IX. Lab 9. Digital Electronics I: Logic Levels and Gates

A. Introduction

1. Logic levels

Digital electronics makes use of two logic levels, “0” and “1”. The ranges of voltage which corresponds to a “0” or to a “1” depend on the logic family. 7400 family TTL logic, the dominant logic family in the 1970s and 1980s, set a standard for logic levels that is still widely used today. 7400 logic must be operated with a power supply voltage $V_s = 4.5$ to 5.5 V. Logic gates in the 74HCT family use CMOS circuits and can operate with a power supply voltage from 2.0 to 6.0 V. 74HCT components are designed to work with the same logic levels as TTL when operated with a 5.0 V supply. These logic levels are illustrated in Figure IX-1(a) for the case of 5 Volt supply. Two more popular logic families are the CD4000 and 74HC families. These are CMOS logic families that have similar thresholds: 0 to 30% of V_s for a logical “0”, and 70% to 100% of V_s for a logical “1”. These thresholds are shown in Figure IX-1(b) for the case $V_s = 5.0$ V. Components in the 74HC family can be operated with a supply voltage $V_s = 2.0$ to 6.0 V. Components in the CD4000 family can be operated with a supply voltage $V_s = 3.0$ to 15.0 V.

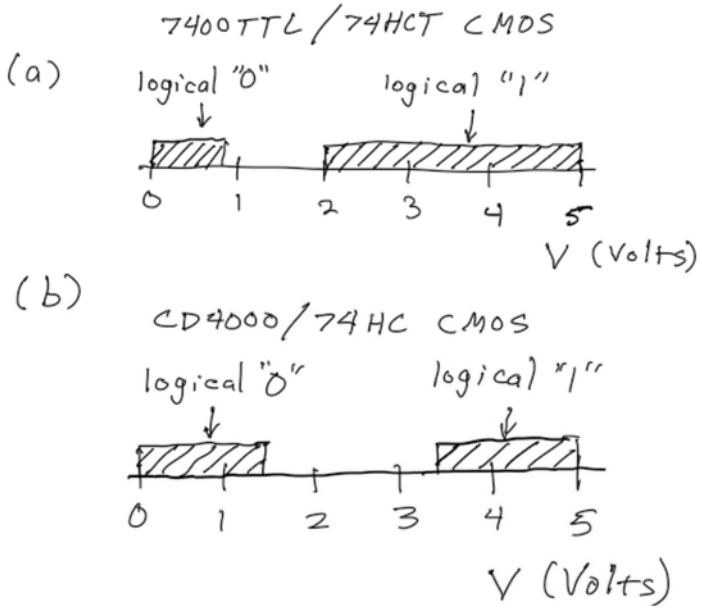


Figure IX-1. Logic levels for (a) 7400 or 74HCT logic families, and (b) CD4000 and 74HC logic families, in all cases with a power supply voltage of 5.0 V.

The main advantage of the TTL/74HCT logic levels shown in Figure IX-1(a) is compatibility with TTL-standard electronics, which is sometimes required. The CMOS logic levels shown in Figure IX-1(b) have better immunity to noise, due to the larger gap between the voltage range for “0” and the voltage range for “1”.

2. Active *vs.* passive pull-up

Logic gates switch their output between a low impedance connection to ground and a low impedance connection to the positive supply voltage V_s. There are two distinct ways to do this. One way is to connect a "pull-up" resistor and transistor switch in series between V_s and ground, with the output taken at the junction between the resistor and transistor, as shown in Figure IX-2(a). In that case the resistor and transistor essentially form a voltage divider. If the transistor is off, its source-drain resistance is very high compared to R_{pullup}, so the resistor "pulls up" the output voltage to be very close to V_s, *i.e.* to a logical 1 state. If the transistor is on, its source-drain resistance is very low compared to R_{pullup}, so the transistor connection "pulls down" the output voltage to be very close to ground, *i.e.* to a logical 0 state. This arrangement is called a *passive pull-up.*

