

UNIVERSITY OF CALIFORNIA AT BERKELEY
COLLEGE OF ENGINEERING
DEPARTMENT OF ELECTRICAL ENGINEERING AND COMPUTER SCIENCE

Lab 1

Transistors

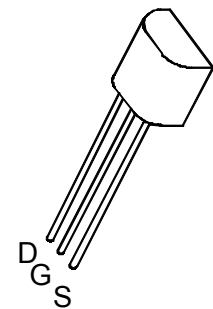
1 Motivation

In this lab you will get to build a logic gate and a latch from actual transistors. Additionally, this lab exercise will reorient you with respect to the lab instruments. We assume that you have had a basic electronics lab experience in the past, and that in particular you have used power supplies, meters, and done simple wiring with proto-boards. If not, see you TA for extra assistance.

2 Introduction

Nearly all digital-logic circuits these days are built from CMOS transistors. However, this is easy to forget, given that most designers work at a higher level of abstraction, typically the gate-level or module-level. But the transistors themselves have a strong effect on the performance, cost, and power consumption of these circuits. Likewise in EECS150, although the main emphasis of this course is not transistor-level design, it is important to have a firm grasp of at least the basics of transistor circuits to be a better digital designer.

The transistors we will use for this lab come in small plastic packages with three wire leads. The actual transistor is enclosed in plastic to protect it from the environment (and your hands). The leads make connection to the source (S), gate (G), and drain (D) terminals of the transistor. The body (or bulk) of the transistor is internally connected to the source terminal. This built-in connection is convenient for most logic circuits. However, as we shall see later, it limits their use. The illustration to the right shows you the transistor pin-out. It is the same for both the nFET and pFET. Remember that because of the internal connection of body to source, the source and drain terminals are different from one another with these devices and you need to wire them up correctly for your circuits to function.



The first part of the lab exercise uses the standard nand gate configuration using 2 nFETS and 2 pFETS (figure 1). The second part uses a data latch circuit similar to the one presented in class. It is modified somewhat to eliminate the “transmission gates”, since these are not possible to implement with transistors with built-in body to the source connections. For transmission gate implementation we would need to connect the body to ground in the nFETs and to Vdd in the pFETs. This latch (figure 2) uses two tri-state inverters and a normal inverter. The tri-state inverter used here is the functional equivalent to an inverter followed by a transmission gate.

The latch works as follows. When the clock signal is high, the first tri-state inverts D and passes it to the inverter that drives the output Q. Simultaneously, the second tri-state floats its output and thus has no effect on the input to the inverter (node X). When the clock is low, the input is disconnected from the input to the inverter. Meanwhile, the output, Q, is inverted by the second tri-state and again by the inverter, forming a

ring of two inversions. Therefore the latch holds Q at whatever value was last seen at D while the clock was high.

The transistors that we will be using operate over a wide range of voltages. For this lab, let V_{dd} be 3 volts. By keeping the supply voltage down to this low level, we will limit the currents in the circuits, letting them stay cooler, slower, and consequently easier to measure.

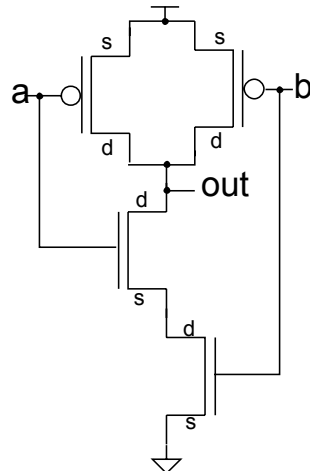


Figure 1: NAND gate – transistor-level.

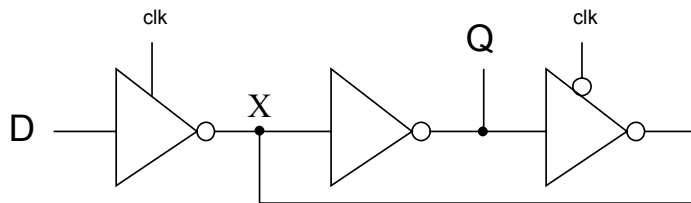


Figure 2: D-type Transparent Latch.

