplied with all the admittances (i.e. the inverse of the impedances) connected to that node, as well as any explicit current source drawing current *out* from node v_0. Then place on the right hand side the sum of all voltages appearing on the other end of the admittances multiplied with that admittance, as well as any explicit current sources supplying current *into* v_0. So for this example we get:

$$v_0\left(\frac{1}{Z_1}+\frac{1}{Z_2}+\frac{1}{Z_3}\right)+I_4=v_1\frac{1}{Z_1}+v_2\frac{1}{Z_2}+v_3\frac{1}{Z_3}+I_5 \qquad (5.23)$$

So let's set up the Kirchhoff current law equations for the two nodes v_i and v_o for teh schematics in figures 5.8 and 5.9. (v_{sig} is a source and comes from an ideal voltage source, so it's not an unknown and can thus be ignored here.) So for node v_i you may look at figure 5.8 where some labels are marked in colour for easier understanding. To simplify the following expressions a bit we combine the parallel resistors into $R_L^* := R_O||R_L$ and parallel capacitors $C_L^* := C_L||C_O$. Also useful will be an expression $R_I^* = R_I||R_{SIG}$. We construct the equation by placing all contributions to current flowing *out* of v_i on the left hand side and all contributions to currents flowing *into* v_i on the right hand side: we multiply v_i (marked in green) with all admittances (i.e. impedances need to be inverted!) connected to it (marked in orange) and place that term on the left hand side. Furthermore we would add any explicit current sources of currents entering v_i on the left hand side as well (see the derivation of equation for v_o.) For the right hand side we multiply the same conductances with the voltages on their other side (marked in red) and we would add any explicit current sources of currents *into* v_i. Note that the node Gnd is defined as zero and thus is ignored. This yields equation (5.24). Doing the same for node v_o (colour marked figure 5.9) yields equation (5.25). Note the explicit current source added to the left hand side, as it is a current flowing *out* of v_o.

$$\begin{aligned} v_i\left(\frac{1}{R_{SIG}}+\frac{1}{R_I}+s\left(C_{FB}+C_I\right)\right) &= v_{sig}\frac{1}{R_{SIG}}+v_osC_{FB} && (5.24)\\ v_o\left(\frac{1}{R_L^*}+s\left(C_{FB}+C_L^*\right)\right)+v_iG_M &= v_isC_{FB} && (5.25)\end{aligned}$$

Solving this to get the transfer function $\frac{v_o}{v_{sig}}$ yields a rather complicated expression:

$$\begin{aligned} \frac{v_o}{v_{sig}} &= -G_MR_L^*\frac{R_I}{R_I+R_{SIG}}\;\frac{(1-s\frac{C_{FB}}{G_M})}{1+sB+s^2A}\\ B &= \left[R_L^*(C_{FB}+C_L^*)+R_I^*\left(C_I+C_{FB}(1+G_MR_L^*)\right)\right]\\ A &= R_I^*R_L^*\left[(C_{FB}+C_L^*)(C_{FB}+C_I)-C_{FB}^2\right]\end{aligned} \qquad (5.26)$$

This is result is OK to make a Bode plot with a maths program on a computer but way too complicated to easily read any directly useful information out of it.

However, making some assumptions of extreme cases we can glean some useful practical information.

5.2.4 Dominant Pole Approximation

For low pass filters, oftentimes one is mostly interested in the pole with the lowest frequency. If that first pole p_1 occurs at a frequency much lower (for practical purposes a factor 4 lower) than any other poles or zeros, the shape of the cut off 'knee' and thus the -3dB frequency and the filter's band width (BW) is entirely determined by that **dominant pole** alone.

In mathematical terms, one can find a dominant pole directly from the polynomial, if the first order parameter is much bigger than any higher order parameters, provided all pole frequencies are bigger than 1Rad. So if the transfer function's denominator is $b_0 + b_1 s + b_2 s^2 ... b_m s^m$, then if there is a dominant pole that pole is approximated by

$$\forall\{i > 1 : |b_i| << |b_1|\} \Leftrightarrow \frac{1}{\omega_{p_1}} \approx b_1 \tag{5.27}$$

Just to illustrate *why* that is the case, we can look at a second order example, where the polynomial can be expressed as:

$$1 + \left(\frac{1}{\omega_{p_1}} + \frac{1}{\omega_{p_2}}\right) s + \frac{1}{\omega_{p_1}\omega_{p_2}} s^2 \tag{5.28}$$

So if $\omega_{p_2} >> \omega_{p_1} > 1$ then the term $\frac{1}{\omega_{p_1}} + \frac{1}{\omega_{p_2}}$ is dominated by $\approx \frac{1}{\omega_{p_1}}$ and $\frac{1}{\omega_{p_1}\omega_{p_2}} << \left(\frac{1}{\omega_{p_1}} + \frac{1}{\omega_{p_2}}\right)$.

If one makes a stronger assumption about the dominant pole, namely that its frequency is smaller than any other poles' or zeros' by a factor corresponding to the DC gain $\forall\{i > 1, j : \omega_{p_1} * A_{DC} < \omega_{p_i} \wedge \omega_{p_1} * A_{DC} < \omega_{z_j}\}$, then any pole or zero occur only at frequencies beyond the **unity gain frequency** ω_T (i.e. where $|A(j\omega_T)| = A * \frac{1}{\omega_T} = 1$, because of the -20dB roll off) and for all practical purposes one can simply treat the filter as a first order filter having a single pole only.

Hunting for a dominant pole in circuit schematics terms one can make some assumptions that eliminate the temporal dynamics from all but a single node (such as 'if this resistor is zero' or 'this capacitor is negligibly small relative to others'), assuming that the dominant pole occurs at that node. Let's get back to our example for this:

5.2.5 Transconductance Amplifier: Dominant Pole at Output

For instance if the input node of the general transconductance amplifier is directly driven from the ideal voltage source and thus not affected by high frequency, i.e. if we consider $R_{SIG} = 0$, then only v_o will experience any influence of frequency and equation (5.26) becomes a simple first order low pass transfer function with one pole and one zero.

$$\frac{v_o}{v_{sig}} = -G_M R_L^* \frac{1 - s\frac{C_{FB}}{G_M}}{1 + s\left(R_L^*(C_{FB} + C_L^*)\right)} \tag{5.29}$$

So one pole and one zero:

$$\omega_{p_1} = \frac{1}{R_L^*(C_{FB} + C_L^*)} \tag{5.30}$$

$$\omega_{z_1} = \frac{G_M}{C_{FB}} \tag{5.31}$$

Note that the unity gain frequency ω_T for this transfer function is:

$$\omega_T = |A_{DC}|\omega_{p_1} = \frac{G_M}{C_{FB} + C_L^*} \tag{5.32}$$

and thus as long as $G_M > \frac{1}{R_L^*}$ we can state that:

$$\omega_{p_1} < \omega_T < \omega_{z_1} \tag{5.33}$$

So the zero after the unity gain frequency does usually not have an important impact.

5.2.6 Transconductance Amplifier: Dominant Pole at Input

On the other hand if we consider the output to be simply $v_o = G_M R_L^* v_i$ and only v_i to be affected by frequency, i.e. we modify (5.25) to

$$v_o \frac{1}{R_L^*} + v_i G_M = 0 \tag{5.34}$$

Note that this is a rather strong assumption! So this really only works if the pole resulting at the input node is dominant. If on the other hand v_o starts to decline at lower frequencies, then the negative feedback current flowing through C_{FB} to the input node will be much diminished and the pole we derive here will be pushed to higher frequencies in the real system, i.e. when using the full equation (5.26). So be aware of that limitation!

OK, but if we accept that the input node is affected by the dominant pole and not the output node directly, then (5.26) becomes:

$$\frac{v_o}{v_{sig}} = -G_M R_L^* \frac{R_I}{R_{SIG} + R_I} \frac{1}{1 + sR_I^*\left(C_I + C_{FB}(1 + G_M R_L^*)\right)} \tag{5.35}$$

5.2.7 Transconductance Amplifier: Miller Approximation for Dominant Pole at Input

Figure 5.10 offers a simplified view that yields the same result, where the serial capacitor between input and output can be changed into equivalent capacitors at the input and output towards Gnd where:

$$C_{MI} = (1 + A_{DC})C_{FB} \quad = \quad (1 + G_M R_L^*)C_{FB} \tag{5.36}$$

$$C_{MO} = \left(\frac{1}{A_{DC}} + 1\right) C_{FB} \tag{5.37}$$

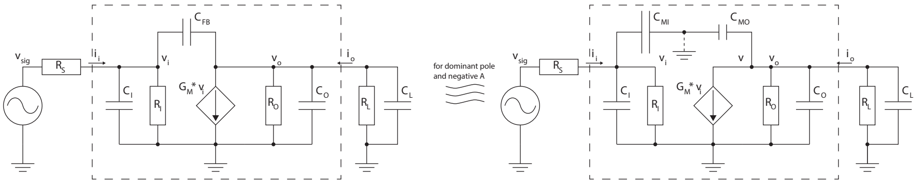


Figure 5.10: An illustration of the 'Miller effect' that makes an the feedback capacitor C_{FB} appear much bigger at the input of an inverting amplifier. To derive a dominant pole at the input node, it can be replaced with two capacitors in series of the right size to create a virtual Gnd between them. This way they can be considered capacitors adding in parallel to the input and an output capacitors. If the dominant pole is at the input this can directly be used to derive this dominant time constant. (Caveat: This is safely true only as long as the DC relation $v_o = v_i G_M(R_O||R_L)$ is not impaired. So if that pole at node v_i is not clearly dominant, this approximation becomes imprecise.)

This is under the assumption that the DC gain is negative/inverting and the gain between v_i and v_o is not yet impaired by a pole or zero: so this simplification can be safely adopted to find a dominant pole at the node v_i of a low pass filter. With this simplification the input section and output section remain independent stages and one can derive a separate transfer function for the input stage, while the transfer function from v_i to v_o is equivalent to its DC transfer function $\frac{v_o}{v_i} = G_M R_L^*$. The time constant for the input stage then is

$$\begin{aligned}\tau_i &= R_I^* \left(C_I + C_{FB}(A_{DC} + 1)\right) \\ &= R_I^* \left(C_I + C_{FB}(G_M R_L^* + 1)\right)\end{aligned} \tag{5.38}$$

And the resulting transfer function is exactly the same as (5.35).

Note that using the Miller approximation for a dominant pole at the output using C_{MO} does not work quite so well: the condition that $\frac{v_o}{v_i} = G_M R_L^*$ is not valid at the cut off frequency. Thus the analysis for a dominant pole at the output in equation (5.29) does NOT yield a pole that is $\frac{1}{R_L^*\left(C_O + C_{FB}(\frac{1}{A_{DC}}+1)\right)}$. Quite the contrary: from equation (5.29) the pole frequency is higher as C_{FB} appears without influence of $G_M R_L^*$ and the zero frequency $\frac{G_M}{C_{FB}}$ may move the cut off to even higher frequencies[1].

5.2.8 Cut Off Frequency by Open-Circuit Time Constant Approximation

The **Open-Circuit Time Constant Approximation** is a more general method to get an approximate cut-off or -3dB frequency. There is no guarantee though that this cut-off frequency will be dominant. So it is good to approximate the BW, but one should not assume first order behaviour for the frequency response beyond that point.

[1] In digital circuits the Miller effect onto the output node *is* in fact used, albeit slightly differently. See 7.1.2!

************* Will be filled in later in the semester *****************

5.3 Stability

Linear filters where the signal is only processed in a feed forward fashion in all stages will not have stability issues. However, real electronic filter implementations will often include circular signal dependencies, i.e. contain some feedback loop. Such a feedback loop together with some amplification can potentially lead to self sustaining oscillations or even a 'run away' oscillation with infinitely increasing amplitude.

Unfortunately, even two of the one-transistor amplifying stages (see chapter 9.2) do have an element of feedback, since there is a capacitor between output and input for the CS and CD (i.e. SF) stages. Thankfully, the problem for the CS stage is mostly academic, but somewhat more real for the CD stage. We'll have a look at it as an example later on (subsection 9.4.2).

Furthermore, explicit negative feedback is an often used method in analog circuit design to obtain a precise gain. The designer is well advised to pay close attention to stability issues when employing this method. One would at first glance think that negative feedback per se is actually unproblematic, right? Unfortunately, negative feedback paired with delay/phase shift can in effect turn into positive feedback under the right (i.e. wrong) conditions. So we'll look at some analysis tools to pinpoint the problem in the following.

5.3.1 Looking at 2nd order equation with complex poles

So far we have discussed transfer functions with real poles and zeros. Another example Bode plot thereof with a second order polynomial denominator, no zeros and DC gain $A_{DC} = 1$ is shown in figure 5.11. However, it can occur that the polynomial in the denominator does have complex solutions and then the behaviour becomes more complicated and can indicate stability problems. We can tackle this for 2nd order polynomials in the denominator, i.e. transfer functions that give rise to complex poles. For this the 2nd order polynomial is by convention written as a function of two parameters, the **quality factor** Q and the **characteristic frequency** ω_0:

$$H(s) = ...\frac{...}{1 + \frac{1}{\omega_0 Q}s + \frac{1}{\omega_0^2}s^2} \tag{5.39}$$

So to find the poles one has to solve:

$$0 = 1 + \frac{1}{\omega_0 Q}s + \frac{1}{\omega_0^2}s^2 \tag{5.40}$$

Using the quadratic formula to find the poles p_1, p_2, i.e solutions for s:

$$p_1, p_2 = \frac{-\frac{1}{\omega_0 Q} \pm \sqrt{\frac{1}{\omega_0^2 Q^2} - 4\frac{1}{\omega_0^2}}}{2\frac{1}{\omega_0^2}} = \frac{-\frac{\omega_0}{Q} \pm \sqrt{\frac{\omega_0^2}{Q^2} - 4\omega_0^2}}{2} \tag{5.41}$$

Looking at the expression under the square root, it becomes negative when

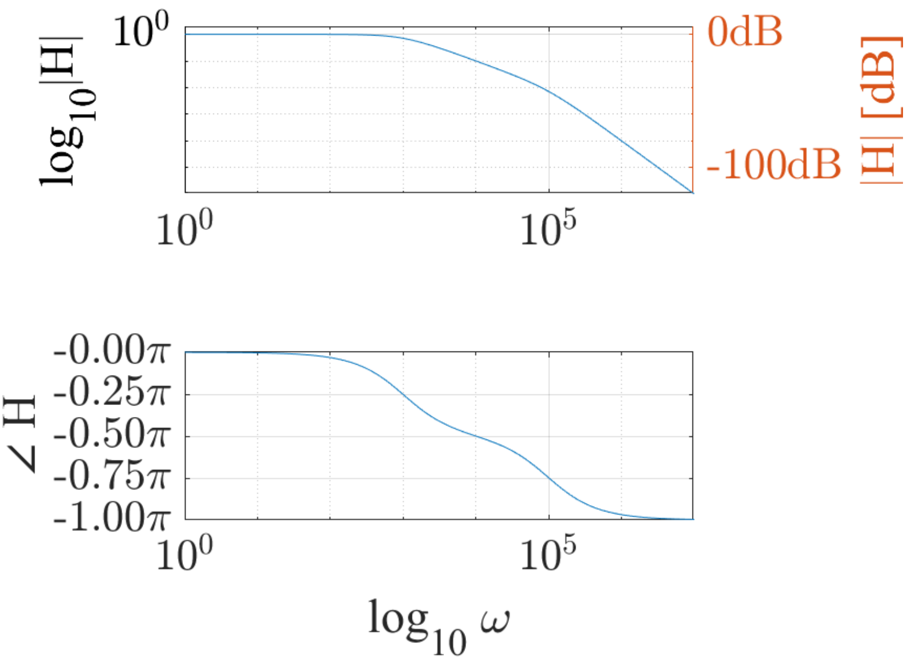


Figure 5.11: 2nd order low pass with $Q = 0.1$ and $\omega_0 = 10^4$Rad.

$$\frac{1}{Q^2} < 4 \Leftrightarrow |Q| > 0.5 \tag{5.42}$$

In that case p_1 and p_2 will be a conjugate complex pair, and their magnitude will be exactly $|p_{1,2}| = \omega_0$. Note that the pole frequencies are also complex numbers, still simply the inverse of the poles: $\omega_{p_i} = -p_i$. So figure 5.11 shows an example Bode plot for a smaller $Q = 0.1$ and $\omega_0 = 10^4$, with two real poles. The poles are centred around ω_0. Then figure 5.12 shows an Bode plot example where $Q = 0.5$ and $\omega_0 = 10^4$. Now there are still two real poles at the exact same frequency $\omega_{p_1} = \omega_{p_2} = \omega_0$. The Bode plot still looks like we are used to, following the instructions for drawing Bode plots with real poles. Since there are two poles at $\omega = 10^4$, we get a drop of -6dB at that point and the roll off will immediately be $\frac{40\text{dB}}{\text{dec}}$.

Now if Q increases some more we get a *conjugate complex* (!) pair of poles. There starts to be a subtle deviation from Bode plots with just real poles, hardly noticeable at first. What happens is that the gain at the cut off frequency increases, i.e. that the -6dB point, which is at ω_0 for two identical real poles, i.e. $Q = 0.5$, is shifted to slightly higher frequencies, while the roll off for very large frequencies remains in place. Another way of saying this is that the 'knee' becomes sharper. This continues until the gain is at -3dB at ω_0, i.e. like for a 1st order LP, but with a second order roll off of $\frac{40\text{dB}}{\text{dec}}$. That point is reached when $\angle\omega_{p_{1,2}} = \pm 45^o$. So the angles of the conjugate complex pair of pole frequencies are $\angle\omega_{p_1} = -\angle\frac{1}{\omega_{p_1}} = -45^o$ and $\angle\omega_{p_2} = -\angle\frac{1}{\omega_{p_2}} = 45^o$ (see figure 5.13), the

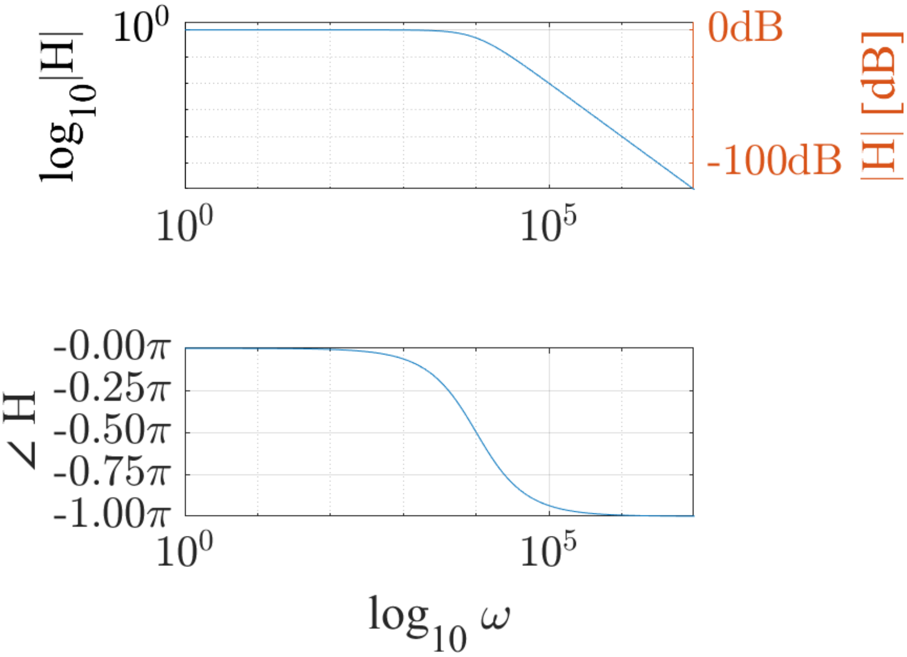


Figure 5.12: 2nd order low pass with $Q = 0.5$ and $\omega_0 = 10^4$Rad. Highest Q resulting in two real poles, i.e. actually the very same pole twice.

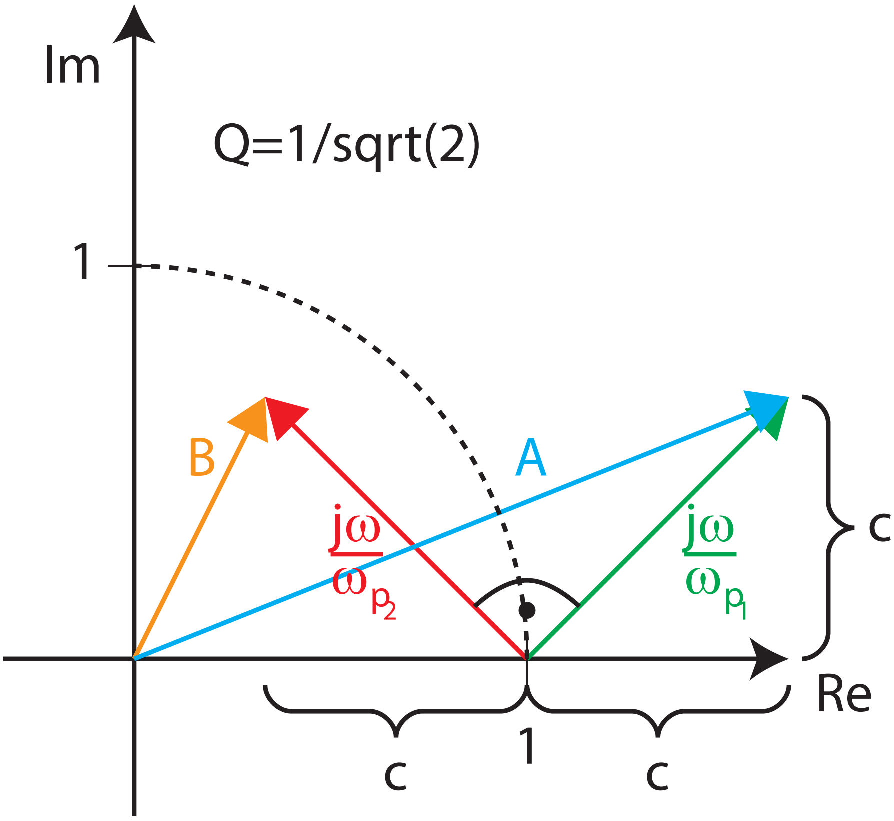


Figure 5.13: Illustration of what happens to the denominator $A * B$ in the complex plain of a 2nd order low pass filter transfer function with $Q = \frac{1}{\sqrt{2}}$ for increasing $j\omega$. Maximally flat response.

imaginary $\pm\sqrt{\frac{\omega_0^2}{Q^2} - 4\omega_0^2}$ part of the poles is the same magnitude as the real part $-\frac{\omega_0}{Q}$. The condition for that is:

$$\left|-\frac{\omega_0}{Q}\right| = \left|\sqrt{\frac{\omega_0^2}{Q^2} - 4\omega_0^2}\right| \Leftrightarrow Q = \frac{1}{\sqrt{2}} \Leftrightarrow \angle\frac{1}{\omega_{p_{2,1}}} = \pm 45^o \tag{5.43}$$

This is the point of maximally flat response, i.e. the BW gets extended by the slight resonance around and beyond the cut off frequency, but the gain still does never exceeds the DC gain . See figure 5.14!

To proof this let's look at how the denominater $(1 + \frac{j\omega}{\omega_{p_1}})(1 + \frac{j\omega}{\omega_{p_2}})$ of the transfer function behaves for increasing ω for when $Q = \frac{1}{\sqrt{2}}$ and then for when $Q > \frac{1}{\sqrt{2}}$. Let us define a variable name for the two multipliers of the denominator $A := 1 + \frac{j\omega}{\omega_{p_1}}$ and $B := 1 + \frac{j\omega}{\omega_{p_2}}$, as well as a variable c that is the magnitude of the real and imaginary part (they are the same length) of the expression $\frac{j\omega}{\omega_{p_{1,2}}}$, i.e. $c := Re(\frac{j\omega}{\omega_{p_{1,2}}}) = Im(\frac{j\omega}{\omega_{p_{1,2}}})$. Then it can be shown that:

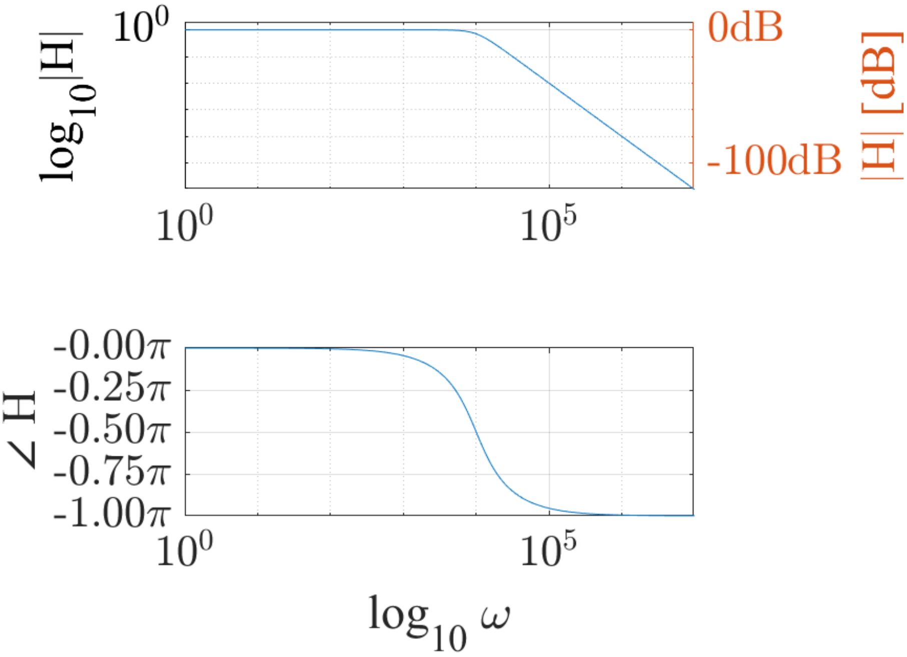


Figure 5.14: 2nd order low pass with $Q = \frac{1}{\sqrt{2}} = 0.71$ and $\omega_0 = 10^4$Rad. Two complex poles, a conjugate complex pair. Maximally flat response, i.e. no resonance just yet.

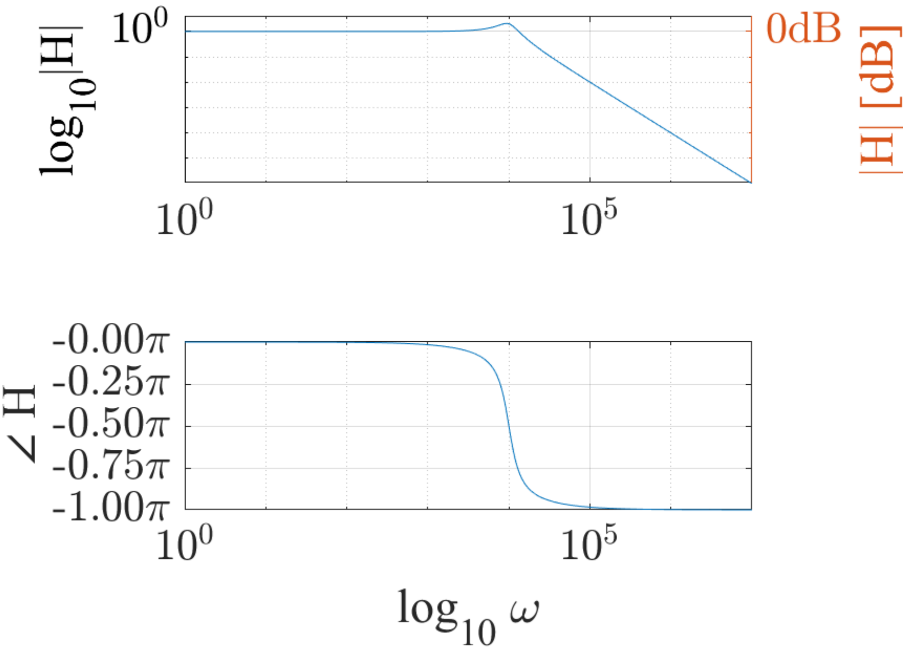


Figure 5.15: 2nd order low pass with $Q = 2$ and $\omega_0 = 10^4$Rad. Substantial resonance peak (not so prominent on the logarithmic scale: the peak looks much more pronounced on a linear scale)

$$|A * B|^2 = 1 + 4c^4 \tag{5.44}$$

So the magnitude of the denominator of the transfer function grows proportionally with $\sqrt{1+4c^4}$ if $Q = \frac{1}{\sqrt{2}}$. In other words it is never smaller than 1, and thus its inverse is never bigger than 1, so no resonance and no 'hump' in the transfer function.

But if $Q > \frac{1}{\sqrt{2}}$ is just a little bigger, then we have $|\text{Im}(\omega_{p_1})| > |\text{Re}(\omega_{p_1})|$ or $|\text{Im}(\frac{1}{\omega_{p_1}})| > |\text{Re}(\frac{1}{\omega_{p_1}})|$ or $\text{Im}(\frac{j\omega}{\omega_{p_{1,2}}}) < \left|\text{Re}(\frac{j\omega}{\omega_{p_{1,2}}})\right|$. We define thus the ratio:

$$r := \frac{\left|\text{Re}(\frac{j\omega}{\omega_{p_{1,2}}})\right|}{\text{Im}(\frac{j\omega}{\omega_{p_{1,2}}})} \tag{5.45}$$

And

$$Q > \frac{1}{\sqrt{2}} \Leftrightarrow r > 1 \tag{5.46}$$

It can be shown that:

$$\frac{1}{|A * B|} > 1 \Leftrightarrow \frac{2(r^2-1)}{(r^2+1)^2} > \text{Im}(\frac{j\omega}{\omega_{p_{1,2}}})^2 \tag{5.47}$$

so for $r > 1$ this equation has a solution (because $\text{Im}(\frac{j\omega}{\omega_{p_{1,2}}})^2$ is always bigger than zero and $\frac{2(r^2-1)}{(r^2+1)^2}$ is bigger than zero only if $r > 1$) and there exists an upper limit for ω where $\frac{1}{|A*B|} > 1$ and we do have some resonance at frequencies up to that limit[2], which becomes visibly pronounced in the Bode plot close to the cut off frequency shortly before the roll off starts as in the example Bode plot in figure 5.15. Interestingly the band where you do get $\frac{1}{|A*B|} > 1$ is widest for $r = \sqrt{3}$.

So now we have analysed a second order low pass transfer function and derived a condition for it having some resonance, most pronounced around the cut off region. Note that 2nd order low pass functions, even if they do express some resonance, will still not be in danger of a 'run away' amplification: the resonance will always be limited, i.e. $\frac{1}{|A*B|} < \infty$.

But what to do if the circuit analysis yields a higher order denominator? Well, there is a bunch of methods to approximate a higher order polynomial with a lower order polynomial. We have mentioned the dominant pole approximation already (See page 61), which approximates a polynomial with a first order polynomial, if the higher order parameters of the polynomial are small (if the pole and zero frequencies are bigger than 1Hz!). You can do the same, if the polynomial's parameters of order higher than two are much smaller than the first two, and reduce the polynomial to a second order polynomial, ignoring the higher orders.

There is also a mathematical way of approximating with a 2^{nd} order function where you use an equivalent pole frequency ω_{eq} as the 2nd pole frequency. This method however, requires that you *know* what all the higher order poles are. It

[2] and if $r < 1$ the condition cannot be fulfilled, ergo no resonance up to that point of r.

