In this section, we will give just a little taste of how the computer digital logic in the computer works by designing a NOT gate with P-channel and N-channel transistors.

DESIGN OF A NOT GATE

0:00 / 3:57 1.0x

DR. JONATHAN VALVANO: Hi.

In this video, we will introduce digital logic--

not to teach you everything about it, but give you a glimpse

of how things work.

In digital logic, we have two states.

The true state is going to be mapped to a high voltage--

in our case, 3.3 volts--

and a false state is going to be mapped to 0 volts.

And the way this is going to work is around two transistors.

The first is a P-channel.

The P-channel has three pins, it has a gate, it has a source,

and it has a drain.

And the way the P-channel works is, if there is a positive voltage--

a positive voltage--

between the source down to the gate,

then this transistor will be on, and

can conduct current down, from source to drain.

The other type of transistor also has three pins.

It's called an N-channel.

And the N-channel has a gate.

And this one's the drain.

And this one's the source.

But it works differently.

In the N-channel, a positive voltage between the gate and the source will

turn the N-channel on.

But now the current will go in this direction, from drain to source.

And what we're going to do is to combine these two

transistors to make a gate.

And the first one we're going to build is an NOT gate.

The NOT gate has an input, which can be True or False, and an output, which

is the opposite.

So True goes to False, and False goes to True.

And the way we're going to build it is we're going to tie the two gates

together here.

And that will be our input.

And we're going to tie the two drains together here.

And that will be our output.

And then, we're going to tie this source, here, to 3.3 volts.

And we're going to tie this one to ground.

Now let's see how it works.

If the input were True, OK-- so if the input were True--

that means it is 3.3 volts.

And that will create--

across this transistor, you'll see there's 3.3 volts in this direction.

And this transistor will then become on.

But over here, we see we've got 3.3 volts to 3.3 volts.

And so there's no voltage across the source gate of the P. And this one

will be off.

And if this transistor's off, and that one's on, we can see--

there's 0 volts--

that becomes 0 volts.

The other case is also easy to see.

Now if the input were False, that means there'd be 0 volts here.

And on the N-channel, we have 0 volts to 0 volts, which is no drop.

And so this one will go off.

But over here, we have 3.3 volts down to 0 volts.

And so we see a positive voltage between the source

and gate of the P-channel.

And it will come on and conduct.

So if this transistor is on, and conducts, this 3.3 volts shows up at

the output as 3.3 volts.

Help

Transistors made with metal oxide semiconductors are called MOS. In the digital world MOS transistors can be thought of as voltage controlled switches. Circuits made with both p-type and n-type MOS transistors are called complementary metal oxide semiconductors or CMOS. The 74HC04 is a high-speed CMOS NOT gate, as shown in Figure 4.4.

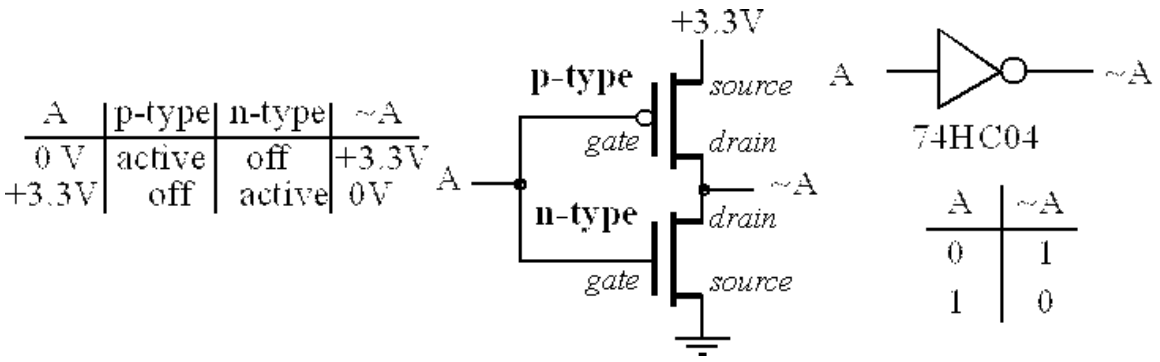

Figure 4.4. CMOS implementation of a NOT gate.

There are just a few rules one needs to know for understanding how CMOS transistor-level circuits work. Each transistor acts like a switch between its source and drain pins. In general, current can flow from source to drain across an active p-type transistor, and no current will flow if the switch is open. From a first approximation, we can assume no current flows into or out of the gate. For a p-type transistor, the switch will be closed (transistor active) if its gate is low. A p-type transistor will be off (its switch is open) if its gate is high.