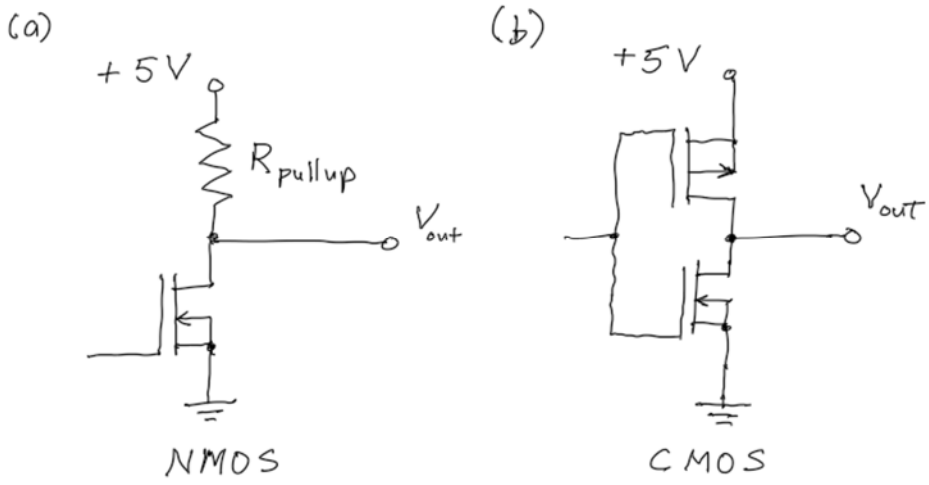


Figure IX-2. Logic gate output with (a) passive pull-up, and (b) active pull-up.

Figure IX-2(b) illustrates a CMOS logic output stage, which has an *active pull-up.* The operating principle is the same, except that a PMOS transistor switch has taken the place of the pull-up resistor. Since the PMOS and NMOS transistors have oppositive polarity, only one of them can be on at any given in time. If the NMOS transistor is on, its low drain-source resistance connects the output to ground, *i.e.* it produces a logical 0 state. If the PMOS transistor is on, its low drain-source resistance connects the output to V_s, *i.e.* it produces a logical 1 state.

Logic with passive pull-up was used mostly in the very early days of semiconductor logic, for instance with RTL logic in the 1960s. Since then, logic has almost exclusively used active pull-up. You'll see one reason for this later in this lab.

B. Procedure

1. Rules when using logic

- Never apply a voltage outside the range of the power supply. For the logic gates we'll use this means that you must keep the signals between 0 and +5 V. Applying voltages outside this range can damage the logic gates.
- Never apply logic signals to gates which do not have the power supply voltage V_s connected and turned on. This can also damage the gates.
- Power your circuits between $V_s = +5$ V and ground only. Before connecting your logic gates, check the supply voltage and make sure it is close to 5 V. If it isn't, adjust the voltage or find another supply that does produce +5 V and use that instead.
- As always, put together your circuit with the power off, and only turn the power on when the circuit is complete. Turn the power off when making any changes to the circuit.