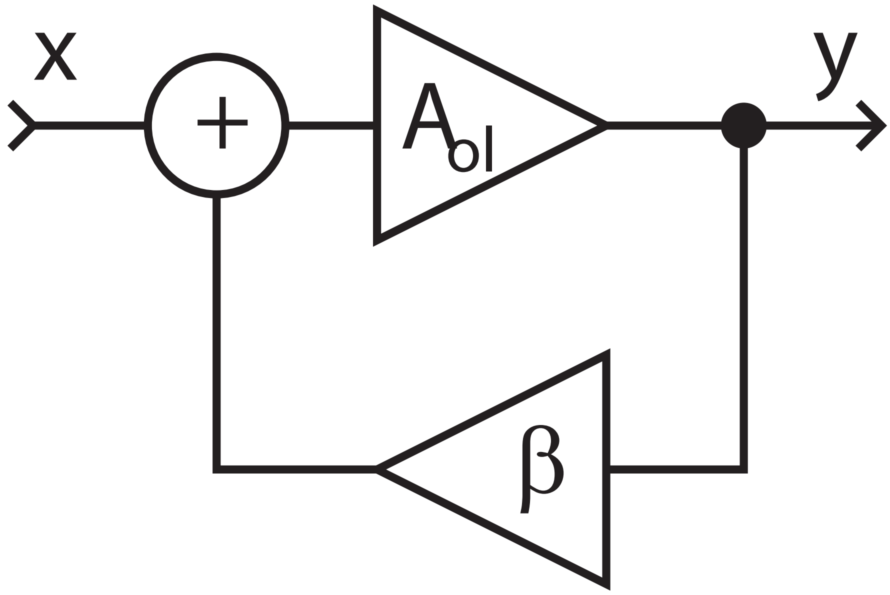


Figure 5.16: General concept of a feedback loop. $A_{ol}(x(s))$ is the **open loop gain** and frequency dependent transfer function. β is often usually a frequency independent constant, but can be frequency dependent as well. The **resulting closed loop gain** A_{cl} is stabilising ($A_{cl} < \infty$) if the total loop gain $A_{ol}\beta < 1$, it is smaller than $A_{cl} < A_{ol}$ if the loop gain is negative $A_{ol}\beta < 0$, and is approaching $A_{cl} \approx \frac{1}{\beta}$ if the loop gain is $A_{ol}\beta << -1$

is described in equation (5.57). However, this not very helpful when faced with a complex circuit analysis: once you have solved it and found all poles, you have already finished the hard part. And replacing the full version with a 2nd order approximation using ω_{eq} to find a Q-value and to assess stability may turn out quite wrong, if the real first two poles are NOT clearly dominant.

Instead we shall in the following look at a more applied method to gauge stability, where it is sufficient to have the transfer characteristics (i.e. the Bode plots) of some circuit building blocks derived from simulation, i.e. it is not necessary to have an explicit root form representation of the transfer function.

5.3.2 Looking at negative feedback, loop gain, and phase margin

A different approach to analyse stability issues of electronic filters due to a feedback loop is possible even for higher order filters without dominant poles if one can pinpoint the feedback loop and disconnect it. This becomes trivial if the circuit is designed that way to begin with. That is, one starts with a filter with a known transfer function (i.e. with a so called (usually frequency dependent) **open loop gain** A_{ol} and connects it in an explicit feedback loop. In fact this is a popular method of designing amplifiers with a specific **closed loop gain** $A_{cl} := \frac{y}{x}$, using a generic operational amplifier with a very high gain in a (negative) feedback loop as illustrated in figure 5.16. Such a feedback loop is stabilizing for any specific frequency if the **loop gain** $A_{ol}\beta < 1$ at that frequency. In particular it will stabilize at an absolute gain smaller than $|A_{ol}|$ if the feedback loop is negative, i.e. $A_{ol}\beta < 0$, and will stabilize with resonance at a level higher than $|A_{ol}|$ for $0 < A_{ol}\beta < 1$. But it will NOT stabilize and end up oscillating uncontrolled if $A_{ol}\beta \geq 1$.[3] More precisely:

$$A_{cl} := \frac{y}{x} = \begin{cases} \infty \text{ (unstable)} & \text{if } A_{ol}\beta \geq 1 \\ \frac{A_{ol}}{1-A_{ol}\beta} & \text{if } A_{ol}\beta < 1 \end{cases} \tag{5.48}$$

and notably for the clearly stable cases, i.e. where the loop gain is very negative ($A_{ol}\beta << 1$):

$$A_{cl} \approx -\frac{1}{\beta} \qquad \text{if } A_{ol}\beta << -1 \tag{5.49}$$

This is actually a method of deriving specific gains from a high gain operational amplifier: in practical terms it is often difficult to design a CMOS amplifier with a specific open loop gain. So rather design one with as high a gain as possible and uses resistive division or capacitive division in a switch cap design to construct a specific β to get a rather precise closed loop gain $A_{cl} \approx \frac{1}{\beta}$, where the approximation becomes more precise, the higher the open loop gain is. The deviation of A_{cl} from $\frac{1}{\beta}$ is often referred to as **closed loop gain error**:

$$A_{cl} - \left(-\frac{1}{\beta}\right) = \frac{A_{ol}}{1-A_{ol}\beta} + \frac{1}{\beta} = \frac{\beta A_{ol} + 1 - A_{ol}\beta}{\beta(1-A_{ol}\beta)} = \frac{1}{\beta}\frac{1}{1-A_{ol}\beta} \tag{5.50}$$

So far so good, but what happens when frequency dependencies enter the picture? Let's just consider the case where there is only frequency dependency in $A_{ol}(\omega)$ but no frequency dependency in $\beta(\omega) = \beta$. Even if one designs such a system to be stable at DC and low frequencies with a negative loop gain well within the 'safe zone' $A_{ol}(0\text{Rad})\beta < 0$, well, the situation becomes complicated when starting to include complex number arithmetic into the equation. The kind of circuits we look at here are low pass filters. For those, increase in frequency brings a reduction in gain and at the same time a phase shift. Let's assume we have a save stable behaviour at DC with $A_{ol}(0\text{Rad})\beta < 0$, i.e. the phase shift of the loop gain to begin with is $\angle A_{ol}(0\text{Rad})\beta = -180^o$. At higher frequencies we may get additional phase shift and possibly at a particular frequency ω_{critical}, that phase shift might accumulate to an additional $-\pi\text{Rad} = -180^o$. If this happens the loop gain at that frequency is now once more real and *positive* $A_{ol}(\omega_{\text{critical}})\beta > 0$ (i.e. $\angle A_{ol}(\omega_{\text{critical}})\beta = -360^o$). As we know from (5.48), if the loop gain is positive the loop can still be stable if, and only if, it is smaller than 1! Thankfully, phase shift in a low pass filter comes together with a roll off in gain. So if we are lucky the roll off ensures that $A_{ol}(\omega_{\text{critical}})\beta < 1$, but there is in fact a 'race' between roll off and phase shift.

So in negative feedback closed loops (!) we can be sure to be safe when

[3] For $A_{ol}\beta > 1$, $A_{cl} = \frac{A}{1-A\beta}$ is as a matter of fact also a fixed point, but it is a 'repulsive' fixed point, i.e. the circuit actually receives recursive positive feedback and will be forced away from that fixed point at even the smallest deviation and become unstable. So from a random starting point it would not be able to stabilize at all and instead be driven to ∞...

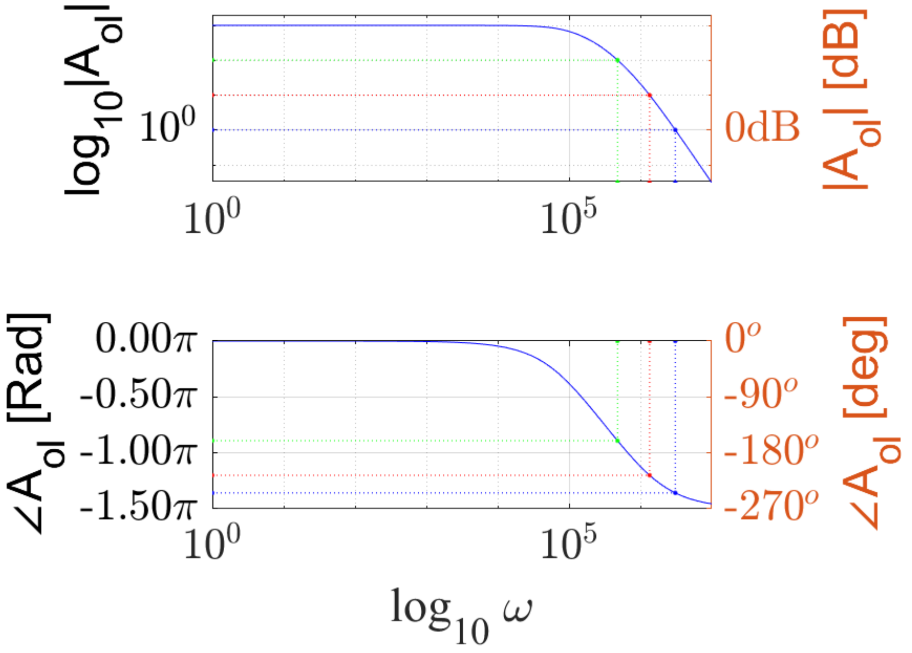


Figure 5.17: Illustrating varying the magnitude of $|\beta|$ to shift the unity gain frequency of the loop gain to lower frequency and thereby increase the **phase margin (PM)**. The transfer function used is a 3^{rd} order low pass filter without zeros, where $A_{ol} = 1000$, $\omega_{p_1} = 10^5$, $\omega_{p_2} = 10^5.5$, and $\omega_{p_3=10^6}$. The blue dotted line marks $\beta = -1$, the red $\beta = -\frac{1}{10}$ and the green $\beta = -\frac{1}{100}$. The green results in additional phase shift of -0.78πRad $= -139^o$ at the unity loop gain, i.e. as $A_{ol} = \frac{1}{\beta}$, leading to a positive phase margin of 41^o. The other phase shifts at unity loop gain exceed -180^o and do thus have negative PM and are guaranteed unstable.

1. A_{ol} is 2^{nd} or first order. In that case the added phase shift will always remain smaller than $-\pi = -180^o$, i e. the total phase shift will never quite reach the critical value $-2\pi = -360^o$. Note that there may still be *resonance* for 2nd order transfer functions (see the section 5.3.1 on the Q-factor) but that resonance will be limited.

2. if we make sure that the decrease in gain is sufficient and $|A_{ol}(\omega_{\text{critical}})\beta| < 1$. Also here there may still be limited resonance or even instability (i.e. unlimited or 'run away' resonance) if you have very high gain still close to ω_{critical}: This only avoids a guaranteed instability e.g the rule is required but not sufficient)! So usually one well advised to aim at $|A_{ol}(\omega_{\text{critical}})\beta| << 1$ by a good margin (see 'phase margin' (PM) later in this section).

The latter can be achieved by the choice of β: beta needs to be small enough to avoid this, i.e. the closed loop gain needs to be small enough at DC to give the roll off a head start over the phase shift. See figure 5.17 for an illustration:

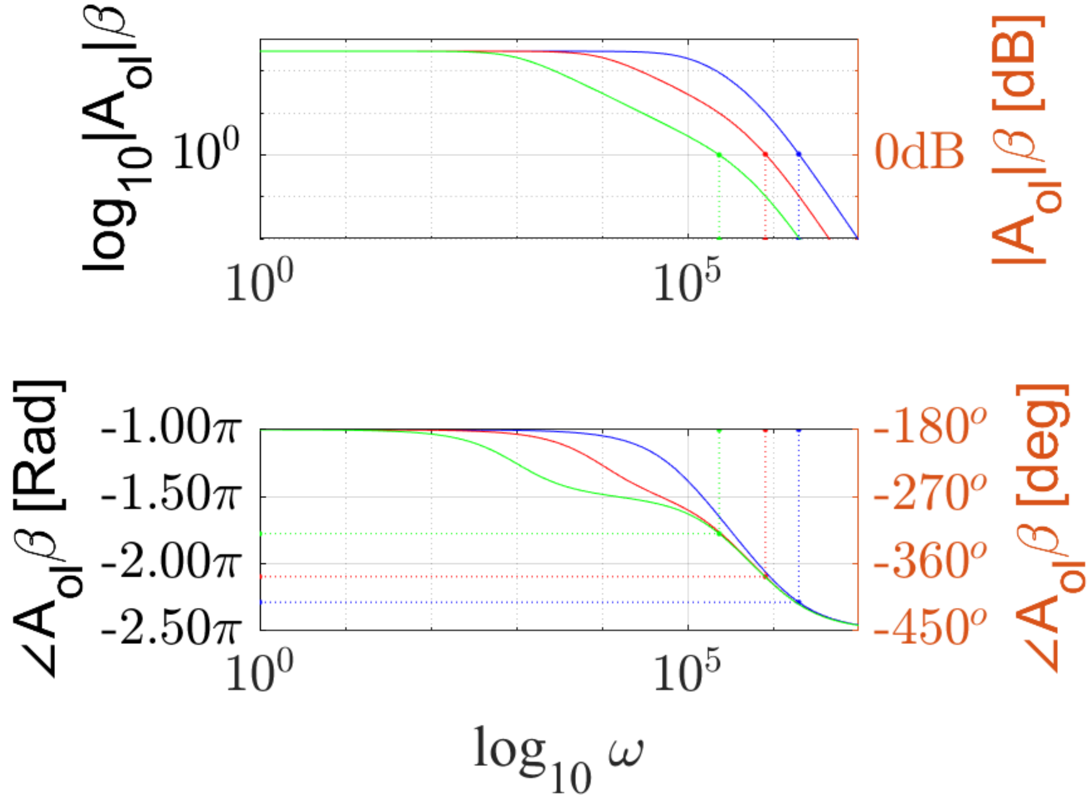


Figure 5.18: Illustrating pole separation to shift the unity gain frequency ω_T of the loop gain to lower frequencies increasing phase margin. The three bode plots are from a 3^{rd} order low pass loop (!) gain with no zeros and have all $A_{ol}(0)\beta = -300$, and $\omega_{p_2} = 10^{5.5}$ and $\omega_{p_3} = 10^6$. However, ω_{p_1} varies and is $\omega_{p_1} = 10^5$ for the blue, $\omega_{p_1} = 10^4$ for the red, and $\omega_{p_1} = 10^3$ for the green curve. So as the magnitude reaches 0dB, the phase shift is at $-2.29\pi\text{Rad} = -412^o$, $-2.10\pi\text{Rad} = 377^o$ and $-1.78\pi\text{Rad} = -319^o$. So only the green loop gain will lead to a stable feedback loop with a phase margin (PM) of $-319^o + 360^o = 41^o$

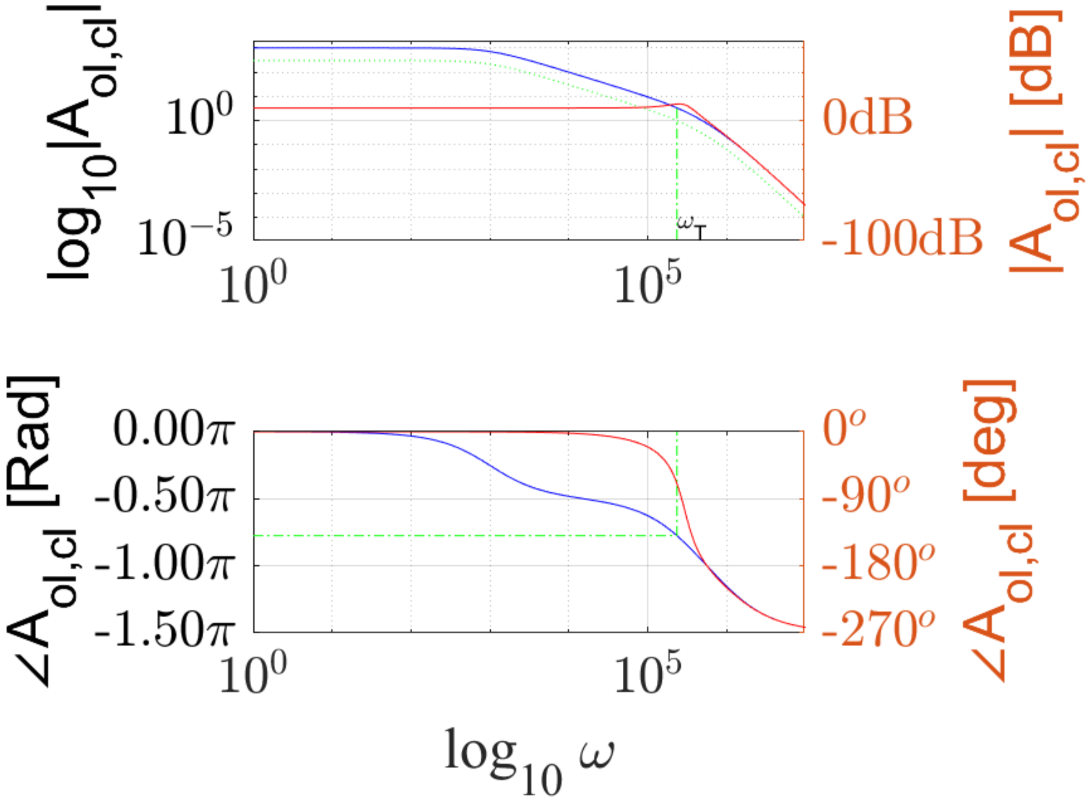


Figure 5.19: Illustrating the BW of a closed loop negative feedback system, for $A_{ol}(0) = 1000$, $\beta = -\frac{3}{10}$, $p_1 = 10^3$, $p_2 = 10^5.5$, and $p_3 = 10^6$. A_{ol} is in blue and A_{cl} in red. The dotted green line shows the loop gain and the dash-dotted line marks the unity gain of the loop gain and the additional phase shift at that frequency, which is -2.43Rad=-139^o. The phase margin here is thus 41^o.

the phase shift at the loop gain unity gain frequency ω_T is where the loop gain $A_{ol}\beta = 1$ and consequently where $A_{ol} = \frac{1}{\beta}$ (as marked in figure 5.17) and thus the point where the additional phase shift should not yet have reached -180^o (or the total phase shift should not have reached -360^o) for the feedback system to avoid guaranteed instability. However, usually one is not at liberty to choose β freely: it is after all the parameter that sets the close loop gain $A_{cl} \approx \frac{1}{\beta}$.

Another method to keep the additional phase shift above -180^o when the loop gain reaches $A_{ol}\beta = 1$ is **pole separation**, i.e. making sure A_{ol} has a clearly dominant pole, for instance by adding capacitance to the node corresponding to the lowest pole. This way one gives again a head start to the roll off, but this time starting it a t lower frequency instead of lowering its starting point. See figure 5.18 for an illustration of the effect on the loop gain if you lower the frequency of the first pole. If there is a clearly dominant pole, the roll off will start at that pole's frequency and the additional -180^o phase shift will only occur somewhere beyond the second pole, at which point the roll off after the dominant pole will have had much 'time' to reduce the magnitude, hopefully below the loop gain's **unity gain frequency** ω_T, i.e. the frequency at which the magnitude of the loop gain $|A_{ol}(\omega_T)\beta| = 1 \Leftrightarrow |A_{ol}| = |\frac{1}{\beta}|$ has shrunk to unity. Also this method has a trade off: shifting the first pole of the loop gain to a lower frequency may also affect the closed loop bandwidth.

The **phase margin (PM)** is a measure of how little resonance the feedback loop system is expected to express. It is the difference between the phase shift of the loop gain at the unity gain frequency and the critical total $= -360^o = -2\pi\text{Rad}$ phase shift.

$$\text{PM} = \angle A_{ol}(\omega_T)\beta - \angle A_{ol}(\omega_{\text{critical}})\beta = \angle A_{ol}(\omega_T)\beta + 2\pi \tag{5.51}$$

The more positive the PM, the more likely the system will be stable, and if the PM of a second order LP filter (also for higher order systems if all other poles are way beyond ω_T) is bigger than 65^o , it follows that $Q < \frac{1}{\sqrt{2}}$ (maximally flat response) and there will be no resonance. In fact there is a few more exact statements one can make, assuming that the A_{ol} is second order.

NOTE that the loop gain's $A_{ol}\beta$ bandwidth is the same as the open loop A_{ol} BW, but NOT THE SAME as the closed loop $A_{cl} = \frac{A_{ol}}{1-A_{ol}\beta}$ bandwidth: generally the cut off of A_{cl} is at the frequency where the open loop gain is reduced to $A_{ol} = \frac{1}{\beta}$. So the BW of A_{cl} will also be reduced when the BW of A_{ol} is reduced by pole separation but will still be larger. See figure 5.19 for an illustration. It shows A_{ol} in blue and A_{cl} in red, again for A_{ol} being a third order LP filter with no zeros where $A_{ol}(0) = 1000$, $\beta = -\frac{3}{10}$, $\omega_{p_1} = 10^3$, $\omega_{p_2} = 10^5.5$, and $\omega_{p_3} = 10^6$. So it corresponds to the green graph in figure 5.18. With a PM of 41^o it does experience some resonance. Both the magnitude and the phase plot approximate $\frac{1}{\beta}$ for very low frequencies and A_{ol} for very high frequencies, more precisely the separating frequency of the two is the frequency at which $A_{ol} = \frac{1}{\beta}$:

$$A_{cl}(\omega) \approx \begin{cases} \frac{1}{\beta} & \text{if } A_{ol} >> \frac{1}{\beta} \\ A_{ol} & \text{if } A_{ol} << \frac{1}{\beta} \end{cases} \tag{5.52}$$

Assuming 2^{nd} order open and closed loop gain

Assuming that A_{ol} is a second order LP filter with no resonance (i.e. $Q_{ol} < 0.5$ and the open loop poles are real) one can deduce some properties of the closed loop system. For instance, one can deduce the poles and Q_{cl} and ω_{0cl} from the poles of A_{ol}.

Let's say that

$$\begin{aligned} A_{ol}(s) &= A_{ol}(0)\frac{1}{(1+\frac{s}{\omega_{p_1}})(1+\frac{s}{\omega_{p_2}})} \\ &= A_{ol}(0)\frac{1}{1+\left(\frac{1}{\omega_{p_1}}+\frac{1}{\omega_{p_2}}\right)s+\frac{1}{\omega_{p_1}\omega_{p_2}}s^2} \end{aligned} \tag{5.53}$$

Then we solve for the A_{cl} poles. Let's define variables for nominator and denominator of the open loop gain $\frac{A_n}{A_d} := A_{ol}$. A bit of manipulation of equation (5.53) reveals that we only need to solve for the denominator to be zero. For stable cases we get from 5.48:

$$\begin{aligned} A_{cl} &= \frac{\frac{A_n}{A_d}}{1-\frac{A_n}{A_d}\beta} \\ &= \frac{A_n}{A_d\left(1-\frac{A_n}{A_d}\beta\right)} \\ &= \frac{A_n}{A_d - A_n\beta} \end{aligned} \tag{5.54}$$

solving for $0 = A_d - A_n\beta$ is the same as solving for $0 = 1 - \frac{A_n}{A_d}\beta$, consequently we solve:

$$0 = 1 + A_{ol}\beta \tag{5.55}$$

to get the closed loop poles

$$\begin{aligned} 0 &= 1 - A_{ol}(0)\frac{1}{1+\left(\frac{1}{\omega_{p_1}}+\frac{1}{\omega_{p_2}}\right)s+\frac{1}{\omega_{p_1}\omega_{p_2}}s^2}\beta \\ 0 &= 1 - A_{ol}(0)\beta + \left(\frac{1}{\omega_{p_1}}+\frac{1}{\omega_{p_2}}\right)s+\frac{1}{\omega_{p_1}\omega_{p_2}}s^2 \\ &\Rightarrow \\ p_{cl1}, p_{cl2} &= \frac{-\left(\omega_{p_1}+\omega_{p_2}\right) \pm \sqrt{\left(\omega_{p_1}+\omega_{p_2}\right)^2 - 4\omega_{p_1}\omega_{p_2}\left(1-A_{ol}(0)\beta\right)}}{2} \\ &\Rightarrow \\ Q_{cl} &= \frac{\sqrt{\omega_{p_1}\omega_{p_2}\left(1-A_{ol}(0)\beta\right)}}{\omega_{p_1}+\omega_{p_2}} \\ \omega_{0cl} &= \sqrt{\omega_{p_1}\omega_{p_2}\left(1-A_{ol}(0)\beta\right)} \end{aligned} \tag{5.56}$$

PM	35.09	40.65	50.35	54.18	60.75	64.02	70.23	75.54	79.55	85.00
$\frac{\omega_T}{\omega_{p_{ol2}}}$	1.48	1.20	0.85	0.74	0.58	0.50	0.37	0.27	0.20	0.10
Q	1.60	1.35	1.05	0.97	0.82	0.75	0.63	0.53	0.45	0.32

Table 5.2: For a second order A_{ol}: Relation between the phase margin PM, the ratio of the loop gain $A_{ol}\beta$ unity gain frequency ω_T and the open loop 2^{nd} pole frequency $\omega_{p_{ol2}}$ (For higher order A_{ol} on may employ the 2^{nd} order approximation as derived with (5.57), but if the higher order poles are close to $\omega_{p_{ol2}}$ the PM in this table will deviate significantly), and the closed loop Q-factor.

One can also establish a relation between Q_{cl}, PM, $\frac{\omega_T}{\omega_{p_{ol2}}}$ as shown in table 5.2

Approximating higher order with 1^{st} order or 2^{nd} order

So if one is interested in these precise statements that are possible for 2^{nd} order A_{ol} and A_{cl} and in general wants to define a Q for higher order functions, one can approximate higher order functions with 2^{nd} order or even just 1^{st} order. (The later makes sense in case of a very dominant pole.)

One can more generally approximate n^{th} order root form transfer function with an x^{th} ($x < n$) order root form transfer function by replacing the x^{th} pole frequency multiplier in the denominator $(1 + \frac{s}{\omega_{p_x}})$ and all higher frequency pole multipliers where $\omega_{p_i} \geq \omega_{p_x}$ and higher frequency zero multipliers $\omega_{z_i} \geq \omega_{z_y}$ in the nominator (where we denote ω_{z_y} as the smallest (!) of these zero frequencies with) with a single multiplier in the denominator $(1+\frac{s}{\omega_{\text{eq}}})$ where ω_{eq} is computed as:

$$\omega_{\text{eq}} \approx \frac{1}{\sqrt{\sum\limits_{\omega_{p_x} \geq \omega_{p_i}} \frac{1}{\omega_{p_i}^2} - \sum\limits_{\omega_{z_i} > \omega_{p_x}} \frac{2}{\omega_{z_i}^2}}} \tag{5.57}$$

To write this out:

$$A\frac{(1 + \frac{s}{\omega_{z_1}})...(1 + \frac{s}{\omega_{z_m}})}{(1 + \frac{s}{\omega_{p_1}})...(1 + \frac{s}{\omega_{p_n}})} \approx A\frac{(1 + \frac{s}{\omega_{z_1}})...(1 + \frac{s}{\omega_{z_{y-1}}})}{(1 + \frac{s}{\omega_{p_1}})...(1 + \frac{s}{\omega_{p_{x-1}}})(1 + \frac{s}{\omega_{p_{\text{eq}}}})} \tag{5.58}$$

This formula is obtained by assuming that the pole- and zero frequencies larger than ω_{p_x} are in general larger than 1, as well as only considering the poles and zeros at higher frequencies than ω_{p_x} and solving for the frequency where the magnitude of that expression H being $|H| = \frac{1}{\sqrt{2}}$. For example when approximating a second order expression with a first order, i.e. approximating the cut off frequency ω_H with the frequency ω_{eq} where the magnitude is reduced to $|H| = \frac{1}{\sqrt{2}}$:

$$|H(\omega_{eq})|^2 = \frac{1}{2} \quad = \quad \frac{(1+\frac{\omega_H^2}{\omega_{z1}^2})(1+\frac{\omega_H^2}{\omega_{z2}^2})}{(1+\frac{\omega_H^2}{\omega_{p1}^2})(1+\frac{\omega_H^2}{\omega_{p2}^2})} \tag{5.59}$$

$$= \quad \frac{1+\omega_H^2\left(\frac{1}{\omega_{z1}^2}+\frac{1}{\omega_{z2}^2}\right)+\omega_H^4\left(\frac{1}{\omega_{z1}^2\omega_{z2}^2}\right)}{1+\omega_H^2\left(\frac{1}{\omega_{p1}^2}+\frac{1}{\omega_{p2}^2}\right)+\omega_H^4\left(\frac{1}{\omega_{p1}^2\omega_{p2}^2}\right)} \tag{5.60}$$

$$\Rightarrow \omega_H \quad \approx \quad \omega_{eq}\frac{1}{\sqrt{\frac{1}{\omega_{p1}^2}+\frac{1}{\omega_{p2}^2}-\frac{2}{\omega_{z1}^2}-\frac{2}{\omega_{z2}^2}}} \tag{5.61}$$

(5.61) ignores the higher than second order terms ω_H^4 of (5.60), which yields a good approximation if the if the higher order poles and zeros are of much higher frequencies than the first order ones.

However, this method depends on us knowing all poles and zeros and is thus of limited use when working with an unsolved nth order polynomial. Then again, one can do the slightly more crude approach to ignore higher order terms outright in the polynomial form, if they are much smaller than their predecessors, indicating that the higher order poles and zeros occur at much higher frequencies, and hopefully beyond the unity gain so that they will not have an important influence.

And maybe even better: one can use simulation tools to derive a Bode plot of a full circuit and determine the PM that way.

Part III

Digital Circuits Basics

Chapter 6

Boolean Logic

The basis of digital electronics is Boolean logic or algebra. Boolean logic describes binary valued functions operating on inputs and outputs with the values 'true' and 'false' or '0' and '1'. The fundamental operands these functions are comprised of are 'AND', 'OR', and 'NOT'. Commonly used symbols for these functions are shown in table 6.1. Their function can be expressed by **truth tables**, see table 6.2. A complete truth table for a given function lists all possible inputs together with the corresponding output.

One can combine these basic functions into longer expressions to derive more complex functions.

Just like other algebras, Boolean logic is governed by a set of rules for evaluation and manipulation of expressions. These are summarized in table 6.3.

	unambiguous	ASCII compatible
AND	$a \wedge b$	$a * b$
OR	$a \vee b$	$a + b$
NOT	$\neg a$ or $\overline{a}$	$\sim a$

Table 6.1: Symbols used for Boolean operands. a and b are Boolean variables.

a	$\bar{a}$
0	1
1	0

a	b	$a\wedge b$
0	0	0
0	1	0
1	0	0
1	1	1

a	b	$a\vee b$
0	0	0
0	1	1
1	0	1
1	1	1

Table 6.2: Truth tables for the basic Boolean functions

$a\wedge b\vee c = (a\wedge b)\vee c$	$a\vee b\wedge c = a\vee(b\wedge c)$	(priority)
$a\wedge b = b\wedge a$	$a\vee b = b\vee a$	(commutativity)
$(a\wedge b)\wedge c=a\wedge(b\wedge c)$	$(a\vee b)\vee c=a\vee(b\vee c)$	(associativity)
$\bar{\bar{a}}=a$		(involution)
$a\wedge\bar{a}=0$	$a\vee\bar{a}=1$	(completness)
$a\wedge a=a$	$a\vee a=a$	(idempotency)
$a\wedge 1=a$	$a\vee 0=a$	(boundedness)
$a\wedge 0=0$	$a\vee 1=1$	(boundedness)
$a\wedge(a\vee b)=a$	$a\vee(a\wedge b)=a$	(absorbtion)
$a\wedge(b\vee c)=(a\wedge b)\vee(a\wedge c)$	$a\vee(b\wedge c)=(a\vee b)\wedge(a\vee c)$	(distributivity)
$\overline{a\vee b}=\bar{a}\wedge\bar{b}$	$\overline{a\wedge b}=\bar{a}\vee\bar{b}$	(deMorgan)

Table 6.3: Rules for Boolean function evaluation and manipulation.

Chapter 7

Combinational Logic Circuits

The digital transistor model of a voltage controlled switch can be used to implement Boolean functions. Basic functions in digital electronics wire diagrams can be represented as **logic gates**. The symbols for the basic logic functions are shown in figure 7.1. The symbols for XOR and XNOR have also been added. Boolean variables are usually represented by voltages on electrical nodes[1] and distinguished as binary either 'true' and 'false', or respectively '1' and '0', if they are 'high' or 'low' with respect to given thresholds. Usually 'high' signals are defined as close to Vdd and above a 'high' threshold, while 'low' voltages are close to Gnd and below the 'low' threshold. The two thresholds are distinct from each other leaving a region in the middle between Vdd and Gnd, that are uncertain states. When functioning properly and given enough time to settle, digital voltage signals should not end up in this no-man's land, but when, for instance, signals are in transition or when different logic circuits with different definitions of 'high' and 'low' are improperly connected such mishaps can occur and the abstraction of the digital circuit as Boolean logic may break down. Thankfully, the fact that a whole signal range is associated with 'high' and 'low' provides usually a good error margin, and a digital gate does 'auto correct' inputs that stray from the ideal logical levels Vdd and Gnd, back to more ideal digital signals. Thus, not a single one of a CPU's billions of logic signals is misinterpreted under normal operating conditions.

