Physics 310
Lecture 8a – Digital Circuits

Mon. 3/12 Wed. 3/14 Thurs. 3/15 Fri. 3/16	**Ch 11:** Digital Circuits **Ch's 11 & 12:** Digital Circuits **Lab 8:** Digital Circuits More of the same **Quiz** Ch's 11 & 12	 **HW 8:** Ch11 Pr.2,8,9*; Ch12 Pr 1,5 **Lab 8 Notebook**
Mon. 3/29	**Ch 14.1, .6-.10; pp 373-374 (**Sampling Frequency**); 12.6:** ADC & DAC	**Project Proposal**

Project Proposals Due Next Monday

Study List for Quiz #8:

1. Logic gates: AND, NAND, OR, NOR, XOR, XNOR, and NOT (inverter).
2. Know how to use the truth tables for logic gates.
3. Data and JK Flip-Flops

Logic Gates

- Symbols
- Truth Tables
- TTL & CMOS

Boolean Algebra

- can make anything with NAND or NOR gates!
- De Morgan's theorem

Numbering systems & codes – decimal, binary, BCD

(Open –collector & three-state logic – NO TIME!)

Flip-Flops

- RS
- Clocked RS – also the different types of clocking
- Data – count-by-two ciruit
- JK – count-to-16 circuit

BCD-to-decimal Decoder & Seven-segment Display

Count-by-3 circuit design (the decade counter is similar!)

Synchronous vs. ripple counters

1. Could you please go over the NOR and NAND gates? I'm not sure I really understand how they work and what their purpose is.
2. Could you go over the different logic lines?
3. Why are there so many different types of flip-flops?

4. 1. Could you go over or explain the Schmitt Trigger inverter as a simple low power oscillator?
5.
6. 2. What are components in the logic line actually used for? Just a premade "circuit" to supply a certain voltage to an instrument?

Physics 310
Lecture 8a – Digital Circuits

7.
8. 3. Working with Boolean Algebra, like solving theorems 10 to 17. I don't understand how they work.
9. Can we go over an example of power dissipation and how you figure out the power requirements of a logic device. – to appreciate why there are multiple families out there, it's worth *appreciating* the power dissipation issue, but we won't get into really calculating it.
10.
11. How these logic lines the only ones there are or can more be discovered and created? These are certainly the big ones that get used a lot; perhaps there are some other ones, but they are not as prevalent (and for the sake of compatibility, one doesn't want too much diversity.)
12. Can we just go over identifying and building different logic lines?
13. Could we try to work through some examples of boolean algebra on our own or with partners?

Chapters 11 & 12 Introduction.

You've already met the circuitry that performs the basics of what I'll call 'continuous' mathematics – adding (or subtracting), integrating, and differentiating (yes, there are even multipliers / dividers). But there's a whole other side to mathematics – discreet logic. You may be most familiar with this side in the outgrowth of Mathematic in what became Computer Science – all those IF…Then, or Do…Until, or And, Or, Not,… Just as there are electronic components to do the 'continuous' kind of math, there are components to do the discrete kind. For that matter, just as the 'continuous' math components are built of mostly transistors, so are the discrete math components.
The 555 timer provided our first occasion to think more in terms of discreet logic states of a "signal" (Hi or Lo) rather than the specific values the signal may assume. There are a handful devices that have been designed to perform the basic logical operations like you might employ when writing code or doing an online search AND, OR, NOT, NOR, and NAND, in addition to these, the flip-flop rounds out the building blocks of a logic, a.k.a., digital circuit. These are the subject of chapters 11 and 12 of our text.

Digital Circuitry is all about manipulating two states and only two states. In their native tongue (that of voltages), these states may be a Hi voltage (something near the positive rail) and a Lo voltage (something near ground). Depending on context, these two states are variously referred to as

$$\left.\begin{matrix} ON & HI & True & 1 \\ OFF & Lo & False & 0 \end{matrix}\right\} usually \begin{cases} \text{voltage near positive rail} \\ \text{ground} \end{cases}$$

Physics 310
Lecture 8a – Digital Circuits

11-2 Gates

The devices that handle those logical connectors, are called "logic gates," or just plain "gates." Here's a quick survey of the basics.

