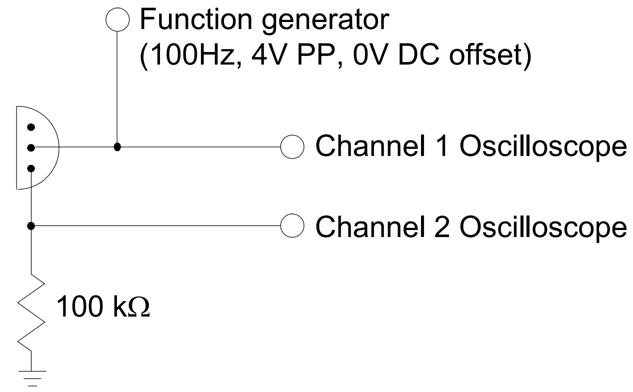
Physics 309 Lab 3
Bipolar junction transistor

The purpose of this third lab is to learn the principles of operation of a bipolar junction transistor, how to characterize its performances, and how to use this widespread device in the construction of current/voltage amplifiers as well as logical gates.

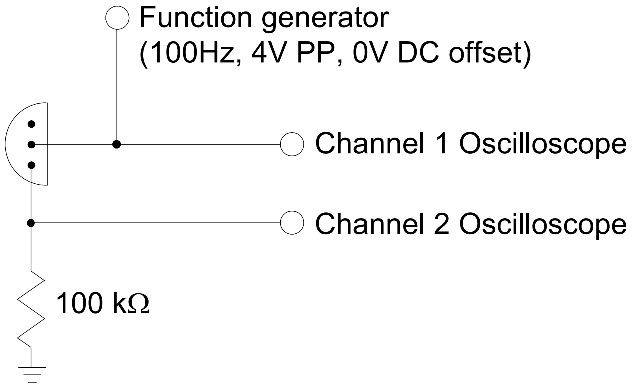
A. Identifying transistor terminals and type.

Start with the oscilloscope, a 100kΩ resistor, a 1μF-100Ω filter, and a 2N3904 transistor.

Build the following circuit:



Turn the transistor around:



Do you see any difference in the signal at the scope for the two circuits?

Hint: You should get a waveform much like in the diode experiment for both circuits. One has slightly larger amplitude on Ch2, however, indicating a much higher value of I_0.

Introduce an RC filter, reduce the function generator frequency to 1Hz, and acquire data for both transistor orientations. Fit both sets of data. How do the values of V_0 compare? Is this expected? What about the values of I_0? Is this expected? Why or why not?

Hint: I_0 will be much bigger for the base-collector junction. Although the emitter is much more heavily doped than the collector, the geometry of the transistor, with the tiny emitter surrounded by the base, in turn surrounded by the relatively gigantic collector ensures that I_0 will be higher for the base-collector junction.

From you measurements determine which transistor terminal is the base, which one is the collector, and which one is the emitter. Is this an NPN or a PNP transistor?

Explain your answers in you lab book. Check then with the TA before proceeding.

B. The common emitter amplifier.

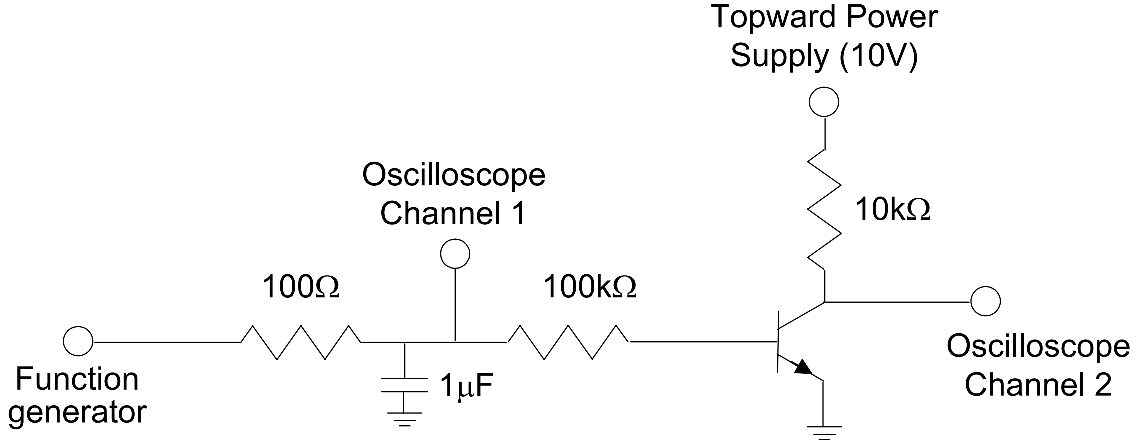
- Configuring the Topward Electric Instrument power supply:
Turn on the power supply. The instrument will provide a constant voltage, set by the two left control knobs, until a maximum amperage is reached. The supply will then output the constant amperage, lowering the output voltage accordingly.

