**S. Y. B. Tech (EE)**

**Trimester: V Subject:** Analog and Digital Integrated Circuits

**Name: Class:**

**Roll No: Batch:**

**Experiment No:**

**Name of the Experiment:** CMOS logic gates.

**Marks**

**Teacher’s Signature with date**

**Performed on:**

**Submitted on:**

**Aim**: **CMOS logic Gates**

**Pre-requisite:**

**Objectives:**

* Measure and Verify output voltage practically and theoretically.
* Calculate resolution, step size and few more specifications.

**Components and equipment required:**

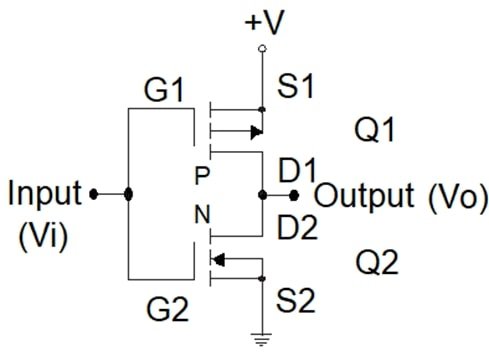
~~Bread Board, Op-Amp-OP-07C, Resistors, Dual Power Supply and, DMM~~

**Theory:**

[https://eepower.com/technical-articles/basic-cmos-logic-gates/#](https://eepower.com/technical-articles/basic-cmos-logic-gates/)

## The CMOS Inverter or NOT Gate

A NOT gate reverses the input logic state. Figure 1 shows a NOT gate employing two series-connected enhancement-type MOSFETS, one n-channel (NMOS) and one p-channel (PMOS).



##### ***Figure 1.*** A CMOS NOT gate.

The input is connected to the gate terminal of the two transistors, and the output is connected to both drain terminals.

Applying +V (logic 1) to the input (Vi), transistor Q2 is “on,” and transistor Q1 remains “off.” Under this condition, the output voltage (Vo) is close to 0 V (logic 0).

Connecting the input to ground (Vi = 0 V), transistor Q2 is “off,” and transistor Q1 is “on.” Now, the output voltage is close to +V (logic 1).

Table 1 summarizes these results.

|  |  |
| --- | --- |
| A | Y |
| 0 | 1 |
| 1 | 0 |

##### ***Table 1.*** The truth table for a NOT circuit.

## The CMOS NAND Gate

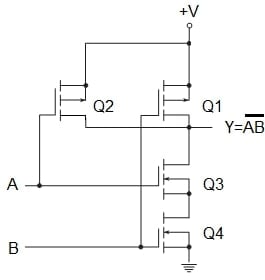
NAND denotes NOT-AND.

Table 2 shows the truth table for a NAND circuit.

|  |  |  |
| --- | --- | --- |
| A | B | Y |
| 0 | 0 | 1 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

##### ***Table 2.*** The truth table for a two-input NAND circuit.

Figure 2 shows a CMOS two-input NAND gate. P-channel transistors Q1 and Q2 are connected in parallel between +V and the output terminal. N-channel transistors Q3 and Q4 are connected in series between the output terminal and ground.



##### ***Figure 2.*** A CMOS two-input NAND gate.

With Q3 and Q4 transistors ”on” and Q1 and Q2 transistors “off,” the output is a logic 0. This condition happens when both inputs, A and B, are logic 1, confirming the lowest row in the above truth table.

With logic 0 in inputs A and B, Q3 and Q4 transistors are “off,” and Q1 and Q2 transistors are “on,” producing a logic 1 output. This is consistent with the first row of the truth table.

When one of the inputs is a logic “1” and the other one is a logic “0”, either Q3 is “off” and Q2 is “on” or Q4 is “off” and Q1 is “on.” The output in both cases is a logic “1,” validating the second and the third rows of the truth table.

## The NOR Gate

NOR signifies NOT-OR.