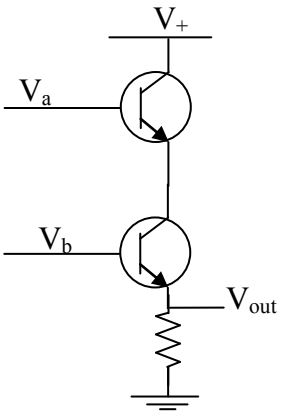
AND: *The output is "true" / Hi voltage only if both of the two inputs are "true"/ Hi voltage; otherwise, the output is "false" / Lo voltage. Of course that's the same way AND works in English too: 'True or False: I have an apple AND a banana: _____.*

Before slide: work out truth table for Apple AND Banana.

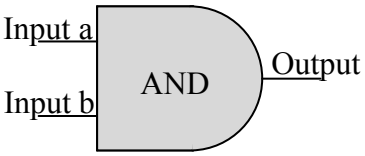
Truth Table

Input A	Input B	Output
0	0	0
1	0	0
0	1	0
1	1	1

Simplified inner workings



Symbol



I present the "simplified inner workings" not because we're going to focus that much 'under the hood' of these gates at this moment, but so you can appreciate that gates *can* be made of transistors and have a passing familiarity with how. Thinking through the "inner workings", each transistor's base voltage needs to be above its emitter's voltage for the transistor to 'turn on' and conduct current; so as long as one or the other base voltage is grounded, negligible current passes through the resistor, and so V_{out} is essentially ground.

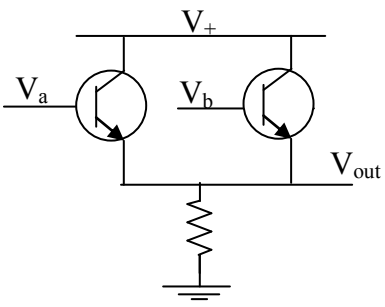
OR: *The output is "true" / Hi voltage if either one or the other (or both) input is "true" / Hi.* Of course, this is how OR is used in English too: *True or False: I have an apple OR a banana:_____.*

Before slide, work out truth table for Apple OR Banana

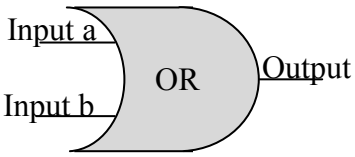
Truth Table

Input a	Input b	Output
0	0	0
1	0	1
0	1	1
1	1	1

Simplified Inner Workings



Symbol



Thinking through the "inner workings", if either one of the transistors' bases is set to a high voltage, then that one will allow current to pass through the resistor and thus allow V_{out} to be a higher voltage than ground.

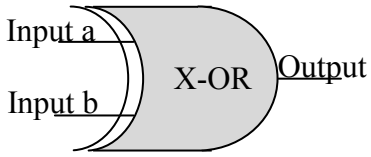
Now, in English, there's some ambiguity to "or": "do you have an apple or a banana?" what if you have *both* is the answer yes or no?

X-OR: (eXclusive OR). *The output is "true" / Hi voltage if one or the other input is "true" / Hi voltage,* but not if both are. In English, we usually just use "or" for this and judge by the context; for example, 'let me know if you've selected your fruit; either an apple or a banana.'

Truth Table

Input a	Input b	Output
0	0	0
1	0	1
0	1	1
1	1	0

Symbol

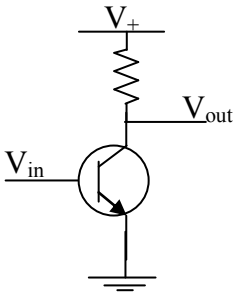


NOT: *Output is the opposite of the single input.* Again, this is the same way we can (juvenilely) use NOT in English: *It is True that I have an apple….NOT.*

Truth Table

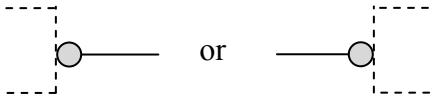
Input	Output
1	0
0	1

Simplified Inner Workings

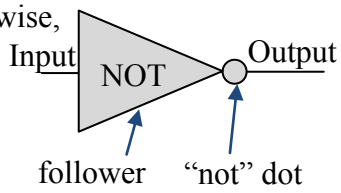


Symbol

When in series with another gate,



Otherwise,



Another name for a "NOT gate" is an "Inverter."

Three useful gates are made by tacking a Not on the end of an AND, an OR, and an X-OR.

