1. OBJECTIVES

- Learn the fundamentals of automated schematic capture and digital simulation.
- Understand the role of digital capture and simulation play in the Development of Digital Circuits.
- Become familiar with some representative Digital Logic documentation techniques.

2. INTRODUCTION

In order to translate the statement of a problem into a working design there are generically 6 phases:

1. Translation of the Problem Statement into Design Requirements (e.g., Boolean Equations, a Truth Table or State Diagram).
2. Circuit Design
3. Schematic Capture and Simulation
4. Fabrication of a Prototype
5. Testing and Debugging of the Prototype
6. Documenting the Working Circuit

Of these, the corresponding ECE 85 lectures emphasize the first two (requirements and design). The ECE 85 Lab emphasizes the remaining four. This Laboratory Session will introduce you to Automated Schematic Capture and Simulation, and the use of such programs to debug and document the circuit.
2.1 The “New” Design Approach

Once the circuit design is complete, historically the next step used to involve the fabrication of a prototype. The design would be first implemented on a test circuit (often hand-wired on a breadboard), and debugged to find any errors. Once the errors were identified, a redesign commenced, resulting in either a second breadboard (if the design modifications were extensive), or a pre-production prototype, called a “brassboard”. The brassboard usually used some form of printed circuitry for interconnection. Once the brassboard operated successfully, the final production model was instituted.

This process has been augmented by a new process, for two primary reasons:

1. Circuits today are sufficiently complex such that the (hand) wiring of a complex breadboard introduces many errors; this causes prototype debugging to be delayed due to errors in the prototype (rather than errors in the design).

2. The necessity to bring a new product to production in a timely fashion (this is called “time to market”), forces the process to be more automated, and hence faster.

2.2 Automated Schematic Capture and Simulation

Today’s approach attempts to supplement the breadboard/brassboard stages with computer-based schematic capture and simulation. With this approach, once the designer has completed the circuit design, the circuit is moved to the computer (this is called “schematic capture”) using software called a schematic capture program.

The captured schematic may be imported into a simulation program, a “virtual” test environment. “Virtual Inputs and Outputs” are applied and the circuit is debugged on-screen where modifications are almost instantaneous (this is called “simulation”). In this manner the design may be tested in this virtual environment without the necessity of physical implementation. In addition, any required design modifications may be rapidly implemented via the schematic capture package, the re-design again imported into the simulation package and the simulation repeated, again without actual implementation. Once the design is valid, implementation may proceed with a high confidence of success.

Most schematic capture packages also support other facets of the design process. As examples:

1. The captured schematic, now in digital format, is a useful vehicle for the documentation of the design.

2. The schematic capture package may produce, as an output, a list of parts used in the design. Such a list may be used for pricing, inventory control or component purchasing.
3. The schematic capture package may produce, as an output, an automated list of components and their interconnection. This so-called “netlist” may be imported into another software package that produces the actual printed circuit board.

2.3 Schematic Capture and Simulation in Prototype Implementation

The captured, simulated program also is useful in the implementation of a prototype (a breadboard such as those you will use in this laboratory).

Once the captured schematic is available it contains:

1. The actual components (the gates) each with pin assignments and gate labels (e.g. U1, U2).

2. Interconnection (e.g. Pin 1 of U1 is shown connected to Pin 3 of U2).

3. A “software” logic probe—an icon which may be moved by the user using the mouse to any pin or wire. When so-moved the simulated display will indicate whether the pin or wire is at a Boolean 0 (a Low voltage) or a Boolean 1 (a High voltage).

These three properties are useful in the fabrication of the prototype. The student wiring the board would consult the captured schematic and, as indicated by the schematic, “connect a wire between U1 pin 1 to U2 pin 3”. As an alternative one student could “call out” the impending connection and a second student could make the actual connection. This would facilitate the wiring as well as providing a degree of verification since two persons are validating the interconnection.