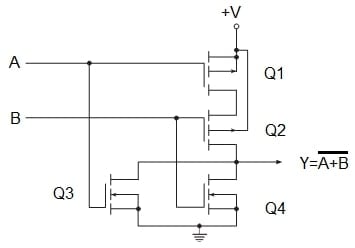
Table 3 shows the truth table for a NOR circuit.

|  |  |  |
| --- | --- | --- |
| A | B | Y |
| 0 | 0 | 1 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 0 |

##### ***Table 3.*** The truth table for a two-input NOR circuit.

The output of a NOR gate is logic 1 with logic 0 in both inputs. The outcomes for other input combinations are logic 0.

Figure 3 shows a CMOS two-input NOR gate. P-channel transistors Q1 and Q2 are connected in series between +V and the output terminal. N-channel transistors Q3 and Q4 are connected in parallel between the output and ground.



##### ***Figure 3.*** A CMOS two-input NOR gate.

When both inputs, A and B, are logic 0, Q1 and Q2 are “on,” and Q3 and Q4 are “off,” and the output is logic 1. This confirms the first row of the truth table above.

With both inputs logic 1, Q3 and Q4 are “on,” and Q1 and Q2 are “off,” producing a logic 0 output that confirms the last row of the truth table.

For the two remaining input combinations, either Q1 is “off” and Q3 is “on” or Q2 is “off” and is Q4 “on”. In these cases, the output is logic 0 which is consistent with the above truth table.

## The AND Gate

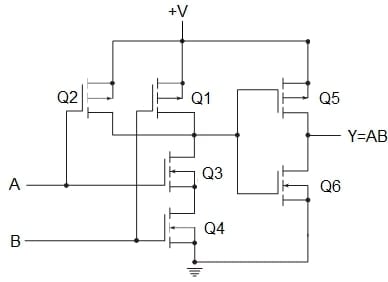
We can say that an AND gate is a NOT-NOT-AND or NOT-NAND. Then, it is just a NAND gate followed by an inverter.

Table 4 shows the truth table for an AND circuit.

|  |  |  |
| --- | --- | --- |
| A | B | Y |
| 0 | 0 | 0 |
| 0 | 1 | 0 |
| 1 | 0 | 0 |
| 1 | 1 | 1 |

##### ***Table 4.*** The truth table for a two-input CMOS AND circuit.

Figure 4 shows a CMOS two-input AND gate.



##### ***Figure 4.*** A CMOS two-input AND gate.

## The OR Gate

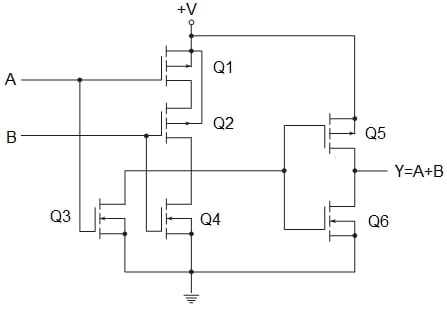
An OR gate is a NOT-NOT-OR or NOT-NOR. Then, it is a NOR gate followed by an inverter.

Table 5 shows the truth table for the OR circuit.

|  |  |  |
| --- | --- | --- |
| A | B | Y |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 1 |

##### ***Table 5.*** The truth table for a two-input OR circuit.

Figure 5 shows a CMOS two-input OR gate.



##### ***Figure 5.*** A CMOS two-input OR gate.

## The Exclusive OR (XOR) Gate

The output of a two-input XOR circuit assumes the logic 1 state if one and only one input assumes the logic 1 state.

An equivalent logic statement is: ”If B=1 and A=0, or if A=1 and B=0, then Y=1.”

In Boolean notation,

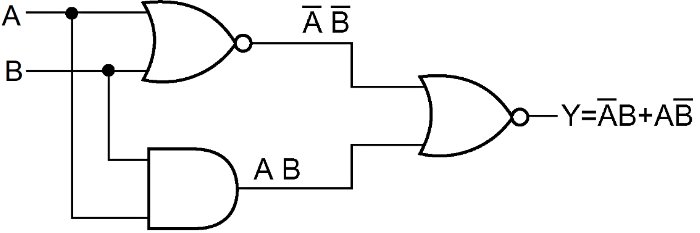
Y=¯AB+A¯BY=A¯B+AB¯

Table 6 shows the truth table for a two-input XOR circuit.