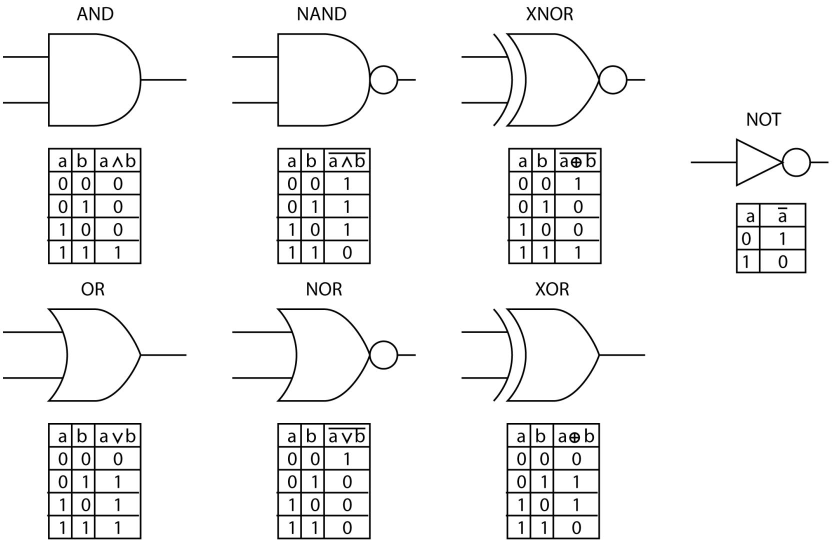
So how can we make a logic gate that implement a Boolean function? There are a few variants. Lets start with the most common one: **complementary pull-up and pull down networks**. The basic idea of a pull-down network is a network of switches/transistors between Gnd and the output node that are controlled by digital input signals at their control input/gate terminal. They are arranged in such a way to open a conductive path from the output to Gnd if and only if the state of the input signals form the appropriate pattern for a Boolean function to be 'false'. Examples are shown in figure 7.2. Vice versa, a pull-up network forms a path only if the inputs are in a state that should make the output go 'high'.

there is a few things we should note about those PD and PU networks.

Observations:

- nFETs are efficient as switches in PD networks, and pFETs in PU net-

[1] Representation by currents is also possible but less popular.



a	b	$a\wedge b$
0	0	0
0	1	0
1	0	0
1	1	1

a	b	$\overline{a\wedge b}$
0	0	1
0	1	1
1	0	1
1	1	0

a	b	$\overline{a\oplus b}$
0	0	1
0	1	0
1	0	0
1	1	1

a	$\bar{a}$
0	1
1	0

a	b	$a\vee b$
0	0	0
0	1	1
1	0	1
1	1	1

a	b	$\overline{a\vee b}$
0	0	1
0	1	0
1	0	0
1	1	0

a	b	$a\oplus b$
0	0	0
0	1	1
1	0	1
1	1	0

Figure 7.1: Equivalent Boolean operators, truth tables and logic gates

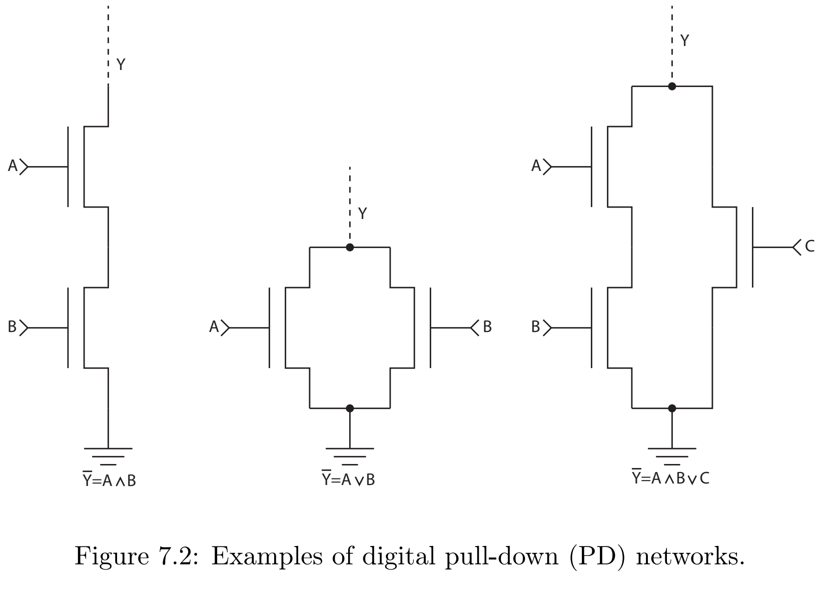


Figure 7.2: Examples of digital pull-down (PD) networks.

works.
This is because it is best to have the gate to source voltage large when a the output undergoes a transition. So for PD nFET the source is normally constant at Gnd [2] and the gate to source voltage difference V_{GS} is Vdd. If a nFET is used in a PU network, its source will be pulled up in a transition, heavily reducing I_{ON} or increasing R_{ON} respectively during the transition. In fact the previous derivations of I_{ON} (4.1) and R_{ON} (4.13) were made under the assumption that the source voltage is at Gnd for nFETs and at Vdd for pFETs.

- Complex Boolean functions that would result in large PD/PU networks are often better implemented with a series of logic gates, rather than a single gate with large PD/PU networks.
  As we will see shortly, large networks that require transistors in series will diminish the propagation delay of a gate AND reduce the quality of the output signal, such that an equivalent implementation with multiple gates in series can oftentimes achieve an overall better performance.

- nFET PD networks can only implement particular Boolean functions, i.e. those that have an equivalent expression where the output being 'low' is a function of non-inverted input variables, e.g. $\overline{Y} = A \vee B \wedge C$. Its the other way round for PU networks implemented with pFETs, i.e. the output being 'high' as a function of all inverted input variables, like $Y = \overline{A} \wedge (\overline{B} \vee \overline{C})$. Note in particular, that an AND or an OR gate does *not* fit that requirement! Thus, it is the NAND and the NOR gate that can directly be implemented as a single stage CMOS logic gate, while AND and OR are composed of two stages, e.g. a NAND followed by an inverter or a NOR followed by an inverter.

One can complement a pull-down network simply with a *pull-up resistor* towards Vdd (figure 7.3 A)). That resistor maintains thus the default of the output being low unless the inputs make the pull down network conductive. This is a possible implementation with some drawbacks. For one, while the output is being pulled low, this circuit consumes a constant current. Furthermore, the resistor needs to be dimensioned correctly such that the PD network in fact is able to pull the output voltage down. Also, actual resistors are costly in layout space in integrated circuits, much more so than transistors.

The latter can be amended by replacing the resistor with a pull up *transistor*, i.e. a pFET transistor with an appropriate constant gate voltage and of appropriate dimensions (figure 7.3 B)). For example on can choose a pFET with its gate voltage at Gnd, which then needs to have a quite small $\frac{W}{L}$ ratio to make its static R_{ON} big enough for the active PD network to be able to pull the output low. So this solves the layout space problem, but not the constant current problem for maintaining a 'low' state at the output.

Thus, the normal solution here is to use *both* a PU *and* a complementary PD network that implement the *same* logic function, i.e. when the PD network is active the PU network is inactive and vice versa (figure 7.3 C)). This is the most

[2] When multiple nFETs are in series this might not be true for nFETs that are not directly connected to the Gnd terminal at the start of the transition. However, if so their source terminal will rapidly be depleted of positive charge and be pulled to Gnd efficiently by the nFET below them.

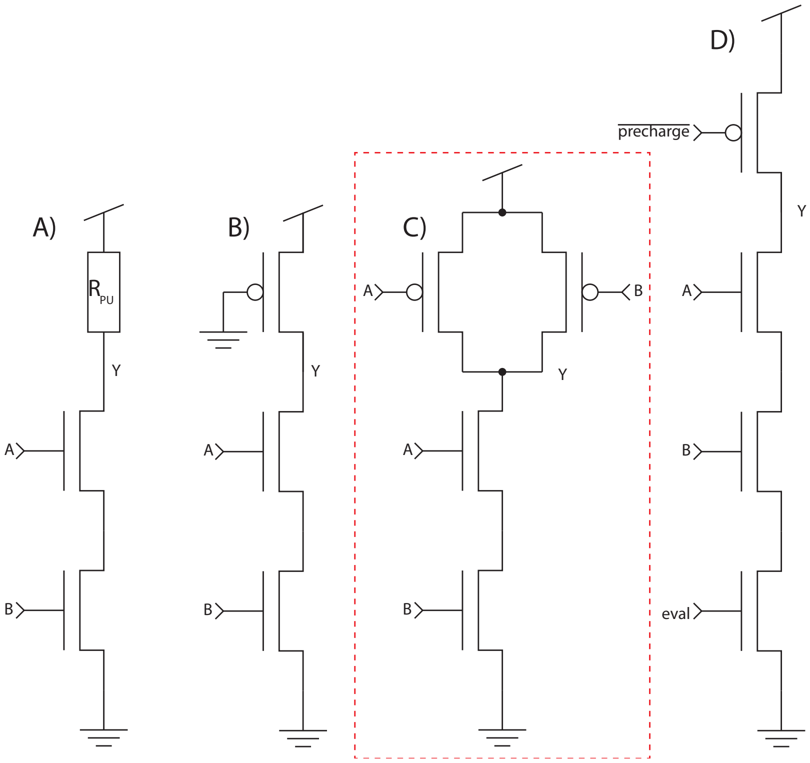


Figure 7.3: Ways of implementing a NAND gate with a nFET pull-down network. The most widely applied is variant C) the **complementary pull-up/pull-down network** version in the red dashed box. A) and B) use a static pull-up, with respectively a resistor and an open pFET. They require that the pull up strength is weak enough for the pull-down network to be able to overcome. D) is a **dynamic logic** implementation. '$\overline{\text{precharge}}$' and 'eval' may simply be the same shorted clock signal, or non-overlapping clock signals.

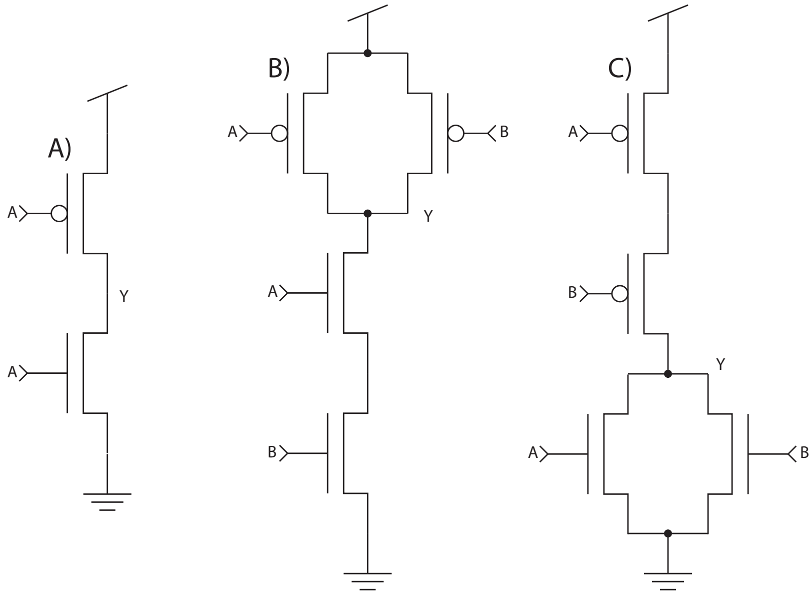


Figure 7.4: Boolean operands that are most easily implemented as logic circuits with complementary PU/PD networks, here with complementary PU/PD networks: A) NOT or inverter, B) 2-port NAND, C) 2-port NOR

prominent solution used in static logic circuits. It does not consume any current to maintain a state (other than leakage!), and is only using small transistors, i.e. it is not excessive in layout space. Figure 7.4 shows the basic Boolean operands' implementation as static logic gates with complementary PU/PD networks.

Figure 7.3 C) shows a **dynamic logic** inverter. We will get back to that variant in section 7.4.

7.1 Balancing PU and PD networks

Let us have a look at the propagation delays of complementary PU/PD static logic. In section 4.2 we have had a first look at the propagation delay t_p when a single transistor pulls a digital signal up from Gnd to $\frac{\text{Vdd}}{2}$ or down from Vdd to $\frac{\text{Vdd}}{2}$. Obviously, logic gates need to do both, so let's distinguish the two by writing t_{pHL} and t_{pLH} for the delay in pulling down and up respectively, and lets us redefine t_p as the *average* of the two:

$$t_p = \frac{t_{pHL} + t_{pLH}}{2} \tag{7.1}$$

We shall use the resistive model of a switched transistor derived in subsection 4.2.2 and equation (4.13) to gauge propagation delays. They are thus proportional to $R_{ON}C_L$ and consequently inversely proportional to parameter $k = \mu C_{OX}\frac{W}{L}$, as well as proportional to the total load capacitance C_L of the output node. Note here that the carrier mobility μ is usually 2 to 3 times bigger for nFETs than for pFETs.

C_L at the output of a gate can be dissected into three main contributors: the capacitance at the output of that gate alone C_{OUT}, the input capacitance C_{IN} of anything it might be connected to, e.g. typically other logic gates, and parasitic capacitances summarized as C_W caused by interconnect of the two, in particular if they are connected by a long cable.

$$C_L = C_{OUT} + C_W + C_{IN} \tag{7.2}$$

7.1.1 Inverter

let's first consider the simplest case of an inverter connected to another inverter. Then the main components of C_{OUT} are:

$$C_{OUT} = 3C_{GDp} + 3C_{GDn} + C_{DBp} + C_{DBn} \tag{7.3}$$

The curious factor 3 for the two C_{GD} is explained by a Miller effect, i.e. the feed forward charge injection one gets from these capacitors with a perfect step input. See the next subsection 7.1.2.
and C_{IN} is:

$$C_{IN} = C_{GSp} + C_{GSn} + C_{GDp} + C_{GDn} \tag{7.4}$$

Be reminded here that all of these capacitors do scale with the width W of the transistors, and C_{GS} does *also* scale with the length L. (See equations (2.25), (2.26), (2.28) and (2.27)).

Considering the respective $\mu_n C_{OX}$ and $\mu_p C_{OX}$ as given for now, one can contemplate to keep $L_{n/p}$ at their minimum[3] and to increase $W_{n/p}$ to reduce R_{ON}, but this will proportionally increase both C_{OUT} and C_{IN} as well. It will *still* reduce t_p though, as it will diminish the relative contribution of C_W. However, this is only worthwhile up to a point where the influence of C_W becomes insignificant. Actually mostly W is kept close to a minimum too, except for logic gates that need to drive long cables or chip pads that are used as connections off-chip. This is because layout space is also a costly commodity and generally it is desirable to cram as many logical gates into an as small as possible space.

Let us first consider the simplest of all digital gates, the inverter, consisting of just one nFET and one pFET (see figure 7.4). R_{ON} is thus derived for a single transistor in each direction (i.e. PD and PU) according to equation (4.13). Considering interconnect capacitance C_W to be negligible, using minimum size transistors will result in a minimal C_L. Further assuming that the minimum transistor dimensions are the same for nFET and pFET (which is usually the case) we get asymmetric propagation delays if $\mu_n > \mu_p$ (which is also usually the case), i.e. $t_{pLH} < t_{pHL}$. This has a number of undesirable side effects:

- The real switching threshold will be skewed: $V_{sw} < \frac{\mathrm{Vdd}}{2}$. Thus, the error margin of a 'low' input signal to be incorrectly identified as 'high' is reduced.

- The worst case propagation delay is much worse than the average t_p. Thus, the clock cycle needs to be slowed down to be able to accommodate that

[3] Note that the name for a given technology actually is based on this minimum transistor length, e.g. Intel 7nm CMOS technology or the like.

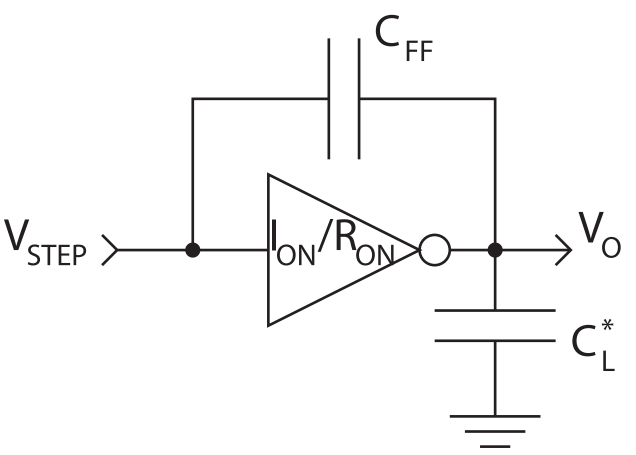


Figure 7.5: Illustration of the Miller effect affecting the output node capacitance of an inverter. Here, the inverter symbol itself does only represent the ideal switches initiating the pull up or pull down via either the current source- or resistor model. C_{FF} represents the capacitors C_{GD} between input and output.

worst case, i.e. slow enough that one can be sure that the logic in question can have a stable output within the clock cycle.

Thus, it is desirable to balance the t_{pLH} and t_{pHL}. This is done by making pFETs wider than nFETs such that $\mu_n C_{OX}\frac{W_n}{L_n} \approx \mu_p C_{OX}\frac{W_p}{L_p}$. So if $\mu_n = x\mu_p$ (where usually $2 \lesssim x \lesssim 3$) we choose for the inverter $W_p = xW_n$. Let us denote the transistor dimensions for a *balanced* inverter as $p = \frac{W_p}{L_p} = \frac{xW_n}{L_p}$ and $n = \frac{W_n}{L_n}$ (Fig. 7.6 A))

7.1.2 Inverter: Miller Approximation of C_{OUT} for Step Input

In digital electronics the feed forward effect of the 'Miller capacitances' between input and inverted output node(s) (e.g. C_{GD} for inverters) is taken into account for approximating capacitance at the output node (in contrast to analog where one is more interested in the Miller effect on the *input* of a transconductance amplifier and its BW), where one assumes a perfect step at the input. For the inverter for instance (see figure 7.5), if we assume the input step to be from Gnd to Vdd, and are interested in the time t_p it takes for the output to go from Vdd to $\frac{\text{Vdd}}{2}$. Then the two C_{GD} become important as a feed forward capacitance, so let's now call their sum C_{FF}. It will inject a charge $Q_{FF} = C_{FF} * \text{Vdd}$. Let's call the total capacitance at the output other than C_{FF} is C_L^*. Then the total charge that needs to be drawn from the output node to move from Vdd to $\frac{\text{Vdd}}{2}$ is then:

$$Q_{\text{Vdd}\rightarrow\frac{\text{Vdd}}{2}} = \frac{\text{Vdd}}{2}\left(C_L^* + C_{FF}\right) + C_{FF} * \text{Vdd} = \frac{\text{Vdd}}{2}\left(C_L^* + 3C_{FF}\right) \qquad (7.5)$$

If that charge is supplied by a current source I_{ON} then the time to reach $\frac{\text{Vdd}}{2}$ (i.e. the propagation delay t_p) is:

$$t_p = \frac{Q_{\text{Vdd}\rightarrow\frac{\text{Vdd}}{2}}}{I_{ON}} = \frac{\text{Vdd}}{2}\frac{C_L^* + 3C_{FF}}{I_{ON}} \tag{7.6}$$

So it's perfectly equivalent to increasing the effect of C_{FF} threefold.

Oftentimes, the same assumption is also made when one uses the model of R_{ON} instead of I_{ON}, resulting in:

$$t_p = \ln(2)R_{ON}(C_L^* + 3C_{FF}) = 0.69R_{ON}(C_L^* + 3C_{FF}) \tag{7.7}$$

even thought this is not correct: the real time constant in that case is still just $R_L^*(C_L^* + C_{FF})$ but the 'starting point' V_{start} is now higher than Vdd since the step charge injection raises the voltage to V_{start}.

$$V_{\text{start}} = \text{Vdd}\left(1 + \frac{C_{FF}}{C_{FF} + C_L^*}\right) \tag{7.8}$$

to get t_p on needs to solve

$$\begin{aligned}
e^{-\frac{t_p}{R_L^*(C_L^*+C_{FF})}} &= \frac{\frac{\text{Vdd}}{2}}{V_{\text{start}}} && (7.9)\\
&= \frac{1}{2\left(1 + \frac{C_{FF}}{C_{FF}+C_L^*}\right)} && (7.10)\\
&= \frac{1}{2}\frac{C_{FF} + C_L^*}{2C_{FF} + C_L^*} && (7.11)\\
t_p &= R_L^*(C_L^* + C_{FF})\left(\ln 2 + \ln\frac{2C_{FF} + C_L^*}{C_{FF} + C_L^*}\right) && (7.12)
\end{aligned}$$

Be that as it may: no-one uses that, but one assumes simply the contribution of $3C_{FF}$ (actually many books use $2C_{FF}$) instead into the already heavily simplified model of R_{ON}, i.e. one uses (7.7). Qualitatively the effect is the same, i.e. that the propagation delay t_p is made slightly longer.

7.1.3 More complex gates

More complex logic gates may have multiple paths by which the output may be pulled up or pulled down. Figure 7.6 shows some examples. Using the resistive model R_{ON}, the total resistance is derived from computing the total R_{ON} of the active PU or PD network. Note that there might be more than just one case to consider, as there might be multiple possible PU or PD paths dependent on how all the inputs switch. In that case it's the worst case scenario that is most important. It may not be possible to balance the transition of the output for *every* possible transition combination of switching inputs. Instead, we shall aim at balancing the *worst case* transition between the PU and PD network. In the figure 7.6 each transistor is marked with a factor xp or xn indicating its respective $\frac{W}{L}$ ratio, p and n being the $\frac{W}{L}$ of a balanced inverter. R_{ON} for each transistor is proportional to $\frac{L}{W}$. If one needs to make sure that the sum of all transistors' R_{ON} in series along each possible path is equal to a single transistor

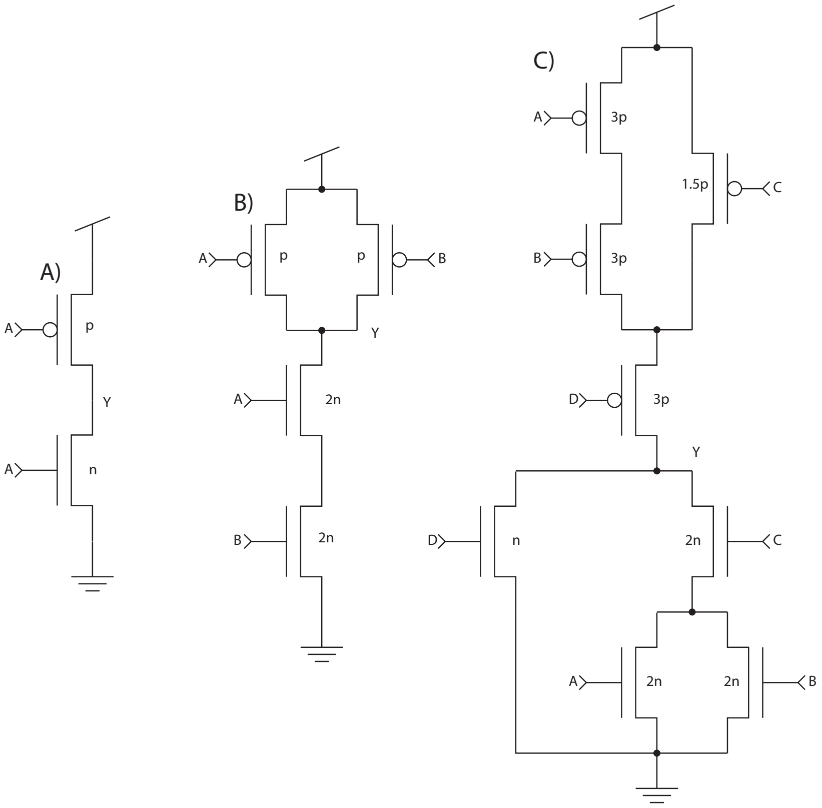


Figure 7.6: Illustrating balancing logic gates for symmetric worst case t_{pHL} and t_{pLH}. n and p indicate the balanced inverter's respective transistor width to length ratios resulting in equivalent R_{ON} for both PU and PD transistors. The ratio $\frac{W}{L}$ needs to be increased when multiple transistors are in series if one wants to maintain the same total R_{ON} for more complex gates. The two examples are a NAND ($\overline{Y} = A \wedge B$, subfigure B)), and the function $\overline{Y} = D \vee (C \wedge (A \vee B))$ (subfigure C)). Note that while the worst case R_{ON} is the same in all three instances, still C_{OUT} at the output node Y is increased for B) and C) due to the wider and more numerous transistors connected to it. Furthermore there are now 'internal' nodes that also may need to be charged during a transition! So B) and C) *are* balanced but still have longer propagation delay t_p.

of a balanced inverter, then the total worst case resistance is equal to that of the balanced inverter. NOTE: this does *only* take care of balancing R_{ON} and making it no bigger than in the case of the inverter. However, C_{OUT} will still increase (and there will be some 'internal' nodes in the gate with node capacitances that may also need charging/discharging), thus the delay proportional to $R_{ON}C_L$ will still increased relative to an inverter!

Observations:

- 'NAND is better than NOR for layout space', that is to say, it's preferable to have the longest serial branch of a gate in the PD network rather than the PU network. That is because $p > n$, so a relative increase in the transistor width of a pFET consumes more space than the same relative increase of an nFET's dimensions.

- Single stage gates with large PU/PD networks are often worse for propagation delay than an equivalent multi-gate implementation. Comparing the propagation delay of figures 7.6 B) and C) with just the *RC* model is quite inaccurate, even if we would extend it to include the internal nodes' capacitances. Just the capacitance at the output C_{OUT} is 1.5 times bigger for C) than for B), so that would only make it 1.5 time slower, but the internal capacitances contain also gate to source capacitances that are way bigger than just the drain to source capacitances of the inverter. So at that point the use of a simulation tool is recommended to gauge propagation delay more accurately and to compare a multi stage implementation of C) to the single stage shown here.

7.2 Dynamic and static power consumption

With today's enormous transistor density in CMOS technology, power consumption and heat dissipation are major challenges. Even if each logic gate dissipates only a weeny tiny bit of power, if you amass a several billion of them on just a few square centimetres, they will turn consume a lot of energy and turn it into concentrated heat. Cooling of CPUs has become a real challenge. The power consumption of digital circuits is often divided into two categories: **dynamic power consumption** that results from the gates actively computing and thus from the energy consumed when digital signals are switching states, and **static power consumption** that is consumed irrespective of activity, i.e. also when the circuit is idle. For an inverter, the major contributors for dynamic switching power are a) the charge needed to affect a state change of the output and is proportional to VddC_{OUT}, and b) a so called **cross current** flowing directly from the inverter's supply/Vdd to Gnd that occurs in the transition of the input signal. b) occurs while the (not infinitely fast) input is somewhere between Vdd and Gnd and is roughly worst while at $\frac{\text{Vdd}}{2}$. That worst current is proportional to Vdd^2 when it's in strong inversion and to $e^{\frac{\text{Vdd}}{U_T}}$ when in weak inversion.

7.3 Driving large external loads C_W

Microscopic integrated circuits do sometimes need to drive some macroscopic loads, typically when driving bus lines or most severely when connected to a pad

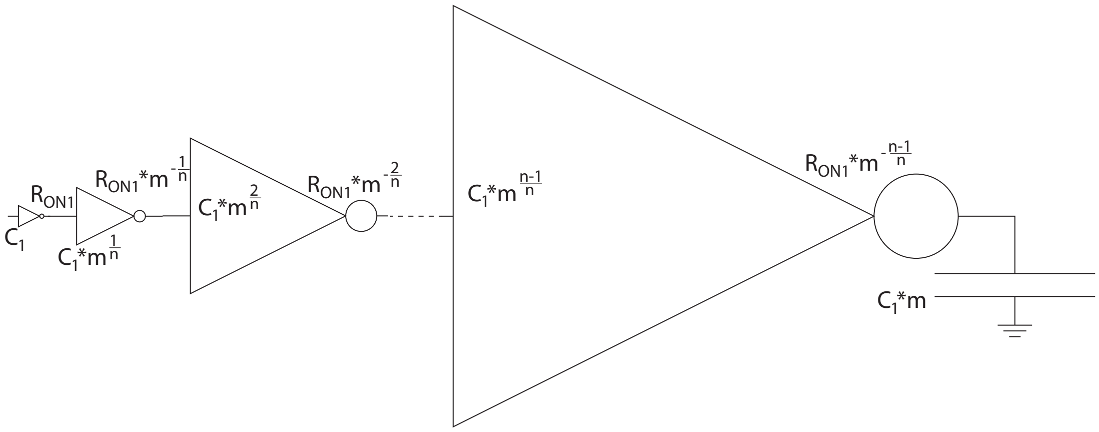


Figure 7.7: Illustrating a cascade of ever bigger inverters for driving a large capacitive load, e.g. when connecting to an off-chip element. Ideally, each stage is e times bigger than the previous for minimal propagation delay and there should be $\ln m$ stages. However, m is often times unknown, so practically 2 to 4 inverters with a size increase of 2-4 per stage are often used and will pragmatically improve the propagation delay, if not perfectly optimize it.

on the chip that in turn is connected via a micro-wire to a conductor on a PCB. In that case one should employ a gigantic inverter (i.e. with large $\frac{W}{L}$ transistors) as a buffer between the internal microscopic world of microscopic capacitances and the macroscopic world of macroscopic capacitances of the PCB. However, this is just dumping the problem onto the digital gate just *before* that inverter that now has to drive an unusually big input capacitance of this humongous inverter.

The best solution to this is actually a cascade of inverters with ever bigger $\frac{W}{L}$ ratio (figure 7.7). If one considers the load of each of those inverters to be dominated by the input capacitance of the next stage, and if an inverter connected to a same size inverter has a gate delay t_{p_1}, then an inverter connected to an x times larger inverter has a propagation delay of

$$t_{p_x} = x t_{p_1} \tag{7.13}$$

So if $C_{W_{ex}}$ is the external load and it is m times bigger than C_1 which is the input capacitance of a 'normal' sized inverter as used in on-chip logic, then the total delay of such an inverter driving $C_{W_{ex}}$ directly is:

$$t_{p_{tot}} = m t_{p_1} \tag{7.14}$$

If one inserts an extra buffer inverter of a size $\sqrt{m}$ times bigger than a 'normal' inverter, that consequently also has a R_{ON} that is $\frac{1}{\sqrt{m}}$ that of a normal sized inverter R_{ON_1} and an input capacitance that is $\sqrt{m}C_1$. So the total delay is now:

$$t_{p_{tot}} = \sqrt{m}t_{p_1} + \sqrt{m}t_{p_1} = 2\sqrt{m}t_{p_1} \tag{7.15}$$

And more generally for $n-1$ inverters beyond the first with the same ratio of increased size between each consecutive stage on gets:

$$t_{p_{tot}} = nm^{\frac{1}{n}} t_{p_1} \tag{7.16}$$

And it turns out (i.e. it can be shown mathematically, but not here) that the optimal factor of increase between the stages $m^{\frac{1}{n}}$ resulting in the minimal delay is e:

$$m^{\frac{1}{n_{opt}}} = e = 2.7183 \tag{7.17}$$

So the optimal number of stages n_{opt} is:

$$n_{opt} = \ln m \tag{7.18}$$

Practically, one uses an increase between stages of 2 to 4 for good results, but since m is often unknown or might vary, as the ASIC is connected to different loads when used, it's not clear a priory how many stages n that are needed. Anyway, any factor $m^{\frac{1}{n}}$, even when > 4 and any number of stages are better than none, in practice.

7.4 Dynamic logic

Dynamic logic circuits (figure 7.3 D)) are not widely used and are finicky, but have their place where ultimate speed is required.

They implement only either a switched PU or PD network, just like variants A) and B) in figure 7.3, but as opposed to A9 And B) that use a static PD/PU resistor or transistor on the opposing side they use switches to toggle between the logic PU/PD network and the opposing PD/PU 'reset', i.e. between a reset/**precharge phase** where the output is set to a default state, and an **evaluation phase** where the logic function is evaluated and determines whether the default state is maintained or switched. The term *dynamic* results from the fact that the 'default' state resulting from the precharge phase is not statically/actively maintained during the evaluation phase if the PD/PU logic does not switch it. Instead, it is only passively stored as a charge on the output node capacitance. Consequently, leakage currents can destroy that state over time if it is not refreshed and/or 'used' quickly enough. Also all kinds of noise can affect it, e.g. coupling noise from nearby switching signals. Furthermore, if the logic PD/PU networks is not completely turning conductive but some individual transistors *are* switched to conductive, charge from internal nodes between transistors may be dumped on the output as well. So in conclusion: if you do not *have* to use dynamic logic, it's better to avoid it.