Once the prototype has been wired the captured, simulated schematic could provide assistance in debugging, in the event the prototype does not operate as designed. Since there is a “virtual” logic probe (the probe icon on the screen) and a “real” logic probe (the logic probe associated with the Logic Development System) the student could place the virtual probe on a particular relevant terminal on the simulated circuit and a real probe on the actual terminal; with this method the circuit may be debugged by comparing the outputs of the two probes until the two disagree, thereby identifying the error(s).

2.4 Multisim

Multisim is a schematic capture and simulation program for Analog, Digital, and Mixed Analog/Digital Circuits. It is one application program of the National Instruments “Circuit Design Suite”, which is a part of a suite of circuit design programs, along with NI Ultiboard. Multisim is one of the few circuit design programs to employ the original Berkeley SPICE based software simulation. Multisim was originally created by a company named Electronics Workbench, which is now a division of National
Instruments. Multisim includes microcontroller simulation (formerly known as MultiMCU), as well as integrated import and export features to the Printed Circuit Board layout software in the suite, NI Ultiboard.

Multisim is widely used in academia and industry for analog and digital circuits education, electronic schematic design, and SPICE simulation.

3. PRELAB

In this lab, we will be using Multisim extensively to perform Simulation and Schematic Capture. We will begin with the basics by limiting ourselves with the 3 fundamental building blocks: AND, OR, and NOT gates, as shown in Figures 3.1 – 3.3.

![2-input AND gate diagram]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Figure 3.1:** AND Gate Boolean Expression: \( O = AB \)

![2-input OR gate diagram]

<table>
<thead>
<tr>
<th>A</th>
<th>B</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>0</td>
<td>0</td>
</tr>
<tr>
<td>0</td>
<td>1</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>1</td>
<td>1</td>
</tr>
</tbody>
</table>

**Figure 3.2:** OR Gate Boolean Expression: \( O = A + B \)
Figure 3.2: OR Gate Boolean Expression: \( O = AB^* + A^*B + AB = A + B \)

**NOT gate truth table**

<table>
<thead>
<tr>
<th>Input</th>
<th>Output</th>
</tr>
</thead>
<tbody>
<tr>
<td>0</td>
<td>1</td>
</tr>
<tr>
<td>1</td>
<td>0</td>
</tr>
</tbody>
</table>

Figure 3.3: NOT Gate Boolean Expression: \( O = A^* \)

1. Launch Multisim. Access “Getting Started with NI Circuit Design Suite” by clicking: **Help > Getting Started** from the Main Menu. Read through the documentation to get a feel for the capabilities of the application software.

2. Read through the Digital Logic Fundamentals tutorial on National Instruments Website, which can be found at:


3. Study the following circuits, as shown in Figures 3.4 and 3.5. Use Boolean Rules for each of the gates to write the output of each circuit in terms of the inputs.

![Figure 3.4: AND-OR-INVERT (AOI) Logic Example 1](image-url)
Figure 3.5: AND-OR-INVERT (AOI) Logic Example 2

NOTE:

1. Inputs are on the Left, and Outputs on the Right (by convention). A Top/Bottom convention may also be used (albeit less frequently).

2. Inputs and outputs are clearly labeled.

3. Each gate is given some unique identification. In the case of automatic schematic capture, the assignment is done by the software.

With this approach, your design may be explained and critiqued by others easily.

4. LABORATORY ASSIGNMENT

4.1 Multisim Tutorial

There are two classes of digital circuits, combinatorial and sequential. Combinatorial digital circuits have static (non-time varying) inputs and outputs. Such circuits can be simulated by applying fixed Boolean inputs and measuring the (fixed) Boolean output(s). All of the circuits that you have been exposed to (so far) in this course have been combinatorial. In this section, we will use Multisim to simulate combinatorial circuits using a variety of methods.

4.1.1. Creating a design
Launch Multisim. This creates a blank design called “Design1”, as illustrated in Figure 4.1.
Figure 4.1: Blank Design with the default name “Design1”.