2. Input logic signals and output signal measurement

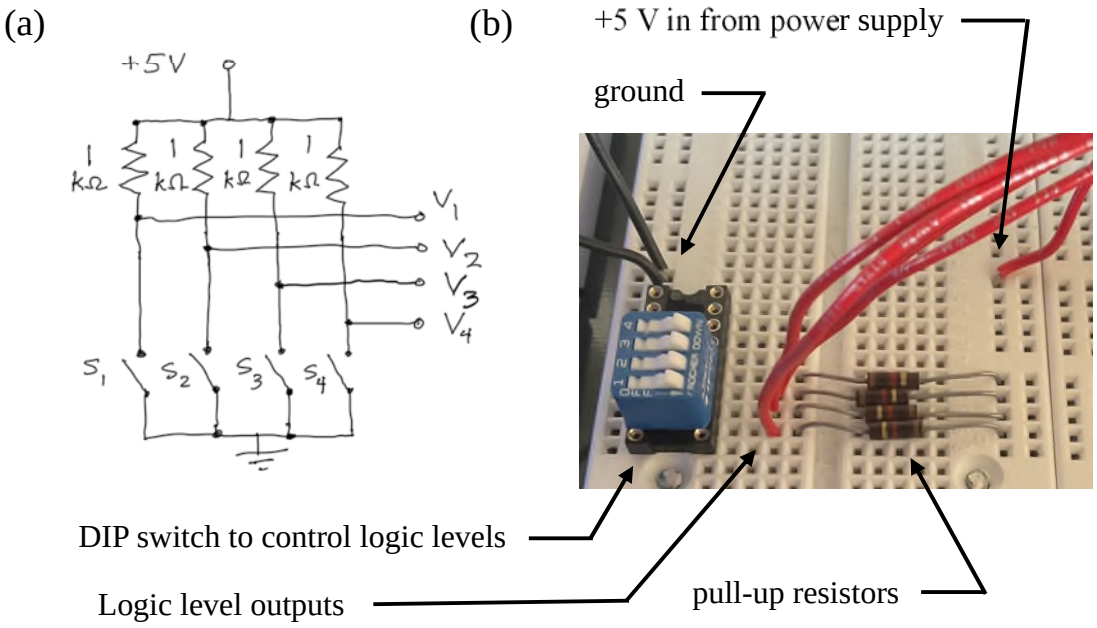


Figure IX-3. (a) Circuit to produce four logical voltage signals (b) realization of the circuit on the breadboard.

In this week's experiment, you'll need adjustable logic signals to use as inputs to your circuits. We have two models of breadboard. Most are the model PB-503. These have an 8 position DIP switch that could potentially be used to generate these signals. However this switch is set up so that the switches can connect only to ground, or only to the +5 V supply. To make our signal source, we need to connect to both, with pull-up resistors on the supply connection. In principle we can add those connections on the breadboard, but once we're on the breadboard it's more convenient to put the switch itself on the breadboard, as shown in Figure IX-3(b).

The model 505A breadboard is in principle a bit better, with an 8-position switch in which each switch can connect either to the +5 V supply or to ground. However the problem with this breadboard is that the +5 V supplies on them are not working. Because of this you'll need to use a separate +5 V supply to power your circuits. You can't use the built-in switches because they would apply voltages outside the 0 to 5 Volt range to your circuits. So, you'll need to build your signal source on the breadboard for the PB505A as well.

Start by identifying the power supply you want to use. For the PB-503's you can probably use the built-in 5 V supply, but check its output voltage first. If it is between 4.75 V and 5.25 V you can use it. Otherwise use one of the bench power supplies. For the PB-505A's you'll need to use one of the bench power supplies. (Alternatively you could use the PB-503 or PB-505A's variable +V supply, but if you do please be careful that it is set to 5.0 V and stays set at 5.0 V.)

Next, build the circuit shown in Figure IX-3(a). For the reasons discussed above, you should build it on the breadboard as shown in Figure IX-3(b). Use a four position DIP switch to control your voltage levels. A DIP switch has the same bottom pins as a DIP (Dual Inline Package) IC. You could just plug the DIP switch directly into the breadboard, but it doesn't connect very firmly and has an annoying tendency to come back out. To alleviate this problem, we've mounted the DIP switches onto IC sockets. These are designed to be used in printed circuit cards, with the pins soldered into the card and the IC inserted into the socket. This makes IC replacement easy since you don't have to unsolder and resolder the IC. In our case, the socket allows the DIP switch to mount firmly into the sockets and the socket pins to mount firmly into the breadboard. (The reason this works is that we've used a high-quality IC socket. If we used a cheap socket it wouldn't work any better than pushing the DIP switch directly into the breadboard.) We've used 14 pin sockets, and cut two of the pins off so that you can plug one side of the socket into the ground line as shown in Figure IX-3(b).

We suggest you orient your switch so that it closes when you push down on the side closest to the ground line. That way, pushing on that side will give a "0" output and pushing on the other side will give a "1" output, which seems "correct" given the positions of the ground and +5 V lines. We also suggest that you cut the leads on your pull-up resistors so that they lay down neatly on the breadboard, as shown in Figure IX-3(b). That way, the legs of the resistors won't short out against each other and change your logic levels.