The transistors used in this lab will roast at powers above 350mW. At 32V, this power can be reached at just over 10mA! Unfortunately the current regulation on the Topward supply is not accurate to the mA level, but setting the output current to the minimum possible value will increase the chances that an incorrectly connected transistor can be disconnected before it roasts. This minimum value is already quite high: 100mA. Note however that it is not the instantaneous power that will kill the transistor but rather a too high power provided over a sufficiently extended period of time; that might give you enough time to react should you notice, for instance, that the transistor is getting hot.

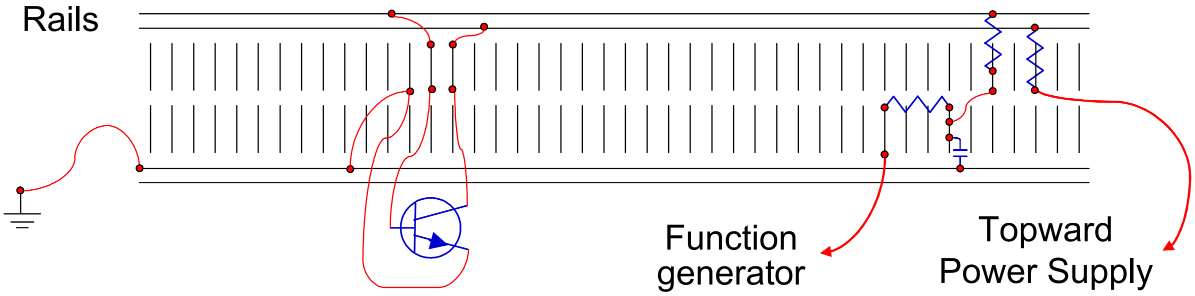
To set the current regulation: jumper together– and + terminals of the supply allowing the maximum current to flow, and adjust the current knob until the current meter reads 0.1A. As discussed, you won't be controlling or reading the current any better than this.

Try, as hard as you can, to avoid roasting the transistor. While its cost is negligible, each transistor has a different gain! So, if the one you are using dies, you have to restart taking data all over again. Also, when the lab period ends, if you are in the middle of a section, take the transistor home with you and bring it back the following time.

- Build the following circuit:



When building your circuit, try to configure the breadboard such that there is no direct path to the power supply of the function generator anywhere near the transistor. See the figure below for an example of a good breadboard: the function generator and power supply go through the resistors before getting to the rails. This means that if the transistor is connected incorrectly to the rails, there is little chance of fire (*do you really mean fire?* A small transistor like this is resistive enough that if you connect it wrong, the most current you can possibly push though it is about 1 amp. This is 30W, and it will certainly cause the transistor's case to deform, and it will smoke, but it will not catch fire. Larger transistors that can pass a large current can heat their internals rapidly enough to explode, but they do not actually burn either. It can melt the breadboard, though).

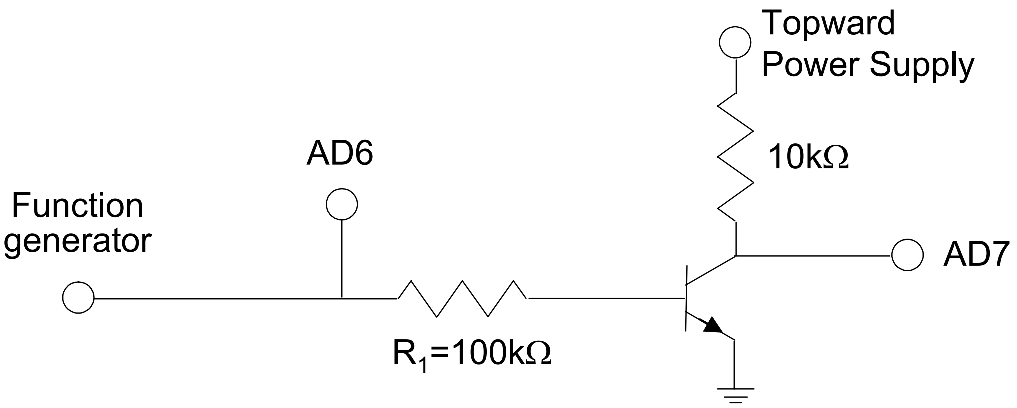


Set the function generator to a triangular 100Hz waveform and the Topward power supply at 10V. Over what range of input values does the amplifier behave roughly linearly? Choose a DC offset and an AC peak-to-peak voltage on the function generator that result to an undistorted, amplified waveform. What is the approximate voltage gain of the amplifier circuit? What is the current gain of the transistor?