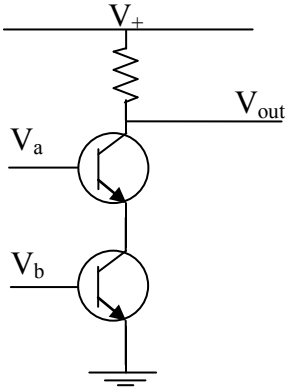
NAND: *Output is "true" / Hi voltage as long as at least one of the inputs is "false" / Lo.* In English: '*it's true that you* don't *have* both *an apple* and *a banana isn't it*?'

Truth Table

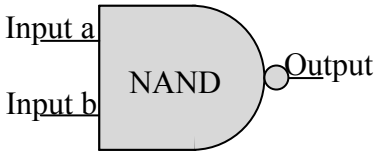
Input a	Input b	Output
0	0	1
1	0	1
0	1	1
1	1	0

Notice that this is the *opposite* Output as for an AND; i.e., it's What you'd get if you followed an AND with a NOT.

Simplified Inner Workings



Symbol



Physics 310
Lecture 8a – Digital Circuits

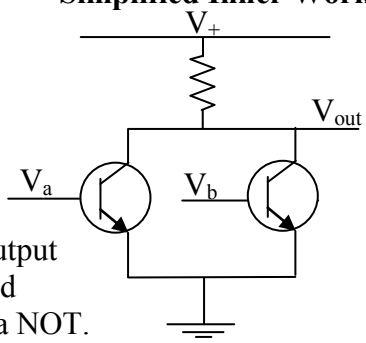
NOR: *Output is true if* neither *of the two inputs is true.* English example: *Is it true that I have neither an apple NOR a banana?*

Before slide, work out truth table.

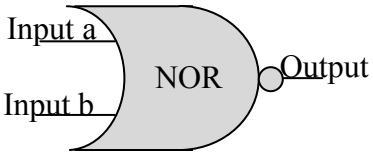
Truth Table

Input a	Input b	Output
0	0	1
1	0	0
0	1	0
1	1	0

Simplified Inner Workings



Symbol



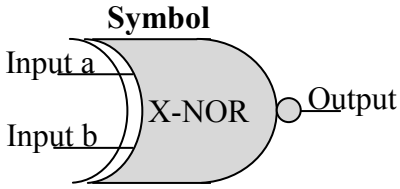
Notice that this is the *opposite* output
As for an OR; i.e., it's what you'd
Get if you followed an OR with a NOT.

X-NOR: (eXclusive NOR). *The output is "true"/Hi voltage as long as the inputs are the same as each other.*

