Transconductance and output resistance

You'll often hear that a transistor amplifies. This page puts a number on how much it can amplify, and shows that what sets the number is the product $g_m r_o$.

So far you've measured currents, and each one answered the question "what does this device do at this bias?". An amplifier cares about something else: **how much the answer changes when you nudge the input.** That's a slope rather than a value, and a transistor has two slopes that matter, one for each terminal you can move.

Symbol	Name	It is the slope of	In words
g_m	**transconductance**	I_D vs V_{GS}	how hard the gate pushes current through the device
g_{ds}	**output conductance**	I_D vs V_{DS}	how much the drain voltage leaks through to the current
$r_o = 1/g_{ds}$	**output resistance**	—	the same slope, written as its reciprocal

g_m is measured in siemens because it's amps per volt, and it's called *trans*-conductance because the volts are at one terminal and the amps at another. That's the reason a transistor can have gain at all. A two-terminal device's conductance can only turn a voltage across it into the current through *itself*, but a transistor lets a voltage on the gate control the current through the drain.

Measuring g_m

g_m is the slope of the I_D vs V_{GS} curve you swept in Lab 03. When you differentiated that sweep in Lab 03, Part 5, you got:

```
== g_m = dI_D/dV_GS   (V_DS = 1.8 V)
   V_GS = 0.90 V   I_D =    63.876 uA   g_m =  376.932 uS   g_m/I_D =   5.90 /V
   V_GS = 1.20 V   I_D =   217.505 uA   g_m =  634.008 uS   g_m/I_D =   2.91 /V
   V_GS = 1.50 V   I_D =   436.080 uA   g_m =  809.693 uS   g_m/I_D =   1.86 /V
   V_GS = 1.80 V   I_D =   696.275 uA   g_m =  914.650 uS   g_m/I_D =   1.31 /V
```

and the same lab printed it next to the model's own value:

```
   g_m at V_GS = 1.8 V               914.6500   915.3117  uS   -0.07 %
```

Your 361-point sweep gives **914.650 µS**, and BSIM's analytic derivative gives **915.3117 µS**, so the two are only 0.07 % apart. Compare that with the threshold voltage, where two reasonable methods disagreed by 86 mV. The difference is that g_m is simply the slope of a measured curve, so there's no choice of definition to argue about.

The square-law estimate

Differentiating the square law gives a tidy expression for the slope:

$$I_D = \frac{1}{2}\mu_n C_{ox}\frac{W}{L}V_{ov}^2 \quad \Rightarrow \quad g_m = \frac{2I_D}{V_{ov}}$$

At $V_{GS} = 1.8$ V, with $V_{TH} = 0.6016$ V from Where saturation starts, the overdrive is 1.1984 V and I_D is 696.275 µA:

$$g_m = \frac{2 \times 696.275\ \mu\ \text{A}}{1.1984\ \text{V}} = 1162.0\ \mu\ \text{S}$$

The model gives **915.3 µS**, so the formula comes out **27 % high**.

Lower down, at $V_{GS} = 0.9$ V, the overdrive is only 0.2984 V:

$$g_m = \frac{2 \times 63.876\ \mu\ \text{A}}{0.2984\ \text{V}} = 428.1\ \mu\ \text{S}$$

The measured value there is **376.9 µS**, so at this lower bias the formula is only **13.6 %** high.

This is the same way the square law failed at the saturation knee, and it has the same cause, velocity saturation. As you push the overdrive up, the current stops growing quadratically, so its derivative stops being $2I_D/V_{ov}$. In the fully velocity-saturated limit $g_m = I_D/V_{ov}$, which at 1.8 V would predict 581 µS, 36 % low. The real value lies between those two limits, in a region that no closed-form formula covers.

> **Use $2I_D/V_{ov}$ as a sanity bracket rather than as an answer.** If your simulator says g_m is half or double that, look for a mistake. If it says 27 % less, that's a 130 nm transistor behaving normally.

g_m/I_D: how designers size devices

Look again at the last column of the g_m printout, set out here as a table:

V_{GS}	I_D	g_m	g_m/I_D
0.90 V	63.876 µA	376.932 µS	**5.90 /V**
1.20 V	217.505 µA	634.008 µS	**2.91 /V**
1.50 V	436.080 µA	809.693 µS	**1.86 /V**
1.80 V	696.275 µA	914.650 µS	**1.31 /V**

Between the first row and the last, the current goes up by a factor of **10.9**, but the g_m you bought with it goes up by a factor of only **2.4**. You can think of g_m/I_D as an exchange rate: it says how much transconductance one ampere of bias current buys you, and it falls steeply as you push the gate harder.

In the other direction, toward lower V_{GS}, it climbs. The Lab 03 deck `spice/gm_ro.spice` steps the same device down to $V_{GS} = 0.7$ V:

```
V_GS = 0.7 V
i(vdc) = -1.06419e-05
@m.xmc.msky130_fd_pr__nfet_01v8[gm] = 1.437123e-04
```

143.7123 µS from 10.6419 µA works out to **13.50 /V**, which is ten times the efficiency of the 1.8 V bias, using a sixty-fifth of the current.