Figure 7.3 D) shows an implementation of a NAND gate. Note that it is advantageous to implement the switched PD network rather than the PU network, since pFETs are usually chosen to be wider and thus need more space. Here, only a single pFET is needed instead of the two for a static logic NAND gate with complementary PU and PD, but an extra nFET is needed to switch the PD network. So again a total of four transistors, but now more nFETs which oftentimes are chosen smaller.

Pros and Cons:

+ Fewer transistors for gates more complex than a NAND or NOR, and fewer pFETs already for NAND and NOR with switched PD network. (good for layout density)

+ With non-overlapping precharge and evaluation phases there is no cross current (good for power)

+ Less parasitics contributing to C_{OUT} and typically only one C_{GS} contributing to C_{IN} instead of two (good for speed and power)

- Always switching even when evaluation state stays the same (bad for power)
- Sensitive to noise and timing due to 'passive' maintaining of state (bad for design effort)
- Can consequently generate errors when run too slowly or due to noise (e.g. cross talk) in evaluation phase

Let us look at an example in figure 7.8. Note that for simplicity sake, we use a single CLK and $\overline{\text{CLK}}$ signal instead of eval and $\overline{\text{precharge}}$. A simple inverted CLK signal will work as well, but remember that to avoid cross currents one needs to make the active phases of CLK and $\overline{\text{CLK}}$ non overlapping.

The logic gate representation of the multiplexer is shown in the upper right. The function of a multiplexer is to route one of multiple digital signals to an output. Here the multiplexer chooses between signals A and B according to signal S. If S is high, O will be the same as A. If S is low, O will be the same as B. The circuit is implemented with three NAND gates. There is a potential problem when using the same CLK/eval signal for the consecutive stages: If the evaluation phase starts simultaneously for the second stage, it may be that the first stage outputs are still in their precharge state and only beginning to be pulled low if their evaluation state is supposed to be low. Thus the PD network in the second stage is open and may start to pull the output low by mistake. To avoid this, the evaluation phase of the second stage *needs* to be delayed and for any consecutive stages the CLK/eval signal needs to be **staggered** this way!!! One can, for example, run the CLK signal through a double (static) inverter before it is applied to the next stage and one needs to make very sure that this delay is sufficient: inverters may be faster than the NANDs here, so the double inverter adds some safety margin. But note that there always is mismatch/variability between CMOS elements after production and with billions of such gates a very unlucky outcome for one of them can never be ruled out entirely.

Thus, there is a dynamic logic variant that avoids this particular problem and the need for clock staggering entirely, called **domino logic**: it makes sure that the default precharge state seen by the next stage is *low* if that stage is employing a PD logic network in the evaluation phase. This can be achieved in either of two ways. The first is to still employ only stages with PD networks but to interpose a static inverter between stages. The multiplexer circuit can thus be implemented like shown in figure 7.9. Now the default precharge signals seen by the second stage are low. Consequently the PD network cannot start pulling low prematurely. Only if the first stage has completed an eventual change of its default stage, can a change in the next stage occur. In fact, the

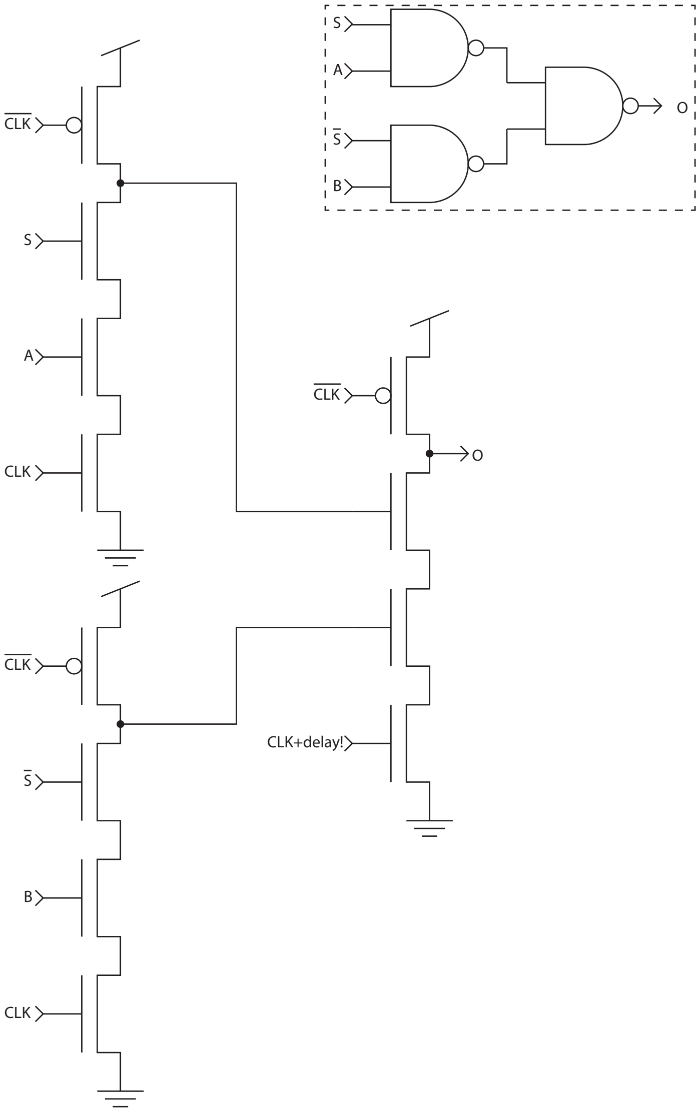


Figure 7.8: Generic dynamic logic implementation of a digital multiplexer

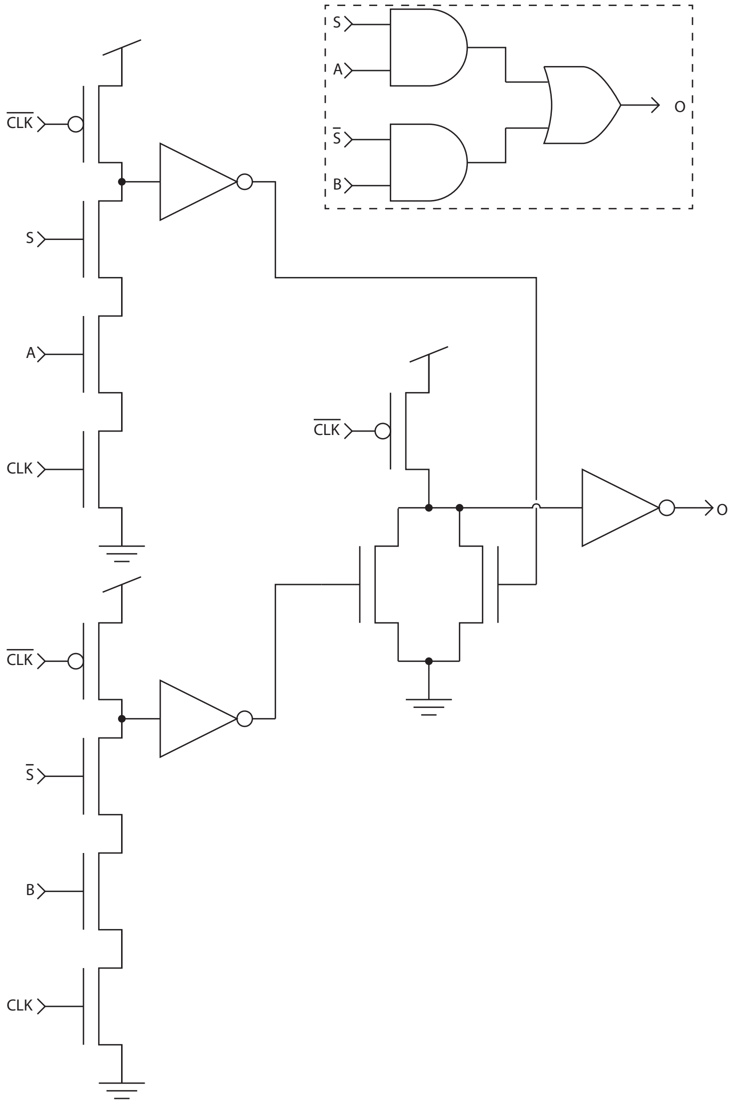


Figure 7.9: Domino logic implementation of a digital multiplexer

switches in consecutive stages initiating the evaluation phase can be omitted entirely, i.e. the nFET receiving the CLK signal, and the correct function is maintained. This has been done in the illustration. *However*, this invites a significant cross current during the pre-charge phase, i.e. when the pre-charge pFET starts conducting and the inputs from previous stages are still in their evaluation phase for a split-(pico)second!!! Thus, it may be worth keeping the evaluation switch and a non-overlapping CLK signal also for consecutive stages. Also if the stage does receive inputs from static logic, i.e. not the appropriate domino logic circuits, the evaluation phase switch/nFET needs to be kept. So typically this might be the case for a first stage that interfaces to static logic. In the illustration, the two gates in the first stage have kept their evaluation switch.

Note that due to the extra inverter, this solution requires more transistors and becomes slower. Also, the generic smallest building blocks are now ORs and ANDs rather than NORs and NANDs, as reflected in the gate representation of the MUX in the figure. Here, the latter is no big impediment, as there is only three gates needed in either case.

A variation of the domino logic is the so called **NP domino logic** that avoids the inverters and thus the extra propagation delay and extra transistors. The multiplexer implementation is shown in figure 7.10. Here, consecutive stages are implemented toggling between PD and PU networks for the logic function. Thus the default precharge state of a PD network gate is high, and if connected to a PU network gate will prevent premature pull up of its output signal. The same domino effect results from this without the inverters. A disadvantage layout-wise, is that PU networks require the usually bigger pFETs. So the layout space requirement will be bigger as opposed to the generic dynamic logic implementation in figure 7.8.

So dynamic logic is typically employed when extremely high speed is required. Another application area is in asynchronous circuits. Remember that the length of the clock cycle in synchronous circuit is determined by the worst case propagation delay of combinational logic executing a computation where the result is needed at the end of the clock cycle. Dual rail Dynamic logic can implement a different kind of combinational logic that can actually *signal* one it has finished a computation. Thus, rather than using a clock that is determined by the worst case delay, a combinational logic block can simply signal the next stage that the result is ready, and a sequence of processing steps can be executed as fast as possible, i.e. not as slow as the worst case would require. ******

7.5 Example of advanced combinational logic circuits

- decoders
- encoders
- multiplexers
- demultiplexers
- ALU

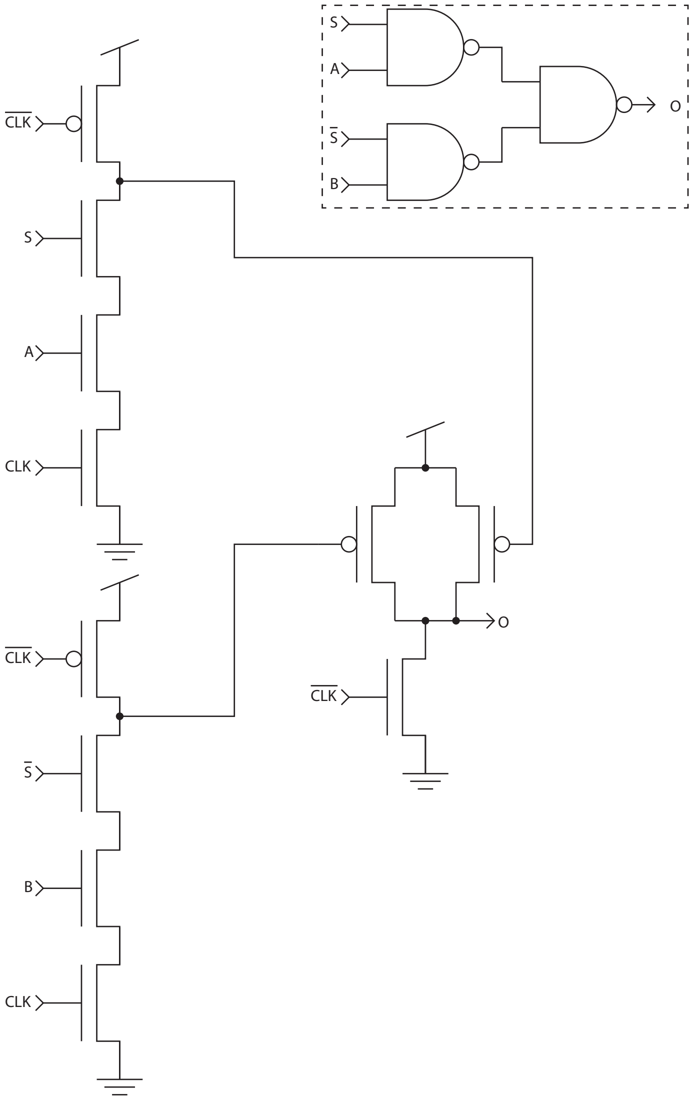


Figure 7.10: NP domino logic implementation of a digital multiplexer

Chapter 8

Sequential Logic Circuits

8.1 Asynchronous latches

Combinational logic circuits implement Boolean functions. That is to say, combinational logic does not contain any form of memory or feedback connections. In contrast, sequential logic circuits do contain feedback loops and thus memory elements and an internal state such that the output does not only depend on the input, but also on the internal states and thus the history or sequence of inputs.

So a truth table describing a sequential logic circuit will not just describe the output for all possible inputs, but will need to describe the outputs for all possible inputs, *as well* as for all possible internal states and the former state of the output. Such a truth table is called a **characteristic table**. Note that one can actually design a circuit that oscillates or ends up in a **metastable state** (i.e. where some signals end up not being 'digital' anymore but get stuck at a voltage between Gnd and Vdd), if there is a circular loop of digital signals that cannot achieve a consistent stable state. The typical example would be an inverter with its output shorted to its input. It will eiother settle on a non-digital voltage somewhere close to $\frac{\text{Vdd}}{2}$ or oscillate wildly.

A simple useful sequential logic circuit is a 1-bit storage element. One quite straight forward implementation is *two* inverters connected in a loop (figure

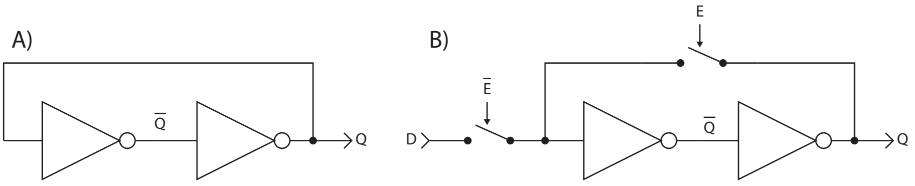


Figure 8.1: A) principle of a double inverter loop storage cell. State Q can be 0 or 1 and is maintained indefinitely, without input to change it. B) The D-latch or transparent latch rectifies this with two inputs: when $E = 1$ then the state is maintained, and when $E = 0$ the latch becomes a 'transparent' double inverter that just conveys the input D.

D	E	Q	Q
0	0	0	0
0	0	1	0
0	1	0	0
0	1	1	1
1	0	0	1
1	0	1	1
1	1	0	0
1	1	1	1

E	Q
0	D
1	Q

Table 8.1: D-latch characteristic table, with full explicit table on the left and an abbreviated version on the right.

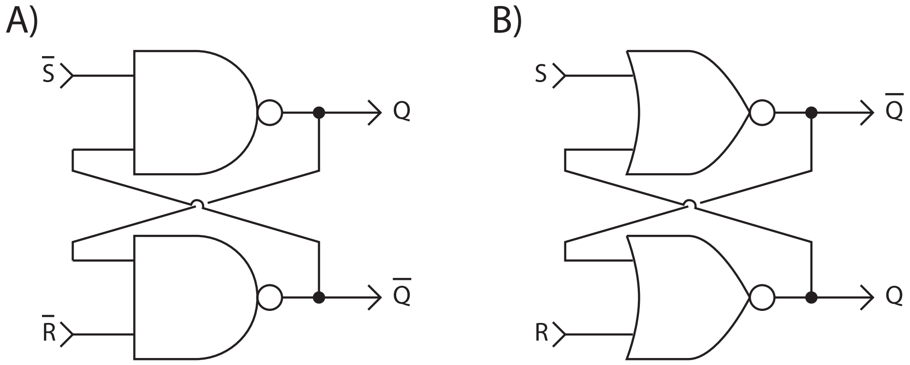


Figure 8.2: Another popular implementation of a latch is the RS-latch. When S is active and L is inactive, $Q <= 1$ and $\overline{Q} <= 0$ and vice versa. When both S and R are inactive, the state is maintained. When both R AND S are active that is actually 'illegal' and the state in a general definition of a RS-latch is not defined. For the specific implementations A) both $Q <= 1$ AND $\overline{Q} <= 1$ and for B) it's $Q <= 0$ AND $\overline{Q} <= 0$.

R	S	Q	Q	$\overline{Q}$
0	0	0	0	1
0	0	1	1	0
0	1	0	1	0
0	1	1	1	0
1	0	0	0	1
1	0	1	0	1
1	1	0	0	0
1	1	0	0	0

R	S	Q	$\overline{Q}$
0	0	Q	$\overline{Q}$
0	1	1	0
1	0	0	1
1	1	0	0

Table 8.2: NOR gate implementation of RS-latch characteristic table, with full explicit table on the left and an abbreviated version on the right.

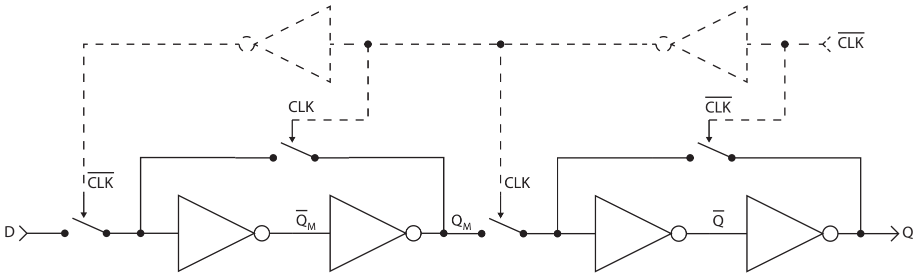


Figure 8.3: D-flipflop implemented with master slave D-latch pair. The inverters on top are not strictly necessary, but ensure a safe timing of CLK and $\overline{CLK}$: it's important that the left hand D-latch latches *before or simultaneously with* the lefthand side latch going transparent, i.e. if the left hand side becomes transparent *first* it could lead to a signal feed through at the negative flank of CLK.

8.1 A)). This loop will indefinitely maintain one of two possible states, either $Q = 1$ or $Q = 0$. In order to be able to control the stored state figure 8.1 B) introduces a 'gating' control signal E. If $E = 0$, the feedback loop is broken and the output Q adapts the state of the input signal D. This storage element is known as **D-latch** or **transparent latch**. Its characteristic table is given in table 8.1.

8.2 Synchronous flip-flops

Most digital electronics is **synchronous**. Synchronous means that there is a master digital signal, the so called clock (CLK), that defines the timing of state changes. Unlike a 'gating' signal it does not toggle between a phase where signals are latched and another phase where storage cells become transparent, i.e between a phase where storage elements act as storage elements and a phase where they behave like combinational logic. Instead a clock signal makes synchronous storage elements latch the state of an input signal in a singular moment in time, i.e. when there is a *transition* of the clock signal (e.g. from 0 to 1, or vice versa, or even at both transitions). The input to synchronous storage elements may then change freely while the CLK is high or low without affecting the storage cell's content. Only at the exact moment of the appropriate *transition* will the storage cell take a 'snapshot' of the input signal and store it.

The most common synchronous storage cell is the D-flipflop. It's implementation with two D-latches is depicted in figure 8.3. It is often referred to as master-slave configuration of D-latches, where the first D-latch is the 'master'. While the CLK signal is HIGH the master keeps the input D as it was as CLK went HIGH and the slave is transparent. When the CLK goes LOW again the slave now latches the state that had been stored in the master, i.e. no change occures at the output Q despite the master now being transparent. Only at the next transition of CLK from LOW to HIGH will the new state of the input D occure at the output Q. So the effect is that of a storage element that changes state only and exactly at the time of a LOW to HIGH transition, i.e the rising

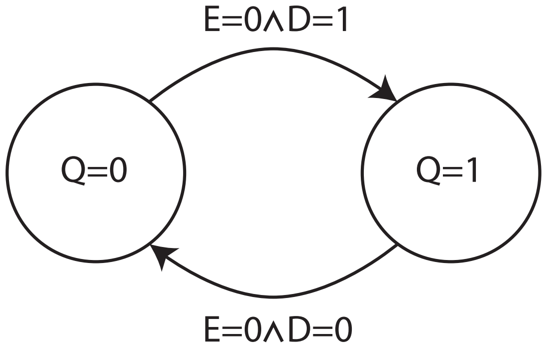


Figure 8.4: The state transition graph of a D-latch. This is an asynchronous graph, i.e. there is no implicit condition of a rising CLK signal for state transitions and the entire condition is explicitly stated with the state transition arrows. Note that any condition that is not listed along an arrow leading from a state will not change that state.

flank of the CLK signal.

8.3 Finite state machines

Besides a circuit schematic or a characteristic table there is another way to describe the behaviour of a sequential logic circuit, i.e. a **state transition graph**. This is a graph consisting of blobs and arrows between those blobs. The blobs are labelled with state identifiers and the arrows with conditions for state transitions. Thus the state transition diagram for the D-latch looks like depicted in figure 8.4.

The states in such a graph could theoretically be defined as the combination of all digital signal nodes in the circuit. In that manner one could even describe purely combinational logic with state-transition graphs too. However, usually one uses only the states of digital signal nodes that are able to maintain their state 'autonomously', i.e. their state is not fully defined by the the nodes defined as inputs to the circuit, but dependent on signal history. Another way of defining them is to say that they are the nodes that are needed in *addition* to the inputs to fully describe the state of the circuit after some settling time. So in short: the state of storage elements in the circuit.

The asynchronous state transition graph of the D-flipflop is given in figure 8.5. The state is defined by the two storage elements, the D-latches. The graph becomes simpler as a synchronous state transition graph (figure 8.6). Only looking at what happens at the rising flank of the clock signal, Q and Q_M are interdependent and it's sufficient to only consider Q as the content of the D-flipflop.

The value of state transition graphs is not just as a way to describe a sequential logic circuit, but also as a means to describe an abstract function and to construct a sequential logic circuit with that specific function. Consider the example of a simplistic traffic light controller giving rise to the state transition graph on figure 8.7. This traffic light has but two states: either it is green for

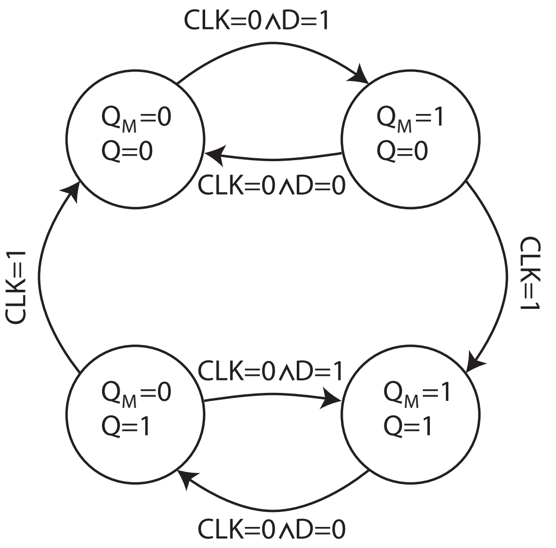


Figure 8.5: The state transition graph of a D-flipflop in asynchronous form. The D-flipflop is the basic synchronous storage element but here we give the graph that derives it from asynchronous D-latches. Seen as a whole with only its output Q as binary state variable, it can be described with a synchronous version of a state transition graph in figure 8.6. This asynchronous version however, makes the potential race condition between the input D and the clock signal CLK more visible: The timing of CLK and D if they change very close to eachother in time is critical. If they change simultaneously, e.g. when $Q_M = 0, Q = 1$, it is not defined what should happen. It could be both a transition to $Q_M = 1, Q = 1$ or to $Q_M = 0, Q = 0$. To avoid this conundrum, synchronous circuits are carefully designed such that inputs are certain to be stable when the trigger flank of the clock occurs. (Such race conditions are generally a problem in asynchronous logic design if the change of two different signals lead away from a state to two different states.)

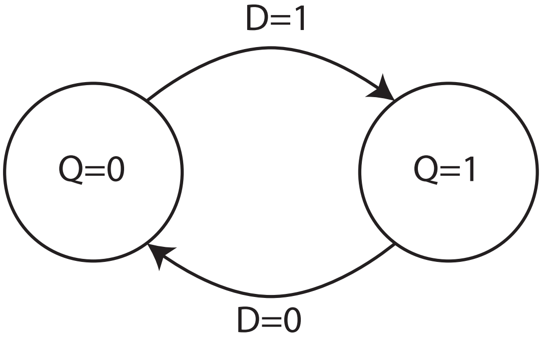


Figure 8.6: Synchronous D-flipflop state transition graph. This is the most usual form of state transition graphs, where the *states* are entirely defined by the content of synchronous storage elements. All state transitions have the implicit additional condition that they only occure at a specific transition of a clock signal.

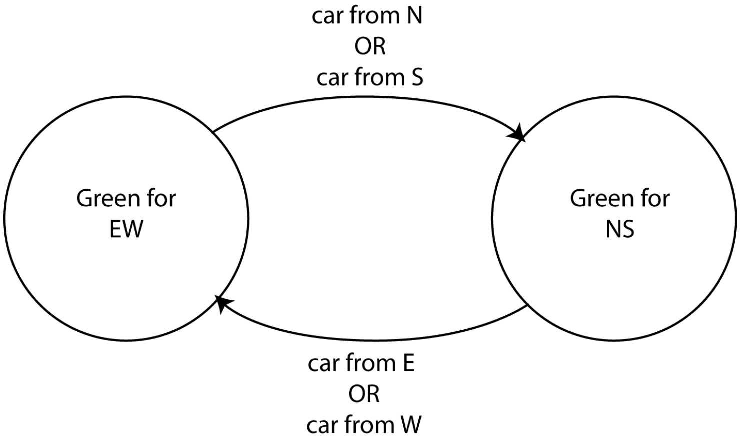


Figure 8.7: State transition graph of a traffic light controller that only changes state when a car is detected waiting in the blocked direction.

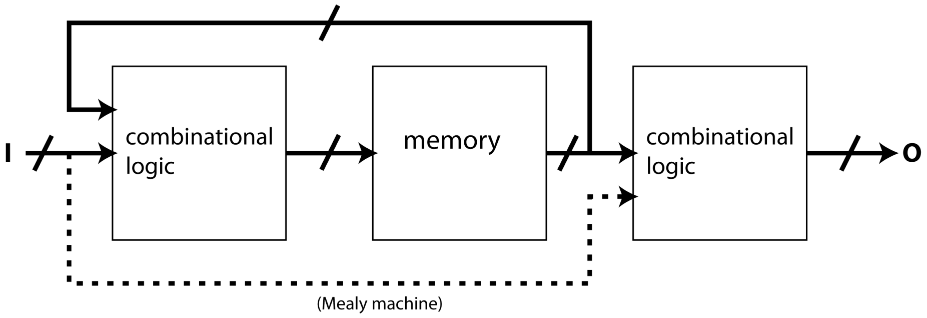


Figure 8.8: General structure of a **finite state machine (FSM)** that allows a direct realization from a state transition graph. The state is stored in the memory which consists of the necessary number of bit storage cells. The feedback combinational logic implements the state transition conditions. The combinational logic to the output generates control signals from the state in the memory. This is known as a **Moore FSM**. A variant is the **Mealy FSM** that allows inputs to be used directly to generate output control signals (dashed arrow). This may allow to use fewer states but can be problematic if the inputs are not synchronized to the same CLK as the memory of the FSM. In that case the outputs may not be synchronized to that CLK either.

the NW direction and red for EW, or vice versa. It maintains its state unless one or more cars are waiting in the blocked direction. The implicit CLK signal is very slow in electronics terms, let's say the clock cycle leasts for a whole 20 seconds, so the minimum duration for the light tos switch from one state to the other is 20 seconds.

There is a systematic way of deriving a logic circuit of the structure of a **finite state machine (FSM)** (see figure 8.8) from such a state transition graph.

First one assigns a different number to each of the states. In our example we can choose to assign '0' to 'green EW' and '1' to 'green NS', for example. Then one encodes these numbers as binary codes. Here, since we only have two states we can use a single bit and call that state variable 'GEW'. If there are more states on needs to use an appropriate number of 1-bit storage cells/variables. Binary numbers are the most straight forward way of encoding the states, but other encoding schemes might be chosen as well and might result in a different, and possibly simpler FSM. Anyhow, any encoding scheme works that assigns a unique bit-pattern to each state.

Then one also needs to encode the (sensor) input as binary signal. Here we have two sensors the state of which we encode as two Boolean variables 'EW' and 'NS'. The variables are '0' if no car is waiting and '1' if a waiting car is detected. Thus we can create a characteristic table and then a characteristic function, i.e. list all possible combinations of internal state and input together with the resulting next state. The characteristic table is shown in table 8.3. From it one can derive a characteristic function for each state bit by for example use all rows where that bit turns '1', use a and gate for each row 'and-ing' the input states and internal states of each of those rows and 'or-ing' them together:

EW	NS	GEW	GEW
0	0	0	0
0	0	1	1
0	1	0	0
0	1	1	0
1	0	0	1
1	0	1	1
1	1	0	1
1	1	1	0

Table 8.3: Characteristic table traffic light controller

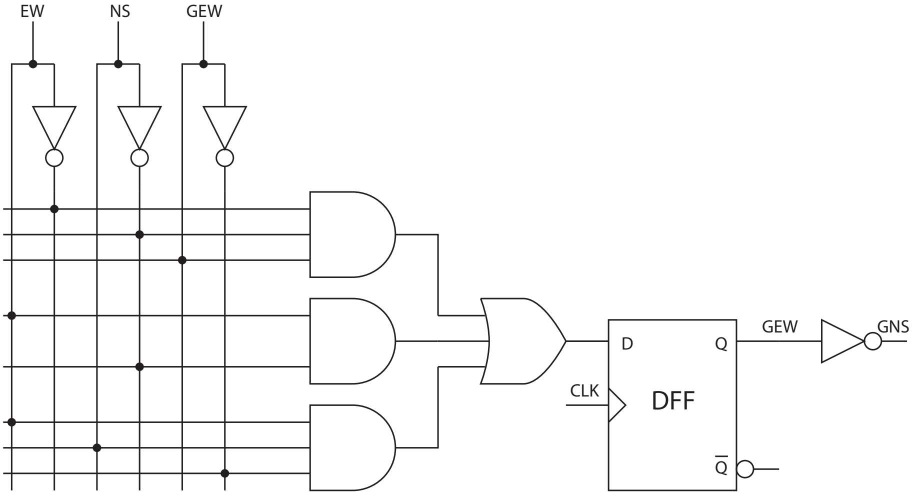


Figure 8.9: Schematic of the traffic light final state machine

$$\begin{aligned}
GEW = & \\
& (\overline{EW} \wedge \overline{NS} \wedge GEW) \\
& \vee (EW \wedge \overline{NS} \wedge \overline{GEW}) \\
& \vee (EW \wedge \overline{NS} \wedge GEW) \\
& \vee (EW \wedge NS \wedge \overline{GEW})
\end{aligned} \tag{8.1}$$

This expression can then be used to make the combinational logic feedback circuit in the FSM and will get a functionally correct FSM. Optionally one can first try to simplify that expression somewhat. Here for example the middle two lines $(EW \wedge \overline{NS} \wedge \overline{GEW})$ and $(EW \wedge \overline{NS} \wedge GEW)$ can be reduced to a single one $(EW \wedge \overline{NS})$:

$$\begin{aligned} GEW = &\\ &(\overline{EW} \wedge \overline{NS} \wedge GEW)\\ &\vee (EW \wedge \overline{NS})\\ &\vee (EW \wedge NS \wedge \overline{GEW}) \end{aligned} \tag{8.2}$$

This results in the schematic in figure 8.9. Equation (8.2) is implemented as the combinational logic deriving input D to the one bit storage cell, the D-flipflop. Just for illustration a single inverter is added at the output as some output cobinational logic generating the control signal that also turns the light in the NS direction green. This signal can of course also be taken directly from the $\overline{Q}$ port from the D-flipflop.