Truth Table

Input a	Input b	Output
0	0	1
1	0	0
0	1	0
1	1	1

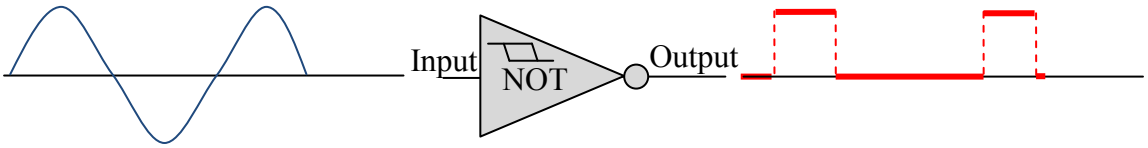
Symbol

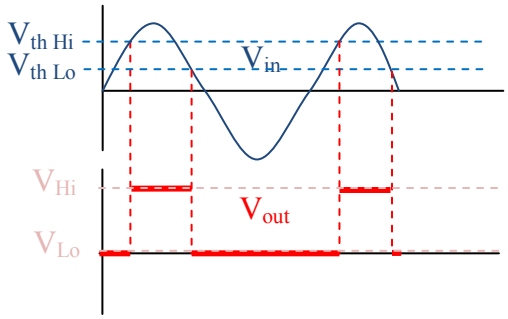


Physics 310
Lecture 8a – Digital Circuits

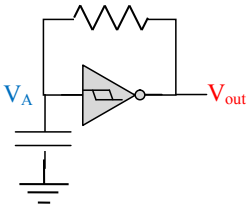
Schmitt Trigger & Hysteresis

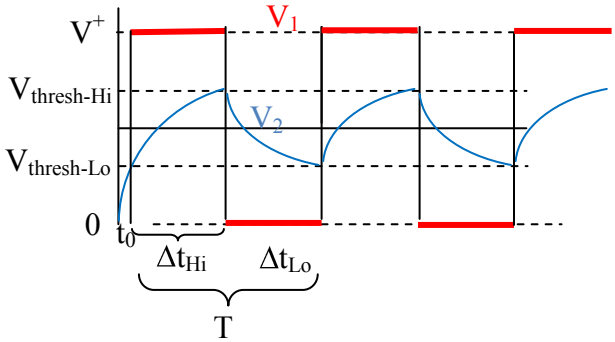
Saying that these are "two-state" devices is all good and well, but it's not like they *really* deal in 1's and 0's; they *really* deal in voltages which have a continuum of possible values. Just as with the comparators, there needs to be a *threshold* value; any voltage higher than it counts as "Hi"/ "True", or 1, and any voltage lower than it counts as "Lo"/ "False" or 0. Better yet, to be a little insulated from noise on the inputs, these devices should be Schmitt triggered. That is to say, they should have two thresholds; if the input crosses the upper one on the way up, it counts as "Hi" or crosses the lower one on the way down it counts as "Lo." We can see that play out for a sine-wave input to a NOT gate / inverter.





Someone asked, so just go over first part:
As has already been seen, in Chapter10, inverters with hysteresis can be used to generate square waves. Here's one more example.





$$T = \Delta t_{Hi} + \Delta t_{Lo}$$

$$V_A(t_o) = V_{th-Lo} = V^+(1-e^{-t_o/RC}) \Rightarrow t_o = -RC\ln\left(1-\frac{V_{th-Lo}}{V^+}\right) = RC\ln\left(\frac{V^+}{V^+ - V_{th-Lo}}\right)$$

$$V_A(t_o+\Delta t_{Hi}) = V_{th-Hi} = V^+(1-e^{-(t_o+\Delta t_{Hi})/RC}) \Rightarrow t_o+\Delta t_{Hi} = RC\ln\left(\frac{V^+}{V_{th-Hi}}-1\right)$$

$$\Delta t_{Hi} = RC\ln\left(\frac{V^+}{V_{th-Hi}}-1\right) - RC\ln\left(\frac{V^+}{V^+-V_{th-Lo}}\right)$$

$$\Delta t_{Hi} = RC\ln\left(\frac{\frac{V^+}{V_{th-Hi}}-1}{\frac{V^+}{V^+-V_{th-Lo}}}\right)$$

$$V_A(\Delta t_{Lo}) = V_{th-Lo} = V_{th-Hi}e^{-(\Delta t_{Lo})/RC} \Rightarrow \Delta t_{Lo} = RC\ln\left(\frac{V_{th-Hi}}{V_{th-Lo}}\right)$$

$$T = \Delta t_{Hi} + \Delta t_{Lo}$$

$$RC\ln\left(\frac{\frac{V^+}{V_{th-Hi}}-1}{\frac{V^+}{V^+-V_{th-Lo}}}\right) + RC\ln\left(\frac{V_{th-Hi}}{V_{th-Lo}}\right) = RC\ln\left(\frac{\frac{V^+}{V_{th-Hi}}-1}{\frac{V^+}{V^+-V_{th-Lo}}}\left(\frac{V_{th-Hi}}{V_{th-Lo}}\right)\right) = RC\ln\left(V^+ - V_{th-Hi}\left(\frac{1}{V_{th-Lo}}-\frac{1}{V^+}\right)\right)$$

Zoom in / Zoom out. Before we get to thinking about really *using* these gates, there are two distinct areas of background we should explore. While you're somewhat familiar with continuous mathematics, you may not have a lot of practice with discrete / logic mathematics; so we'll pull back and survey Boolean Algebra. Before we do that though, we'll zoom in and consider some of the real details of these logic gates, what's really under the hood so we can appreciate

a) how they relate to other circuitry devices and principles we've already met and

b) how different design choices impact their exact behavior – what's a cheap gate design, what's a fast gate design,…

11-3 Survey of IC Gates

Now it's time to look under the hood: not all AND's are "created equal." When you select a vehicle, first you decide if it's going to be a bike, a car, a plane… After that, you still have choices of makes and models. Similarly, when you select a logic device, first you decide if it's going to be an AND or an X-OR,… but then you still have two different lines (TTL & CMOS) and different families within them TTL: 74, 74L, 74LS, 74ALS from which to choose. Here are the key distinguishing features.