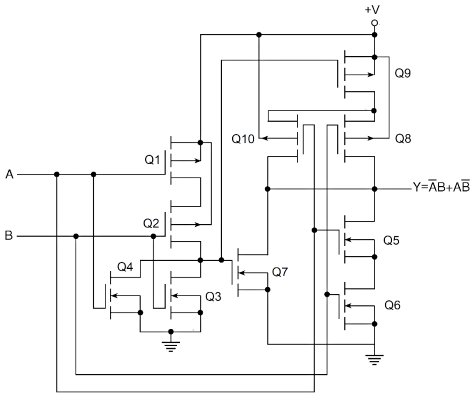
|  |  |  |
| --- | --- | --- |
| A | B | Y |
| 0 | 0 | 0 |
| 0 | 1 | 1 |
| 1 | 0 | 1 |
| 1 | 1 | 0 |

##### ***Table 6.*** The truth table for a two-input XOR circuit.

Figure 6 shows a two-input logic diagram, and figure 7 shows a CMOS circuit to satisfy the Boolean equation.



##### ***Figure 6.*** A logic block diagram for the XOR circuit.

**

##### ***Figure 7.*** A CMOS two-input XOR gate.

Transistors Q1, Q2, Q3, and Q4 comprise the NOR gate. Transistors Q5 and Q6 make the ANDing of inputs A and B, and transistor Q7 supplies the ORing of the NOR output with the ANDed output. Transistors Q8, Q9, and Q10 complement the arrangement of transistors Q5, Q6, and Q7, inverting the result.

## About the Basic CMOS Logic Gates

Combinations of n- and p-channel transistors allow the construction of logic building blocks.

The inverter, NAND, and NOR logic building blocks are the backbone of most digital logic families.

Two primary connections are the two-input NAND gate and the two-input NOR gate.

A NAND gate places two n-channel transistors in series to ground and two p-channel transistors in parallel connected to +V. Only when both inputs are logic 1, the output goes to logic 0.

A NOR gate arranges two n-channel transistors in parallel so that either one can pull the output to ground (logic 0) for a logic 1 (+V) input. It also places two p-channel transistors in series, which must work together to pull the output to logic 1 for logic 0 in both inputs. The output will go to logic 1 only when both A and B are logic 0.

AND and OR gates are NAND and NOR gates followed by an inverter.

An important function that is often needed in logic design is the Exclusive-OR (XOR), with the Boolean expression.

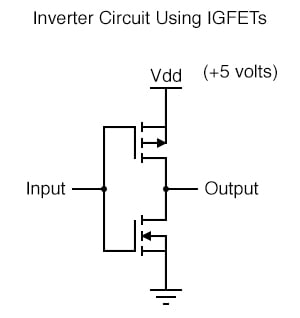
Y=¯AB+A¯BY=A¯B+AB¯

The XOR is not a primary gate but constructed by a combination of other logic gates.

<https://www.allaboutcircuits.com/textbook/digital/chpt-3/cmos-gate-circuitry/>

### Field-Effect Transistors

Field-effect transistors, particularly the insulated-gate variety, may be used in the design of gate circuits. Being voltage-controlled rather than current-controlled devices, IGFETs tend to allow very simple circuit designs. Take for instance, the following inverter circuit built using P- and N-channel IGFETs:



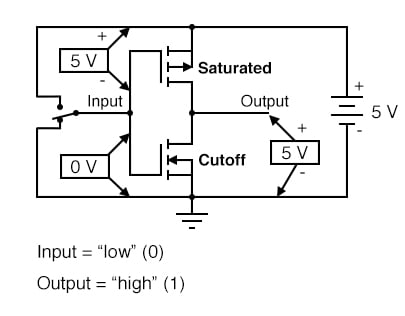
Notice the “Vdd” label on the positive power supply terminal. This label follows the same convention as “Vcc” in TTL circuits: it stands for the constant voltage applied to the drain of a field effect transistor, in reference to ground.

### Field Effect Transistors in Gate Circuits

Low Input

Let’s connect this gate circuit to a power source and input switch, and examine its operation. Please note that these IGFET transistors are E-type (Enhancement-mode), and so are normally-off devices.

It takes an applied voltage between gate and drain (actually, between gate and substrate) of the correct polarity to bias them on.