Save the file with the desired name via the Menu Bar **File > Save As** to use the standard **Windows Save Dialog**, as shown in Figure 4.2 to avoid any potential loss of work. Navigate to the directory where you want to save your design, enter the desired filename, and click the **Save** button. The default file extension for Multisim design files is `ms[XX]`, where `[XX]` represents the version number. At the time of the writing, Multisim 14.1 is being used, meaning that the file extension is `.ms14`.

**NOTE:** Create a separate directory on your flash drive to for exclusively storing your Multisim designs and related files. This will ensure that (1) your files will be accessible from any lab computer, and (2) your Multisim files will not be mixed in with other files. It is also a good idea to periodically make backup copies of your files as protection against data loss.

A previously created design can be opened via **File > Open**. In the Dialog Window, navigate to the directory in which the design is stored, select the file, and click the Open button.
4.1.2 Draw a Schematic Diagram of the Circuit

**Placing Components**

A schematic diagram comprises one or more circuit components, interconnected by wires. Optionally, signal “sources” may be connected to the circuit inputs, and “indicators” to the circuit outputs. Each component is selected from the Multisim library and placed on the drawing sheet in the Circuit Window (also called the Workspace). The Multisim library is organized into “Groups” of related components (Transistors, Diodes, Misc Digital, TTL, etc.). Each Group comprises one or more “Families”, in which the components are implemented with a common technology. For designing and simulating digital logic circuits in ECE 85L, two groups are to be used: “Misc Digital” (TIL Family only) and “TTL”.

The “Misc Digital” Group has three families of components, of which family “TIL” contains models of generic logic gates, flip-flops, and modular functions. These components are technology independent, which means that they have only nominal circuit delays and power dissipation, unrelated to any particular technology. Generic components can be used to test the basic functionality of a design, whereas realistic timing information requires the use of technology – specific part models, such as those in the TTL group.
To place a component on the drawing sheet, select it via the Component Browser, which is opened via the Component Tool Bar or the Menu Bar. From the Menu Bar, select **Place > Component** to open the Component Browser window, illustrated in Figure 4.3. You can also open this window by clicking on the **Misc Digital** icon in the Component Toolbar. On the left side of the window, select “**Master Database**”, **Group “Misc Digital”**, and **Family “TIL”**. The component panel in the center lists all components in the selected family. Scroll down to and click on the desired gate (NOT in Figure 4.3); its symbol and description are displayed on the right side of the window. Then click the **OK** button. The selected gate will be shown on the drawing sheet next to the cursor; move the cursor to position the gate at the desired location, and then click to fix the position of the component. The component can later be moved to a different location, deleted, rotated, etc. by right clicking on the component and select the desired action. You may also select these operations via the Menu Bar Edit Menu. Place down 2 NOT gates, 2 AND gates, and an OR gate, as shown in Figure 3.4.

![Component Browser: Misc Digital, TIL Family, NOT Gate Component Selection.](image)

**Figure 4.3:** Component Browser: Misc Digital, TIL Family, NOT Gate Component Selection.

After a component has been positioned, the component browser is redisplayed and additional components can be placed by repeating the above actions. When the last component has been placed, click the **Close** button to close the component selection.
window. You may return to the component browser at any time to add additional components.

The TIL family part naming convention is as follows. Look at the symbol and function panels on the right side of the component window to determine the specific function of a selected component.

1. Basic Logic Gates: \textbf{AND}_x, \textbf{NAND}_x, \textbf{NOT}, \textbf{NOR}_x, \textbf{OR}_x, \textbf{XNOR}_x, \textbf{XOR}_x, \text{where } x \text{ is the number of gate inputs (2-8).}

2. Flip-Flops: \textbf{D}_FF, \textbf{JK}_FF, \textbf{SR}_FF, \textbf{T}_FF, \text{and latches \textbf{D}_LATCH, \textbf{SR}_LATCH, \text{plus versions that have active-high or active-low asynchronous Set and Reset pins (ex. \textbf{D}_FF\_POSSR \text{ and } \textbf{D}_FF\_NEGSR, respectively).}

3. Standard Digital Modules: \textbf{Multiplexers, Decoders, Encoders, Counters, ALU, BCD-to-7 Segment Decoder, Registers, Shift Registers, etc.}

4. \textbf{DIGITAL\_PULLUP, DIGITAL\_PULLDOWN}: To pull up a pin to a constant 5 V (Logic 1) or down to ground (Logic 0), respectively.