The climb has a ceiling, though. Deep below threshold the drain current is exponential in V_{GS}, and the derivative of an exponential is proportional to the exponential itself, so g_m/I_D levels off at

$$\frac{g_m}{I_D}\Big|_{\max} = \frac{1}{nV_T}$$

Here $V_T = kT/q = 25.8649$ mV at 27 °C, and n comes from the subthreshold slope you measured in Lab 03, **85.6 mV/decade**, so $n = 85.6/(2.3026 \times 25.8649) = 1.437$. That puts the ceiling for this device at

$$\frac{1}{1.437 \times 0.0258649} = 26.9 \text{ /V}$$

At $V_{GS} = 0.7$ V you're at 13.50, **half the ceiling**, which is the textbook definition of **moderate inversion**. That's the region between the square law and the exponential, where most low-power analog circuits operate and where neither hand formula works.

The single number g_m/I_D captures the whole design trade. If you bias the device hard, you get speed and small devices, but poor current efficiency. If you bias it softly, you get more g_m per microamp, at the cost of area and bandwidth. Professional analog designers choose g_m/I_D **first** (10 to 15 /V for a low-noise amplifier, 5 for something that has to be fast) and let the geometry follow from it.

r_o and intrinsic gain

g_m tells you how much signal current you get for a given input. To turn that current back into a voltage you have to push it through a resistance, and the largest one available is the transistor's own r_o. It comes from the slope of the saturation region, which looks flat but, as Where saturation starts showed, isn't quite.

When you multiply the two, the units of current cancel and you're left with a voltage gain:

$$A_{v,\,\max} = g_m r_o = \frac{g_m}{g_{ds}}$$

This is the **intrinsic gain**, the most voltage gain one transistor can produce at a given bias, whatever you build around it. It's an upper limit because it assumes you loaded the transistor with something infinitely better than itself. `spice/gm_ro.spice` measures it at four lengths, each at $V_{GS} = V_{DS}$ = 1.8 V:

L	I_D	g_m	$r_o = 1/g_{ds}$	$g_m r_o$	in dB
0.15 µm	2644.190 µA	2468.806 µS	3.588 kΩ	**8.86**	18.9 dB
0.5 µm	1196.020 µA	1460.824 µS	24.008 kΩ	**35.07**	30.9 dB
1 µm	696.275 µA	915.312 µS	64.130 kΩ	**58.70**	35.4 dB
4 µm	197.974 µA	274.342 µS	386.764 kΩ	**106.11**	40.5 dB

Compare the two ends of the table. The minimum-length device has **9.0× the transconductance** of the 4 µm one, yet it gives only **one twelfth of the gain**, because its output resistance is 108 times lower. Channel-length modulation is the reason: when the drain voltage shortens a short channel slightly, that's a *large* fractional change in its length, so g_{ds} is large.

This is the most important trade in analog IC design, and it runs in the opposite direction to digital design. The digital track, DD101 through DD104, wants every transistor as short as the process allows, because short transistors are fast. An analog designer who needs gain deliberately draws long devices and pays for them in area and speed. Both use the same process and the same transistor, with opposite instincts about its length, and when the two teams share a die, this is what they argue about.

> **Intrinsic gain in the tens is normal** for a modern process. If a hand analysis promises a gain of 500 from one transistor, it used a value of λ from a textbook written when channels were microns long.

You'll run `spice/gm_ro.spice` yourself in Lab 04, Part 3.

Where these numbers go next

The inverter. Take an NMOS and a PMOS, tie their gates together and their drains together, and you have two transconductances driving one node, loaded by two output resistances in parallel:

$$A_v = -(g_{mn} + g_{mp})(r_{on} \parallel r_{op})$$

Both devices push on the output, so their g_m values add, but their output conductances add as well, so the result is a weighted average of the two devices' own $g_m r_o$ rather than a sum. The CMOS inverter measures a gain of **−116.0341** at L = 0.5 µm, well above the 35.07 in the table above, because it works at a different bias from the table's, where each transistor's own $g_m r_o$ is about 116. That page has the numbers.

The mirror. A saturated transistor is very nearly a current source, and r_o tells you how nearly. An L = 1 µm device holding 50 µA changes its current by **1.46 %** when its drain moves from 0.9 V to 1.2 V. That device is the heart of the current mirror, and this page's r_o column sets the mirror's error budget.

Check yourself

1. Why can a transistor have voltage gain when a resistor can't?
2. At V_{GS} = 1.2 V, what does $2I_D/V_{ov}$ predict for g_m, using V_{TH} = 0.6016 V? Is the measured value above or below it?
3. You need a g_m of 1 mS from as little current as possible. Should you bias near threshold or with a large overdrive? What do you give up?
4. Going from L = 0.15 µm to L = 4 µm, g_m falls and $g_m r_o$ rises. Explain each in one sentence.

Next: Sizing with W and L.