Next, you'll need a way to measure the logical output states of your circuits. Both the PB-503 and the PB-505A have a set of 8 logic indicators that allow you to do this. The logic indicators on both types are buffered, so they won't load down your signal source.

On the PB-502, the indicators light if the input voltage is more than 1.4 V (indicating logical "1") and stay unlit if the input voltage is less than 1.4 V (indicating logical "0").

The logic monitors on the PB505A are more complicated. They determine whether the input signal is a "0", a "1", or neither by comparing it to correct logical threshold voltages shown in Figure IX-1. To do this there are two comparators for each input. One compares V_{in} to $V_{"0",\max}$, the maximum voltage for a logical 0. If $V_{in} < V_{"0",\max}$ the "low" LED lights and the "high" LED remains unlit. The other comparator compares V_{in} to $V_{"1",\min}$, the minimum voltage for a logical 1.

If $V_{in} > V_{"1",min}$, the "high" LED lights and the "low" LED remains unlit. If V_{in} is between $V_{"0",max}$ and $V_{"1",min}$ neither LED lights. This arrangement is useful if "three-state" logic is present. For such logic the possible output states are "0", "1", and "high-Z". In the high-Z state, the logical output has neither a low-impedance path to $+V_s$ nor a low-impedance path to ground. This allows some other component of the circuit to determine the logical level of that line.

If the logic monitor switches on the PB505A are set to "TTL" and to "+5", the logic threshold levels used are those for the 7400 series TTL standard (indicated in Figure IX-1(a)). If the monitor switches are set to "CMOS" the logic threshold levels used are those for the 74HC standard (indicated in Figure IX-1(b) for +5 V supply). For CMOS, the logic thresholds can also be set to the level for a supply voltage equal to the "+V" output of the PB-505A by putting the switch into the "+V" position.

Next, connect a variable voltage source to one of the logic indicators on your breadboard. If you are using the PB-505A, set the monitor to "TTL" (since you'll be using TTL-compatible ICs). Set the voltage to "+V" and set the +V power supply so that it puts out 5.0 V. (The reason to do this is that the +5 V supply is putting out the wrong voltage, and that could cause the threshold voltages to be wrong.)

If you are using the PB-503, measure the voltage needed to light the LED for each of the first four channels. Verify that these numbers are not too far away from the specified 1.4 V.

If you are using the PB-505A, measure the maximum voltage for which the "low" LED lights and the minimum voltage at which the "high" LED lights for the first four channels. Verify that these numbers are not too far away from the specified thresholds for TTL logic.

Finally, connect up your four logic source outputs to the first four channels of the breadboard logic monitor. Verify that you can set each output to either "0" or "1" with the switch, as read by the logic monitor.

For your report: Include your measurements of the voltage thresholds. Include a couple of photos of your logical voltage source being read out on the LEDs for different combinations of zeros and ones.

3. Quad NAND gate

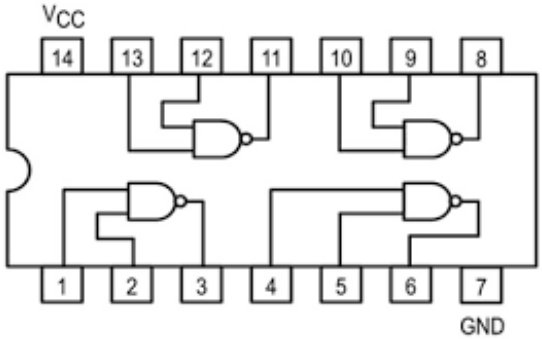


Figure IX-4. Pin-out for the 74HCT00N Quad NAND gate.