Reduce the function generator frequency to 1Hz and change the 100Ω resistor in the filter to a 10kΩ one; at this point, you can acquire data with the computer. Model your transistor as having a constant gain and fit your data. What is the voltage gain of the amplifier and what is the current gain of the transistor? Plot your residuals. Is there structure to the residuals? If so, does this come from a failure of the instrumentation, experimental design, or the model?

- Gain as a function of I_B.

Build the following circuit, setting the supply to 10V and adjusting the AC and DC voltages of the function generator to produce an amplified waveform that appears free from distortion on the scope. You should be using a 10Hz sine wave.

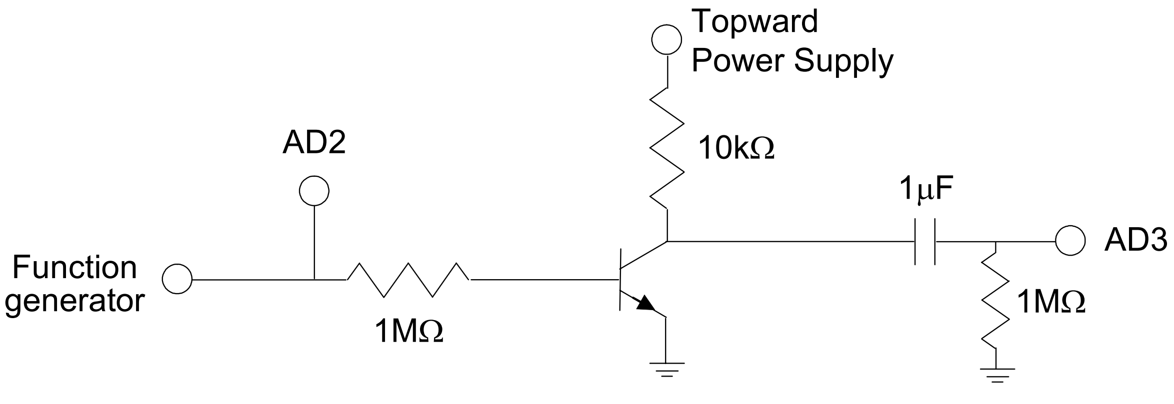


Take data with the computer system and fit the output waveform. Solve for the gain.

Change the 100kΩ resistor to higher resistor values, up to 10MΩ. Each time you change resistor, fit the resulting waveform and solve for the gain. Plot the gain vs. the base current. Outside of the lab, look up the gain as a function of base current for you transistor. How does your graph compare to the published results?

- Gain as a function of V_C.

Build the circuit below, which contains a high-pass filter whose purpose is to allow only the AC component of the signal to the ADC amplifier (cranking up the voltage out of the Topward supply, we would soon go out of the pre-amp range, even on the ±10V scale).



Use a sine waveform. Set the DC offset to 2V and the AC PP to 1V. Set the frequency to 100Hz. Take the data with the computer acquisition system for V_C ranging from 5 to 32V. Fit the amplitude of the sine waves and solve for the current gain of the transistor at each value of V_C. Plot the gain as a function of V_C. Pretty flat, eh???

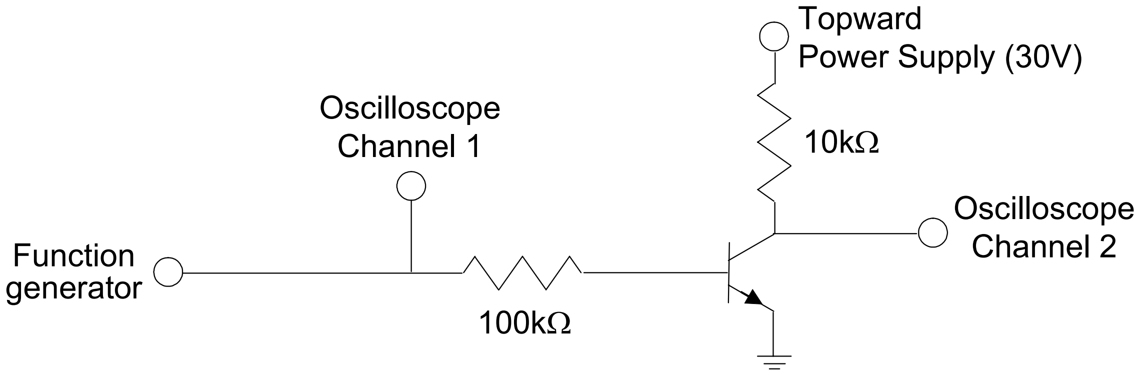
C. AC response of the transistor.