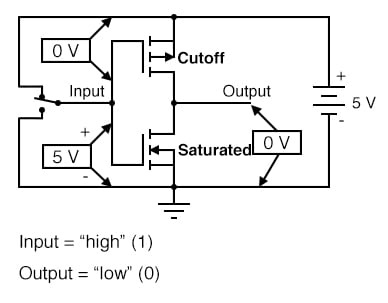
The upper transistor is a P-channel IGFET. When the channel (substrate) is made more positive than the gate (gate negative in reference to the substrate), the channel is enhanced and current is allowed between source and drain.

So, in the above illustration, the top transistor is turned on. The lower transistor, having zero voltage between gate and substrate (source), is in its normal mode: off.

Thus, the action of these two transistors are such that the output terminal of the gate circuit has a solid connection to Vdd and a very high resistance connection to ground. This makes the output “high” (1) for the “low” (0) state of the input.

#### High Input

Next, we’ll move the input switch to its other position and see what happens:



Now the lower transistor (N-channel) is saturated because it has sufficient voltage of the correct polarity applied between gate and substrate (channel) to turn it on (positive on gate, negative on the channel). The upper transistor, having zero voltage applied between its gate and substrate, is in its normal mode: off.

Thus, the output of this gate circuit is now “low” (0). Clearly, this circuit exhibits the behavior of an inverter, or NOT gate.

### Complementary Metal Oxide Semiconductors (CMOS)

Using field-effect transistors instead of bipolar transistors has greatly simplified the design of the inverter gate. Note that the output of this gate never floats as is the case with the simplest TTL circuit: it has a natural “totem-pole” configuration, capable of both sourcing and sinking load current.

Key to this gate circuit’s elegant design is the complementary use of both P- and N-channel IGFETs. Since IGFETs are more commonly known as MOSFETs (**M**etal-**O**xide-**S**emiconductor **F**ield **E**ffect **T**ransistor), and this circuit uses both P- and N-channel transistors together, the general classification given to gate circuits like this one is CMOS: **C**omplementary **M**etal **O**xide **S**emiconductor.

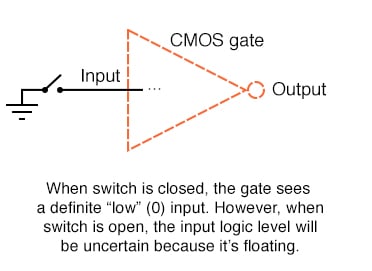
### CMOS Gates: Challenges and Solutions

CMOS circuits aren’t plagued by the inherent nonlinearities of the field-effect transistors, because as digital circuits their transistors always operate in either the saturated or cutoff modes and never in the active mode. Their inputs are, however, sensitive to high voltages generated by electrostatic (static electricity) sources, and may even be activated into “high” (1) or “low” (0) states by spurious voltage sources if left floating.

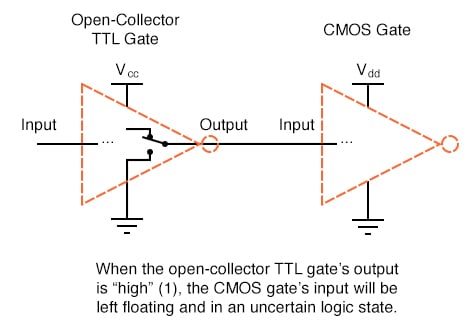
For this reason, it is inadvisable to allow a CMOS logic gate input to float under any circumstances. Please note that this is very different from the behavior of a TTL gate where a floating input was safely interpreted as a “high” (1) logic level.