Bring a 74HCT00N quad NAND gate and a 7400N quad NAND gate to your bench. The pin-out for the two ICs is the same and is shown in Figure IX-4. These ICs are in a DIP-14 package, *i.e.* a DIP package with 14 pins. As was the case for the analog chips, each chip has a small cut-out at one end. Looking from the top side of the chip, pin 1 is the first pin to the left of the cut-out, and the remaining pins are number 2, 3, 4.,, down one side of the chip and then 8, 9, 10,… up the other side of the chip. For all of the 14 pin 74-series ICs, the power connections are on opposite corners of the chip: ground on pin 7, and the positive supply on pin 14. In this case the chip contains four NAND gates, each with two inputs and one output. The pin connections are as shown in the figure.

Insert the 74HCT00N into your breadboard. This is a 74HCT family CMOS IC. When you do this, the two sides of the IC will straddle a gap in the breadboard in the same way as the op-amps you worked with. Wire up a power connection and a ground connection to the IC. Connect two of your logic voltage sources to the inputs of one of the NAND gates, leaving them also connected to the logic indicators. Connect the output of the NAND gate to another logic indicator. Finally, connect all six of the unused input to each other and to ground. (The reason to do this is that if the inputs are floating, the IC can draw much more power than if the inputs are 1 or 0. You should always ground unused inputs for circuits for a permanent application. It's kind of optional for temporary breadboard circuits.)

For your report: Measure the output of the NAND gate for all four possible input states (00, 01, 10, and 11). Summarize your results with a truth table. Does it agree with what you expect? Finally, measure and report the actual output voltages of the NAND gate for all four states. You should find a result that is very typical for CMOS logic gates.

Next replace the 74HCT00N with the 7400N. (Don't forget to turn off the power when doing this.) This is the 7400 TTL family version of the quad NAND gate. Since it has the same pinout and runs on the same supply voltage as the 74HCT00N, you don't need to change your circuit.

For your report: Repeat the steps you just did. Do you still get the same truth table, experimentally? Is the output voltage for the TTL version of the gate the same as the CMOS version, or different?

4. NAND gate circuits

The NAND gate is an example of a *universal logic gate.* This means that any Boolean logic function can be implemented only with NAND gates. In this part of the lab we'll work through two examples of this.

For this part, let your output 1 be the value of Boolean variable A, output 2 the value of variable B, output 3 the value of variable C, and output 4 the value of variable D. Leave your four sources connected to the #1, 2, 3, and 4 logic indicators, and connect your circuit output to the #8 logic indicator.

a) <u>NAND implementation of OR gate</u>

Build a circuit using NAND gates that implements the function A OR B. Measure the output result for all possible inputs.

For your report: Give your measured results in the form of a truth table.

b) A four input NAND gate circuit.

(ii) Build the circuit shown in Figure IX-5. Measure the output Y for all possible input values of A, B, C, and D.

For your report: Give your measured results in the form of a truth table. Include a representative photo of the logic indicator output for one input example. What logic function does this circuit carry out?

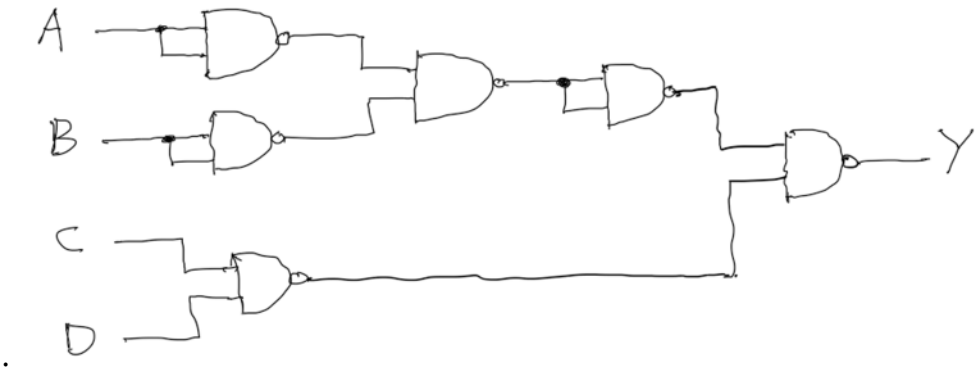
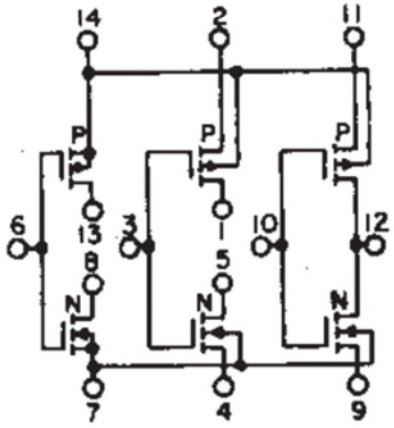