Fanout: How many other similar chips can be connected before the signal degrades too much. To be more specific, these are all transistor based devices, and as you're familiar, there are inherent diode-voltage drops on the order of 0.6V. If you string too many of these in series, there's a chance that the "Hi" output of the last one won't be that high anymore.

Physics 310
Lecture 8a – Digital Circuits

Threshold Level: For that matter, what counts as "Hi" and what counts as "Lo"? Above/below 2.5 V, above 5 V, above 10 V,…?

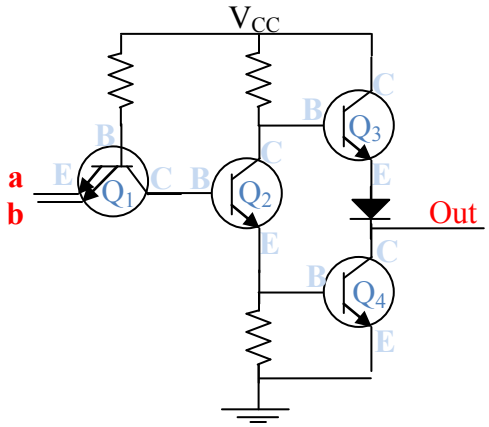
Actually, one given device may have *four* answers to that question. On its
input, a TTL component powered by a +5V rail (and ground) may recognize
Lo < 0.8V
Hi > 2.0 V,
while on its
output it may produce
Lo=0.4V
Hi = 2.4V
– just to give a little margin for error and for voltage drops along the way from one component to the next.

Propagation Delay, T_d: How long after the input flips values will the output flip?

Power Dissipation, P_d: When deciding how to power a circuit, the dissipation of each component needs to be considered so you don't 'undersize' the power supply (just can't source as much current as all the devices want to suck out of it.) We won't get into the details here, but the basic question is usually how much *current* is drawn by a circuit (depends on the voltages and impedances to ground) and a secondary one is how *hot* will it get (depends on how much current passes through each resistor.)

Table 11.16 gives you an idea of the different qualities of the different lines and families. To get a little bit of an appreciation why there should be these differences and more generally how these work, we'll peek under the hood of a few logic gates. You need understand them only enough so that they're *plausible.* You won't be modeling them.

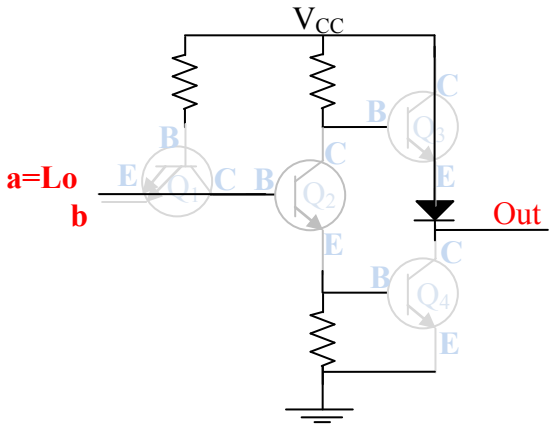
Transistor-Transistor Logic (TTL):
Q_1: This shows a peculiar device, a transistor with *two* emitters. Taking that literally, the device is a chunk of semiconductor built like a "Y": the Collector is at the bottom, the Base is in the middle, and then there are two different branches to Emitters.
Here's the actual schematic followed by the mechanical-switch version for the two possibilities.



To draw current, need emitter lower than base voltage; in this design, the base is connected to the + rail, so that's taken care of, now we need to control the emitter voltage to 'turn onn and off' the transistor.

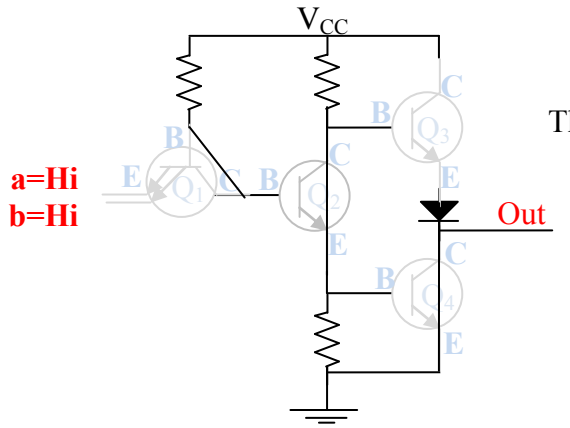
Physics 310
Lecture 8a – Digital Circuits

If one Q_1 Emitter is held "Lo"…
Then it *can* draw current. It only takes one Lo emitter to draw current through Q_1. Q_1 is on, and so pulling its Collector / Q_2's Base below *it's* emitter, and so turn off Q_2. Then Q_3's base floats Hi and so it turns On and Q4's base float's low and it's off.



