University of California at Berkeley College of Engineering Department of Electrical Engineering and Computer Sciences

EECS150 Spring 2000 J. Wawrzynek E. Caspi

Transistor Notes

This document is a condensed review of transistors (simplified RC model).

A MOSFET¹ transistor is a 3-terminal device that acts as a switch. The *gate* voltage controls whether the path from *drain* to *source* is an open circuit² ("off") or a resistive path ("on"). The gate itself looks like a capacitor.

There are 2 kinds of MOSFETs:

- An *n*-type MOSFET (*nfet*) is on for high gate voltage, and off oherise. When on, an nfet passes a 0 level cleanly (it degrades a 1).
- A p-type MOSFET (pfet) if on for low gate voltage, and off otherwise. When on, a pfet passes a 1 level cleanly (it degrades a 0).



A typical static CMOS gate is built of two complementary networks:

- A *pull-down* network composed of nfets, with sources connected to GND
- A *pull-up* network composed of pfets, with sources connected to VDD

Only one of these networks may be on at a time. Thus the output will be electrically connected either to GND or to VDD, not to both. If both networks are on simultaneously, the electrical path from VDD to GND will cause excessive current draw

¹ MOSFET = Metal Oxide Semiconductor Field Effect Transistor

² An open circuit is often denoted as *high impedence* or *high-Z*.

and may damage the circuit (this can happen if the inputs to a CMOS gate are tri-stated and floating – bad bad bad).

The simplest such structure is the *CMOS inverter*: 1 nfet, 1 pfet. More complex gates, *e.g.* NAND and NOR, require more transistors.



The pull-up and pull-down networks are always *duals*. To construct the dual of a network, perform the following transformations:

- Exchange nfets for pfets (and vice versa)
- Exchange series connections for parallel connections (and vice versa)

The series/parallel transformation requires looking at the hierarchical structure of the network. For instance, the dual of a parallel pair of series pairs (4 transistors) is a series pair of parallel pairs (draw it). With this methodology of duals, designing a CMOS gate requires designing only one of the networks (pull-up or pull-down), then producing a dual as the other network.

The pull-up network represents the computation of a function's 1-valued outputs. The pull-down network represents the computation of 0-valued outputs. Remember that pfets in the pull-up are activated by false inputs, whereas nfets in the pull-down are activated by true inputs. In whichever network you choose to design, place transistors in series to represent AND conditions, in parallel to represent OR conditions. For example, consider the NAND above, which computes (ab)'. The function is 0 when A and B are both true, Hence the pull-down network is a series connection. Alternatively, the function is 1 when either A or B is false, hence the pull-up network is a parallel connection.

The *delay* of a static CMOS gate really reflects how fast it can switch its output. In the simple model, we assume that all transistors turn on/off when their gates transition through VDD/2. The delay of a gate is measured from that point until the output transitions through VDD/2.

How does a gate switch its output? At any time, either the pull-up or the pull-down network is on, creating a resistive electrical connection from either VDD or GND (respectively) to the output. That path is a source of current which either charges or discharges the *load capacitance* at the output – namely the capacitance of any wires and gate inputs connected to that output (recall that the inputs of a static CMOS gate are transistor gates, which are effectively capacitors). Thus the switching of a gate's output can be modeled by an RC circuit. R's come from on transistors, and C's come from gates loading this output.

The delay of an RC circuit is proportional to the so-called *RC time constant*, τ =RC. Larger R means longer delay. Larger C means longer delay. Recall that series resistances sum, whereas parallel resistances sum in reciprocals ($1/R_{1||2} = 1/R_1 + 1/R_2$). The particular configuration of on transistors will determine the total R. Similarly, parallel capacitances sum, so the total C depends on how many other gate inputs are fed from the output (the *fanout*). In particular, keep in mind that a gate outputting (fanning out) to *n* other gates will be *n* times as slow as the same gate outputting to only 1 other gate, because the load capacitance is *n* times as large.



Charging a load capacitance $0 \rightarrow 1$

Discharging a load capacitance $1 \rightarrow 0$

Note that the design of static CMOS circuits (presented above) is distinctly different from the design of transmission-gate circuits. A *transmission-gate* is an nfet and pfet in parallel, controlled by opposite gate signals. With both kinds of transistors are present, a transmission gate can pass a 0 or 1 equally well and can thus be used as a switch for arbitrary data. Pull-ups, pull-downs, and dual networks are not necessary.

For more examples and discussion, see solution handouts for Homework #5 and Quiz #5.



Transmission Gate