
Instructions for the Lab Report

High-Speed Devices EITP01

Markus Hellenbrand
markus.hellenbrand@eit.lth.se

November 5, 2019

1 Overview and Hand-In Instructions

During the lab session, we measure the s -parameters of a lateral nanowire MOSFET, the cross-section of which is very similar to that of the planar 2D MOSFETs that were covered in the lectures. You will now use a small-signal model to experimentally determine the high-frequency performance and the intrinsic parameters of the transistor. Your results should be handed in in the form of a complete lab report; hints and technical support are provided below.

The report should contain a theory part, where you briefly explain the small-signal model and its relation to the physical structure and behaviour of the transistor as well as the most important metrics, which characterise a MOSFET's high-frequency operation. These metrics are the gain(s) of the transistor, its cutoff frequency f_T , maximum oscillation frequency f_{\max} , and the stability factor k , so these have to be determined in the report. To do so, include a figure, where you plot the forward current gain h_{21} , the unilateral power gain U , the stability factor k , and the maximum stable and maximum available gain (depending on the stability factor k – cf. lecture slides) as a function of the frequency. It is sufficient to determine f_T and f_{\max} experimentally from the measured gains and it is alright to cut the low-frequency part of the figure, if it is too noisy.

To determine the small-signal model parameters, you first have to derive expressions for the y -parameters, which describe the small-signal model in Fig. 1. You will then see, how (with some approximations) the different parameters can be determined by fitting your equations to the measured data. These fits should be included in the lab report in the form of proper figures to document your procedure. (All fits can be done with linear frequency axes. Do not forget proper labels and units on the axes. Theory and some figures are obviously not the only two parts of the report, but you know how to write a report...)

Hand in the report as a pdf document to navya.sri.garigapati@eit.lth.se no later than one week after your lab date. You can also contact Navya if you have questions prior to submission. Review (supervisor) and correction (students) times are one week for each respective iteration and you can make corrections for a maximum of two times to be approved. Lab Reports can be handed in in pairs, but not in larger groups.

2 Model, Hints, and Technical Help

2.1 Small-Signal Model

You should use the model provided in Fig. 1 to derive the y -parameters. The data that you receive will already be de-embedded, so it directly corresponds to the model. The data is not normalised.

Hint 1: Begin by subtracting R_g , R_d , and R_s in the z -parameter representation (cf. lecture slides). We cannot measure these values experimentally on the given sample, since they are too small, but you can assume $2\ \Omega$ for each of the R_x . The y -parameters for the remaining network are straightforward. In fact, if you do not begin by subtracting R_g , R_d , and R_s , everything becomes very messy.

Hint 2: The parasitic capacitances in Fig. 1 can be determined from the *off-state* measurement.

Hint 3: The capacitances are expected to be in the fF range. This means that the denominator $1 + j\omega^2 R_i^2 C_{gs,i}^2$, which should show up in your equations, can be approximated to 1 for the fits.

Hint 4: Convince yourselves of the simplification that was made for C_{gd} by comparing the values that you get from the off-state and the on-state measurement.

Hint 5: Skip the low frequencies when fitting to y_{21} and y_{22} , since the effects at these frequencies are not covered by the model in Fig. 1. Some of the other fits are quite sensitive to the range where you fit, so always check the fits graphically.

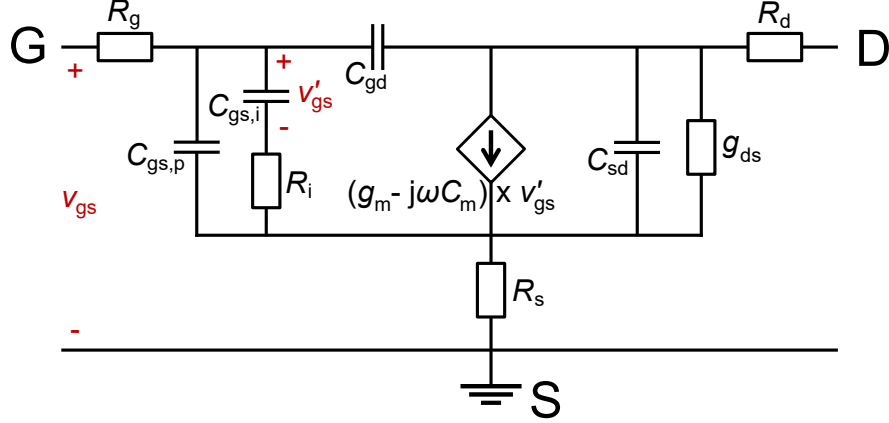


Figure 1: Appropriate small-signal model for the analysis of the measurement data.

A brief explanation of the physical origin of the different parameters is provided in Table. 1.

Table 1: Model parameters for Fig. 1 and their meaning.

Parameter	Corresponds to
R_g, R_d, R_s	Gate, drain, and source resistance, respectively.
$C_{gs,p}$	Parasitic source-gate capacitance.
$C_{gs,i}$	Intrinsic source-gate capacitance.
R_i	Intrinsic channel resistance. This is not a real resistance but instead models the time delay of carriers moving in the channel.
C_{gd}	Gate-drain capacitance. Since this capacitance is usually dominated by parasitics, we do not discern between intrinsic and parasitic capacitance here.
g_m	Transconductance. Modelled as a controlled current source.
C_m	Mutual gate-drain capacitance. Takes into account phase lag and models charge conservation.
g_{ds}	Output conductance.
C_{sd}	Source-drain capacitance. Contains an intrinsic and a parasitic part. Since this capacitance is in parallel with the (large) g_{ds} , the corresponding intrinsic capacitance can be neglected.

2.2 Technical Support

You can perform the data analysis with a computer program of your choice, but we recommend *MatLab*, which for students is available for download or accessible on the department computers. For *MatLab*, we can provide some useful functions, which facilitate handling the measured data. The functions are accessible for download from the course homepage and have the general form $a2b(A)$. a , b , and A here stand for a

certain set of parameters (s, y, z) and the functions automatically transform data (A) from a -parameters to b -parameters. If the function name contains an n in front of the corresponding parameter, it means that the function expects/returns the parameter set with the respective normalisation. The input should always be a matrix of the form (note the order of the indices) $[a_{11} \ a_{21} \ a_{12} \ a_{22}]$, where each a_{xy} corresponds to a vector. The output will be in the same order.

Example: The function $s2nz(\mathbf{S})$ expects a matrix of s -parameters and converts them to normalised z -parameters.

Further basic (standard) *MatLab* functions that can be helpful to get started:

- `csvread('filename.csv', x)` reads a file of the csv format with name 'filename' and skips the first x lines of the file (which can e.g. contain column names).
- `polyfit(x, y, n)` fits a polynomial of degree n to a vector y as a function of a vector x . The coefficients a_x of the polynomial will be returned as a vector $[a_n, a_{n-1}, a_{n-2}, \dots, n_1]$.