I do this one

If both Q1 Emitter's are "Hi"…
Then it *can't* draw current; Q_1 is off, and it's collector / Q_2's base is allowed to float up through Q1's collector and base toward Vcc, so Q_2 turns on. That allows it to pull Q_4's base up to about 0.6V / turn Q_4 on, and meanwhile pull Q_3's base *down* to 0.6V which is roughly where it's emitter is if Q_4 is, thus Q_3 is off.



Thus, this circuit has the behavior of a NAND.

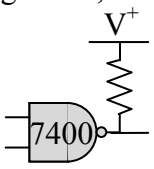
Input a	Input b	Output
0	0	1
1	0	1
0	1	1
1	1	0

Big voltage swings, long time to respond. One important point about TTL circuits like this is that the voltage drops across the transistors' terminals vary from nearly 0 volts to nearly the rail. In so doing, like with a charging up capacitor, opposite charges build up across the junctions between the n-type and p-type slabs in the transistors. As with a capacitor, the more you want to charge it up, the longer it takes.

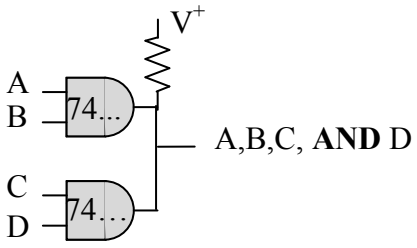
Physics 310
Lecture 8a – Digital Circuits

Open Collector-Output (from 11-7).

Alternatively, the 74-series TTL gates *don't* have the equivalent of Q_3. What that means is the output *either* gets pulled down to ground when Q_4 is on or it's like a loose wire. In order to get the output to be high then, the user needs to include a "pull-up" resistor on the output. For example:



You might wonder why someone would ever use such a chip, but there's a real virtue. Each of these chips only handle two input signals, but what if you want to handle more; say, you want to flag when A, B, C, AND D are all Hi. You can do this by using multiple gates in parallel, but then you can't have them fighting over the output's value

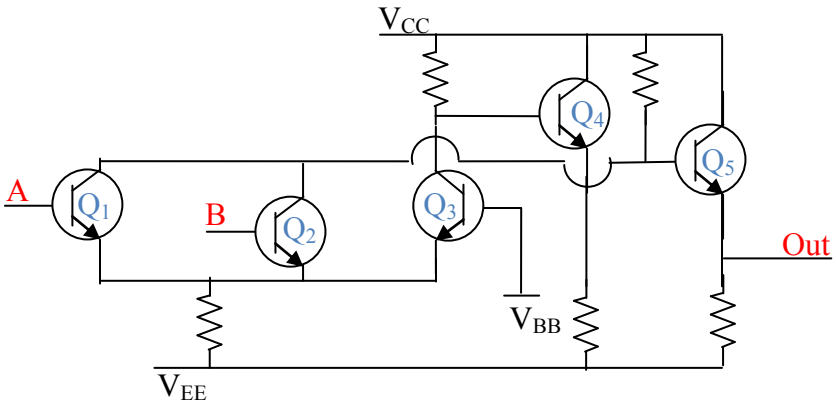


Three-State Logic (from 11-7)

Another approach to letting multiple gates run in parallel is taking turns 'activating' their outputs. Three-State or tri-state ™ chips have an extra input that simply tells it when to connect its output. That way each of a number of gates can take turns controlling the shared output line.

Emitter-coupled Logic (ECL)

Emitter-coupled devices minimize this effect by using transistors in a way that doesn't so dramatically change the voltages across them. Thus ECL devices can responded more quickly. Here's the ECL version of a NOR.