Figure IX-5. A circuit consisting of six NAND gates, with four inputs and one output.

5. CMOS logic gates from individual transistors.

(a)



(b)

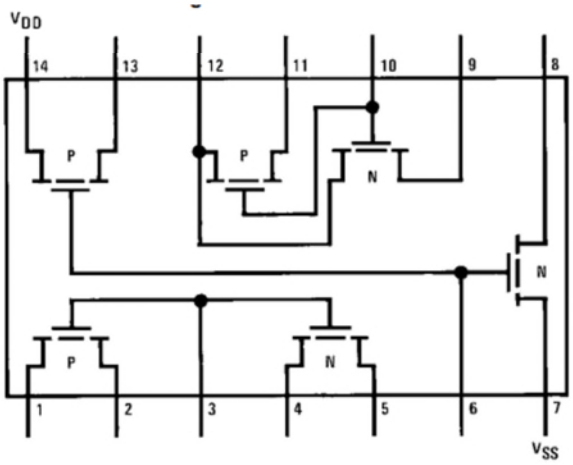


Figure IX-6 (a) Pin-out of the CD4007 dual complementary CMOS pair plus inverter. (b) Same diagram except showing the pins in the order you see them on the IC. For this diagram note that all p-channel substrates are connected to V_{dd} and all n-channel substrates are connected to V_{ss}, but the connections are not shown on the diagram.

For this part of the lab you will use individual transistors that are available in the CD4007 IC. It has three pairs of complementary MOSFET transistors with pin-out and internal wiring as shown in Figure IX-6. Note that the pair connected to pins 9 through 12 is connected internally as an inverter, while the other two pair are not. For this reason you have access to the individual pmos

and nmos transistors connected to pins 2 though 8 and 13 and 14, but not to the pair configured as an inverter.

a) Inverter with passive pull-up

Using one of the nmos transistors from the CD4007, build the inverter with passive pull-up shown in Figure IX-7. Drive the input of this circuit with the TTL output of a function generator and view the output on an oscilloscope. Verify that the circuit does invert. Now, turn the frequency up as high as you can. What happens now?

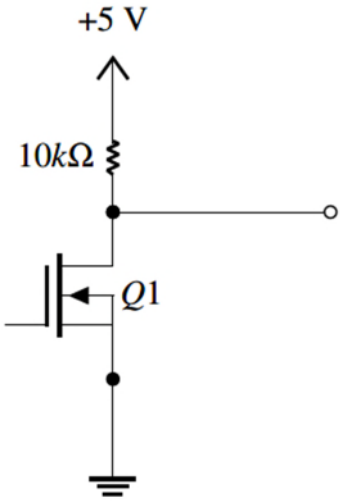


Figure IX-7. NMOS inverter with passive pull-up.

For your report: Include a representative oscilloscope trace showing input and output at a low frequency and at a high frequency. Give a qualitative description of the circuit behavior. Also, measure the risetime and fall time of the circuit output and include the result in your report.

b) Inverter with active pull-up

Using two transistors in the CD4007, build the inverter with active pull-up shown in Figure IX-8. Again, drive your circuit with the TTL output of a function generator and display the input and output on an oscilloscope. Verify that the circuit does invert, and again examine the behavior of the circuit at low and high frequency.