Transistors are used to amplify very high frequencies. All wireless devices depend on transistor circuitry to amplify high-frequency signals coming to and from their antennae (AM radio, 500kHz; wireless router, 4GHz; microwave oven, 50GHz). In this section, we will study how the transistor responds to high frequency signals. All transistors have a maximum frequency value at which they can operate. We will investigate and model the high-frequency response of our transistor.

This lab section will use the oscilloscope only, as the circuitry in the computer data acquisition card does not have a speed anywhere near to what is needed in order to take data at frequencies that would look high to the transistor itself.

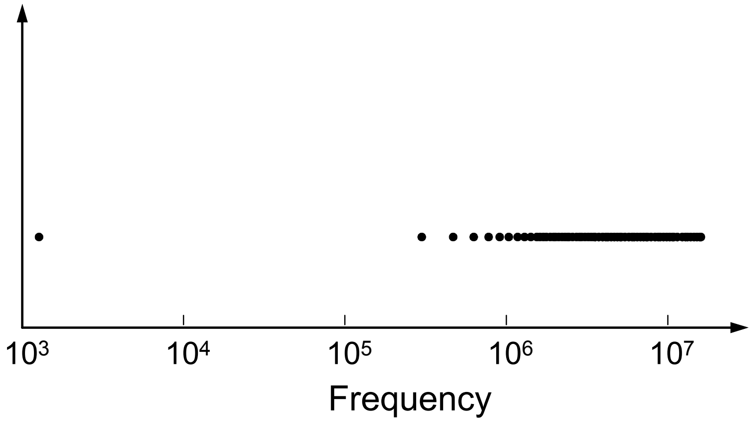
Note also that, the higher the frequency, the more significant capacitance and inductance of the wires become. Therefore, we want to minimize any unneeded cabling. The circuit should not be connected to the computer data acquisition system at all and there should be no T's connected to any BNC cable connectors.

Build the following circuit:



- AC mode on the oscilloscope.
In the earlier labs, we have kept the scope on DC signals in order to measure the real undistorted signal. Setting the scope on AC mode puts a capacitor in series with the signal input; this capacitor blocks any DC component of the signal. Normally we don't want to do this, because we want to measure exactly what is "happening" to our circuitry. The exception to this approach is when you have a small AC signal with a large DC offset. In this situation, in DC mode you cannot zoom-in enough to get a good view of the AC component of the signal (the DC offset would push the waveform of the top of the oscilloscope screen).

Set channel 2 to AC. Set the function generator to 1.0V PP, with a 1.6 DC offset, and set the Topward supply to 30V. Change the function-generator output waveform to a sine wave. Measure the amplitude of the output of the amplifier as well as the relative phase between the input signal and the output from 1Hz to 10MHz. try to get data for 30 or 40 different frequencies. Space your data points in a manner appropriate for plotting on a frequency log scale. In other words, if you take data with a constant spacing in frequency, almost all of you data will be between 1MHz and 10MHz; instead, what you want to get is the same number of data points in each decade!



Enter the frequency, output amplitude, and phase into a spreadsheet. Solve for the gain of the transistor. Save your file as a space or tab-delimited text file, so that you can plot your results in Gnuplot. Plot the gain as a function of frequency. Set the x-axis (frequency) to a log scale. Can you think of a model function to fit this behavior to? This might require resorting to the so-called *hybrid parameter* description.

Determine the input frequency at which the transistor will have a current gain of 1.

D. Transistor logic.

Transistor logic is the basis of practically all digital devices. In transistor logic circuits, a set voltage (V_{CC}) represents 1, while ground represents 0 or 'false'. In this section, we will be building a number of logic gates. The final logic device will be comprised of all the other devices that you make, so do not disassemble any of your working circuits. Just build your new circuits besides your old ones; also, you need to keep your circuits neat and small, otherwise you will quickly run out of space on the breadboard.

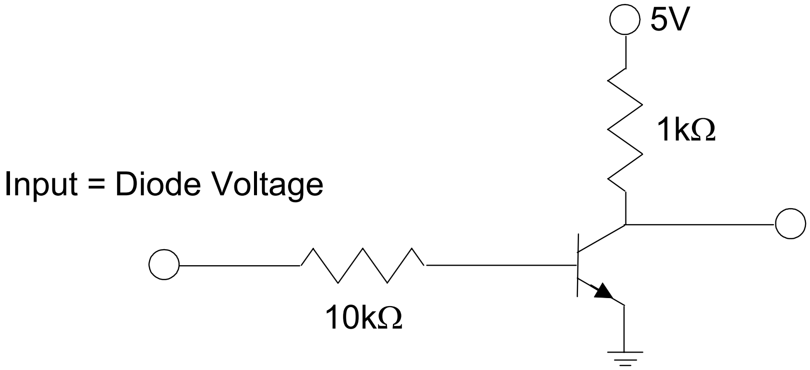
The circuits around your breadboard contain a number of useful analysis tools. Use the Molex cable (big, white end on a grey cable) to connect the breadboard to the triple output power supply (the same that powers the computer data acquisition amplifier). To the left of you breadboard are wire sockets giving +5V, +15V, -15V and ground. We will use 5V as our 1 or 'true'. Connect 5V to one breadboard rail, and ground to the other one.