Figure 4.4 shows the schematic diagram with 5 placed components. Note that each placed gate as a “Designator” (\textbf{U2}, \textbf{U3}, \textbf{U4}, \textbf{U5}), which can be used when referring to that gate. You can change a designator by right clicking on the component, selecting \textbf{Properties}, and entering the desired name on the \textbf{Label Tab}.

\begin{figure}[h]
\centering
\includegraphics[width=\textwidth]{schematic_diagram.png}
\caption{Schematic Diagram with all Placed Components.}
\end{figure}
Drawing Wires
Wires are drawn between component pins to interconnect them. Moving the cursor over a component pin changes the pointer to a crosshair, at which time you may click to initiate a wire from that pin. This causes a wire to appear, connected to the pin and the cursor. Move the cursor to the corresponding pin of the second component (the wire follows the cursor) and click to terminate the wire on that pin. If you do not like the path selected for the wire, you may click at a point on the drawing sheet to fix the wire to that point and then move the cursor to continue the wire from that point. Terminating a wire is accomplished by double clicking a partially drawn wire at a point on the workspace. You may also initiate or terminate a wire by clicking in the middle of a wire segment, creating a “Junction” at that point. This is necessary when a wire to be fanned out to more than one component input. Wires can also be created by selecting Place > Wire from the Tool Bar. A partially-wired circuit, including one junction point, is illustrated in Figure 4.5. Complete the wiring of the circuit, as indicated by the first Prelab Problem.

![Circuit Diagram](image)

**Figure 4.5:** Partially wired circuit, with 1 junction point.

4.1.3 Generating test input patterns

To drive circuit simulations, Multisim provides several types of “Sources” to generate and apply patterns of logic values to digital circuit inputs. Sources are placed on the Schematic sheet and connected to the circuit inputs in the same way as circuit components, selecting them from the “Digital_Sources” Family of the “Sources”
Group in the Component Browser. Note that there is a Place Source shortcut icon in the toolbar.

There are three basic digital sources:

1. **DIGITAL_CONSTANT**: This is a box with a constant Logic 1 or 0 output, and is used when the logic values are not to be changed during simulation. To change the Output Value, right click on the box, select Properties, selected the desired value on the Value Tab, and click the OK button.

2. **INTERACTIVE_DIGITAL_CONSTANT**: This is a clickable box that can be connected to a circuit input. Clicking on the box toggles its output between 0 and 1. This can be used to interactively change a circuit input during simulation.

3. **DIGITAL_CLOCK**: This is a box that produces a repeating pulse train (square waveform), oscillating between 0 and 1 at a specified frequency. To set the frequency and duty cycle, right click on the box, select Properties, select the desired Frequency and Duty cycle value on the Value Tab, and click the OK button.

Figure 4.6 shows the circuit of Figure 4.5 with an INTERACTIVE_DIGITAL_CONSTANT connected to each input. Note that the initial state of each is Logic 0. Since this circuit has only 2 inputs, all 4 input patterns can be produced (to generate a truth table for the circuit) by manually toggling the inputs. To change which Key binding toggles the input, double click on Key = Space, and under the Value Tab, change the Key for toggle option to A and B, respectively, for the two INTERACTIVE_DIGITAL_CONSTANTs. This allows for the ‘A’ key to toggle the first input HIGH or LOW, and the ‘B’ key to toggle the second input HIGH or LOW.
4.1.4 Connect circuit outputs to indicators