8.4 Example list of more advanced sequential logic circuits

For now and for this compendium, we will name some more sequential logic circuits only as a list. It will be outside the scope of the course to go into the design specifics of these. These are some examples of electronics that can still be usefully described as FSMs. For even more complex sequential electronics, one can for example use hierarchical FSMs that interact with eachother, i.e. it's not very useful to describe an entire CPU or Microcontroller as FSM, but parts of it can be, e.g. the arithmetic logic unit (ALU).

- counters
- appliance controllers (dishwashers, fridges, thermostats, ...)
- control circuit for SAR ADCs
- ALU

Part IV

Analog Circuit Basics

Chapter 9

Single Ended Amplifiers

9.1 Low frequency/DC analysis

Figure 9.1 shows a version of a general small signal representation of an amplifier. This particular model is convenient if R_O is large and one can think of it as a transconductance amplifier. Normally it is considered connected to a signal source represented by a voltage source and its output resistance R_S and a load represented by load resistance R_L. When deriving the equivalent parameters R_I, G_M, and R_O from a concrete circuit model, R_S and R_L may influence the result, e.g R_S may influence the derivation of R_O, etc.

So for any single ended amplifier circuit, one can find a linearized version of it that complies with the structure in Fig. 9.1. We define and find R_I, G_M, and R_O at a particular biasing point/point of operation as:

$$\begin{aligned} R_I &:= \frac{v_i}{i_i} = \frac{\partial V_I}{\partial I_I} && (9.1) \\ G_M &:= \frac{i_o}{v_i} = \frac{\partial I_O}{\partial V_I} \text{ for } V_O \text{ held constant and } v_o = 0\ ! && (9.2) \\ R_O &:= \frac{v_o}{i_o} = \frac{\partial V_O}{\partial I_O} \text{ for } V_{SIG} \text{ held constant and } v_{sig} = 0\ ! && (9.3) \\ & && (9.4) \end{aligned}$$

These definitions indicate two ways of finding these parameters. For both, you first have to find the point of operation in the large signal model, i.e. the bias currents and voltages V_I, I_I, V_O, and I_O. Then you either:

1. continue in the large signal domain and find the derivatives by applying small test signals.

2. replace the circuit elements with their small signal equivalents and find the small signal ratios by applying small test signals.

So for example to find $\frac{\partial I_O}{\partial V_I}$ in the large signal domain you connect an ideal voltage source to the output that holds it constant at its point of operation V_O, and measure or compute ∂I_O when applying an infinitesimally small ∂V_I to the

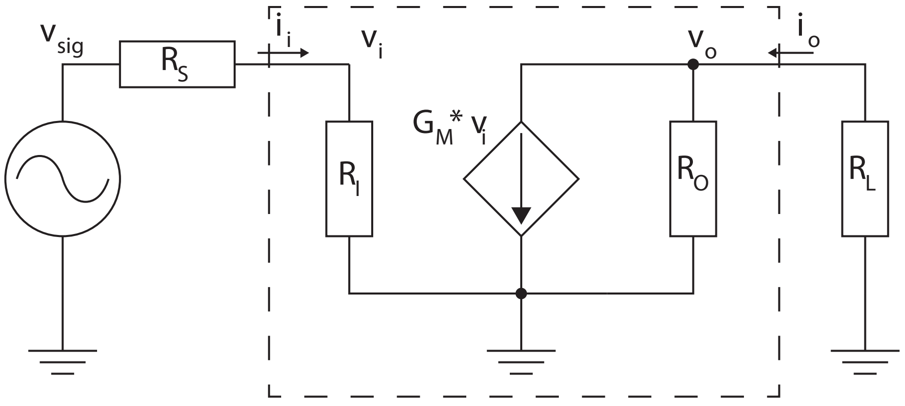


Figure 9.1: One possibility for a general linear (or small signal) model of an amplifier is depicted inside the dashed box. This particular model is convenient if R_O is large and one can think of it as a transconductance amplifier. Normally it is considered connected to a signal source represented by a voltage source and its output resistance R_S and a load represented by load resistance R_L. Both R_S and R_L can have an influence when deducing the amplifier parameters R_O and R_I respectively.

input. Or you replace all circuit elements with their small signal equivalents, connect v_o to Gnd, and compute $\frac{v_o}{i_o}$

We will refer to the maximal voltage gain of the amplifier, i.e with the ideal (but usually unachievable) configuration where $R_S = 0$ and $R_L = \infty$, as **intrinsic gain** A_O:

$$A_O = -G_M R_O \tag{9.5}$$

Be aware that we defined $G_M := \frac{i_o}{v_i}$ and that if you want to know what $\frac{i_o}{v_{sig}}$ is, you have to take into consideration that there is a resistive voltage division happening between R_S and R_I, i.e. $v_i = v_s \frac{R_I}{R_S+R_I}$. Consequently we have:

$$\frac{i_o}{v_{sig}} := G'_M = G_M \frac{R_I}{R_S + R_I} \tag{9.6}$$

Also be aware that the voltage gain $\frac{v_o}{v_i}$ of this amplifier is severely influenced by R_L. So if you design the circuit to have a large R_O in order to achieve a large voltage gain, then you have to pay close attention that R_L is of the same order as R_O or ideally even bigger, because:

$$\frac{v_o}{v_i} = -G_M(R_O||R_L) \tag{9.7}$$

So if you want to know the voltage gain A' starting from v_{sig} taking both R_S and R_O into account then it becomes:

$$A' = \frac{v_o}{v_{sig}} = -G'_M(R_O||R_L) = \frac{R_I}{R_S + R_I} A_O \frac{R_L}{R_O + R_L} \tag{9.8}$$

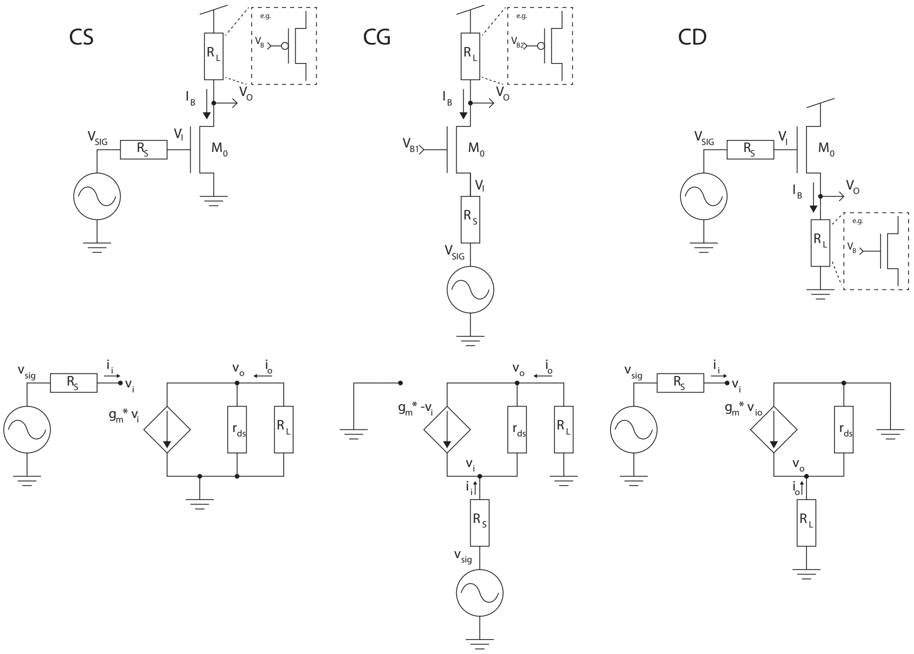


Figure 9.2: The common source (CS), common gate (CG) and common drain singke FET amplifier configurations. Large signal representation on top and the corresponding small signal representations below.

9.2 Single transistor amplifier configurations

In the following we will introduce three single transistor gain stages, simply by using different pairs among the transistor pins as input and output. Then by deriving the small signal R_I, G_M, and R_O under specific biasing conditions we can derive the gain of all these stages form the general transconductance amplifier model in figure 9.1 and transfer function (9.8).

9.2.1 Common Source (CS) gain stage

Now you might actually realize that this general transconductance amplifier model (figure 9.1) resembles our small signal transistor model considerably. Most obviously the **Common Source (CS)** amplifier we have already introduced as a small signal analysis example in figure 3.6 chapter 3 fits very nicely into this scheme. We simply declare $v_i := v_g$, $v_o := v_d$, and $v_s := 0$ is Gnd and voilà: we have ourselves an amplifier just like the one in figure 9.1. The name 'common source' actually comes from the fact that the source terminal is the common reference for both input and output. It is depicted in figure 9.2, in the left column labelled CS. The amplifier parameters are listed in table 9.1. In CMOS technology the initial load is usually provided by another FET transistor with a fixed $V_B = V_{GS}$. If the output is further connected to another circuit's input, that circuit's input resistance would also need to be added in parallel for the total R_L. If however, it is not connected to anything else, the r_{ds} of that **biasing transistor** is all of R_L. The bias current is then the current provided by that transistor and needs to be matched approximately by the CS transistor (setting an appropriate DC-level for the input signal V_{SIG}) to obtain an appropriate point of operation. If the currents are not matched, the output will simply saturate towards either Gnd or Vdd, and the circuit cannot operate as intended.

Note that the previously defined **intrinsic gain** often refers to the intrinsic gain of a CS connected single transistor, i.e. the best gain you can possibly achieve under a certain biassing condition. Thus, using the values for the CS stage in table 9.1, the intrinsic gain for a single transistor is:

$$A_0 = g_m r_{ds} \tag{9.9}$$

The CS amplifier achieves good voltage gain because of the high output resistance together with the good transconductance. However, it cannot drive loads with small resistances without loosing that gain. In the case of a very small resistance as load it is better to think of it as a transconductance amplifier, i.e. where it delivers a reliable current $i_o \approx g_m v_i$ to the load.

9.2.2 The Common Gate (CG) gain stage

Now, a FET is a three terminal device[1] and can be hooked up differently. In fact, there are two more configurations that also work as amplifiers, and you might already suspect what they shall be called. First, the **Common Gate (CG)** gain stage/amplifier where $v_i := v_s$ and $v_o := v_d$, and $v_g := 0$ is connected to Gnd.

[1] assuming V_B and V_S are shorted and/or that V_B is fixed at Gnd.

	R_I	G_M	R_O
CS	∞	g_m	r_{ds}
CG	$\frac{r_{ds}+R_L}{1+g_m r_{ds}}$	$-(\frac{1}{r_{ds}}+g_m)$	$r_{ds}+R_S(g_m r_{ds}+1)$
CD	∞	$-g_m$	$\frac{1}{g_m}\|\|r_{ds}=\frac{r_{ds}}{1+g_m r_{ds}}$
Cascode	∞	$g_{m1}r_{ds1}\frac{\frac{r_{ds2}}{1+g_{m2}r_{ds2}}}{r_{ds1}+\frac{r_{ds2}}{1+g_{m2}r_{ds2}}}(\frac{1}{r_{ds2}}+g_{m2})$	$r_{ds2}+r_{ds1}(g_{m2}r_{ds2}+1)$

Table 9.1: Equivalent linear amplifier parameters for single transistor amplifier configurations.

The CG amplifier with an ideal voltage source as signal behaves much the same as the CS amplifier without inverting the output, i.e. a decent voltage amplifier. However, due to its low input resistance it is heavily affected by signal source output resistance R_S. It is better suited and thought of as a **current conveyor**: if one uses a current source i_{sig} with R_S in parallel rather than a voltage source v_{sig}, and if $R_S >> \frac{1}{g_m}$, then since $R_I \approx \frac{1}{g_m}$ (as long as $R_L \leq r_{ds}$), the input voltage is $v_i \approx \frac{i_{sig}}{g_m}$. And since the transconductance is approximately $G_M \approx -g_m$, the output current will be approximately $i_o \approx i_{sig}$ and this current will be little affected by a resistive load (as long as it is smaller than the considerable output resistance). This is thus useful to convey a current from a source with not so high output resistance to an output with very high output resistance $R_O = r_{ds}$.

9.2.3 The Common Drain (CD) stage or Source Follower (SF)

And finally the **Common Drain (CD)** stage, better known as **Source Follower (SF)**, where $v_i := v_g$, $v_o := v_s$ and $v_d := 0$ is the common Gnd reference. The parameters for all three are listed in table 9.1.

The CD/SF stage is not really an amplifier, since its voltage gain is close to 1. As the name 'source follower' suggests, it is used as a follower. In the small signal world the output is simply a copy of the input, with a small gain error. In the large signal world it follows its input with an offset, i.e. the gate to source voltage V_{GS} that is determined by the biassing current in the branch.

9.3 The cascode gain stage

A **Cascode** gain stage depicted in figure 9.3 is a CS and CG in series. The total circuit's output resistance is that of the CG stage, with the output resistance of the CS stage as the CG's signal source resistance. The total input resistance is infinite. The total transconductance is that of the CG stage multiplied with the voltage gain of the CS stage with the input resistance of the CG stage as load when $R_L = 0$. Using index 1 for the CS transistor and index 2 for the CG transistor:

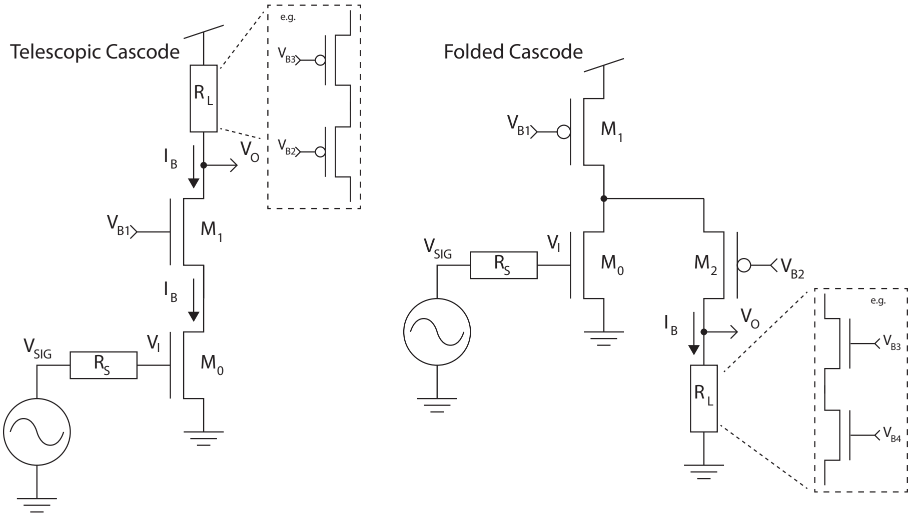


Figure 9.3: The telescopic and folded cascode gain stage. The telescopic variant modulates the current into a CG stage directly from a CS stage. The telescopic variant does the same by subtracting from the input current to the CG stage of the opposite transistor type. The small signal representations are equivalent baring an extra parallel r_{ds} loading the CS drain.

$$\begin{aligned}
R_O &= r_{ds2} + R_{O1}(g_{m2}r_{ds2}+1) = r_{ds2} + r_{ds1}(g_{m2}r_{ds2}+1) && (9.10)\\
G_M &= -A_1'|_{(R_L=0)}\left(\frac{1}{r_{ds}} + g_m\right) && (9.11)\\
&= g_{m1}r_{ds1}\frac{R_{I2}|_{(R_L=0)}}{r_{ds1} + R_{I2}|_{(R_L=0)}}\left(\frac{1}{r_{ds2}} + g_{m2}\right) && (9.12)\\
&= g_{m1}r_{ds1}\frac{\frac{r_{ds2}}{1+g_{m2}r_{ds2}}}{r_{ds1} + \frac{r_{ds2}}{1+g_{m2}r_{ds2}}}\left(\frac{1}{r_{ds2}} + g_{m2}\right) && (9.13)\\
R_I &= \infty && (9.14)
\end{aligned}$$

If one makes the assumptions that in general $g_m r_{ds} >> 1$ and $g_{m1} \approx g_{m2}$ and $r_{ds1} \approx r_{ds2}$ one can simplyfy thes set of equations to:

$$\begin{aligned}
R_O &\approx g_m r_{ds}^2 && (9.15)\\
G_M &\approx g_m && (9.16)\\
R_I &= \infty && (9.17)
\end{aligned}$$

So note that the actual gain with a load R_L, i.e. $A' = -G_M(R_O||R_L)$, will depend a lot on that R_L. If one uses just a single pFET on top of a nFET cascode stage, $R_L = r_{dsp}$. If the drain resistance of a pFET is similar to that of the nFETs (i.e. $r_{dsp} = r_{dsn} = r_{ds}$) then:

$$A' = -G_M(R_O||R_L) = g_m(g_m r_{ds}^2||r_{ds}) \approx g_m r_{ds} \qquad (9.18)$$

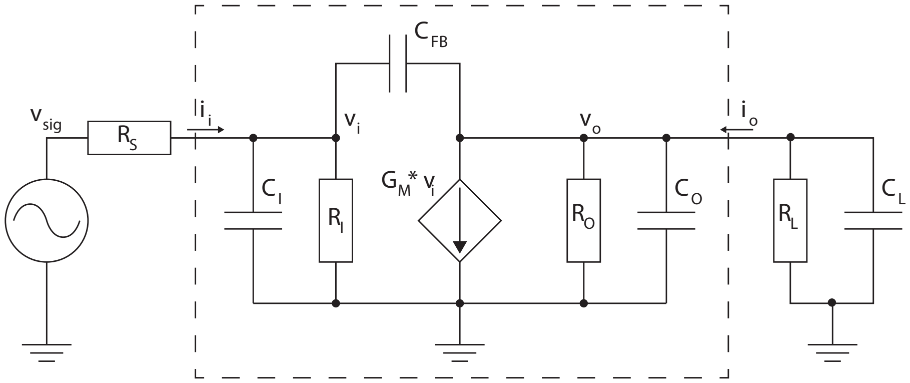


Figure 9.4: General transconductance amplifier model for frequency analysis

And A' is practically no bigger than that of a CS stage. Interestingly though, such a configuration actually can increase the *bandwidth* as compared to just the CS stage, as we will see in subsection 9.4.4. If however, we really want more gain, the solution is to use *two* biassing transistors, i.e. by mirroring the cascode structure (the 'insets' in figure 9.4). The output resistance of two biassing transistors in series can be derived exactly like the one of the cascode gain stage. So now we have:

$$R_L = g_m r_{ds}^2 \tag{9.19}$$

And consequently:

$$A' = -G_M(R_O||R_L) = g_m(g_m r_{ds}^2||g_m r_{ds}^2) = \frac{1}{2} g_m^2 r_{ds}^2 \tag{9.20}$$

9.4 Frequency response

So far we have neglected the 'parasitic' capacitors in the small signal model. Important will be C_{GS}, C_{GD}, C_{DB}, and C_{SB} . Mostly we will consider the bulk being connected to the source, so often-times we can ignore C_{SB}. In general our three terminal amplifier model (fig 9.1) will gain an input capacitor and an output capacitor to Gnd, as well as a capacitor between input and output, resulting in figure 9.4.

9.4.1 Assuming dominant pole at input or output

We have already derived its transfer function (5.26) in section 5.2 and can use the results for the three cases. If we just care for the two cases where either the dominant pole is at the input node or at the output node we can use the simpler expressions in equations (5.35) and (5.29) yielding tables 9.2 and 9.3.

9.4.2 Stability of single transistor stages

Note that the CS and CD/SF stages do in fact have a feedback capacitor from the output to the input node. So this is a possible source of instability which

	$\tau_i = \frac{1}{\omega_{p_i}}$
CS	$R_S\left(C_{GS} + C_{GD}(g_m(r_{ds}\|\|R_L) + 1)\right)$
CG	$\left(R_S\|\|\frac{r_{ds}+R_L}{1+g_m r_{ds}}\right) C_{GS}$
CD	$R_S\left(C_{GD} + C_{GS}\left(1 - g_m\left(R_L\|\|\frac{1}{g_m}\|\|r_{ds}\right)\right)\right)$

Table 9.2: Single transistor amplifiers time constants when the input node dominates

	$\tau_o = \frac{1}{\omega_{p_o}}$	ω_{z_o}
CS	$(r_{ds}\|\|R_L)(C_{GD} + C_{DB} + C_L)$	$\frac{g_m}{C_{GD}}$
CG	$[(r_{ds} + R_S(g_m r_{ds} + 1))\|\|R_L]\,(C_{GD} + C_{DB} + C_L)$	NA
CD	$\left(R_L\|\|\frac{1}{g_m}\|\|r_{ds}\right)(C_{GS} + C_{SB} + C_L)$	$\frac{-g_m}{C_{GS}}$

Table 9.3: Single transistor amplifiers time constants when the output node dominates

might be worthwhile to check. Note that we have derived the complete transfer function for the general transconductance amplifier model in equation (5.26). It is actually a second order LP function (If $R_I^* > 0$, so that normally means $R_S > 0$). Thus, combining that with the definition of the Q-factor in (5.39) one gets:

$$Q = \frac{\sqrt{A}}{B} = \frac{\sqrt{R_I^* R_L^*\left[(C_{FB} + C_L^*)(C_{FB} + C_I) - C_{FB}^2\right]}}{R_L^*(C_{FB} + C_L^*) + R_I^*\left(C_I + C_{FB}(1 + G_M R_L^*)\right)} \tag{9.21}$$

CG stage

Just to confirm that we do not have a problem *without* the the feedback capacitor (among the three that is the **CG stage**), abbreviating $\tau_i = R_I^* C_I$ and $\tau_o = R_L^* C_L^*$, we are left with:

$$\begin{aligned} Q^2 = \frac{\tau_i \tau_o}{(\tau_i + \tau_o)^2} &= \begin{cases} < \frac{1}{4} \text{ if } \tau_i \neq \tau_o \\ = \frac{1}{4} \text{ if } \tau_i = \tau_o \end{cases} \\ \Rightarrow Q \leq \frac{1}{2} & \end{aligned} \tag{9.22}$$

The above is true because it is generally true that

$$\frac{ab}{(a+b)^2} = \begin{cases} \frac{1}{4} \text{ if } a = b \\ < \frac{1}{4} \text{ if } a \neq b \end{cases} \tag{9.23}$$

for positive a and b (i.e. $a, b \in \mathbb{R}^+$).

So with $Q \leq \frac{1}{2}$ the stage is always stable with real poles, i.e. no resonance or even flat bandwidth extension.

CS stage

Including C_{FB}, we get:

$$\begin{aligned} Q^2 &= \frac{(\tau_i + R_I^* C_{FB})(\tau_o + R_L^* C_{FB}) - R_I^* R_L^* C_{FB}^2}{\left(\tau_o + R_L^* C_{FB} + \tau_i + R_I^* C_{FB}(1 + G_M R_L^*)\right)^2} \\ &\leq \frac{(\tau_i + R_I^* C_{FB})(\tau_o + R_L^* C_{FB})}{\left(\tau_o + R_L^* C_{FB} + \tau_i + R_I^* C_{FB}(1 + G_M R_L^*)\right)^2} \end{aligned} \tag{9.24}$$

If the 'Miller term' $1 + G_M R_L^*$ is bigger than or equal to 1 (as it is for the **CS stage**) we can write further (again using (9.23)):

$$\begin{aligned} Q^2 &\leq \frac{(\tau_i + R_I^* C_{FB})(\tau_o + R_L^* C_{FB})}{\left(\tau_o + R_L^* C_{FB} + \tau_i + R_I^* C_{FB}\right)^2} \leq \frac{1}{4} \\ &\Rightarrow Q \leq \frac{1}{2} \end{aligned} \tag{9.25}$$

CD/SF stage

However, for the CD/SF stage the 'Miller term' is close to zero, as $G_M R_L^* = -g_m(\frac{1}{g_m}||r_{ds}||R_L) \approx -1$. So if it were exactly zero ($1 + G_M R_L^* = 0$), we can write:

$$\begin{aligned} Q^2 &= \frac{(\tau_i + R_I^* C_{FB})(\tau_o + R_L^* C_{FB})}{\left(\tau_o + R_L^* C_{FB} + \tau_i\right)^2} - \frac{R_I^* R_L^* C_{FB}^2}{\left(\tau_o + R_L^* C_{FB} + \tau_i\right)^2} \\ &= \frac{\tau_i \tau_o + \tau_i R_L^* C_{FB} + \tau_o R_I^* C_{FB}}{\left(\tau_o + R_L^* C_{FB} + \tau_i\right)^2} \end{aligned} \tag{9.26}$$

In this case Q is no longer limited and it can be shown that the condition for $Q > \frac{1}{2}$ can be met if:

$$\begin{aligned} \frac{1}{4} &< Q^2 \\ &\Leftrightarrow \\ \frac{1}{2}\left(\frac{R_I^* C_I}{R_L^*(C_L^* + C_{FB})} + \frac{R_L^*(C_L^* + C_{FB})}{R_I^* C_I}\right) - 1 &< \frac{2 C_L^* C_{FB}}{C_I(C_L^* + C_{FB})} \end{aligned} \tag{9.27}$$

This condition is certainly met when

$$R_I^* C_I = R_L^*(C_L^* + C_{FB}) \tag{9.28}$$

, because then the left hand term becomes zero and the right hand term is always bigger than zero.

So specifically for the CD/SF stage this *sufficient condition* derived here (i.e. it's *one* scenario that can lead to resonance and instability, but not necessarily the only one) using the specific parameters involved this is:

$$R_S C_{GD} \approx (C_{GS} + C_{SB} + C_L)(\frac{1}{g_m}||r_{ds}||R_L) \tag{9.29}$$

Which you can get if:

$$R_S g_m \approx \frac{(C_{GS} + C_{SB} + C_L)}{C_{GD}} \tag{9.30}$$

That is that the ratio of the output resistance of the signal source and the output resistance of the CD/SF stage is similar to the ratio of the CD/SF input capacitance (excl. feedback cap) and the total capacitance at the CD/SF stage output (incl. feedback cap). So if one is so unfortunate to observe some resonance in a SF then one can try to reduce R_S or to increase C_L to get out of this funk.

Take home message

So the **take home message** here is that CS and CG by themselves do not give rise to resonance or instability, while a CD/SF stage can.

9.4.3 Intrinsic CS/transistor speed

When it comes to maximum frequency that can be achieved by a CS stage under ideal circumstances there is (analogous to the maximum possible gain, the *intrinsic gain*) the so called **transition frequency** ω_T, or more descriptively also known as **unity (current) gain frequency**. For the current gain considered here, it is the frequency at which the output current from the drain (connected to small signal Gnd) of a CS configured FET is no bigger than the input current. So if one assumes that the load connected to the drain includes a capacitive load C_L that is of similar magnitude than the input capacitance C_I, also the voltage gain will be smaller or equal to 1. So an effective upper limit to the unity gain bandwidth of such a CS stage (just as the intrinsic gain is the upper limit for the gain). The condition of unity current gain translates as:

$$\begin{aligned} |i_o| &= |i_i| \\ g_m|v_i| &= |v_i s(C_{GS} + C_{GD})| = |v_i|\omega_T(C_{GS} + C_{GD}) \end{aligned} \tag{9.31}$$

so it follows that:

$$\omega_T = \frac{g_m}{C_{GS} + C_{GD}} \tag{9.32}$$

9.4.4 Cascode gain stage frequency behaviour

The frequency behaviour of the Cascode gain stage can no longer be fully described by the general transconductance amplifier model, because of a internal node and capacitors not represented in that model.

Still, if we assume a dominant pole on either the input node or the output node, we can draw a few conclusions. The cascode gain stage offers some interesting trade offs. Let's look at it again a s a CS and a CG stage in series. Let us furthermore again simplify our ruminations to the two cases where a) the input node gives rise to the dominant pole, and b) the output node hosts a clearly dominant pole.

In case a) we can look at the CS stage with its load resistance being the input resistance R_I to the CG stage. The later heavily depends on the R_L at the cascode output. We have lernt that we get maximal gain when R_L matches $R_O = g_m r_{ds}^2$. In that case the input resistance of the CG stage is of the order of $R_I \approx r_{ds}$. Thus the cut off at the input node appears at the same frequency as for a CS stage alone with a single transistor as load. So the cascode gives us its significantly increased gain without loss of BW. On the other hand if we use but a single transistor as load for the cascode, we have seen earlier that we do 'only' get approximately the gain of a single CS stage again. However, now the BW at the input node corresponds to a CS stage with a very small load resistance since the CG's $R_I \approx \frac{2}{g_m}$. Thus the dominant pole time constant appears at about $R_S(C_{GS} + 2C_{GD})$, i.e. the 'Miller term' $(g_m(r_{ds}||R_L) + 1) * C_{GD}$ is reduced to just $2 * C_{GD}$ so the BW is extended by about the gain of the equivalent single CS stage.

In case b) we are unfortunately not so well of: choosing a big $R_L \approx g_m r_{ds}^2$ will still increase the DC gain A'_{DC} but at the price of a proportional decrease in BW. Or by choosing a small $R_L \approx r_{ds}$ will only increase the BW back to the level of a CS stage, as well as the DC gain.

Chapter 10

Differential Amplifiers

Differential amplifiers (i.e. with differential input, while 'fully differential' amplifiers also have differential output) refer their input not to global -Vss or Gnd but to an explicit reference input $V-$. This makes the differential amplifier more flexible than the single ended variants we have looked at so far. Advantages are immunity to correlated noise at the two inputs and the possibility to use them in feedback circuits, as well as rending the concept of negative inputs simple.

10.1 Differential pair

FET differential amplifiers are usually based on the differential pair. Its core are two transistors in parallel ideally connected to a current source at their sources, but realistically this is usually a third transistor with a fixed bias voltage at its gate and thus a good approximation of a good current sources with high output resistance. See figure 10.1.

The intuitive explanation of its function is that the total current is limited and divided between the two parallel transistors. That balance tips rapidly in favour of the transistor with the higher gate voltage input and is largely only dependent on the *difference* of the input voltages and only to a minimal degree by the absolute input values. So in the small region where the inputs are close and both branches receive part of the bias current the circuit acts like a **differential transconductance amplifier**. And if the two inputs are different enough that the entire bias current just flows through one branch, the circuit acts as a **comparator** that compares two analog inputs and turns them into a digital output. Comparators are for instance essential components of analog to digital converters (ADC).

If we consider its operation as a differential transconductance amplifier (making sure we are close to the correct large signal point of operation/biasing point, where each branch conducts exactly half of the bias current!) figure 10.2 shows the small signal equivalent. Note that the voltage controlled current source of the parallel transistors has been separated into two separate sources for the negative and positive contributions of the two control terminals (i.e. gate and source). As we do have two inputs now, let's look at two separate scenarios of correlated changes in these inputs, that we can linearly combine to reflect any change of the two inputs: Let's linearly separate all possible inputs into:

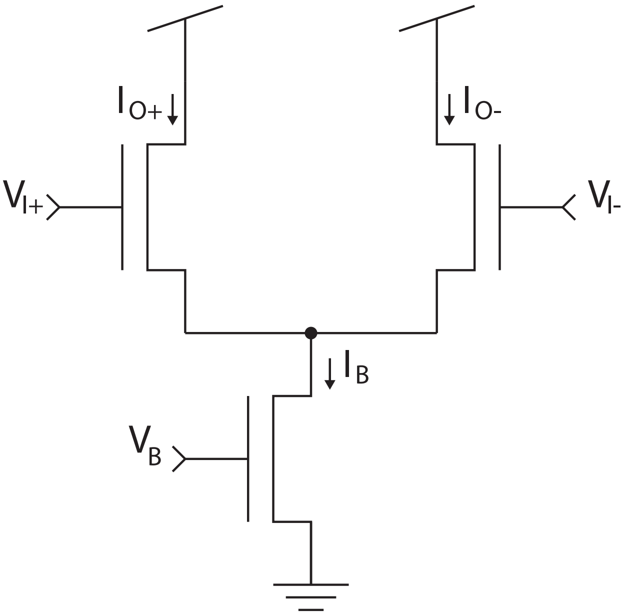


Figure 10.1: Differential pair basics with differential current output

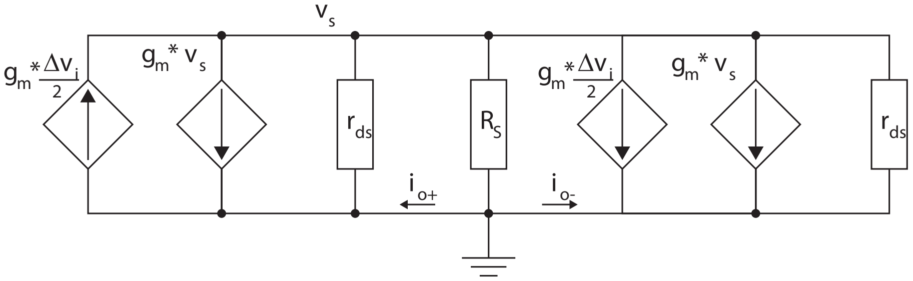


Figure 10.2: Small signal differential pair for balanced differential input Δv_i with differential current output.