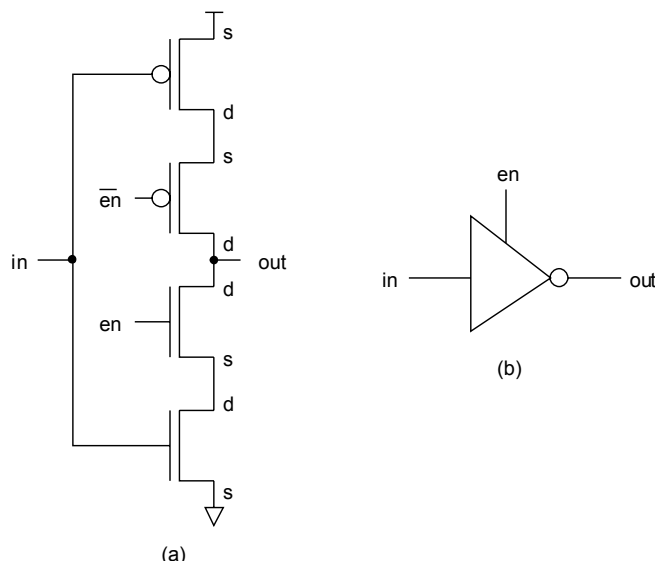


Figure 3: Tri-state Inverter

3 Prelab

1. Read and understand the entire lab handout.
2. Review the circuits involved in this lab. Make sure that you understand how they are supposed to work. Drawing the circuits out for yourself will help.
3. Make sure that you understand (remember) how proto-boards are internally connected. If you don't ask your TA. Plan out how you will wire up your circuits on the board. With careful planning you can minimize the amount of wiring that you will have to do in the lab, saving time and eliminating some chances for mistakes. Draw a little sketch with the transistors in the proper places to use as a guide in the lab.
4. Think through the experiments and what you will expect to see at each step.

4 Procedure

Part I: NAND Gate

1. Wire it up. Using the standard 4 transistor nand-gate circuit discussed in class (figure 1) as a model and your planned layout from the prelab, wire up a two-input nand-gate with the discrete transistors on your protoboard.
2. Static test of logic function. Connecting the inputs to either Vdd for logic ONE or ground for logic ZERO, verify that your circuit correctly implements the nand function by observing its output voltage with a voltmeter or an oscilloscope.
3. Measure V_{in} versus V_{out} . Holding one input at a constant logic HIGH, measure and plot the V_{in} versus V_{out} relationship for the other input, with input values covering the range from ground to Vdd. Take at least 15 data points, focusing your effort at the region of high slope. Plot your results with MS excel.
4. Repeat for the other input.

Part II: Latch Circuit.

1. Wire it up. Based on the circuit diagram shown above, and your planned layout from the prelab, wire up the transparent data-latch. Implement one extra inverter to generate clock-bar.
2. Verify tri-state action and measure D-Q delay.
 - a. Set the pulse generate to output a square wave with a frequency of around 100KHz and a maximum voltage close to Vdd when driving the D input to your latch. Set the clock signal to put the latch into "transparent" mode and verify the presence of the square wave at its output Q.

- b. Disconnect the feedback wire and verify the signal after the second tri-state, then reconnect the feedback.
 - c. Measure and record the “D-to-Q” delay.
 - d. Now, invert the clock signal, and verify that D (or its complement) are not present at the input to the inverter nor at Q.
3. Test latch function with “static” signals.
 - a. Put the latch in transparent mode. Deassert the input. Put the latch in latch mode. Check the output while you change the input. What do you see?
 - b. Repeat with a logic ONE at the input. Is everything ok? Is it really a latch?
4. Measure clk-Q delay with high-frequency clock signal. This test uses two square waves of the same frequency, one delayed relative to the other. Our signal generators only have one output, but we can use “trigger out” as one output and the normal output delayed as the other. The first square wave we will use as the D signal to our circuit, and the other delayed by 1/4 cycle as the clock. (Perhaps you should sketch this arrangement, to be sure that you understand it). Using the oscilloscope to monitor the signals, set up the signal generator to generate two square waves then connect them to your circuit (one to D and the other to clock). You will not have control over the level of the trigger out signal, but as long as it is around 3 volts (plus or minus 1 or so), things should be fine. Adjust the level of the other to close to 3 volts. This arrangement is what you need to measure the clock-to-Q delay for latching a logic ONE. Use the oscilloscope to make this measurement. Now repeat for latching a logic ZERO. You can arrange this by inverting one of the normal signal generator output.

John Wawrzynek and Yatish Patel (Spring 2002).

5 Checkoff

Name: _____

Name: _____

Section: _____

Part I – NAND Gate

1. Your NAND circuit. TA: _____ (10%)
2. Your plots of V_{in} versus V_{out} . One for $V_{in}=\text{ONE}$ and another for $V_{in}=\text{ZERO}$. TA: _____ (10%)

Part II – Latch Circuit

1. Your latch circuit. TA: _____ (30%)
2. The measured D-to-Q delay _____ TA: _____ (10%)
3. The clock-to-Q delay for logic ONE _____ TA: _____ (10%)
4. The clock-to-Q delay for logic ZERO _____ TA: _____ (10%)
5. Your setup (including scope waveforms) for measuring the clock-to-Q delay. TA: _____ (20%)

Total

50% off if the lab is handed in 1 week late TA: _____ (-50%)

Total Score TA: _____ (100%)