Hardware: Getting Started

Breadboard (Protoboard)

A typical breadboard (also called a protoboard) is shown in Figure 1. In this figure there are two long rows and four rows of columns. Each of the pins in a long row are electrically connected to all the other pins in that row. Similarly, the pins in the short columns are connected electrically to the other pins in that column. A chips is placed across the gap between two rows of pin columns as shown. For more info on the functioning of a breadboard, see http://www.kpsec.freeuk.com/breadb.htm.

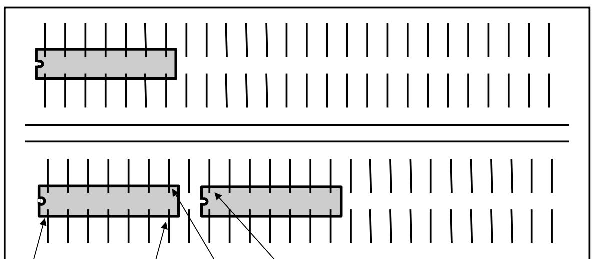


Figure 1. Connection on a typical breadboard (protoboard).

A breadboard board, augmented with a few other parts, can provide you a means to build and verify the proper operation of digital circuits. Some of the useful devices are described below.

Switches: Switches are the means to supply inputs to your circuits. These inputs represent variables in Boolean expressions. Each switch circuit generally also contains a resistor.

LEDs: Light Emitting Diodes (LEDs) are used in a simple circuits to show the values of the outputs. The outputs represent variables in Boolean expressions. Each LED circuit generally also contains a resistor.

Power: You will need a power supply to supply each of the components with the relevant Vcc (3.3 V) and ground (0 V) connections.

CLK: A clock circuit is needed as an input device for many circuit elements. Later this semester, you will add a clock circuit that includes an SPDT switch, two resistors and an SR-latch.

ICs: Integrated Circuits (also called ICs or chips) will provide the logical and other functions to create our digital circuits. Each of the ICs that we use is composed of a rectangular plastic case with two rows of metal pins along the long sides. A transparent IC is shown in Figure 1. The notch indicates the location of pin 1. Looking down on the chip, with the writing on the chip facing up and the pins facing away, turn the chip so that the notch is on the left, as shown in Figure 1. Pin 1 is on the left below the notch.

Warning: This may seem trivial, but do **not** turn your board on until you are ready to test/debug your design. This lowers the probability that you will do the following **WRONG** things: connect power to ground or connect outputs together. If you do either, you may destroy your power supply or from one to several chips.

Trouble-Shooting Suggestions

Your circuit will probably not work the very first time. Therefore, trouble-shooting is a basic skill that you will have to learn. The followings are some helpful suggestions:

- Utilize neat (well organized) circuit diagram.
- Your circuit diagram should identify the IC (chip) type and pin numbers for each of the gates in the circuit..
- In a large circuit, identify, build, and debug a small portion of the circuit at a time. This will give you a higher likelihood of having the entire circuit work when it is finally completed. To isolate problem(s), try to backtrack by testing individual blocks in a similar fashion. In other words, decompose your design and your construction.
- Start trouble-shooting at the point, at which the logic/voltage value is wrong in the circuit and continue to check your circuit backward, making sure that at each point the logic/voltage value agrees with your desired value.
- Construct your circuit neatly. Use appropriate length wires for connections, i.e. use short wires for short distance connections. Your circuit should not look like a bowl of spaghetti. For power and ground, use the rows of connected pins to help avoid the spaghetti.
- Use a long wire to an LED circuit as a logic probe.
- Check all IC insertions to make sure all the pins are in the holes correctly and the ICs do not have any missing legs.
- Make sure that the power and the ground are properly connected to all IC's **before** applying live power.
- **DO NOT** jam long bare wire ends into the breadboard holes and never put more than one wire into one hole. Each of these can ruin the board.
- **DO NOT** connect power to ground. This will cause the permanent damage to the power supply.
- **DO NOT** connect the power supply to the breadboard with reverse polarity. This will cause the permanent chip damage.
- **DO NOT** connect an output of any gate to the output of another gate, to a switch, to the power (Vcc), or to ground. These situations will cause excessive currents and result in permanent damage to the chip.

Hardware: Getting Started

Breadboard Wiring and Testing

For any circuit you have designed to implement a logic equation or function, software (like Quartus) allows you to simulate its logical correctness before wiring the solution on a breadboard. Quartus has only a voltage simulation, and thus a voltage table must be constructed to match the simulation output.