To facilitate studying the digital circuit output(s), Multisim provides a variety of “Indicators”. For digital simulation, the most useful are digital “Probes”, Hex Displays, and the Logic Analyzer instrument. A probe, illustrated in Figure 4.6, displays a single digital value as HIGH or LOW (the probe is “illuminated” in Figure 4.6, indicating a HIGH condition). The PROBE Family of the Indicators Group includes a generic PROBE_DIG and several PROBE_DIG_color indicators (color = BLUE, GREEN, ORANGE, RED, YELLOW). The probe in Figure 4.6 is PROBE_DIG_BLUE. This circuit can be verified by manually changing the three INTERACTIVE_DIGITAL_CONSTANT inputs to each of the 4 possible combinations, and recording the probe value for each combination to create a truth table.

4.1.5 Run the simulation

A simulation is initiated by pressing the Run button (Green Arrow) in the Tool Bar or via the Menu Bar under Simulate > Run. Alternatively, simulation can be initiated from a Word Generator by pressing the Cycle, Burst, or Step buttons. Simulate the circuit, and toggle the A and B inputs to determine all possible combination of outputs (create a Truth Table). What logic function has been created using this combination of AND, OR, and NOT gates?
When you are finished simulating the circuit, stop the simulation by pressing the **Stop** button (**Red Square**) or via the Menu Bar under **Simulate > Stop**. Whenever you are finished simulating a circuit, be sure to stop the simulation since many simulations are CPU intensive and will continue to run until it is formally stopped.

You may capture any window and paste it into a Word or any other document for generating reports. An individual window is captured by pressing the **ALT** and **Print Screen** keys concurrently. You may then “Paste” the captured window into a document via the editing features of that document. To capture a circuit diagram in the main window, the simplest method is via the Menu Bar **Tools > Capture Screen Area**. This produces a rectangle whose corners can be stretched to include the screen area to be captured; the “Copy” icon on the top left corner is pressed to copy the area, which may then be pasted into a document. The circuit images in this document were copied in this manner.

4.1.6 Save the design and close Multisim

The simplest way to save a design is to click the Save icon in the Design Toolbar on the left side of the window, directly above the design name. Alternatively, you may use the standard Menu Bar **File > Save**. As mentioned earlier, you should save all designs in a special course directory on your flash memory device. Save the circuit as **ECE85L-Lab2-Circuit1.ms14**.

Multisim is exited in the same manner as any other Windows program.

4.1.7 Alternative Input Sources and Output Probes

Reopen your **ECE-Lab2-Circuit1.ms14** file.

Another mechanism to apply all possible input combinations to a circuit is to connect a **DIGITAL_CLOCK** to each input, with the frequency of the first clock to set some value N, the second to 2N, etc, where N represents the fundamental frequency. This produces a pattern that varies between 11, changing to 10, 01, and 00, before repeating. **DIGITAL_CLOCK** sources are also used to drive the clock inputs of sequential circuits.

Modify your circuit by removing the **INTERACTIVE_DIGITAL_CONSTANT**s, and replace them with 2 **DIGITAL_CLOCK** sources from the **Sources Group, DIGITAL_SOURCES Family**. Double click on the two sources, and change the Frequency to 1 Hz and 2 Hz, respectively, as indicated in Figure 4.8. Place red digital probes after each clock source and re-run the simulation. Can you re-verify the Truth Table in this manner?
Figure 4.8: DIGITAL_CLOCK Sources being used in place of INTERACTIVE_DIGITAL_CONSTANTS.

One additional mechanism for generating digital patterns is the “Word Generator” instrument, which can be added to the circuit via the Shortcut icon on the right side of the main window, or via the menu bar under Simulate > Instruments > Word Generator. Figure 4.9 shows a Word Generator (XWG1) connected to the two circuit inputs. The Word Generator produces a sequence of patterns, each containing from 1 to 32 bits.
To specify the patterns and the rate at which they should be produced, double click on the Word Generator symbol, producing the window shown in Figure 4.10. The Control Buttons on the left of this window allow patterns (1) to be continuously applied and repeated (Cycle button), (2) a single set of patterns to be applied (Burst button), or (3) a single pattern to be applied (Step button). The patterns displayed in this window will be applied in the order listed, and can be entered manually or generated automatically.