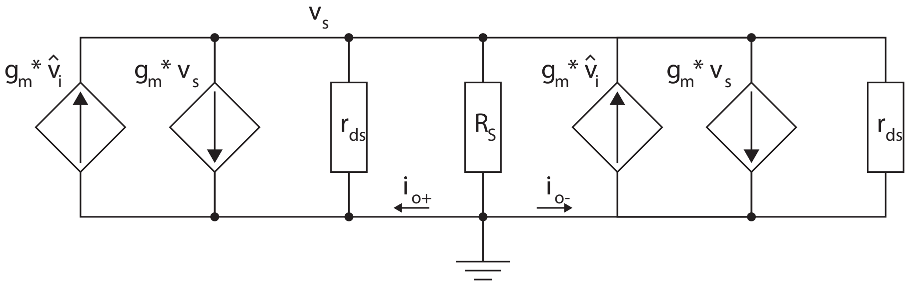


Figure 10.3: Small signal differential pair for common mode input Δv_i with differential current output.

$$\begin{aligned} v_{i+} &= \hat{v}_i + \frac{\Delta v_i}{2} \\ v_{i-} &= \hat{v}_i - \frac{\Delta v_i}{2} \end{aligned} \tag{10.1}$$

where we define the average of the input voltages as he **common mode input**

$$\hat{v}_i := \frac{v_{i+} + v_{i-}}{2} \tag{10.2}$$

and the difference of the input voltages as **differential input**

$$\Delta v_i := v_{i+} - v_{i-} \tag{10.3}$$

So in the manner of linear systems, we can investigate the effects of changes of Δv_i and $\hat{v}_i$ separately and treat arbitrary changes as a linear combination of the two separate effects.

Let's first consider what happens if only the difference of the inputs Δv_i changes but their average remains constant, i.e. the plus terminal receives plus half the difference and the negative terminal minus half the difference. Note in figure 10.2, instead of writing a minus sign at the negative terminal input the direction of the current has been inverted. And finally the resistor R_S is the output resistance of the bias current source, so in the case of a FET it will be that transistor's r_{ds}.

If we write down the Kirchhoff current equations for node v_s and for the two output currents i_{o+} and i_{o-} we get:

$$\begin{aligned} i_{o+} + v_s\left(g_m + \frac{1}{r_{ds}}\right) &= g_m\frac{\Delta v_i}{2} \\ i_{o-} + v_s\left(g_m + \frac{1}{r_{ds}}\right) &= -g_m\frac{\Delta v_i}{2} \\ v_s\left(2g_m + 2\frac{1}{r_{ds}} + \frac{1}{R_S}\right) &= \frac{\Delta v_i}{2}(g_m - g_m) = 0 \end{aligned} \tag{10.4}$$

You notice that the current being balanced that way means that there the only small signal current flowing into node v_s is the two differential current sources that cancel each other out, so v_s will actually be constant in this scenario, i.e. $v_s = 0$ (compare the last line of (10.4)). We get the result for the differential output current by subtracting the first two lines of (10.4).

$$\begin{aligned} \left(i_{o+} + v_s\left(g_m + \frac{1}{r_{ds}}\right)\right) - \left(i_{o-} + v_s\left(g_m + \frac{1}{r_{ds}}\right)\right) &= 2g_m\frac{\Delta v_i}{2} \\ \Delta i_o &= g_m \Delta v_i \end{aligned} \tag{10.5}$$

So at first glance we do get differential output current $\Delta i_o := i_{o+} - i_{o-}$ that is purely dependent on the differential input voltage. Right...? Well, not quite! In fact one has to go back to the large signal world in order to understand why. Let's for now stay with the small signal model though and have a first look at what happens if one changes the **common mode input voltage**, which we define as the average of the two inputs $\hat{v}_i := \frac{v_{i+}+v_{i-}}{2}$, the output currents are changed together in the same direction as well. See figure 10.3. The Kirchhoff current equations in this case are:

$$\begin{aligned} i_{o+} = i_{o-} = \hat{v}_i g_m - v_s\left(g_m + \frac{1}{r_{ds}}\right) \\ v_s\left(2g_m + 2\frac{1}{r_{ds}} + \frac{1}{R_S}\right) = 2\hat{v}_i g_m \end{aligned} \tag{10.6}$$

Resulting in

$$i_{o+} = i_{o-} = \hat{v}_i \frac{\frac{g_m}{R_S}}{2g_m + 2\frac{1}{r_{ds}} + \frac{1}{R_S}} \tag{10.7}$$

In the small signal world, however, this does not affect the differential output current at all because of the superposition principle of linear systems: one can change $\hat{v}_i$ and Δv_i separately or together and the result at the output is just a superposition of the individual effects. So the differential output result will still not change even if $\hat{v}_i$ is different.

However, looking at the situation with the large signal spectacles, changing the output currents together in the same direction means changing the voltage V_S and thus the bias current I_B. Remember that bias parameters are those that set the point of operation aka biasing point. The effect here will be that the small signal parameters have to be re-evaluated: in fact g_m depends on the bias current and will change! So the result of a change of $\hat{V}_I$ (note the use of the large signal parameter here!) will lead to a change in transconductance, i.e. will change the gain of the differential pair. These common mode gain errors are NOT COVERED by the usual derivation of the widely used **common mode rejection ratio (CMRR)** which we will discuss later.

10.1.1 R-load gain

A first extension that allows to read out a differential voltage from the differential output currents of the differential pair can be achieved by adding resistive loads to both branches (figure 10.4).

Deriving the Kirchhoff equations yields:

$$\begin{aligned} v_{o+}(\frac{1}{R_L} + \frac{1}{r_{ds}}) - g_m * \frac{\Delta v_i}{2} + g_m v_s = v_s\frac{1}{r_{ds}} \\ v_{o-}(\frac{1}{R_L} + \frac{1}{r_{ds}}) + g_m * \frac{\Delta v_i}{2} + g_m v_s = v_s\frac{1}{r_{ds}} \\ v_s(\frac{1}{R_S} + \frac{2}{r_{ds}}) = v_{o+}\frac{1}{r_{ds}} + v_{o-}\frac{1}{r_{ds}} \end{aligned} \tag{10.8}$$

And solving them (one can for example again use the shortcut that $v_s = 0$ for balanced differential input and subtract the first two lines):

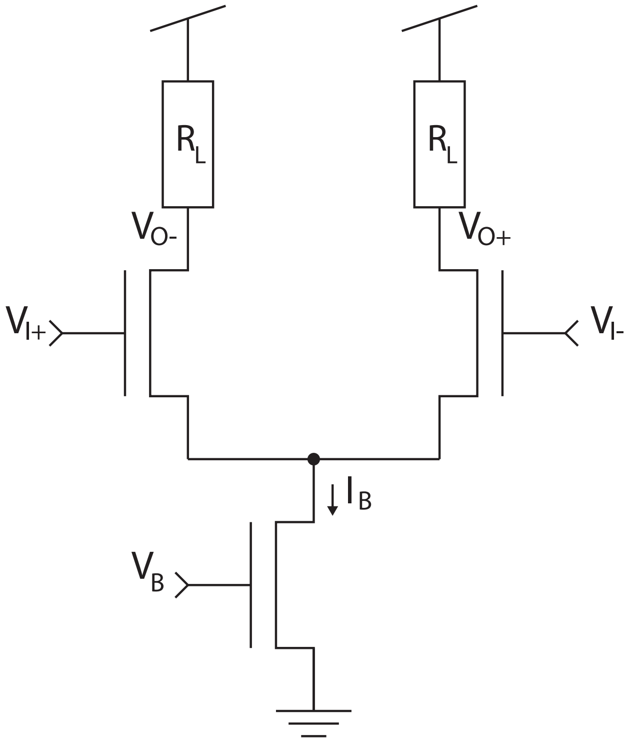


Figure 10.4: Differential pair with resistors as load on each branch and thus differential voltage output.

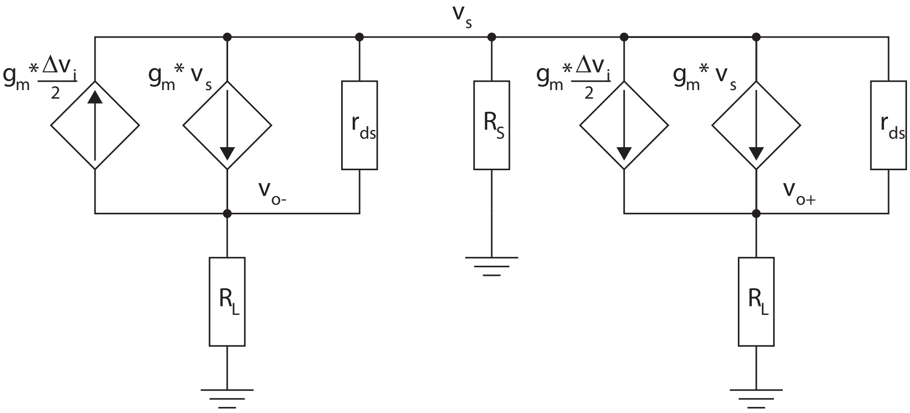


Figure 10.5: Small signal differential pair for balanced input Δv_i with load resistors and differential voltage output.

$$\begin{aligned} (v_{o+} - v_{o-})(\frac{1}{R_L} + \frac{1}{r_{ds}}) - 2g_m * \frac{\Delta v_i}{2} &= 0 \\ \frac{\Delta v_o}{\Delta v_i} &= g_m(R_L||r_{ds}) \end{aligned} \tag{10.9}$$

We will spare ourselves with the detailed analysis of the effects of the common mode input $\hat{v}_i$: by the sheer symmetry of the schematics it is kind of trivial that when changing both inputs, one will change the output voltages but crucially: *not their difference.* So like before one need not worry about effects of the common mode input in the *small signal world*, but again when looking at large changes of the average input voltages, one needs to consider a different point of operation, and ergo a different bias current and g_m, and consequently the *gain* will change in dependence of the large signal average input voltage level!

10.1.2 Current mirror loaded differential pair gain

Maybe the most common basic way of implementing a differential amplifier with a single ended output uses the differential pair with a current mirror (Fig. 10.6). The current mirror mirrors the output current from the left branch of the differential pair into the right branch from the top. So the right branch drain voltage is pulled high, when the left branch wins and gets virtually all the bias current. Vice versa if the left branch gets all the current the current mirror injects no current at all and the right hand side drain voltage is pulled low. And in the interesting case where the input voltages are close enough to each other and *both* branches get a share in the bias current, the circuit behaves like an amplifier, i.e. the difference of their shares is conveyed to the output. Then, with the circuit's output resistance and possibly a load resistance in parallel, that differential current is turned into a proportional output voltage.

The small signal equivalent circuit is depicted in figure 10.7 which gives rise to the Kirchhoff current equations in (10.10)

$$\begin{aligned} v_d(\frac{1}{r_{dsn}} + \frac{1}{r_{dsp}} + g_{mp}) + \frac{\Delta v_i}{2} g_{mn} &= v_s(g_{mn} + \frac{1}{r_{dsn}}) \\ v_s(2\frac{1}{r_{dsn}} + 2g_{mn} + \frac{1}{R_S}) &= v_d \frac{1}{r_{dsn}} + v_o \frac{1}{r_{dsn}} \\ v_o(\frac{1}{r_{dsn}} + \frac{1}{r_{dsp}}) + v_d g_{mp} &= v_s(g_{mn} + \frac{1}{r_{dsn}}) + \frac{\Delta v_i}{2} g_{mn} \end{aligned} \tag{10.10}$$

Solving this completely[1] results in equation 10.11:

$$\frac{v_o}{\Delta v_i} = g_{mn}\left(r_{dsn}||r_{dsp}\right) \frac{2g_{mp}g_{mn} + 2\frac{g_{mp}}{r_{dsn}} + \frac{g_{mp}}{R_S} + \left(\frac{g_{mn}}{r_{dsp}} + \frac{1}{r_{dsn}r_{dsp}} + \frac{1}{2r_{dsn}R_S} + \frac{1}{2r_{dsp}R_S}\right)}{2g_{mn}g_{mp} + 2\frac{g_{mp}}{r_{dsn}} + \frac{g_{mp}}{R_S} + 2\left(\frac{g_{mn}}{r_{dsp}} + \frac{1}{r_{dsp}r_{dsn}} + \frac{1}{2r_{dsn}R_S} + \frac{1}{2r_{dsp}R_S}\right)} \tag{10.11}$$

[1] Not something I recommend repeating ;-) I took me some days to do and recheck several times. I should definitely have gotten some symbolic math software, like Maple. Let me know if you find an error!

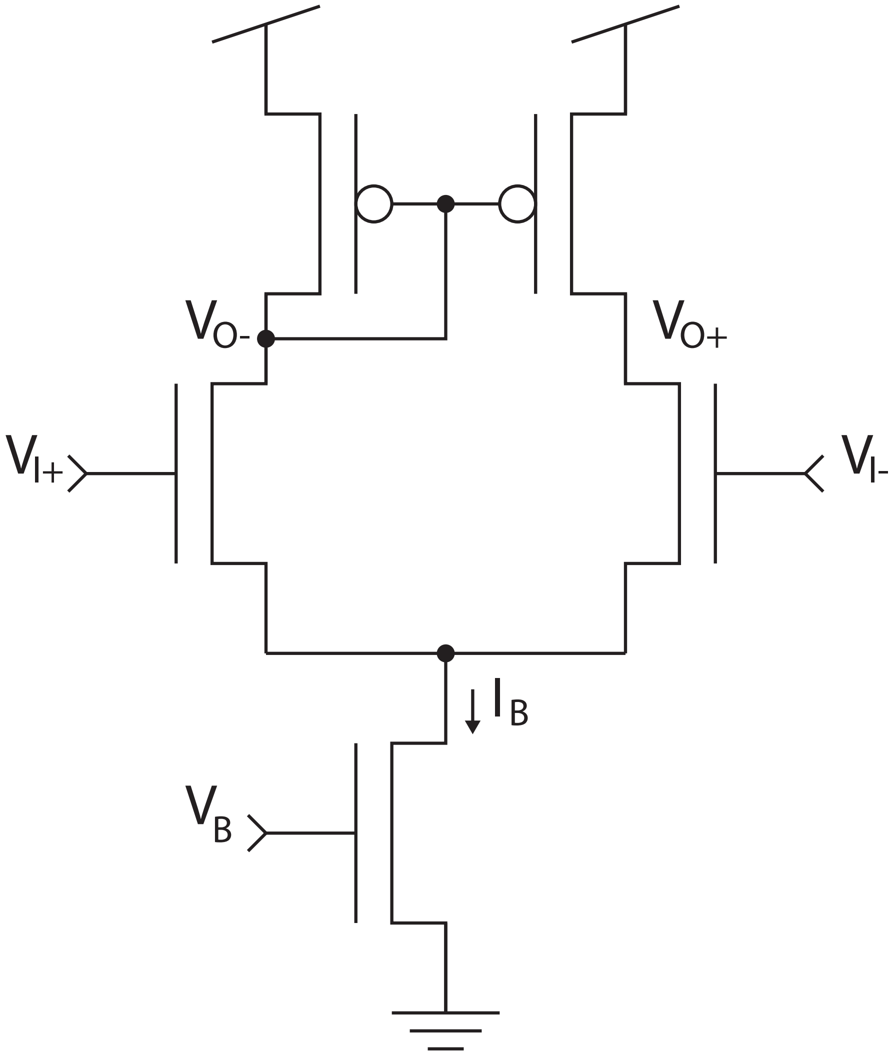


Figure 10.6: The differential pair with a current mirror load, also known as basic differential transconductance amplifier. It mirrors the left hand side current to the right hand side branch. Thus the current in the branches will compete against each other in the output node, either pulling the output high or low.

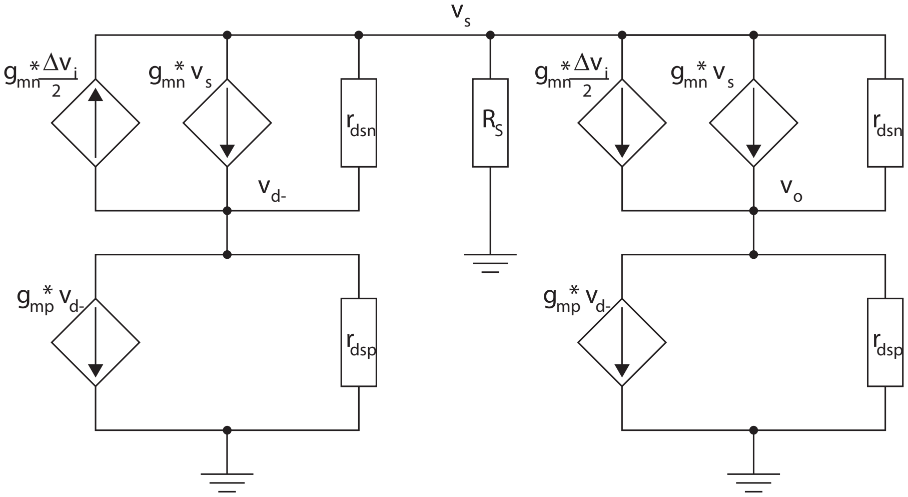


Figure 10.7: The small signal equivalent model for a differential pair with a current mirror load that mirrors the left hand side current to the right hand side branch (Fig. 10.6). Note that the branches are 'folded down' where the lower Gnd terminals correspond to the Vdd terminals in figure 10.6.

The large division term to the right is close to 1 but slightly smaller, if we assume that $\frac{1}{R_S} \approx \frac{1}{r_{dsn}} \approx \frac{1}{r_{dsp}} << g_{mn} \approx g_{mp}$. That's a reasonable assumption if we start from assuming that a basic CS gain stage biased the same as the transistors in the differential transconductance amplifier gets us a gain of, for example, at least 10, then $g_m r_{ds} > 10 \Rightarrow \frac{g_m}{10} > \frac{1}{r_{ds}}$. So if we assume that all resistors are the same and that g_m is at least 10 times larger than $\frac{1}{r_{ds}}$ then (10.11) gives us:

$$\begin{aligned} g_{mn}\left(r_{dsn}||r_{dsp}\right) > \frac{v_o}{\Delta v_i} &> g_{mn}\left(r_{dsn}||r_{dsp}\right)\frac{200+40+2}{200+50+4} = g_{mn}\left(r_{dsn}||r_{dsp}\right) * 0.953 \\ &\Rightarrow \\ \frac{v_o}{\Delta v_i} &\approx g_{mn}\left(r_{dsn}||r_{dsp}\right) \end{aligned} \tag{10.12}$$

So for all practical purposes, we shall assume that the gain is exactly the same as for a CS gain stage: $\frac{v_o}{\Delta v_i} \approx g_{mn}\left(r_{dsn}||r_{dsp}\right)$. Note that any external load R_L would need to be added in parallel with $(r_{dsn}||r_{dsp})$!

10.1.3 Differential transamp G_M and R_O deduced separately

An alternative deduction of the same result for the gain is to deduce expressions for G_M and R_O separately first and then to multiply them to get $A = G_M R_O$. This approach is also documented here, since it might come in handy to also know the parameters G_M and R_O in other contexts.

Let's first deduce G_M. From figure 10.8:

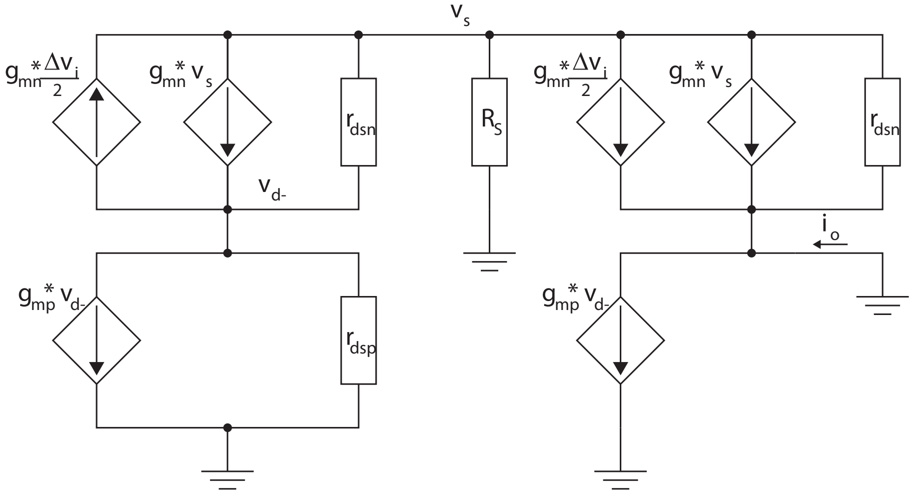


Figure 10.8: The small signal equivalent model for a differential pair with a current mirror load (Differential Transconductance Amplifier), with balanced differential input and the output at a fixed voltage to deduce the total transconductance G_M. Giving rise to (10.13).

$$\begin{aligned} v_d(\frac{1}{r_{dsp}} + \frac{1}{r_{dsn}} + g_{mp}) + \frac{\Delta v_i}{2} g_{mn} &= v_s(\frac{1}{r_{dsn}} + g_{mn}) \\ v_s(2g_{mn} + 2\frac{1}{r_{dsn}} + \frac{1}{R_S}) &= v_d \frac{1}{r_{dsn}} \\ v_d g_{mp} &= v_s(\frac{1}{r_{dsn}} + g_{mn}) + \frac{\Delta v_i}{2} g_{mn} + i_o \end{aligned} \tag{10.13}$$

Solving this yields:

$$\begin{aligned} G_M &= \frac{i_o}{\Delta v_i} \\ &= -g_{mn} \frac{2g_{mn}g_{mp} + 2\frac{g_{mp}}{r_{dsn}} + \frac{g_{mp}}{R_S} + \left(\frac{g_{mn}}{r_{dsp}} + \frac{1}{r_{dsn}r_{dsp}} + \frac{1}{r_{dsn}2R_S} + \frac{1}{r_{dsp}2R_S}\right)}{2g_{mn}g_{mp} + 2\frac{g_{mp}}{r_{dsn}} + \frac{g_{mp}}{R_S} + 2\left(\frac{g_{mn}}{r_{dsp}} + \frac{1}{r_{dsn}r_{dsp}} + \frac{1}{r_{dsn}2R_S} + \frac{1}{r_{dsp}2R_S}\right) + \frac{g_{mn}}{r_{dsn}} + \frac{1}{r_{dsn}^2}} \end{aligned} \tag{10.14}$$

As the division term at the end is only slightly smaller and very close to 1, this expression is usually very close to just

$$G_M \approx -g_m \tag{10.15}$$

For example if we again assume that the transistors are biased to obtain a decent intrinsic gain $g_m r_{dsn} > 10$ and that all resistors $r_{dsn} = r_{dsp} = R_S$ are the same, as well as the transconductances $g_{mn} = g_{mp}$ we get:

$$-g_m < G_M < -g_m * \frac{200 + 40 + 2}{200 + 60 + 5} = -g_m * 0.91 \tag{10.16}$$

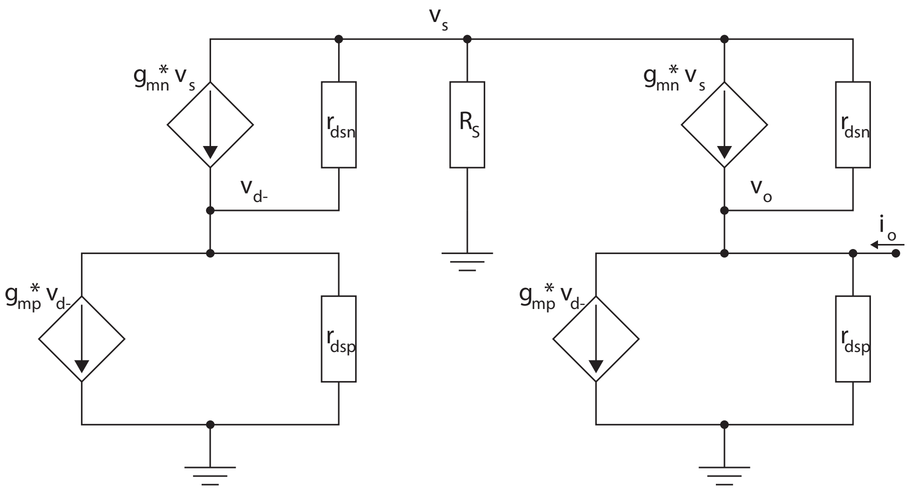


Figure 10.9: The small signal equivalent model for a differential pair with a current mirror load (Differential Transconductance Amplifier), with the input at a fixed voltage to deduce the total output resistance RO. Giving rise to (10.17).

So for practical purposes we will usually just use (10.15).
And then R_O. From figure 10.9:

$$\begin{aligned} v_d(\frac{1}{r_{dsp}} + \frac{1}{r_{dsn}} + g_{mp}) &= v_s(\frac{1}{r_{dsn}} + g_{mn}) \\ v_s(2g_{mn} + 2\frac{1}{r_{dsn}} + \frac{1}{R_S}) &= v_d\frac{1}{r_{dsn}} + v_o\frac{1}{r_{dsn}} \\ v_o(\frac{1}{r_{dsn}} + \frac{1}{r_{dsp}}) + v_d g_{mp} &= v_s(\frac{1}{r_{dsn}} + g_{mn}) + i_o \end{aligned} \tag{10.17}$$

Solving this yields:

$$\begin{aligned} R_O &= \frac{v_o}{i_o} \\ &= (r_{dsn}||r_{dsp})\frac{2g_{mp}g_{mn} + 2\frac{g_{mn}}{r_{dsp}} + \frac{g_{mp}}{R_S} + 2(\frac{g_{mp}}{r_{dsn}} + \frac{1}{r_{dsn}r_{dsp}} + \frac{1}{r_{dsp}2R_S} + \frac{1}{r_{dsn}2R_S}) + \frac{g_{mn}}{r_{dsn}} + \frac{1}{r_{dsn}^2}}{2g_{mp}g_{mn} + 2\frac{g_{mn}}{r_{dsp}} + \frac{g_{mp}}{R_S} + 2(\frac{g_{mp}}{r_{dsn}} + \frac{1}{r_{dsn}r_{dsp}} + \frac{1}{r_{dsp}2R_S} + \frac{1}{r_{dsn}2R_S})} \end{aligned} \tag{10.18}$$

Once more the ratio at the end is an expression very close to 1, this time slightly bigger.

$$R_O \approx (r_{dsn}||r_{dsp}) \tag{10.19}$$

Again with the assumption of a decent intrinsic gain of $g_m r_{dsn} > 10$ and all resistors $r_{dsn} = r_{dsp} = R_S$ and transconductances $g_{mn} = g_{mp}$ being the same we get:

$$(r_{dsn}||r_{dsp}) < R_O < \frac{200+60+3}{200+50+2}(r_{dsn}||r_{dsp}) = 1.04(r_{dsn}||r_{dsp}) \tag{10.20}$$

So for any practical purpose we will usually just use (10.19)

Combining the exact results (10.14) and (10.18) we confirm the result in (10.11) and get the same expression:

$$\begin{aligned} A = -G_M * R_O = g_m(r_{dsn}||r_{dsp})* \\ \frac{2g_{mn}g_{mp} + 2\frac{g_{mp}}{r_{dsn}} + \frac{g_{mp}}{R_S} + \left(\frac{g_{mn}}{r_{dsp}} + \frac{1}{r_{dsn}r_{dsp}} + \frac{1}{r_{dsn}2R_S} + \frac{1}{r_{dsp}2R_S}\right)}{2g_{mp}g_{mn} + 2\frac{g_{mn}}{r_{dsn}} + \frac{g_{mp}}{R_S} + 2\left(\frac{g_{mp}}{r_{dsp}} + \frac{1}{r_{dsn}r_{dsp}} + \frac{1}{r_{dsp}2R_S} + \frac{1}{r_{dsn}2R_S}\right)} \end{aligned} \tag{10.21}$$

Again an expression close to:

$$A = -G_M R_O \approx g_m(r_{dsn}||r_{dsp}) \tag{10.22}$$

And once more for for example $g_m r_{dsn} > 10$ and all resistors $r_{dsn} = r_{dsp} = R_S$ and transconductances $g_{mn} = g_{mp}$ being the same we get:

$$g_m(r_{dsn}||r_{dsp}) * 0.953 < A < g_m(r_{dsn}||r_{dsp}) \tag{10.23}$$

10.1.4 Current mirror loaded differential pair common mode rejection ratio (CMRR)

With the current mirror load the circuit is no longer exactly symmetrical. Thus, it can be expected that a change in the common mode input voltage $\hat{v}_i$, while the input difference Δv_i remains constant, may actually cause the output to move (i.e. *also* already in the small signal world!!!). So if one does a complete analysis of the small signal equivalent circuit once again, but now with $\hat{v}_i$ as the input, one starts from the small signal model in figure 10.10.

The Kirchhoff current equations for the nodes are:

$$\begin{aligned} v_d(\frac{1}{r_{dsn}} + \frac{1}{r_{dsp}} + g_{mp}) + \hat{v}_i g_{mn} &= v_s(g_{mn} + \frac{1}{r_{dsn}}) \\ v_s(2\frac{1}{r_{dsn}} + 2g_{mn} + \frac{1}{R_S}) &= v_d\frac{1}{r_{dsn}} + v_o\frac{1}{r_{dsn}} + 2\hat{v}_i g_{mn} \\ v_o(\frac{1}{r_{dsn}} + \frac{1}{r_{dsp}}) + \hat{v}_i g_{mn} + v_d g_{mp} &= v_s(g_{mn} + \frac{1}{r_{dsn}}) \end{aligned} \tag{10.24}$$

The resulting common mode gain is:

$$\frac{v_o}{\hat{v}_i} = -\frac{\frac{g_{mn}}{R_S}}{2g_{mp}g_{mn} + 2\frac{g_{mp}}{r_{dsn}} + 2\frac{g_{mn}}{r_{dsp}} + \frac{g_{mp}}{R_S} + 2\frac{1}{r_{dsn}r_{dsp}} + \frac{1}{r_{dsn}R_S} + \frac{1}{r_{dsp}R_S}} \tag{10.25}$$

If we again make the assumption that the resistors r_{dsn}, r_{dsp} are equal and all transconductances g_{mn}, g_{mp} are equal and at least 10 times bigger than the inverted resistors (conductance of the resistors), then by only considering the

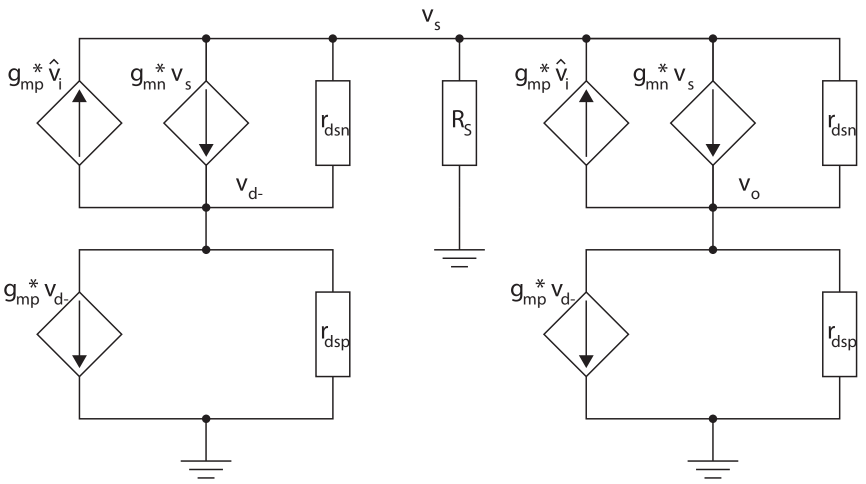


Figure 10.10: The small signal equivalent model for a differential pair with a current mirror load with common mode input $\hat{v}_i$.

term $2g_{mn}g_{mp}$ (which is at least 10 times bigger than any of the others) in the denominator one gets

$$\frac{v_o}{\hat{v}_i} \approx -\frac{1}{2R_S g_{mp}} \tag{10.26}$$

So if also R_S is approximately equal to the other drain source resistances, then the commoin mode gain is approximately equal to the inverse of (four times) the amplifiers differential gain.