- Logical NOT gate.

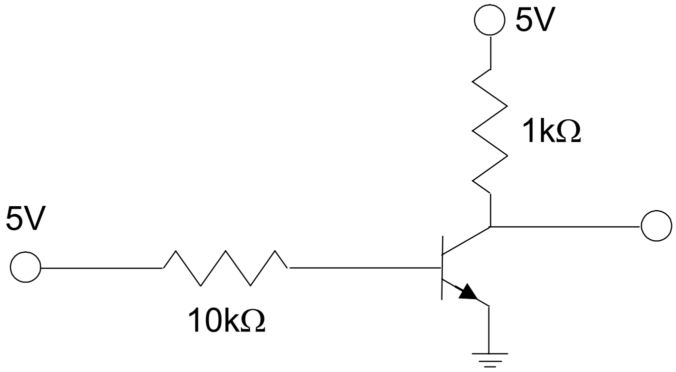
Input	NOT gate output
1	0
0	1

The simplest transistor NOT gate is just a common emitter amplifier. It is used in a different way than in the last section, though. Instead of properly biasing the gate, to facilitate amplification, we only use the circuit in two modes: cut-off and saturation.

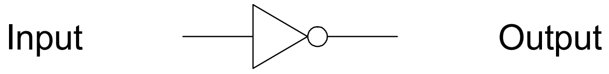
In cut-off mode, no current is allowed through the base. This stops any collector current from being drawn. With no collector current, there is no voltage drop across the collector resistor, and the output is exactly 5V.



In saturation mode, a relatively large current is sent through the base. With a current gain of 100, solve for the voltage of the output. Is this a realistic number? What voltage do you think you will actually get?



Build the NOT gate drawn above, on the far left side of your board. Use the TTL (transistor-transistor logic) output on your function generator to supply an input signal to your NOT gate. Connect channel 1 of the scope to the input signal and channel 2 to the output. Is your NOT gate doing what you expect? If not, check your circuit and eventually talk to a TA! What is the highest input frequency you can use before the device stops working? When drawing transistor logic circuit diagrams, drawing a full common emitter amplifier each time you have a NOT gate gets to be a real pain (there are lots in any digital circuit). The following symbol is used instead (the power supply and ground connections are not shown but are assumed to be there).

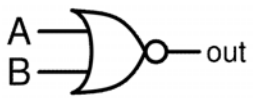


- Logical NOR (NOT-OR) gate

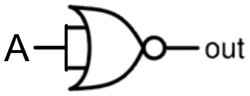
To understand what a NOR gate is supposed to do, it helps to first think of an OR gate. If input A or input B or both of them are 1, then the output is 1. If both inputs are 0, then the output is also 0. you can think of a NOR gate as an OR gate with a NOT gate after it.

Input A	Input B	OR (A,B)	NOR (A,B)
0	0	0	1
0	1	1	0
1	0	1	0
1	1	1	0

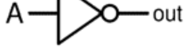
The NOR gate can be used to construct all other Boolean logic operators (1's and 0's). Note that deep down a computer is just a bunch of NOR gates!

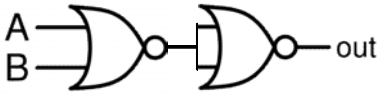


NOR GATE Symbol

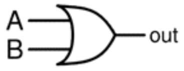


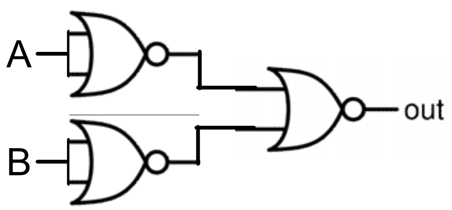
NOT GATE made from 1 NOR

( NOT Symbol)



OR GATE made from 2 NOR's

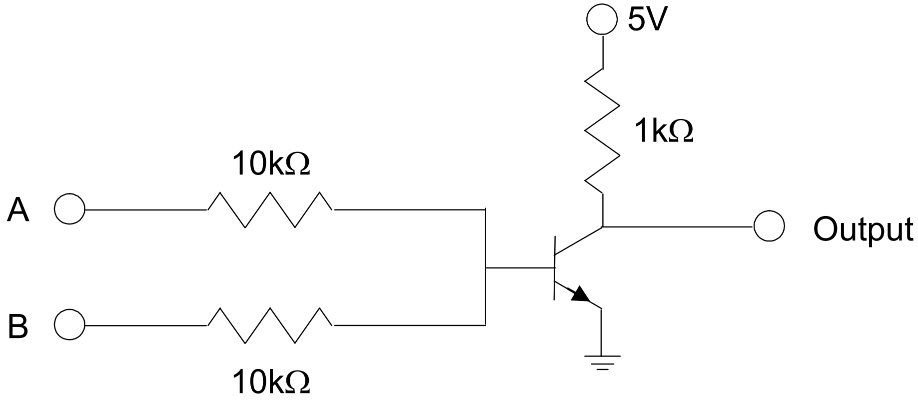
( OR Symbol)