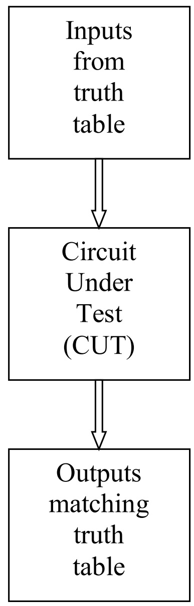


Figure 2. Hardware testing block diagram.

Naturally the question arises as to how the simulated circuit design, known as the Circuit Under Test (CUT), may be tested when built with real hardware. The block diagrams in Figure 2 show the essential components to be constructed on your breadboard to test your solution with hardware.

An input for your circuit is built with a single pole single throw (SPST) switch, a resistor and wires appropriately connected to ground and the Vcc (3.3 V) power supply (as shown in Figures 3). Use **ONLY** the switch circuit on the left in our course. The switch circuit on the right will work, but for various reasons not preferred.

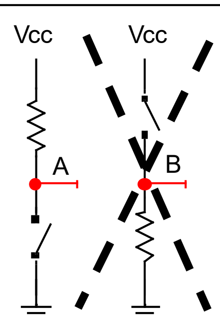


*Figure 3. Input circuit with pull-up resistor (**preferred [and required in 3701], left**) and input circuit with pull-down resistor (right).*

A DIP (dual in-line package) is a package for various electronic components that have two parallel sets of pins. A SIP (single in-line package) is a package for various electronic components that has a single row of pins.

Included in your lab kits is a DIP switch bank, with 8 or 10 separate SPST switches. Multiple inputs can be built from a DIP switch bank, a SIP or DIP resistor pack, and wires connected to Vcc and ground. See Figure 4 and Figure 5 for diagrams of SIP and DIP resistor packs, respectively.

A SIP resistor pack, as shown in Figure 4, has a common pin connected to one side of each of the resistors. This common pin is connected to either ground or Vcc. The other sides of each of the resistors are the connection points to other components.

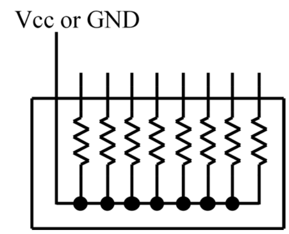


Figure 4. SIP resistor pack.

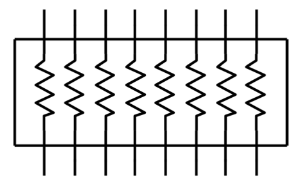


Figure 5. DIP resistor pack.

A DIP resistor pack, as shown in Figure 5, has no common pin connections.

The "CUT" (see Figure 2) is built with IC's properly connected to ground and Vcc, along with appropriate inputs and output wires.

Let's understand how the switch circuit operates, starting with the left circuit of Figure 3. When the switch is open (as shown), the point marked A is connected only to Vcc (3.3 V) through the resistor. Point A will therefore have a voltage of approximately 3.3 V. When the switch is closed, then A is connected directly to ground (through the switch) and to 3.3 V through the resistor. Since resistors "resist," the point marked A will take the value of ground (0 V) rather than 3.3 V, i.e., A takes the voltage from the path of least resistance. The switch has a resistance of approximately 0 Ω. (Ω is an abbreviation for ohm and is the measurement unit of resistance).

The switch circuits of Figure 3 can each be used for either active-high or active-low inputs. If the left circuit of this figure is used for an active-high input, the circuit is in the true (active) configuration (high) when the switch is open; it is in the false configuration (low) when the switch is closed. If the left circuit of this figure is used for an active-low input, the circuit is in the true (active) configuration (low) when the switch is closed; it is in the false configuration (high) when the switch is opened.

If the right circuit of this figure is used for an active-high input, the circuit is in the true (active) configuration (high) when the switch is closed; it is in the false configuration (low) when the switch is opened. If the right circuit of this figure is used for an active-low input, the circuit is in the true (active) configuration (low) when the switch is

Hardware: Getting Started

opened; it is in the false configuration (high) when the switch is closed.

You can use an active-high input (switch) for an active-low signal by just interpreting the off/on backwards.

A DIP switch bank typically has the word “ON” printed on it. Usually, this word is on the upper left hand corner of the switch bank. If a switch is put in the up position (i.e., towards the ON), the switch is closed. Note that the word ON does not necessarily indicate whether the signal is true or false. It actually depends on how the switch circuit is designed.