For automatically-generated patterns, press the Set... button, producing the Settings Window of Figure 4.11. In this example, “Up Counter” is selected to produce a sequence of binary numbers, “Buffer Size” is set to 0004 to limit the sequence to 4 numbers, and “Initial Pattern” is set to 0. Note that the 4 binary numbers are displayed in the Word Generator window of Figure 4.10. The rate at which the patterns are to be applied to the circuit is specified via the Frequency box of the Word Generator window. In Figure 4.10, the frequency has been set to 1 kHz, which means that 1000 patterns will be generated each second, or one pattern every 1 ms. Apply the necessary changes to the Word Generator for Burst Mode, Up Counter, with an Initial Pattern of 0000, and a Frequency of 1 kHz.
Another useful indicator is the Logic Analyzer instrument, shown on the right side in Figure 4.9. A Logic Analyzer is an instrument that captures and displays sequences of
digital values over time, with the sequences displayed as waveforms rather than as tables of numbers. The Multisim Logic Analyzer instrument can capture and display up to 16 signals. Samples are triggered either by an **Internal Clock** (N samples per second) or by an **External Clock**. Figure 4.12 shows the Logic Analyzer display of the 2 Inputs and the Output of the circuit in Figure 4.9, showing the response of the circuit to the burst of Word Generator values shown in Figure 4.10, plus the Internal Clock that was used to capture samples.

The Logic Analyzer must be configured to capture values at the correct sampling times. This is done by clicking on the **Set...** Button in the Clock area under the Logic Analyzer display, opening the Clock Setup window shown in Figure 4.13. Since the Word Generator was configured to generate patterns at a rate of 1 kHz, a **Clock Rate** of **2 kHz** was set for the logic analyzer to make it capture two values for each input pattern.

![Logic Analyzer Display](image)

**Figure 4.12:** Logic Analyzer Display of the 2 Inputs and 1 Output of the Circuit in Figure 4.9, for the 4 Patterns Produced by the Word Generator.
After making the necessary adjustments to the Logic Analyzer, run the simulation and capture the waveforms with the Logic Analyzer. Examine the Output Logic Levels as a function of all possible Input Logic Levels. Can you verify the Truth Table of the circuit? Discuss and explain your results.

4.2 Simulating physical circuits

The simulation techniques discussed thus far used generic logic devices to generate logical 1’s and 0’s for the Inputs and Outputs of the circuit. However, Multisim can be used in a more sophisticated manner to simulate physical circuits by applying virtual voltages and currents to the digital logic devices. Opposed to using the TIL Family of logic from the Misc Digital Group, we will begin simulating Transistor-Transistor Logic (TTL) logic devices from the TTL Group. In this manner, digital logic gates on virtual TTL chips may be used with virtual voltage sources, switches, LEDs, and resistors to produce a more realistic simulation.

4.2.1 TTL Chip Placement

Begin by creating a new Design by pressing File > New > Blank > Create. Using the TTL Group, 74STD Family, we can begin placing gates using the standard TTL logic convention for logic gates. For the AND, OR, and NOT gates, the following chips may be used:

- **NOT**: 7404N HEX INVERTER (6 NOT Gates)
- **AND**: 7408N QUAD 2-INPUT AND (4, 2-Input AND Gates)
- **OR**: 7432N QUAD 2-INPUT OR (4, 2-Input OR Gates)

Under the TTL Group, 74STD Family, select the above Components and place them in your new Design to create the second Prelab circuit, as shown in Figure 4.14. When placing a component, Multisim will prompt you to use a new chip (by clicking on NEW),
or an existing chip with Input A, B, C, etc. For a HEX Inverter, the chip has 6 NOT Gates (NOT A, NOT B, NOT C, etc.) – one of which can be selected by clicking on A, B, C, etc.