AND GATE made from 3 NOR's

( AND Symbol)

Build your NOR gate circuit (see figure below) to the right of the NOT gate. Connect the output to one of the LED drivers on your breadboard logic test and connect each of the inputs to one of the two momentary switches. Test your circuit; is it doing what a NOR gate should do? Describe your results.

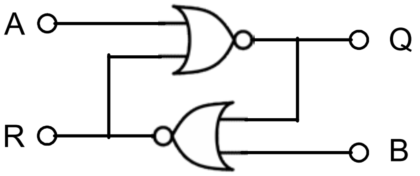


- LATCH circuit

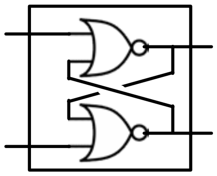
The LATCH circuit (see below) is the fundamental building block of the RAM (Random Access Memory). The LATCH circuit can remember a 1 or a 0, and can be switched back and forth between the 1 and the 0 state. With both A and B equal 0, there are 2 stable states. If Q is 1, the input to the lower NOR is 0. This makes the inputs of the upper NOR (0,0), so its output is 1, giving a self-consistent solution, with the lower NOR outputting 1. Suppose now that Q is 0 and we change input A to 1. This causes the upper NOR to change to 0, and the lower NOR is forced to 1. We have changed the state of the LATCH. Sending A back to 0 does not change the state back again; only setting B to 1 will do it.

A	B	Q	R
0	0	0	1
		1	0
1	0	0	1
0	1	1	0
1	1	0	0

Do not use your first NOR as part of your LATCH. Leave enough space between your first NOR and your LATCH to make another NOR, which you will do in the next section. Lay out each of your NOR's exactly the same way and use short wires; things will be much less confusing. Connect each input to a momentary switch and connect each output to a LED. Verify that your LATCH works properly.

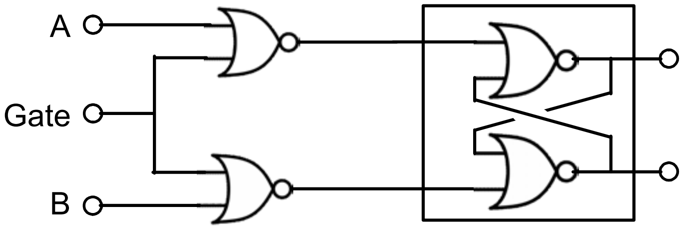


LATCH Symbol

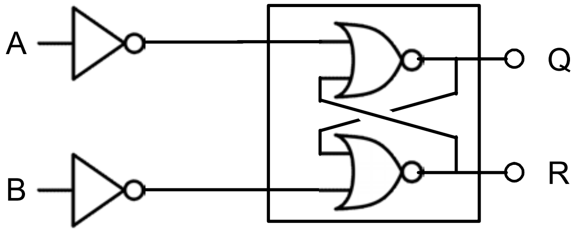


- GATED LATCH circuit

In a memory storage application, you need a way of selecting which bit, or LATCH, you are accessing. If you have only a few bits, each one can have its own wire. If you have a gigabyte, though, you cannot have billions of wires, so gating becomes necessary. The circuit below will behave like a normal LATCH if the Gate is 0; but if the Gate is 1, A and B cannot change the state of the LATCH.



Note also that if you set the Gate permanently to 0, then the circuit above is equivalent to:



Now the 2 stable states are found when A and B are both 1, and setting A or B momentarily to 0 will change states.

Gate	A	B
0	0	0
0	0	1
0	1	0
0	1	1
1	0	0
1	0	1
1	1	0
1	1	1

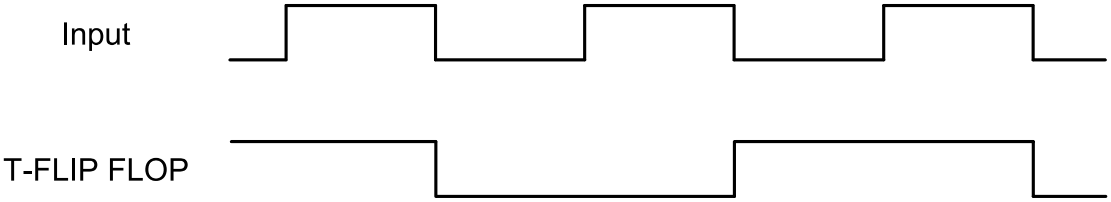
Q	R
0	0
0	1
1	0
0 1	1 0
0 1	0 1
0 1	0 1
0 1	0 1
0 1	0 1

To construct your GATED LATCH build another NOR between your first NOR and your LATCH. Test the function of the LATCH and both NOR gates individually before connecting them all together. This will save a lot of time debugging your circuit.