#### CMOS Problems with Floating Inputs

This may cause a problem if the input to a CMOS logic gate is driven by a single-throw switch, where one state has the input solidly connected to either Vdd or ground and the other state has the input floating (not connected to anything):



 Also, this problem arises if a CMOS gate input is being driven by an open-collector TTL gate. Because such a TTL gate’s output floats when it goes “high” (1), the CMOS gate input will be left in an uncertain state:

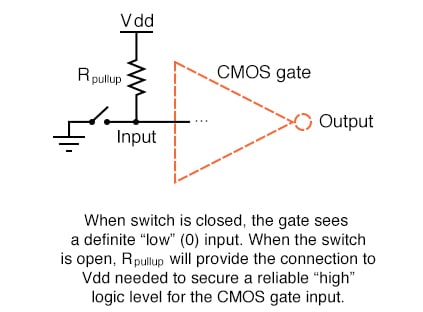


#### Solution to Floating Inputs

Pullup Resistors

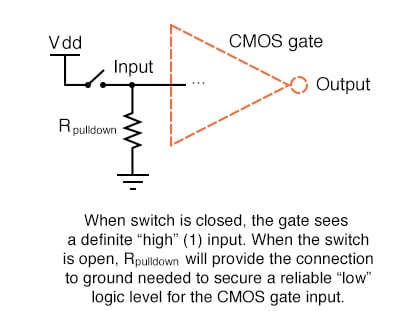
Fortunately, there is an easy solution to this dilemma, one that is used frequently in CMOS logic circuitry. Whenever a single-throw switch (or any other sort of gate output incapable of both sourcing and sinking current) is being used to drive a CMOS input, a resistor connected to either Vdd or ground may be used to provide a stable logic level for the state in which the driving device’s output is floating.

This resistor’s value is not critical: 10 kΩ is usually sufficient. When used to provide a “high” (1) logic level in the event of a floating signal source, this resistor is known as a pullup resistor:

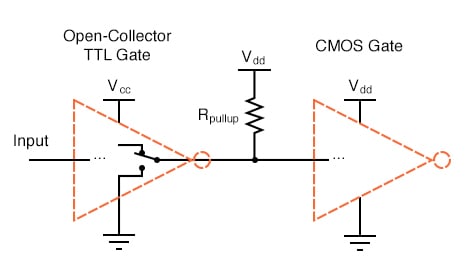


Pulldown Resistors

 When such a resistor is used to provide a “low” (0) logic level in the event of a floating signal source, it is known as a pulldown resistor. Again, the value for a pulldown resistor is not critical:

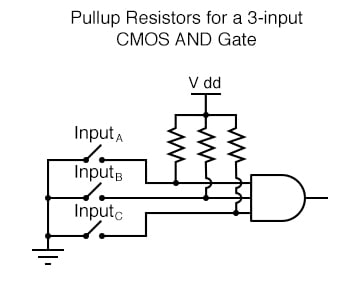


 Because open-collector TTL outputs always sink, never source, current, pullup resistors are necessary when interfacing such an output to a CMOS gate input:



Multiple Pullup and Pulldown Resistors

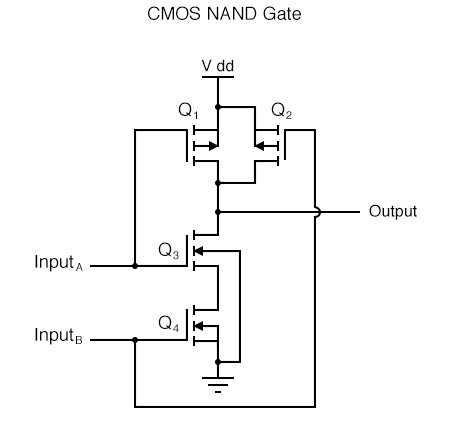
Although the CMOS gates used in the preceding examples were all inverters (single-input), the same principle of pullup and pulldown resistors applies to multiple-input CMOS gates. Of course, a separate pullup or pulldown resistor will be required for each gate input:



 This brings us to the next question: how do we design multiple-input CMOS gates such as AND, NAND, OR, and NOR? Not surprisingly, the answer(s) to this question reveal a simplicity of design much like that of the CMOS inverter over its TTL equivalent.

### CMOS NAND Gates

For example, here is the schematic diagram for a CMOS NAND gate:



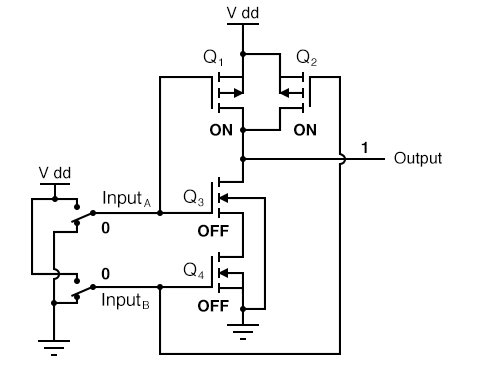
Notice how transistors Q1 and Q3 resemble the series-connected complementary pair from the inverter circuit. Both are controlled by the same input signal (input A), the upper transistor turning off and the lower transistor turning on when the input is “high” (1), and vice versa.