The gate on the n-type works in a complementary fashion, hence the name complementary metal oxide semiconductor. For an n-type transistor, the switch will be closed (transistor active) if its gate is high. An n-type transistor will be off (its switch is open) if its gate is low. Therefore, consider the two possibilities for the circuit in Figure 4.4. If the input *A* is high (+3.3V), then the p-type transistor is off and the n-type transistor is active. The closed switch across the source-drain of

the n-type transistor will make the output low (0V). Conversely, if *A* is low (0V), then p-type transistor is active and the n-type transistor is off. The closed switch across the source-drain of the p-type transistor will make the output high (+3.3V).

The AND, OR, EOR digital logic takes two inputs and produces one output; see Figure 4.5 and Table 4.1. We can understand the operation of the AND gate by observing the behavior of its six transistors. If both inputs *A* and *B* are high, both T3 and T4 will be active. Furthermore, if *A* and *B* are both high, T1 and T2 will be off. In this case, the signal labeled **~(A&B)** will be low because the T3–T4 switch combination will short this signal to ground. If *A* is low, T1 will be active and T3 off. Similarly, if *B* is low, T2 will be active and T4 off. Therefore if either *A* is low or if *B* is low, the signal labeled **~(A&B)** will be high because one or both of the T1, T2 switches will short this signal to +3.3V. Transistors T5 and T6 create a logical complement, converting the signal **~(A&B)** into the desired result of *A*&*B*. We can use the **and** operation to extract, or **mask**, individual bits from a value.

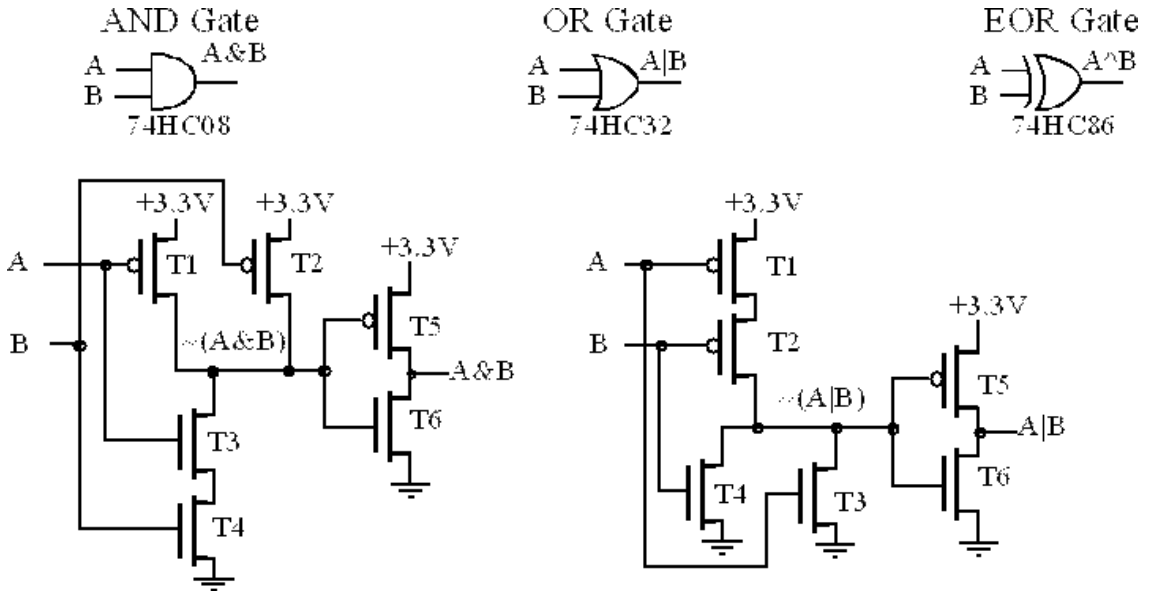

Figure 4.5. Logical operations can be implemented with discrete transistors or digital gates.

A	B	AND	NAND	OR	NOR	EOR	Ex NOR
0	0	0	1	0	1	0	1
0	1	0	1	1	0	1	0
1	0	0	1	1	0	1	0
1	1	1	0	1	0	0	1
Symbol		A&B	~(A&B)	A|B	~(A|B)	A^B	~(A^B)

Table 4.1. Two-input one-output logical operations.