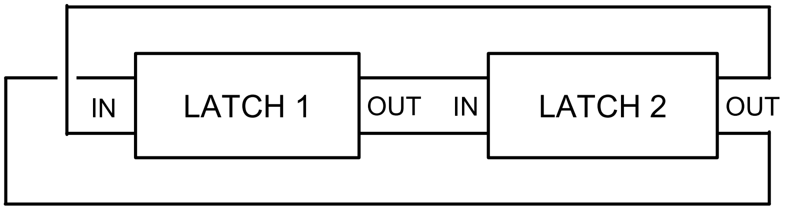
Connect the top of one of the toggle switches to 5V and connect the middle of the switch to the gate. Connect the momentary switches to A and B, and connect the outputs to LED's. Verify that your GATED LATCH works properly. If it does not, and the problem is not obvious, talk to your TA.

- T-FLIP FLOP

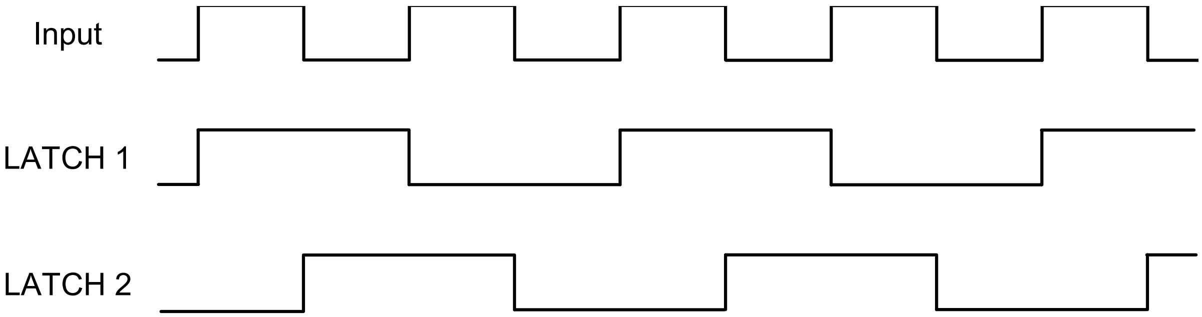
The T-FLIP FLOP is the basic building block for counters (T is for toggle). There are a lot of them inside your watch, for instance, if you have one and it is not spring wound. Even 'analog' watches that run off a battery have a quartz oscillator with an output of usually 32768± 0.6 (=2^{15}) Hz. 15 T-FLIP FLOPS are required to reduce the frequency to the second hand's stepper motor to 1 Hz.). The T-FLIP FLOP changes state as the input signal changes from 1 to 0 (see figure).



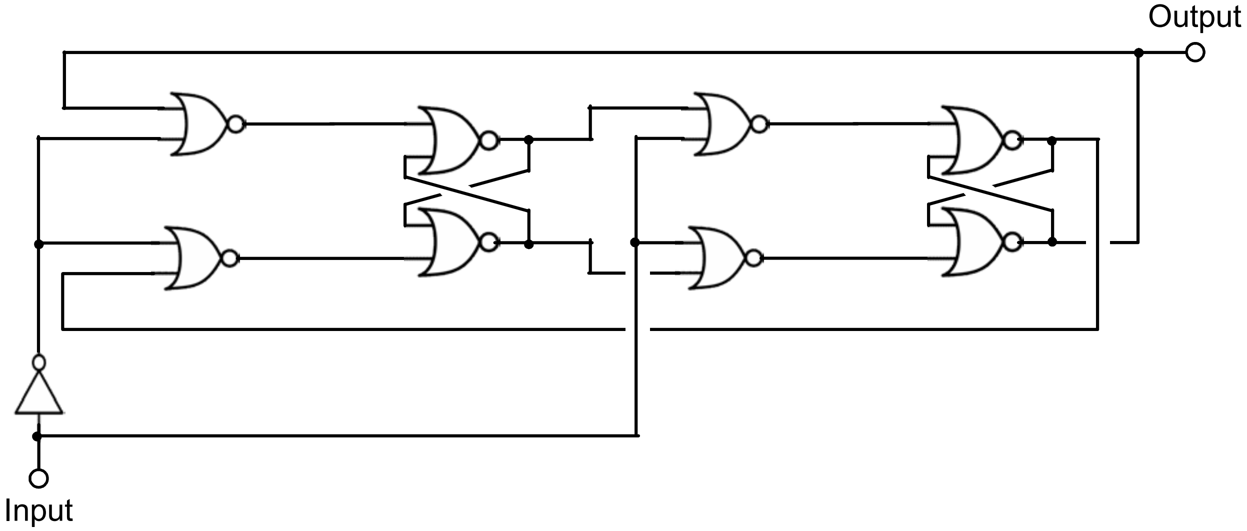
To make this happen, we start with two GATED LATCH circuits that are fighting each other (see figure below). If LATCH 1 has its upper output 0, it sets LATCH's 2 upper output also to 0, which goes to the lower input of LATCH 1. This in turn sets LATCH 1's upper output to 1, and lower output to 0, flipping LATCH2, etc.