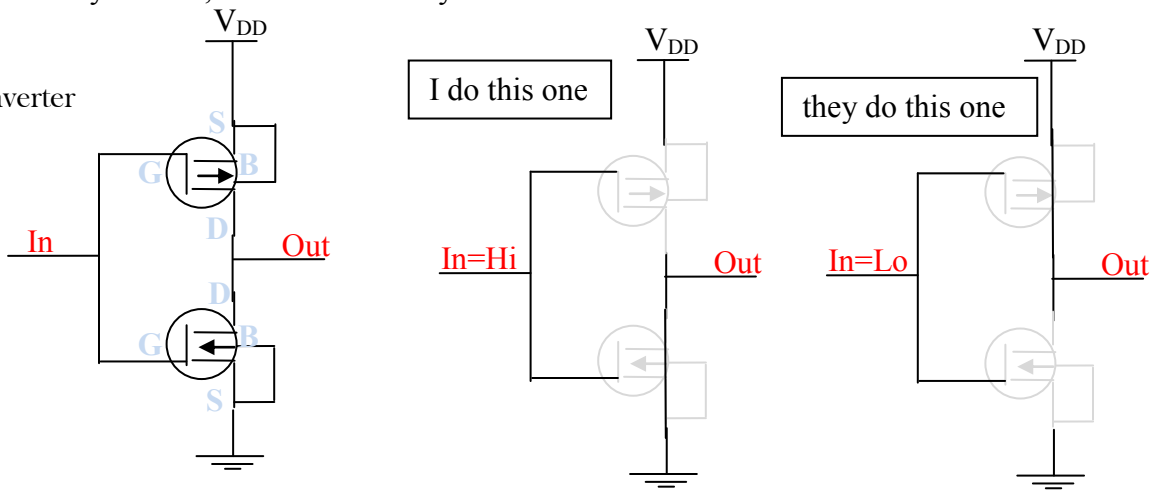
While the subtle way in which the transistors are always held near the brink of transition may not be obvious, the *basic* operation should be clear:

With no voltage applied to A or B, the respective transistors are off and Q5's base voltage is pulled up, thus having it on and pulling the Output Hi, as is appropriate for a NOR.
In contrast, if either A or B is Hi, then the respective transistor is on and Q5's base is pulled Lo which turns it off and allows the output to be pulled Lo.

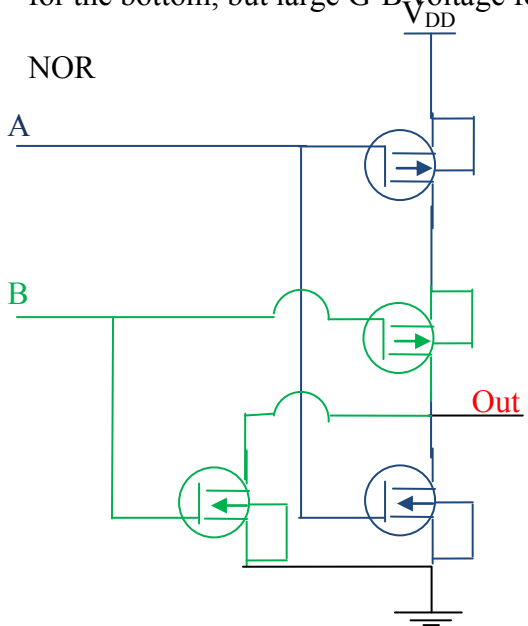
The book notes that this design is *very* quick to respond, ("sub-nanosecond"), but such quick switching can generate high frequency noise in a circuit which can cause other problems.