For your report: Include a representative oscilloscope trace showing input and output at a low frequency and at a high frequency. Give a qualitative description of the circuit behavior. Also, measure the risetime and fall time of the circuit output and include the result in your report. What are the main differences between the circuit with active pull-up and the circuit with passive pull-up. Explain why these differences occur. This provides one of the main reasons that essentially all logic circuits use active pull-up.

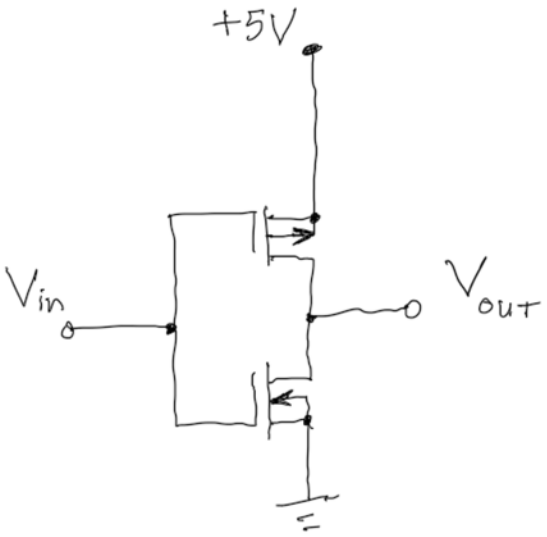


Figure IX-8. CMOS inverter with active pull-up.

c) CMOS NAND gate constructed from individual transistors

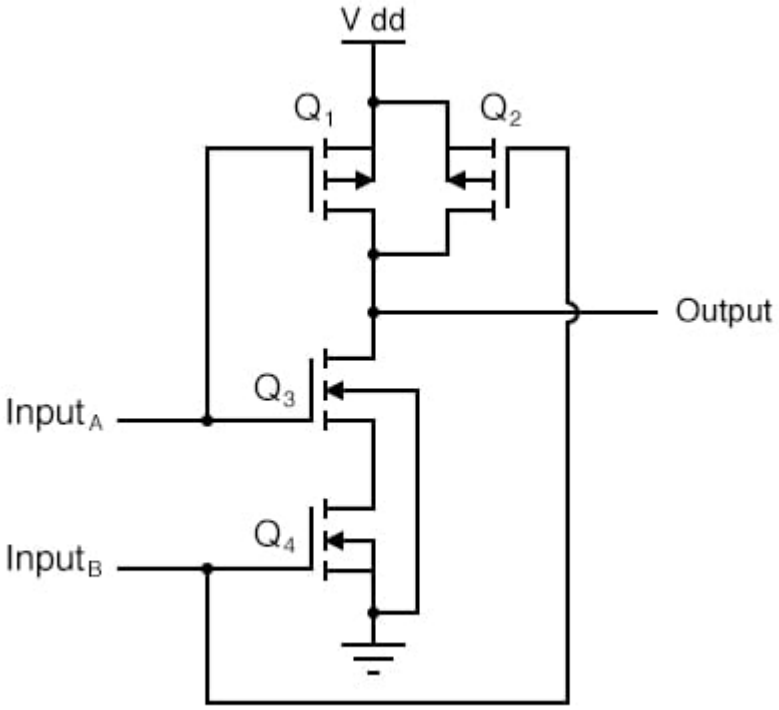


Figure IX-9. Circuit diagram of CMOS NAND gate

Figure IX-9 shows the circuit diagram for a CMOS NAND gate. Build this circuit using four of the transistors in the CD4007 IC. Connect to its two inputs and verify that it does operate as a NAND gate. Comment: this is quite tricky. Pay close attention to both your added circuit connections and to the internal connections within the CD4007. Also, the CD4007 burns our

rather easily. Make sure to leave the power off when constructing or making changes to your circuit, and take some care in making connections so that you have the best chance to have the circuit correct before turning the power on.

For your report: Show the circuit diagram your used with just the CD4007 pins and your external connections, and also another diagram showing the complete circuit with the individual transistor connections. Also show the CD4007 pins on the second diagram. Show your result for the experimental truth table that you produced with your circuit.