But what about the gate? The input is connected directly to the gate on LATCH 2, ad through a NOT gate to the gate on LATCH 1. In this configuration, LATCH 1 can flip LATCH 2 only if the input is 0, and LATCH 2 can only flip LATCH 1 if the input is 1.



To build the T-FLIP FLOP, construct another LATCH. You will save a lot of time if you lay it out on your breadboard exactly as you did for your last LATCH, even if the layout on the first one was not the most efficient. Troubleshooting a new layout will take much more time. Test your second GATED LATCH before connecting everything together. Connect the input to a momentary switch, and the output to an LED. Test you circuit.



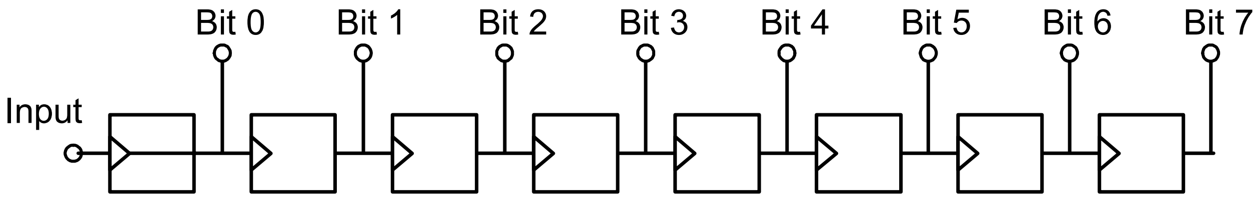
On one of the 7-segment displays (top right of your board), connect the 2 highest bits (marked 8 and 4) to ground. Connect the TTL output to the input of your T-FLIP FLOP, and to the first bit of the 7-segment display. Connect the output to the T-FLIP FLOP to the second bit (marked 2). Set the function generator to 2 Hz. You should see the 7-segment display count from 0 to 3.

T-FLIP FLOP Symbol

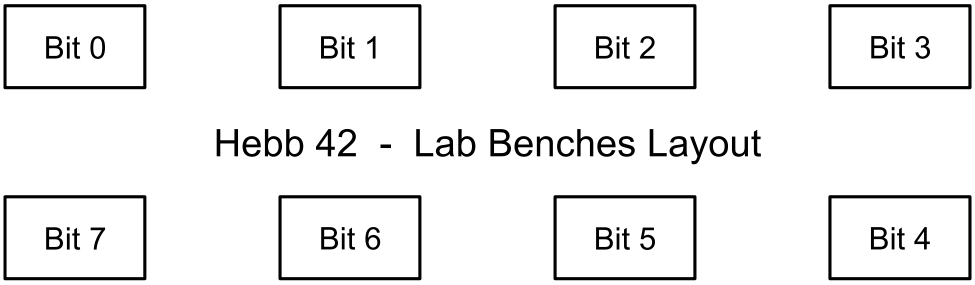
E. The 8-bit counter.

A device that can count 8 bits (256) can be constructed from 8 T-FLIP FLOP’s (see below). The outputs are the digits for an 8 digit binary number, which can be directly entered into the two 7-segment displays.

Of course none of you want to build 8 T-FLIP FLOP’s, so you will do what scientists do when there is more work than they feel like doing: COLLABORATE! If you are done with your T-FLIP FLOP and the other groups are not there yet, give them a hand. Once every group has a functioning T-FLIP FLOP, we will cascade them all to make a counter. Color-coded wires run from each lab bench to the next, to facilitate connections. The counter that will display the result will be on the lab bench next to the printer.



Wire color	Bit
Black	0
Brown	1
Red	2
Orange	3
Yellow	4
Green	5
Blue	6
Violet	7



Hebb 42 - Lab Benches Layout