The so called **common mode rejection ratio (CMRR)** is defined as the absolute ratio between the differential gain and the common mode gain:

$$\text{CMRR} := \left| \frac{\frac{v_o}{\Delta v_i}}{\frac{v_o}{\hat{v}_i}} \right| \tag{10.27}$$

If again we use the approximate terms we get:

$$\text{CMRR} \approx g_{mn}\left(r_{dsn} || r_{dsp}\right) 2g_{mp}R_S \tag{10.28}$$

10.1.5 Current mirror loaded differential pair frequency response

Figure 10.11 shows the small signal model of the current mirror loaded differential pair, now including the most prominent capacitances. Note that some capacitors are neglected. More precisely some terms as specified below are neglected in the following Kirchhoff equations. The capacitors that are included are:

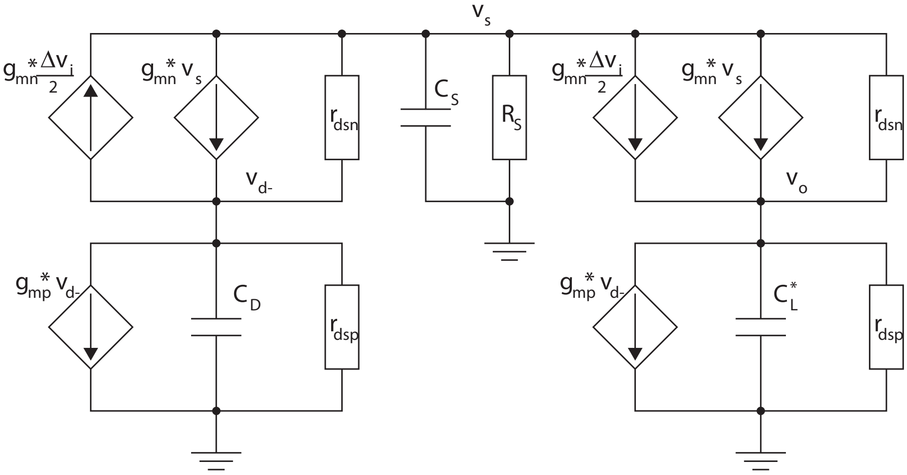


Figure 10.11: The small signal equivalent model for a differential pair with a current mirror load Including prominent capacitors for frequency analysis.

$$
\begin{aligned}
C_S &= 2(C_{GS_n} + C_{SB_n}) + C_{DB_n} + C_{GD_n} && (10.29)\\
C_D &= 2C_{GS_p} + C_{GD_p} + C_{DB_p} + C_{GD_n} + C_{DB_n} && (10.30)\\
C_L^* &= C_{DB_n} + +C_{DG_n} + C_{DB_p} + C_{DG_p} + C_{L_{\text{ext}}} && (10.31)
\end{aligned}
$$

$C_{L_{\text{ext}}}$ is a contribution from the actual load the circuit is connected to, so often of the order of magnitude of C_{GS}, if that circuit is another CMOS circuit of similar dimensions.

The capacitors that are omitted in figure 10.11 and thus the missing terms in the Kirchhoff equations are:

- forward effect of C_{GD_n} between positive input $\frac{\Delta v_i}{2}$ and v_{d-}: the term $sC_{DG_n}\frac{\Delta v_i}{2}$ will be omitted in the following, assuming $C_D >> C_{DG_n}$ it will not affect the voltage v_{d-} significantly.
- feedback effect of C_{GD_p} between v_{d-} and v_o: that is, we shall neglect the term $v_o C_{GD_p}$. This is a bit more daring, as there is some Miller effect to be considered here: v_{d-} is inverted from v_o and v_o is very much amplified. However, this amplification is likely small at the high frequencies where a pole at v_{d-} occurs (see in the following).
- The forward effect of C_{GD_n} and C_{GD_p} on node v_o. This is likely insignificant with external contribution to C_L^* by the circuit this amplifier is connected to.

We solve the problem here for ideal input voltage input signals. If, on the other hand, one wanted to consider the signal source having some output impedance, one can as a first approximation consider the input nodes v_i to have a static input capacitance of $C_{GS_n} + C_{GD_n}$ and thus compute a pole at the input. But once again: we do not do that here in the following, but consider ideal inputs.

Thus the resulting Kirchhoff equations are:

$$\begin{aligned} v_d(\frac{1}{r_{dsn}} + \frac{1}{r_{dsp}} + g_{mp} + sC_D) + \frac{\Delta v_i}{2} g_{mn} &= v_s(g_{mn} + \frac{1}{r_{dsn}}) \\ v_s(2\frac{1}{r_{dsn}} + 2g_{mn} + \frac{1}{R_S} + sC_S) &= v_d\frac{1}{r_{dsn}} + v_o\frac{1}{r_{dsn}} \\ v_o(\frac{1}{r_{dsn}} + \frac{1}{r_{dsp}} + sC_L^*) + v_d g_{mp} &= v_s(g_{mn} + \frac{1}{r_{dsn}}) + \frac{\Delta v_i}{2} g_{mn} \end{aligned} \tag{10.32}$$

But lets go for G_M and Z_O separately, Z_O being what we previously called R_O but which is now an impedance rather than just a resistance.

The Kirchhoff equations to solve for G_M with the small signal output shorted to Gnd are:

$$\begin{aligned} v_d(\frac{1}{r_{dsn}} + \frac{1}{r_{dsp}} + g_{mp} + sC_D) + \frac{\Delta v_i}{2} g_{mn} &= v_s(g_{mn} + \frac{1}{r_{dsn}}) \\ v_s(2\frac{1}{r_{dsn}} + 2g_{mn} + \frac{1}{R_S} + sC_S) &= v_d\frac{1}{r_{dsn}} \\ v_d g_{mp} &= i_o + v_s(g_{mn} + \frac{1}{r_{dsn}}) + \frac{\Delta v_i}{2} g_{mn} \end{aligned} \tag{10.33}$$

Resulting in:

$$\frac{i_o}{\Delta v_i} = G_M = -g_{mn} \frac{\left(g_{mn} + \frac{1}{r_{dsn}} + \frac{1}{2R_S} + \frac{sC_S}{2}\right)\left(2g_{mp} + \frac{1}{r_{dsp}} + \frac{1}{r_{dsn}} + sC_D\right) - \frac{1}{r_{dsn}}\left(\frac{1}{r_{dsn}} + g_{mn}\right)}{\left(2g_{mn} + 2\frac{1}{r_{dsn}} + \frac{1}{R_S} + sC_S\right)\left(g_{mp} + \frac{1}{r_{dsp}} + \frac{1}{r_{dsn}} + sC_D\right) - \frac{1}{r_{dsn}}\left(\frac{1}{r_{dsn}} + g_{mn}\right)} \tag{10.34}$$

And doing the Kirchhoff equations for Z_O with the small signal input being 0:

$$\begin{aligned} v_d(\frac{1}{r_{dsn}} + \frac{1}{r_{dsp}} + g_{mp} + sC_D) &= v_s(g_{mn} + \frac{1}{r_{dsn}}) \\ v_s(2\frac{1}{r_{dsn}} + 2g_{mn} + \frac{1}{R_S} + sC_S) &= v_d\frac{1}{r_{dsn}} + v_o\frac{1}{r_{dsn}} \\ v_o(\frac{1}{r_{dsn}} + \frac{1}{r_{dsp}} + sC_L^*) + v_d g_{mp} &= v_s(g_{mn} + \frac{1}{r_{dsn}}) \end{aligned} \tag{10.35}$$

Solving for Z_O we get:

$$\begin{aligned} \frac{v_o}{i_o} &= Z_O = (r_{dsn}||r_{dsp})\frac{1}{1 + s(r_{dsn}||r_{dsp})C_L^*} \\ &* \frac{\left((\frac{1}{r_{dsp}} + \frac{1}{r_{dsn}} + g_{mp} + sC_D)(2g_{mn} + 2\frac{1}{r_{dsn}} + \frac{1}{R_S} + sC_S) - (\frac{1}{r_{dsn}} + g_{mn})\frac{1}{r_{dsn}}\right)}{\left((\frac{1}{r_{dsp}} + \frac{1}{r_{dsn}} + g_{mp} + sC_D)(2g_{mn} + 2\frac{1}{r_{dsn}} + \frac{1}{R_S} + sC_S) - 2(\frac{1}{r_{dsn}} + g_{mn})\frac{1}{r_{dsn}}\right)} \end{aligned} \tag{10.36}$$

Together this is:

$$\frac{v_o}{\Delta v_i} = -G_M Z_O = g_{mn}(r_{dsn}||r_{dsp}) \frac{1}{1+s(r_{dsn}||r_{dsp})C_L^*}$$
$$* \frac{\left(2g_{mp} + \frac{1}{r_{dsp}} + \frac{1}{r_{dsn}} + sC_D\right)\left(g_{mn} + \frac{1}{r_{dsn}} + \frac{1}{2R_S} + \frac{sC_S}{2}\right) - \left(g_{mn} + \frac{1}{r_{dsn}}\right)\frac{1}{r_{dsn}}}{\left(g_{mp} + \frac{1}{r_{dsp}} + \frac{1}{r_{dsn}} + sC_D\right)\left(2g_{mn} + 2\frac{1}{r_{dsn}} + \frac{1}{R_S} + sC_S\right) - 2\left(g_{mn} + \frac{1}{r_{dsn}}\right)\frac{1}{r_{dsn}}} \tag{10.37}$$

Making the same assumptions again that $g_m r_{ds} >> 1$ is orders of magnitude bigger than 1, we neglect everything except terms of the order g_m^2 or sCg_m. We get:

$$\frac{v_o}{\Delta v_i} = -G_M Z_O \approx g_{mn}(r_{dsn}||r_{dsp}) \frac{1}{1+s(r_{dsn}||r_{dsp})C_L^*} \frac{\left(1+s\frac{C_D}{2g_{mp}}\right)}{\left(1+s\frac{C_D}{g_{mp}}\right)} \tag{10.38}$$

So we have the pole frequency ω_{p_1} resulting from the output impedance Z_O:

$$\omega_{p_1} = \frac{1}{(r_{dsn}||r_{dsp})C_L^*} \tag{10.39}$$

It is quite easily the dominant pole, since the other pole frequency ω_{p_2} (resulting from the expression for G_M) is close to the maximum frequency one of the diff pair transistors can convey under ideal circumstances, the intrinsic speed, i.e. the current unity gain/transition frequency $\omega_T = \frac{g_m}{C_{GS}+C_{GD}}$ (see section 9.4.3, equation(9.32)) (here: $\omega_T = \frac{g_m}{C_L^*}$?):

$$\omega_{p_2} = \frac{g_{mp}}{C_D} \tag{10.40}$$

The sole zero frequency ω_{z_1} also results from the G_M term and occurs at double the frequency:

$$\omega_{z_1} = 2\omega_{p_2} = \frac{2g_{mp}}{C_D} \tag{10.41}$$

The combined effect of these two later terms is that the output current starts to fall proportional with the frequency with an 1:1 ratio or $20\frac{\text{dB}}{\text{dec}}$ at ω_{p_2} and stops at double that frequency at $\omega_{z_1} = 2\omega_{p_2}$. Consequently, $|G_M|$ and thus the output current is *halved* at that point. The intuition here is that the current mirror goes into cut off, so only the output current from the right diff pair branch drives the voltage gain. But once again: this usually happens at way higher frequencies than ω_{p_1}, i.e. beyond the circuit's unity gain frequency.

Index

Bibliography

[1] T. C. Carusone, D. Johns, and K. Martin. *Analog Integrated Circuit Design, international student version*. Wiley, 2013.

[2] A. S. Sedra and K. C. Smith. *Microelectronic Circuits*. Oxford University Press, international 7th edition, 2016.

[3] E. Vittoz. Analog VLSI signal processing: Why, where and how? *Analog Integrated Circuits and Signal Processing*, pages 27–44, July 1994.

Appendices

Appendix A

Exercises

This collection of former exam tasks is still quite limited in the bata edition 0.1! More tasks will appear in the first edition, expected summer 2024. Be on the lookout for it!

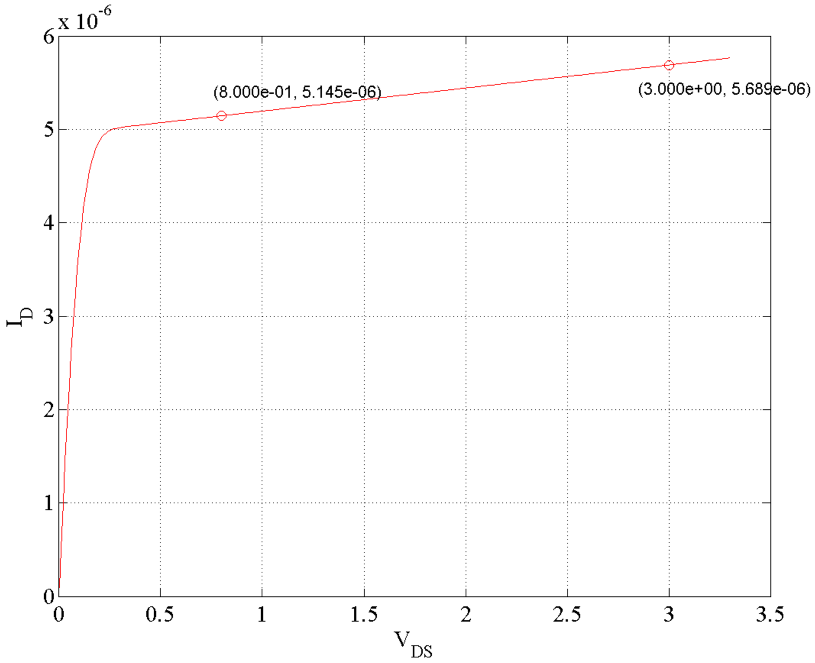


Figure A.1: Exercise: compute λ from I_D vs V_{DS} graph

A.1 Transistor Models

A.1.1

From the I_D vs V_{DS} graph in figure A.1 for a specific nFET with a specific V_{GS}, compute parameter λ.

A.1.2

From I_D vs V_{DS} graph in figure A.2 for a specific nFET with a specific V_{GS}, compute parameter r_{ds} near the origin, i.e. in deep conduction.

A.1.3

An nFET is to be operated in the saturation region at a current of 20μA. It's $k'_n \frac{W}{L} = 70 \frac{\mu \mathrm{A}}{V^2}$ and its threeshold voltage $V_{tn} = 650$mV. (You may assume parameter $\lambda = 0$ and $n = 1$). What is its required gate to source voltage $V_{GS} =$? and minimum drain to source voltage $V_{DS,min} = V_{sat} =$?

A.1.4

Make a matlab function 'function '[Vd]=CMOSdiode(Id,kn,Vtn)' that computes the drain voltage Vd (V_D) on a diode connected nMOSFET (i.e. drain and gate are shorted together) as a function of the input current Id (I_D) at the drain in weak inversion (!!!), and given parameters kn, Vtn (i.e. the MOSFET

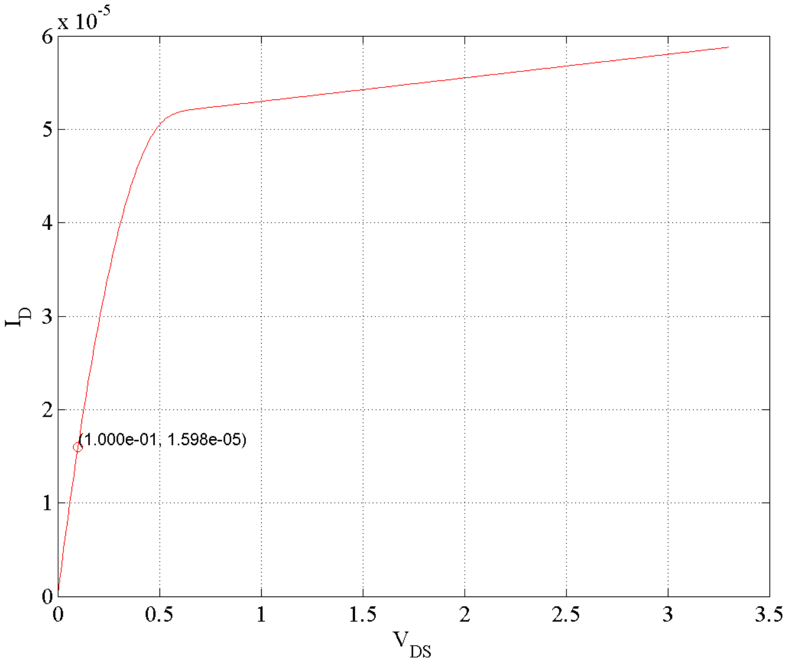


Figure A.2: Exercise: compute r_{ds} near origin from I_D vs V_{DS} graph

transconduction paramter k_n and the threshold voltage V_{tn}) . Ignore channel length modulation $\lambda = 0$ and slope factor $n = 1$. The thermal voltage U_T shall be 26mV. Assume the source voltage $V_S = V_B = 0$.

Also assume that the transistor is in saturation. Could you find out from the result, if that uis actually a correct assumption?

A.1.5

For the schematics in figure A.3 perform a small signal (!) analysis and find an expression for the total output resistance $R_O = \frac{v_o}{v_i}$ and the total transconductance $G_M = \frac{i_o}{v_i}$ and the gain $A = \frac{v_o}{v_i}$! To form your expressions you may use the small signal variables g_m, r_{ds} of the transistor as well as the resistors R_1, R_2. In other words express these terms as functions of those four variables (or a subset of them).

A.1.6

Given process parameters: $\mu_n C_{ox} = 280 * 10^{-6} \frac{\mathrm{A}}{\mathrm{V}^2}$, $\frac{1}{\lambda * L} = \frac{V_A}{L} = 5 * 10^6 \frac{\mathrm{V}}{\mathrm{m}}$, $V_{tn} = 0.5\mathrm{V}$, for an NFET of length $L = 0.4\mu\mathrm{m}$, $V_{ov} = 0.15\mathrm{V}$, and operated at a drain current $I_D = 80\mu\mathrm{A}$, find transconductance, drain resistance in the active/saturation region, intrinsic gain and transistor width: g_m, r_{ds}, A_0, W!

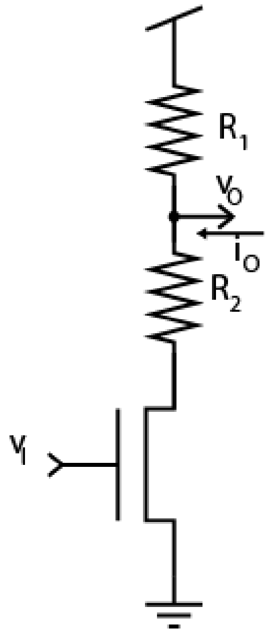


Figure A.3: Schematic for small signal analysis

A.1.7

Make a matlab function 'function '[Vd]=CMOSdiode(Id,kn,Vtn)' that computes the drain voltage Vd (i.e. V_D in the compendium) on a diode connected nMOSFET (i.e. drain and gate are shorted together) as a function of the input current Id (i.e. I_D in the compendium) at the drain in weak inversion (!!!), and given parameters kn (i.e. the MOSFET transconduction paramter k_n), Vtn (i.e. the threshold voltage V_{tn}) . Ignore λ (i.e. $\lambda = 0$) and V_A (the Early effect, i.e. $V_A = \infty$) and n (the slope factor from the EKV model) and you may assume that the transistor is in saturation. The thermal voltage U_T shall be 26mV. Assume the source voltage $V_S = 0$ and that all voltages are referenced to the bulk $V_B := 0$.

Solutions section A.1

Solution A.1.1

r_{ds} is the inverse of the slope in the saturation region, i.e. the inverse of the small signal conductance g_{ds} and λ is simply the conductance divided by the saturation drain current I'_D neglecting λ. Since we do have two points of that slope we can deduce:

$$g_{ds} = \frac{I_{D2} - I_{D1}}{V_{DS2} - V_{DS1}} = (5.689e-6-5.145e-6)/(3.0-0.8) = 0.2473e-6 \quad \text{(A.1)}$$

One could eyeball $I'_D \approx 5e-6$, but more precisely:

$$I'_D = I_{D1} - V_{DS1} * g_{ds} = 5.145e-6 - 0.8 * 0.2473e-6 = 4.95e-6 \tag{A.2}$$

$$\begin{aligned} I_{D1} &= I'_D(1 + V_{DS1} * \lambda) \;\; 1) \\ I_{D2} &= I'_D(1 + V_{DS2} * \lambda) \;\; 2) \\ &\Rightarrow 2) - 1) \\ I_{D2} - I_{D1} &= I'_D(V_{DS2} - V_{DS1}) * \lambda \\ \Rightarrow \lambda &= \frac{I_{D2} - I_{D1}}{I'_D(V_{DS2} - V_{DS1})} = \frac{g_{ds}}{I'_D} \\ &= 0.2473e-6/4.95e-6 = 0.05\frac{1}{\mathrm{V}} \end{aligned} \tag{A.3}$$

Solution A.1.2

As the point labelled on the curve is quite close to the origin one can simply approximate:

$$r_{ds} \approx \frac{V_{DS1}}{I_{D1}} = 0.1/15.98e-6 = 6.26\mathrm{k}\Omega \tag{A.4}$$

Which is slightly overestimated as the curve is marginally steeper right at the origin.

Solution A.1.3

Assuming strong inversion (needs to be verified in the end) we can write

$$V_{ov} = \sqrt{\frac{2I_D}{k_n}} = \sqrt{2 * 20e-6/70e-6} = 0.756\mathrm{V} \tag{A.5}$$

Checking for strong inversion: V_{ov} is of the same order of magnitude as V_t. If it would be orders of magnitude smaller, the result could be wrong! In that case we would likely be in weak inversion. However, here we seem to are safe.

$V_{GS} = 1.406\mathrm{V}$
$V_{sat} = 0.756\mathrm{V}$

Solution A.1.4

In saturation with $\lambda = 0$ the current does not depend on V_{DS} at all, so one simply needs to find the correct V_G.

```
function '[Vd]=CMOSdiode(Id,kn,Vtn)
VT=26e-3 ;%V
Is=2*kn*VT^2;
% Id=Is*exp((Vg-Vtn)/VT) % solve for Vg
% log(Id)=log(Is)+(Vg-Vtn)/VT
Vg=VT*(log(Id)-Log(Is))+Vtn;
```

A diode connected CMOS transistor in strong inversion (so not here) is except for very artificial scenarios actually guaranteed to be in saturation, as $V_{sat} < V_{GS}$. So by shorting $V_{DS} = V_{GS}$, V_{DS} will always be big enough. Here in weak inversion however, for very small input currents assuming saturation the resulting V_{GS} might turn out to be smaller than $4U_T$. Then one would actually need to solve the problem using the conduction rules or full EKV and the real $V_{GS} = V_{DS}$ will turn out to be bigger.

Solution A.1.5

$R_O = (r_{ds} + R_2)||R_1 = \frac{(r_{ds}+R_2)R_1}{R_1+R_2+r_{ds}}$
$G_M = g_m \frac{r_{ds}}{r_{ds}+R_2}$ $A = G_M R_O = g_m \frac{R_1 r_{ds}}{R_1+R_2+r_{ds}}$

Solution A.1.6

With given positive V_{ov} away from moderate inversion we can safely assume strong inversion. Thus:

$$\begin{aligned}
W &= \frac{2I_D L}{V_{ov}^2 \mu C_{ox}} = 2*80e-6*0.4e-6/(0.15^2*280e-6) = 10.16\mu\text{m} \\
g_m &= \sqrt{2I_D k_n} = \sqrt{2*80e-6*280e-6*10.16e-6/0.4e-6} = 1.07\frac{mA}{V} \\
r_{ds} &= \frac{1}{I_D \lambda} = 1/(80e-6*1/5e6/0.4e-6) = 25\text{k}\Omega \\
A_0 &= g_m r_{ds} = 1.07e-3*25e3 = 26.75\frac{\text{V}}{\text{V}}
\end{aligned} \tag{A.6}$$

Solution A.1.7

The assumption that we are in weak inversion and saturation allows to use the simplified equation for that region of operation. However, if the input current 'Id' is too small or too large in relation to the other parameters, these assumptions may actually turn out to be wrong: if 'Id' is too small the resulting 'Vd' may be smaller than $4U_T$ and thus the transistor is actually in conduction and the result is wrong. Or if 'Id' is too large, it might be bigger than I_S and thus the transistor would really be in strong inversion. So the following matlab program only returns the correct results for certain inputs. But since the tasks specifies that one can assume weak inversion and saturation, it is the correct answer.

```
function [Vd]=CMOSdiode(Id,kn,Vtn)
VT=26e-3 ;%V
Is=2*kn*VT^2;%A
% Id=Is*exp((Vg-Vtn)/VT) % solve for Vg
% log(Id)=log(Is)+(Vg-Vtn)/VT
Vg=VT*(log(Id)-Log(Is))+Vtn;
```

To write down the underlying symbolic solution:

$$
\begin{aligned}
I_S &= 2k_n U_T^2 \\
V_G &= U_T(\ln I_D - \ln I_S) + V_{tn}
\end{aligned}
\tag{A.7}
$$

A.2 Linear Circuit Analysis Basics

A.2.1

Draw the Bode plots (both magnitude and phase!) for the following two transfer functions A_1 and A_2. The x-axis should extend at least 1 decade beyond the highest pole or zero frequency. Use a grid like the one in the illustration A.4! Draw the plots as 'piece-wise linear' approximation of the real graphs, like on the lecture slides: for phase transitions use either ±one decade (like in FYS3220) or ±half a decade (like in the graphs on the lecture slides) around the respective pole or zero to complete the full phase shift. For cut-off frequencies in the magnitude plots make an abrupt transition/knee point at the respective pole or zero to change the slope of the magnitude.

$$
A_1 = A_{DC1} \frac{1 + \frac{s}{\omega_{z11}}}{\left(1 + \frac{s}{\omega_{p11}}\right)\left(1 + \frac{s}{\omega_{p12}}\right)} \tag{A.8}
$$

where $A_{DC1} = 10^3$, $\omega_{z11} = 10^6$rad, $\omega_{p11} = 10^3$rad, $\omega_{p12} = 10^8$rad.

$$
A_2 = A_{DC2} \frac{\left(1 + \frac{s}{\omega_{z21}}\right)\left(1 - \frac{s}{\omega_{z22}}\right)}{\left(1 + \frac{s}{\omega_{p21}}\right)\left(1 + \frac{s}{\omega_{p22}}\right)\left(1 + \frac{s}{\omega_{p22}}\right)} \tag{A.9}
$$

where $A_{DC1} = 10^4$, $\omega_{p21} = 10^3$rad, $\omega_{p22} = 10^7$rad, $\omega_{p23} = 10^9$rad, $\omega_{z21} = 10^5$rad, $\omega_{z22} = 10^7$rad

A.2.2

Consider the two Bode plots for transfer functions A1 and A2 in Fig. A.5. Find the corresponding transfer functions in root form and Implement the MATLAB function 'function [A1,A2]=transferF(s)' in file transferF.m which computes both transfer functions in dependency of $s = j * \omega$ (but we shall just use the variable s here).

Solutions section A.2

Solution A.2.1

See figure A.6.

Solution A.2.1

Note that in the following I use the opperands with a '.' in front, such as '.^' and '.*'. In matlab this indicates that the operation should be done element-wise on vectors. This allows variable s to be a vector of values and the returned values

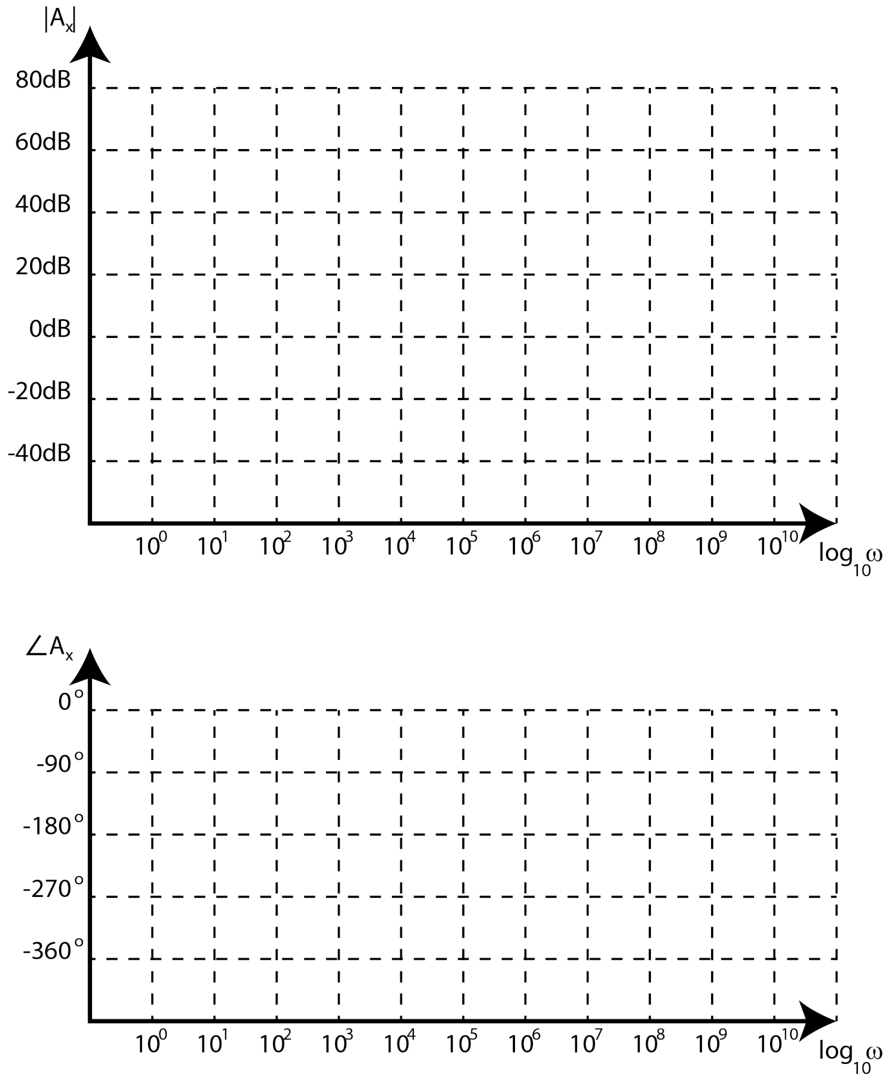


Figure A.4: Template to draw Bode plots in task A.2.1.

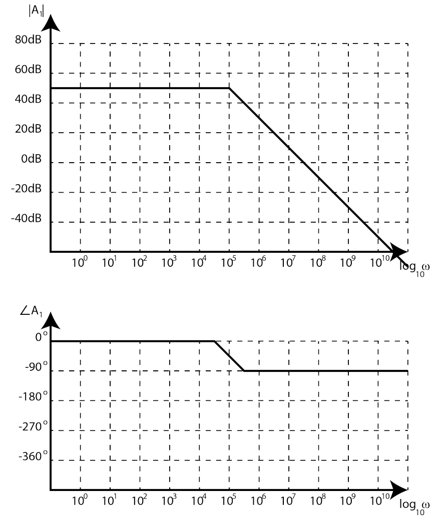


Figure A.5: Bode plots for task A.2.2.