Generally, pull-up inputs are preferred to pull-down inputs since the input resistance (or impedance) to gates are generally much lower between inputs and Vcc than they are from inputs to ground. If you must use a pull-down, use a smaller resister (100 Ω) than you would for a pull-up resistor (~2 kΩ to 5 kΩ is good). A pull-down using a 1 kΩ resistor can result in a low input of about 1 V, which is outside the rated specifications for a low input on our chips. Use **only pull-up resistors** for input switches in this course.

An LED (light emitting diode) circuit can be used to indicate the truth level (or voltage) of any signal. An LED circuit consists of an LED connected in series with resistor (to limit the current). A voltage must be placed across the LED circuit in order to activate (light up) the LED. Figure 6 shows various connections of Vcc and ground with the LED circuit. Only the circuit on the left in Figure 6 will cause the LED to be on..

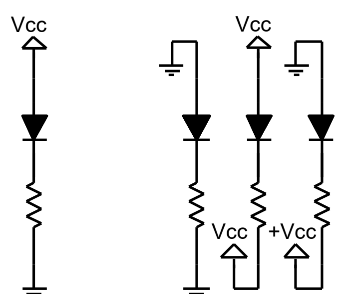


Figure 6. LED circuits. Light will be on in the left circuit and off in the three circuits on the right.

Figure 7 shows how to use an LED circuit to indicate whether a signal is true (LED on) or false (LED off) for both an active-high and an active-low signal.

The series resistors for typical LED circuits should be about 1kΩ. If the resistor value is too high, the LED will be dim. If the resistor value is too low, the output of the gate could be adversely effected.

Multiple LED outputs are built with a DIP or SIP resistor bank connected to a DIP bank of LED’s.

The details of wiring single active-high and active-low signals for the "Inputs" block are shown in Figure 3 and the details of wiring single active-high and active-low “Outputs” block are shown in Figure 7.

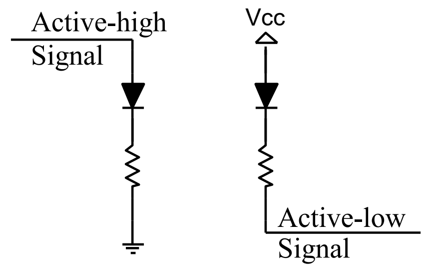


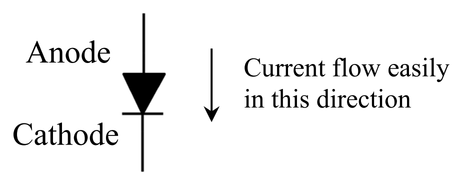
Figure 7. Active-high and active-low output LED circuits.

Note that LED circuits should **NOT** be attached to switch circuits. If they are, the LEDs may be very dim when on and the inputs may have voltages outside of the specified values.

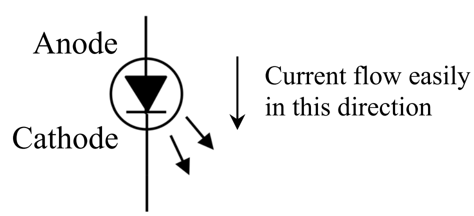
APPENDIX A: LED Details

An LED (light emitting diode) is often drawn like a normal diode as shown below.

It is also sometimes (and more correctly) drawn as shown below.



In either case, current flows easily from the **anode** to the **cathode**. (I’m a dog man, so I remember cathode from bad



cats. When you are bad, you get grounded. So ground the cathode.) The LED turns on when current flows in the indicated direction (and is of sufficient magnitude). If the current is too small or if the current is in the opposite direction (caused by having a higher voltage on the cathode than the anode), the LED is off.

APPENDIX B: Why is my LED Dim?

A dim LED can result because of the fact that high outputs are not always Vcc. The resistor values were chosen by assuming that that Vcc or Ground would be the input to the LED circuit. It is within the specification requirements of 74HCxxx part to have a high output at as low as 60% of

Vcc, depending on the manufacturer. If the output is 60% of Vcc, and the resistor value is unchanged, the LED may be dim.

A dim LED can also occur if the LED series resistor's resistance is too high.

APPENDIX C: Unused Inputs on ICs

When using HC (HCMOS) IC's, unused inputs may be left floating for simple prototyping and in our labs. But in production devices, unused inputs should be tied to V_{CC} or ground to prevent the input from floating. If left to float, the power consumption of the device increases because the input may be at an invalid logic level (between V_{IH} and V_{IL}), causing I_{CC} (the current from the power input) to increase.