Complementary metal Oxide Semiconductor (CMOS) Logic

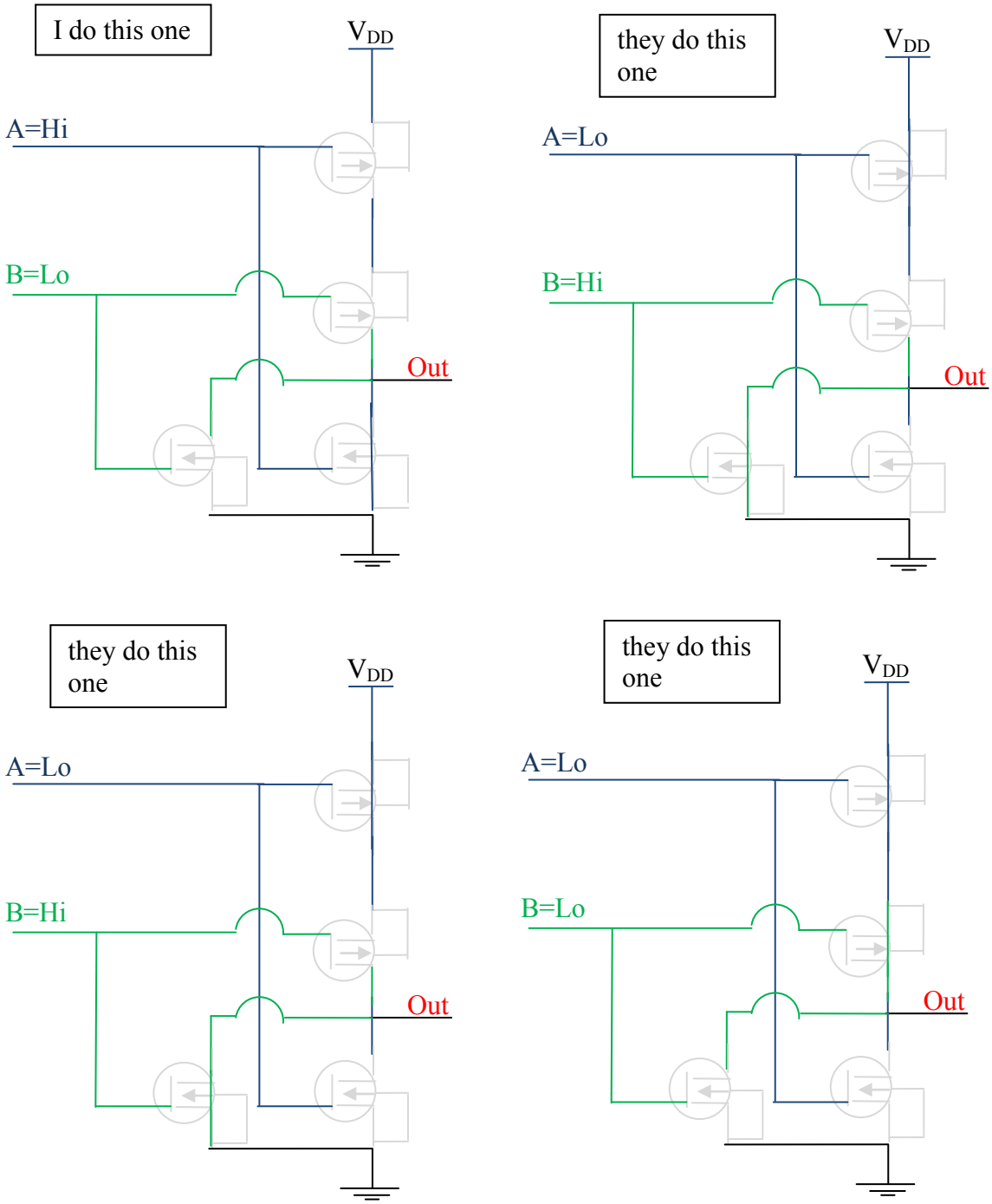
Back when we first met transistors we didn't spend time on CMOS transistors, so here's the key: Whether or not current can pass between the Source and the Drain depends on the voltage difference between the Gate and the Base. The greater this voltage difference, the more easily current passes between Source and Drain. Think of this as a rubber pipe and increasing V_{GB} is like stretching it wider. The down side is that these rely on the fairly slow process of charging up and discharging the gate, so CMOS isn't particularly quick. On the upside, the Gate barely draws any current, so CMOS is fairly efficient.



When the input is Hi, then there's little G-B voltage for the top, but large G-B voltage for the bottom transistor, so output is pulled Lo. When the input is Lo, then there's little G-B voltage for the bottom, but large G-B voltage for the top transistor, so output is pulled



Physics 310
Lecture 8a – Digital Circuits



This brief survey should give you a sense of the differences between different lines of logic gates.

Physics 310
Lecture 8a – Digital Circuits

11-8 Family Interfacing

In general, once you choose a "family" of gates, you want to stick with it since they have different input and output impedances, different voltage thresholds, and even different rail voltages. Still, sometimes you've got to interface CMOS and TTL family chips.

Going from open-collector TTL to CMOS is easiest. The TTL is more than able to supply the minimal current that the CMOS will draw, and using a pull-up resistor on the output allows you to set its Hi value where ever you need to communicate with the CMOS.

Going the other way is a little trickier; the TTL may draw more current than the CMOS can provide and its threshold can be significantly different from that for the CMOS. For this purpose an intermediate "buffer" makes the transition from one logic level to the next and can source the required current.

Now closing the hoods again, let's think about what you can do with them.