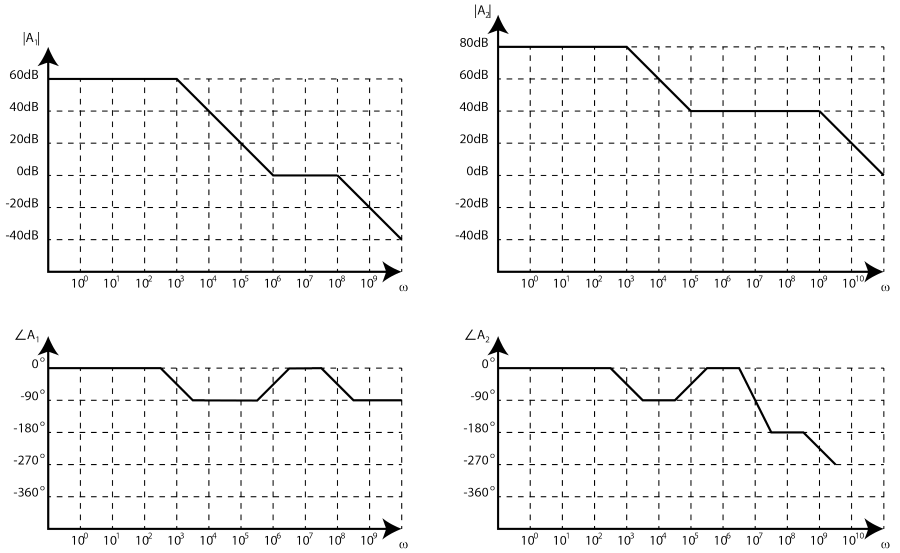


Figure A.6: Solution A.2.1

parameter	180nm		45nm	
	N	P	N	P
μC_{OX} $\left[\frac{\mu\text{A}}{\text{V}^2}\right]$	280	80	290	90
V_t [V]	0.45	0.45	0.45	0.45
Vdd-Vss [V]	1.8		1.8	
λL $\left[\frac{\mu\text{m}}{\text{V}}\right]$	0.08	0.08	0.11	0.15
n	1.6	1.7	1.8	1.8

Table A.1: Process parameter examples

A1 and A2 will also be vectors. Thus, one can immediately use for instance loglogplot(s,abs(A1))' and get the magnitude plot of A1. However, this was not required in the task and you can also just ignore the '.'s.

```
function [A1,A2]=transferF(s)

K=10.^(50./20)
A1=K.*1./(1+s./10.^5);
A2=A1.*(1-s./10.^5)./(1+s./10.^5);

end
```

A.3 Analog Circuit Basics

A.3.1

Let us assume some typical process parameters for a 180nm and a 45nm technology in table A.1. Consider the intrinsic gain of an nFET in 45nm and 180nm technology according to this table. What are the intrinsic gains A45 and A180 of a 45nm and 180nm nFET respectively, if both are biassed at 40μA (assume strong inversion (!) and that this is the drain current without channel length modulation), have a length L45 and L180 that is two times their respective minimum length and are three times as wide as long.

Note in this table is a process specific constant, the same as the inverse of that the book prefers.

Be careful about using all units correctly: one option is to get rid of all the micro and kilo etc. in the units by writing *1e-6 and *1e3 respectively. Submit the result as a MATLAB script in file IntrinsicGain.m. The final result after executing the script needs to be in variables A45 and A180 respectively. To make your file readable, use '45' and '65' in your variable names if they are process specific, e.g. W45 and W180. Use variable names that are similar to the ones used in the equations in the book. Please upload MATLAB script in file IntrinsicGain.m with the final result in variables A45 and A180!

A.3.2

For a nFET with k_n=400 $\frac{\mu\text{A}}{\text{V}^2}$ (pay attention to the 'micro' in the unit!) and thereshold voltage $V_tn = 0.5$V biased at a current of 15μA, and a $C_{GS} = 10$fF

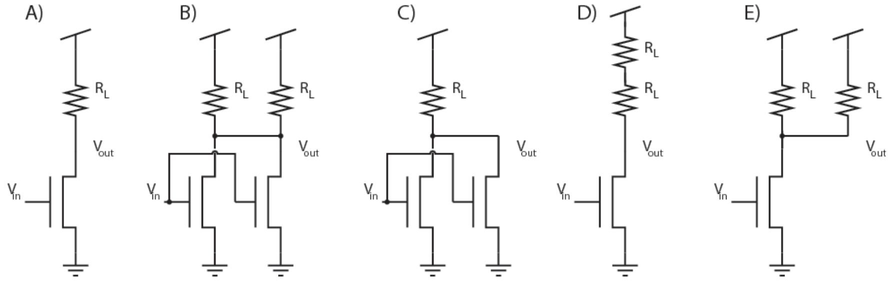


Figure A.7: CS stages to be ordered by their gain with same input bias V_I

and $C_{GD} = 1$fF, compute the current unity gain frequency ω_T in the MATLAB script fT.m, i.e. it shall not be a matlab function with a function header but simply a series of matlab commands at the end of which the result shall be in variable f_T.

A.3.3

Please order these Common source stages according to their small signal DC gain A_{DC}. Assume that all transistors have the same dimensions W and L, and are biased in saturation with the same overdrive voltage V_{OV}, i.e. the same input DC-offset V_I . Also assume that $R_L = r_{ds}$, where r_{ds} is the small signal drain resistance of one of the nFETs in the active region. If two or more CS-stages should happen to have the same ADC you can place them next to each-other in any order.

A.3.4

An nFET differetial pair is loaded with a pFET current mirror and an ideal current source provides a bias current I_B of 100μA (i.e. this is a current mirror loaded differential amplifier, see illustration A.8). Compute its differential small signal gain $A = \frac{v_o}{\Delta v_i}$ if $k_n = 500\frac{\mu\text{A}}{V^2}$ and $k_p = 300\frac{\mu\text{A}}{V^2}$ and and the Early voltage $V_A = \frac{1}{\lambda} = 20$V for both type of transistors.

A.3.5

The illustration A.9 shows the small signal model of a CS gain stage. Given that $g_m = 1\frac{\text{mA}}{\text{V}}$, $r_{ds}||R_L = 30\text{k}\Omega$, $R_{sig} = 4.8\text{k}\Omega$, $C_{gs} = 100$fF and the -3dB cut off is $\omega_{-3\text{dB}} = 100$Mrad, what is the DC gain A_{DC} and C_{gd} ?

A.3.6

Can you list effects/properties that will affect the output current I_O of a current mirror and cause deviation of the output current from the input current I_I (intentional or unintentional). Please make your answers a list of just 1-3 sentences per effect/property.

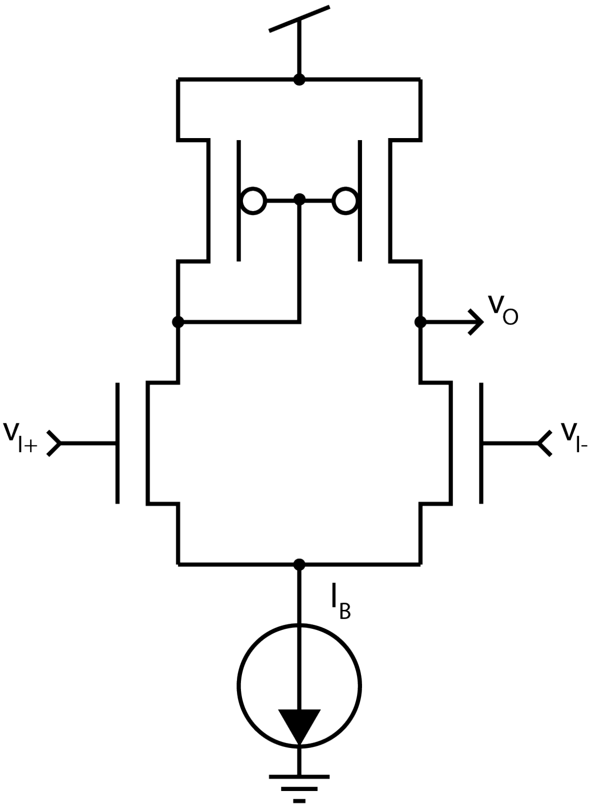


Figure A.8: Schematic of a current mirror loaded differential pair with ideal bias current source

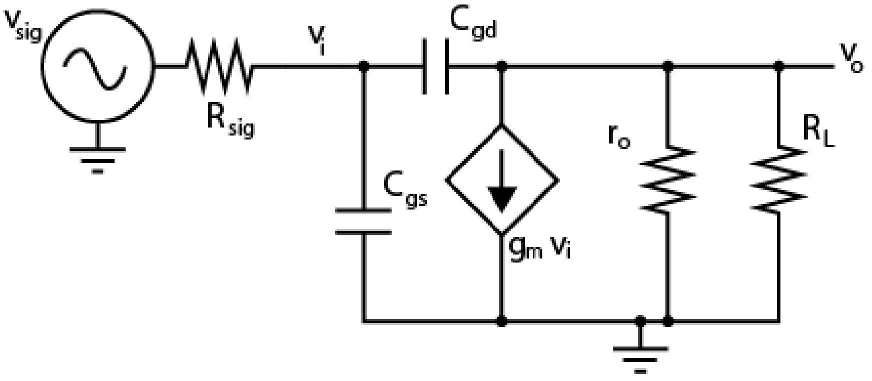


Figure A.9: CS small signal schematics to compute Miller capacitance

A.3.7

(*At the exam this question was somewhat difficult, because students did not really know about using vectors in MATLAB, so in future iterations I will ask for 5 individual results Q1, Q2, Q3, Q4, Q5 for Rsig1, Rsig2, Rsig3, Rsig4, Rsig4*)

Find the Q-factor of a source follower! Use $C_{GS} = 2\text{pF}$, $C_{GD} = 0.1\text{pF}$, $g_m = 5\frac{\text{mA}}{\text{V}}$, $R'_L = 30\text{k}\Omega$, Compute 5 results for 5 values of Rsig=[200 500 5000 50000 125000]Ω what is the Q-factor Q? Upload MATLAB script Qfactor.m which assigns final result to variable Q where Q is a vector with 5 elements corresponding to Rsig=[200 500 5000 50000 125000]!

Solutions section A.3

Solution A.3.1

```
function [A] = IntrinsicGain( Id, L, uCox, W, LamL)
%UNTITLED2 Summary of this function goes here
%   Detailed explanation goes here

%Id=100e-6
%uCox=400e-6
% LamL=(1/5)*1e-6
% W=10e-6
% L=0.5e-6


%gm=Vov*kn;
%Id=0.5*kn*Vov^2
gm=sqrt(2*Id*uCox*W/L)
%lambda=lamL/L
ro=L/LamL/Id

%A=sqrt(2*Id*uCox*W/L)*L/lamL/Id
A=sqrt(2*L/Id*uCox*W)/LamL
end
```

Then you may execute:

*** needs update with numbers from table ***

```
Id=40e-6;
L45=45*2e-9;
L180=180*2e-9;
W45=3*L45;
W180=3*L180;
uCox45=280e-6;
uCox180=270e-6;
LamL45=0.1e-6;
LamL180=0.08e-6;
```

```
[A45] = IntrinsicGain( Id, L45, uCox45, W45, LamL45)

[A180] = IntrinsicGain( Id, L180, uCox180, W180, LamL180)

A45 =

5.8327

A180 =

28.6378
```

Solution A.3.2

```
d=15e-6
kn=400e-6
Cgs=10e-15;
Cgd=1e-15;

%gm=Vov*kn;
%Id=0.5*kn*Vov^2
gm= sqrt(2*Id*kn)

f_T=gm/(Cgs+Cgd)/2/pi


f_T =

1.5850e+09
```

Solution A.3.3

The gain is given trough the total transconductance G_M multiplied with the total output resistance R_O. G_M is multiplied with the total transistors in parallel, and R_O is reduced with the total transistors *and* load resistors in parallel and increases if resistors are in series. Thus:

$$\begin{aligned}
A_A &= \frac{1}{2} g_m r_{ds} \\
A_B &= \frac{1}{2} g_m r_{ds} \\
A_C &= \frac{2}{3} g_m r_{ds} \\
A_D &= \frac{2}{3} g_m r_{ds} \\
A_E &= \frac{1}{3} g_m r_{ds}
\end{aligned} \tag{A.10}$$

$$A_C = A_D > A_A = A_B > A_E \tag{A.11}$$

Solution A.3.4

Once more we assume strong inversion (with μA biasing it is most likely the case) which needs verification at the end.

$$\begin{aligned} r_{ds} &= \frac{V_A}{I_D} = 20/50e-6 = 400\text{k}\Omega \text{ for both p and n} \\ g_{mn} &= \sqrt{2I_Dk_n} = \sqrt{2*50e-6500e-6} = 224\frac{\mu\text{A}}{\text{V}} \\ A &= g_{mn}\frac{r_{ds}}{2} = 224e-6*400e3/2 = 44.8\frac{\text{V}}{\text{V}} \end{aligned} \tag{A.12}$$

Checking for strong inversion:

$$V_{ovn} = \sqrt{\frac{2I_D}{k_n}} = \sqrt{2*50e-6/500e-6} = 0.45\text{V} \tag{A.13}$$

Solution A.3.5

One can safely assume that there is a clearly dominant pole at the input to the amp as there is no C_L, so the Miller cap alone at the input will be much bigger than any capacitance seen at the output.

$$\begin{aligned} A_{DC} &= g_m(r_{ds}||R_L) = 1e-3*30e3 = 30\frac{\text{V}}{\text{V}} \\ C_{tot} &= \frac{1}{\omega_{-3\text{dB}}R_{sig}} = 1/(100e6*4.8e3) = 2083fF \\ C_{tot} &= C_{gd}(A+1) + C_{gs} \\ C_{gd} &= \frac{C_{tot}-C_{gs}}{A+1} = (2083e-15-100e-15)/31 = 64\text{fF} \end{aligned} \tag{A.14}$$

Solution A.3.7

Solution based on Sedra Smith's book, which uses a definition for R'_L that is NOT THE SAME as this compendium's R^*_L, so the equation look different, but is equivalent to the one in this compendium. This will be revised in the course of the Spring term 2024. Check first edition!

```
function [Q] = QinSF(Cgs,Cgd,Rsig,RL,CL,gm)


Miller=gm*RL+1;
b1=(Cgd+Cgs/Miller)*Rsig+(Cgs+CL)/Miller*RL;
b2=((Cgs+Cgd)*CL+Cgs*Cgd)/Miller*Rsig*RL;
Q=sqrt(b2)./b1;
end
```

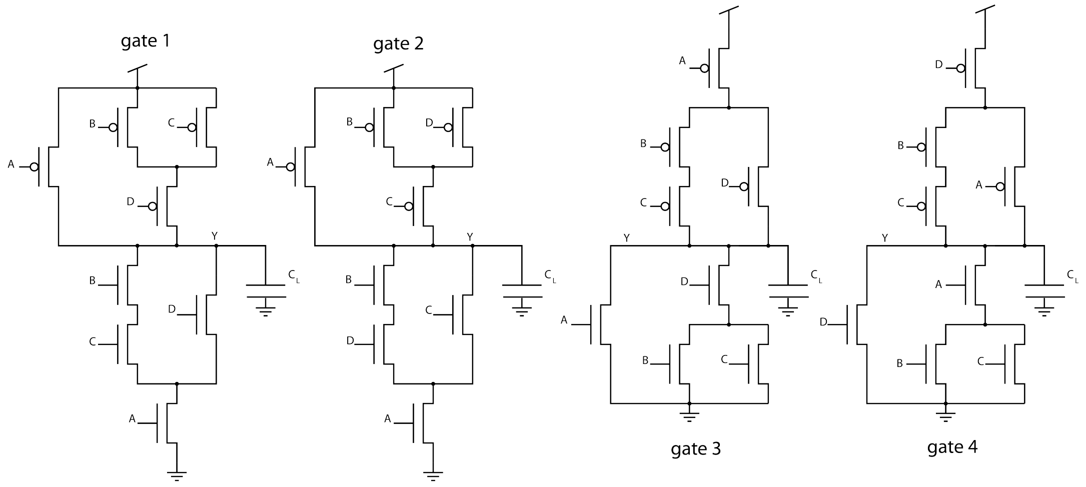


Figure A.10: Digital gates for task A.4.1

```
Rsig=[200 500 5000 50000 125000]

Cgs=2e-12
Cgd=0.1e-12
CL=1e-12
gm=5e-3
RL=30e3

Miller=gm*RL+1;
Q=sqrt(((Cgs+Cgd)*CL+Cgs*Cgd)/Miller*Rsig*RL) ./ ...
((Cgd+Cgs/Miller)*Rsig+(Cgs+CL)/Miller*RL)
%And then executing it


Q =

0.4886    0.7324    1.3005    0.7638    0.5123
```

A.4 Digital Circuit Basics

A.4.1

Please order the digital gates in figure A.10 by their propagation delays (fastest on top, slowest on the bottom) in a particular situation: specifically the delay for output signal Y to switch after the input signal A switches from low to high while input signals B and C are, and remain, high and signal D is ,and remains, low. The transistors have all the same $\frac{W}{L}$ ratio and C_L is the same for all gates. Should two gates have the same delay, just order them next to each other and

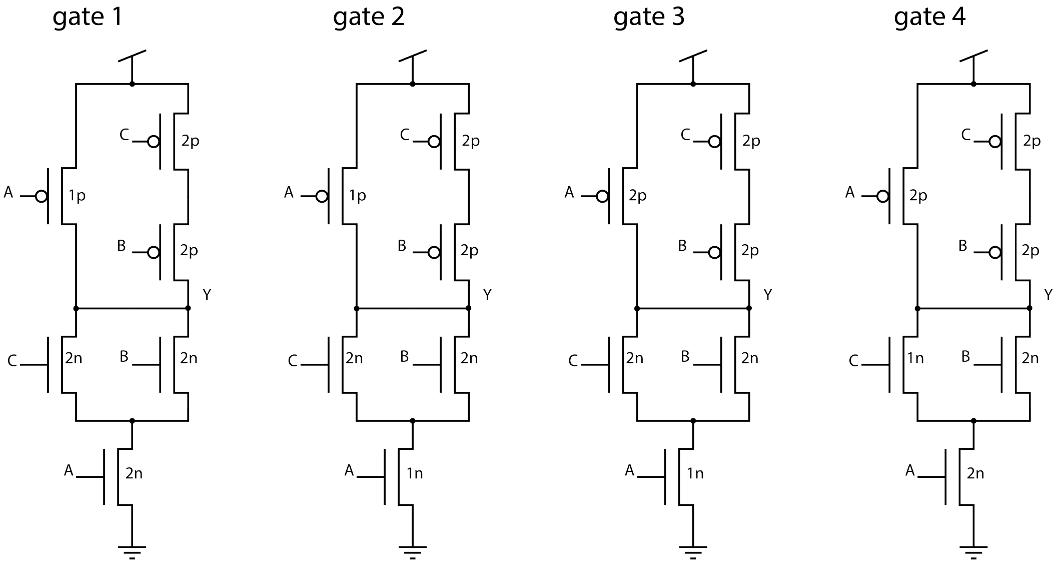


Figure A.11: Digital gates for task A.4.2

it does not matter which one you place before the other.

A.4.2

Which of the gates in figure A.11 switches the fastest when all inputs (A;B;C) go from low to high, and which is second, third and slowest? Consider the 'internal' capacitors on the output node (!) only as listed in the compendium according to equation (7.3), as well as the equivalent resistances of the conducting transistors. Note that the letters 'p' and 'n' indicate the W/L ratio of a balanced inverter, i.e. where both the PUN and PDN of that inverter have the same equivalent resistance when the transistor is conducting/switched on. So here, some of the transistors have double the width W at the same length and thus double the W/L ratio of that inverter. Here we do not consider any load capacitor, neither form the wire or from any subsequent circuit, so only the parasitic capacitors of the transistors connected to the output node Y and all equivalent resistances of conducting transistors.

A.4.3

Implement a static two input AND gate (Note: not a NAND gate!) with complementary PUN and PDN and input signals A and B , using nFETs and pFETs, power supply Vdd and Gnd.

A.4.4

Which of the pull up networks (PUN) in figure A.12 would complement the pull down network (PDN) shown. Multiple answers may be possible.

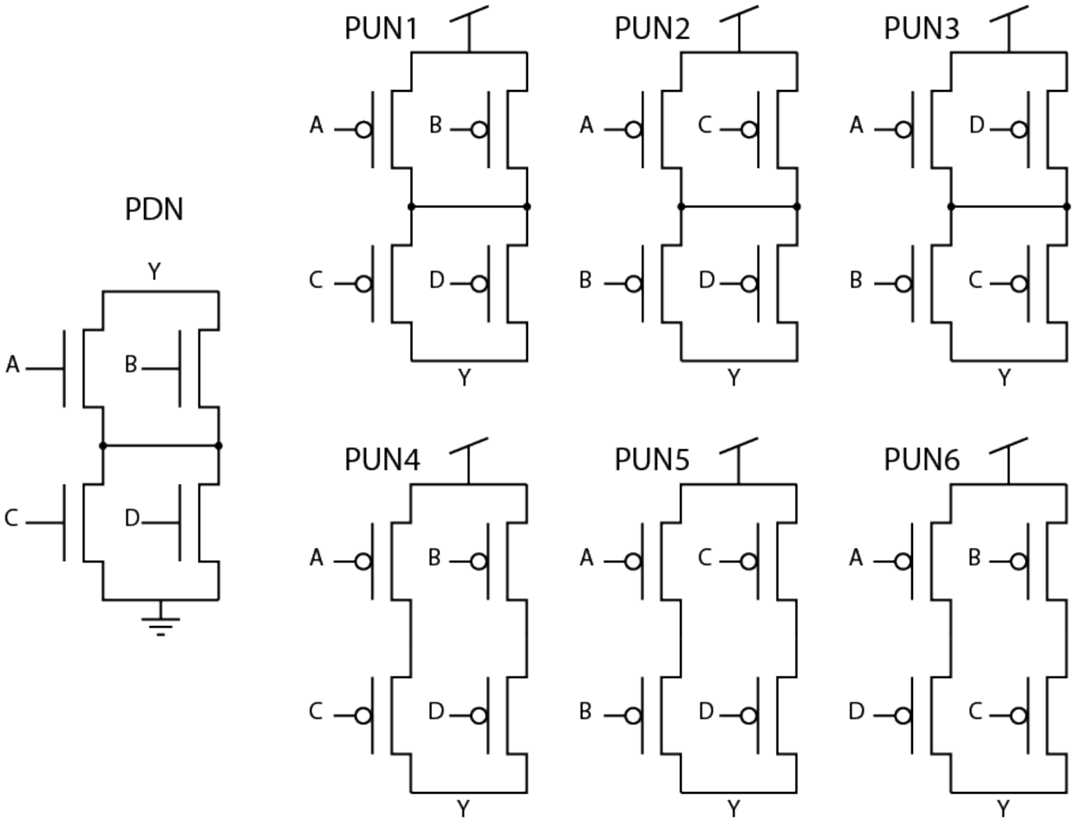


Figure A.12: PDN and corresponding PUN candidates. Illustration task A.4.4

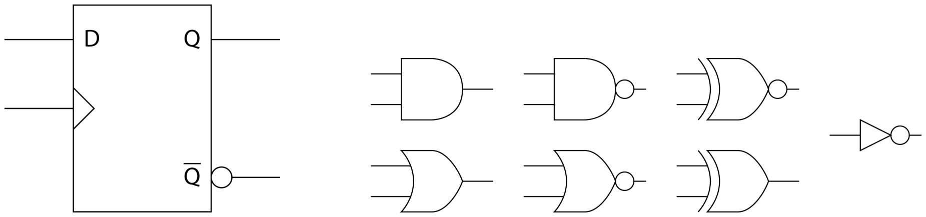


Figure A.13: Symbols to use in task A.4.5. From left to right: DFF, AND, OR, NAND, NOR, XNOR, NOR, NOT.

A.4.5

Make a JK-flipflop using one D-flipflop and combinational logic gates. A JK-flipflop is a synchronous flip flop with two inputs J and K (as well as the clock signal CLK) and two outputs Q and its inverse $\overline{Q}$. If only J is high, Q will go high and $\overline{Q}$ low at the next rising clock flank. If only K is high, Q will go low and $\overline{Q}$ will go high at the next rising clock flank. If neither J nor K are high, the previous state is maintained. If both J and K are high the next state will be the inverse of the previous state, i.e. the state will 'toggle'. It is not important that you use a minimal number of gates.

Use symbols for DFF and gates from figure A.13.

A.4.6

Draw an NP Domino logic schematic implementing the expression

$$Y = (A \vee B) \wedge C \tag{A.15}$$

Use nFETs and pFETs, a Vdd and Gnd power supply and static (!) input signals A, B, C and their inverse $\overline{A}, \overline{B}, \overline{C}$, as well as a clock signal ϕ and its inverse $\overline{\phi}$. The precharge phase happens when ϕ is low and the evaluation phase when ϕ is high. Use footed logic gates! Mark the output node Y! The function Y with three input variables shall not be implemented with a single logic gate with three inputs, but connect multiple gates with just two inputs each together. You might want to use the de'Morgan theorem to transform some of the expression into a form that lends itself more easily to be implemented in NP Domino logic.

A.4.7

Design a simple synchronous (i.e. there is a CLK signal that you do not need to represent explicitly, and state transition only occur synchronously with the rising edge of that clock signal) Moore-type finite state machine (FSM) to control a 'smart' burglar alarm. It shall have three states: 'asleep', 'attentive', and 'alarm'. It receives two binary sensor inputs: 'Movement' and 'human intruder'. It's default state is 'asleep' where it consumes very little energy and only motion sensors are active. They activate signal 'M' which is 0/false when there is no movement and 1/true if there is movement. If movement is detected the system enters its 'attentive' state, turning on a power consuming intelligent sensor system that processes camera images to detect the shape of a human being and distinguishes it from motion triggered by animals or by plants moved by the wind. Humans are signalled by the signal 'H' being 1/true and the system goes to state 'alarm'. On the other hand, if no human shape is detected, 'H' is 0/false and the system goes back to state 'asleep' until the next time it detects motion. An alarm remains active as long as there continues to be motion. If the movement stops, the system goes back to 'asleep'. 1) Draw a state transition graph for this FSM. In the state 'bubbles', do indicate the name and the binary encoding (!) of that state. Attention: if there are fewer states than your binary encoding would allow, make sure to include those 'undefined' states as well with an unconditional transition to the 'asleep' state: that's always good policy in case your circuit ends up in this undefined states by mistake, e.g. during power up of your circuit! 2) Make a state transition table with columns for the 'state now' with as many bits as necessary, with the two one bit input variables H and M, and with the 'next state'. Use binary variable values, 0, 1, and you may use an X for "don't care" states/inputs to simplify. Make sure to include all possible states and inputs in the table.

Solutions section A.4

Solution A.4.1

From fastest to slowest transition: (3,4,2,1)

Solution A.4.2

From fastest to slowest transition: (1,4,2,3)

Solution A.4.3

See figure A.14! The task asks for the transistor level solution, so that is the important one to submit, if this were a exam task. The others er more high level representations for illustration. They are equivalent only if it is understood that they represent complementary PUN/PDN style logic.

Solution A.4.4

PUN5

Solution A.4.5

```
Truth table
J K  D
0 0  Q
0 1  0
1 0  1
1 1  notQ

More explicit
J K Q  D
```

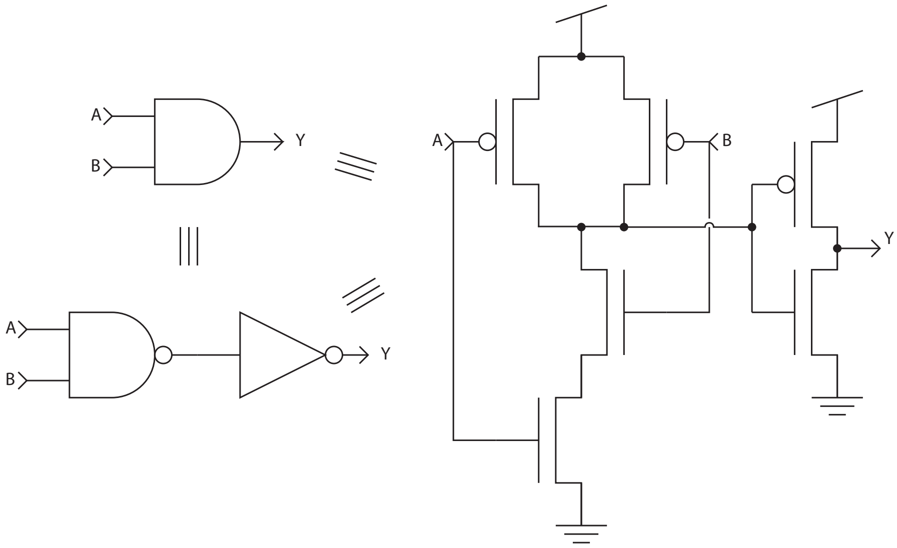


Figure A.14: Solution task A.4.3

0	0	0	0
0	0	1	1
0	1	X	0
1	0	X	1
1	1	0	1
1	1	1	0

Simplest implementation with 2x3-input AND gates and 1x2-input AND gate in the three cases where D is 1

Solution A.4.6

See figure A.15. It uses $\sim C$ and $\sim \phi$ as inversion instead of $\overline{C}$ and $\overline{\phi}$

Solution A.4.7

See figure A.16. It uses $\sim M$ and $\sim H$ etc. as inversion instead of $\overline{M}$ and $\overline{H}$, etc. Note that this is but one possible solution! Other equivalent implementations are possible.

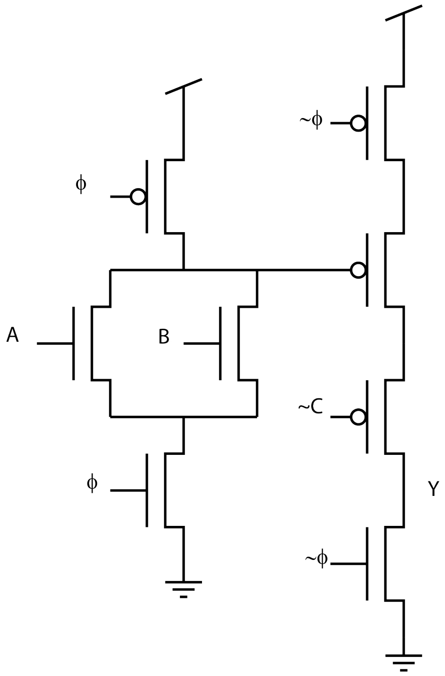
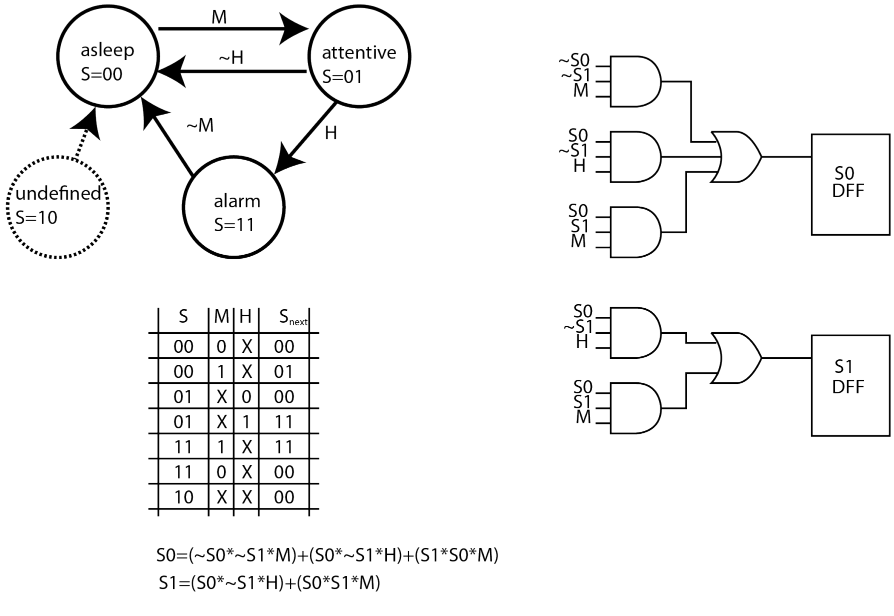


Figure A.15: Solution task A.4.6



S	M	H	S_{next}
00	0	X	00
00	1	X	01
01	X	0	00
01	X	1	11
11	1	X	11
11	0	X	00
10	X	X	00

S0=(~S0*~S1*M)+(S0*~S1*H)+(S1*S0*M)

S1=(S0*~S1*H)+(S0*S1*M)

Figure A.16: Solution task A.4.7