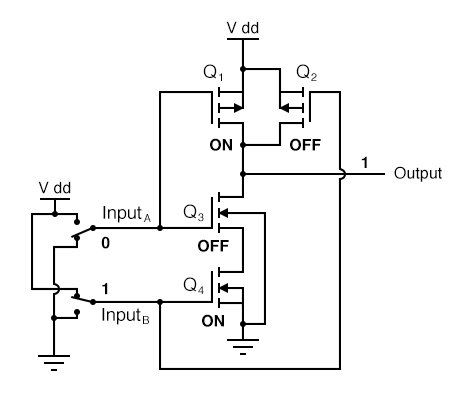
Notice also how transistors Q2 and Q4 are similarly controlled by the same input signal (input B), and how they will also exhibit the same on/off behavior for the same input logic levels. The upper transistors of both pairs (Q1 and Q2) have their source and drain terminals paralleled, while the lower transistors (Q3 and Q4) are series-connected.

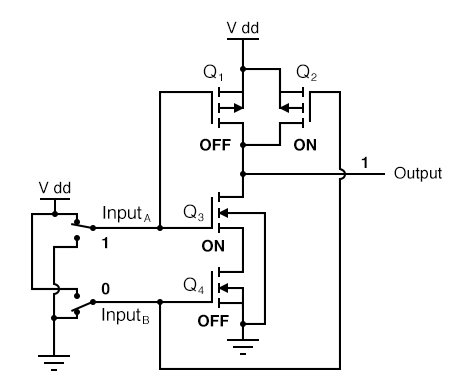
What this means is that the output will go “high” (1) if either top transistor saturates, and will go “low” (0) only if both lower transistors saturate.

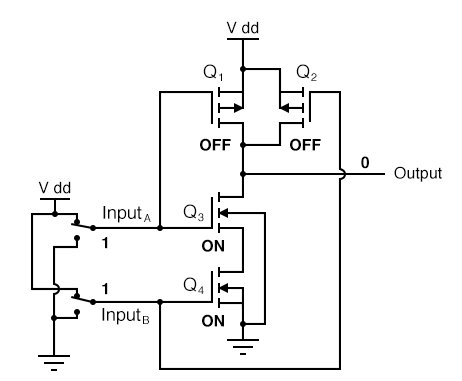
#### CMOS Circuit Behaviors for All Logic Inputs

The following sequence of illustrations shows the behavior of this NAND gate for all four possibilities of input logic levels (00, 01, 10, and 11):



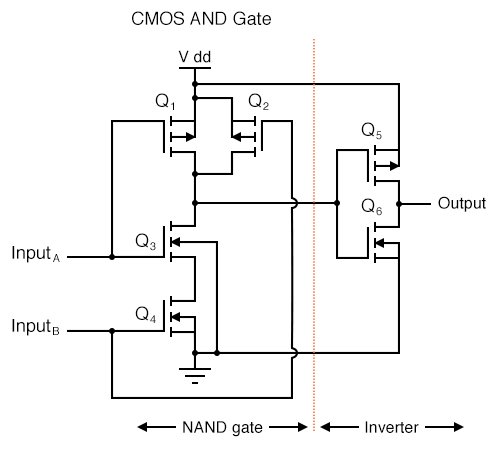






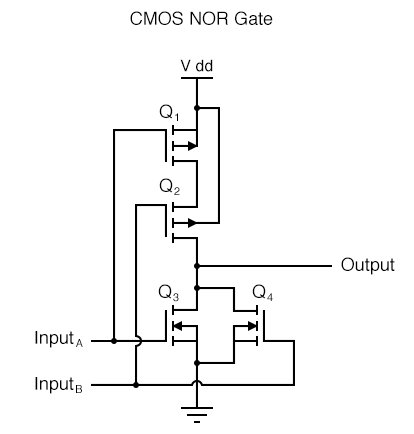
#### CMOS AND gate

As with the TTL NAND gate, the CMOS NAND gate circuit may be used as the starting point for the creation of an AND gate. All that needs to be added is another stage of transistors to invert the output signal:



### CMOS NOR Gates

A CMOS NOR gate circuit uses four MOSFETs just like the NAND gate, except that its transistors are differently arranged. Instead of two paralleled sourcing (upper) transistors connected to Vdd and two series-connected sinking (lower) transistors connected to ground, the NOR gate uses two series-connected sourcing transistors and two parallel-connected sinking transistors like this:

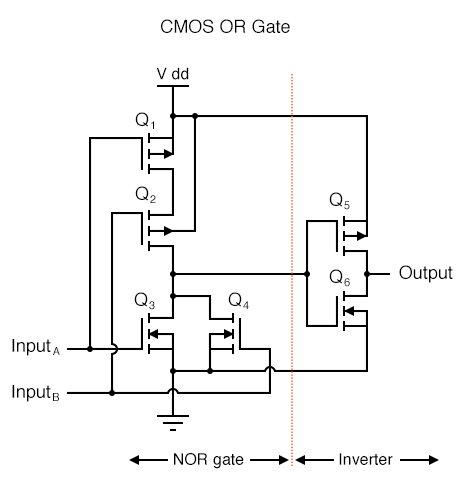


 As with the NAND gate, transistors Q1 and Q3 work as a complementary pair, as do transistors Q2 and Q4. Each pair is controlled by a single input signal. If either input A or input B are “high” (1), at least one of the lower transistors (Q3 or Q4) will be saturated, thus making the output “low” (0).

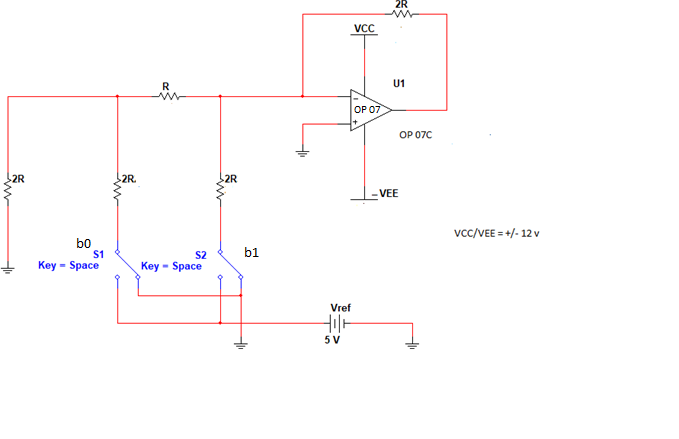
Only in the event of both inputs being “low” (0) will both lower transistors be in cutoff mode and both upper transistors be saturated, the conditions necessary for the output to go “high” (1). This behavior, of course, defines the NOR logic function.

### CMOS OR Gates

The OR function may be built up from the basic NOR gate with the addition of an inverter stage on the output:



### Circuit Diagram:



**Figure 8.3: 2 bit R-2R DAC**

### Design Steps:

Select two resistors of type R and 2R for DAC

Let’s assume R =7.5K so then 2R =15K (by design).

Calculate Output Voltage by the given formula in equation 8.2.

**Calculations: Calculate output voltage for different combinations of b1 and b0**

Vo = ……….(8.2)

Attach separate sheet

**Observation Table:**

|  |  |  |  |
| --- | --- | --- | --- |
| **b1(MSB)** | **b0(LSB)** | **Vtheorotical** | **Vpractical** |
| 0 | 0 |  |  |
| 0 | 1 |  |  |
| 1 | 0 |  |  |
| 1 | 1 |  |  |

**Conclusion:**

**Post Lab Questions:**

1. Why Precision Op-amp is used to perform this practical?

2. What would be the change in resolution formula if we increase the number of bits?

3. If we need to make the resolution better what changes should be made in circuit?

**Additional links for more information:**

<https://www.elprocus.com/digital-to-analog-converter-dac-applications>

http://ume.gatech.edu/mechatronics\_course/DAC\_S06.ppt