Use a single HEX INVERTER chip and a single QUAD 2-INPUT AND chip to create your circuit. Refer to Figure 4.14 for guidance. The gates will be labeled U1A, U1B, U1C, U1D, and U2A, U2B, U2C, U2D for the 7404N and 7408N chips. After placing the logic gates, hook up wires to create the appropriate circuit.

![Multisim Logic Gates](image)

**Figure 4.14:** Multisim Logic Gates using the TTL Group of Components.

### 4.2.2 Input switches

To create a switch to toggle between a Logic Level 1 and a 0, we must generate a 5 V or a 0 V signal without creating a short circuit. This can be performed using 5 V Voltage Sources, Ground, Single Pull Single Throw Switches, and 10 kΩ Pull-up Resistors.

The 5 V DC Power Supply and Ground can be found under **Place > Component > Group: Sources > Family: POWER_SOURCES > Component: VCC** and **GROUND**, respectively. Resistors and switches can be found under **Place > Component > Group: Basic > Family: RESISTOR** or **Family: SWITCH > Component: SPST**, respectively. Using these components, create the necessary input circuits, as shown in Figure 4.15. Double click on the **SPST Switches**, and change the toggle keys to A and B for the two inputs.
Figure 4.15: 5 V and 0 V Inputs (Logic Level 1 and 0, respectively) using Vcc, Ground, SPST Switches, and 10 kΩ Resistors to avoid Short Circuits when using TTL Chips.

Note that the required input current for digital TTL logic devices ranges from the µA to mA range. Thus 10 kΩ pull-up resistors are used to ensure that the current in the circuit is limited to 0.5 mA. This is determined via Ohm’s Law, which states that the Voltage V (V) dropped across a device is equal to the Current I (A) times the Resistance R (Ω). In other words, \( V = I \times R \).

To understand how the current is limited, consider the two switch positions. When the switch is open, there is no direct connection from the voltage source to ground, meaning that current cannot flow. This means that the change in voltage across the resistor is 0 V, since \( \Delta V = 0 \text{ A} \times 10 \text{ kΩ} = 0 \text{ V} \). Since there is no change of voltage across the resistor when the switch is open, the 5 V from the Voltage Supply is copied across the resistor to the input of the logic gate (meaning that the input voltage is 5 V), which is interpreted by the TTL logic gate as a Logic Level 1.

Alternatively, when the switch is closed, there is now a direct connection from the voltage source to ground via the 10 kΩ resistor. The 5 V is dropped across the resistor, meaning that the current \( I = \frac{V}{R} = \frac{5 \text{ V}}{10 \text{ kΩ}} = 0.5 \text{ mA} \). With the current limited to 0.5 mA, a short circuit is averted, since without the resistor, there would be a direct connection from the 5 V source to ground. Furthermore, there is now a 0 V connection.
directly from ground to the input of the logic gate, meaning that it is interpreted as a Logic Level 0.

To summarize, when the SPST switch is open, there is an open circuit, and the input into the logic gate is 5 V (Logic Level 1). Likewise, when the SPST switch is closed, there is a closed circuit, and the input into the logic gate is 0 V (Logic Level 0). In either case, a short circuit is averted, ensuring that the TTL chips do not burn out. Summarize and explain these concepts in your laboratory notebook.

4.2.3 LED Output Indicators

LED Output Indicators can be added to the outputs to determine their respective logic levels, but the current must be limited to 20 mA or less for typical Silicon LEDs to avoid burn out. Current limiting can performed by placing a 330 Ω in series with the Si LED before dropping it to ground, as illustrated in Figure 4.16.

A Red LED can be found for your circuit under Place > Component > Group: Diodes > Family: LED > Component > LED_red. The resistor can be found under Place > Component > Group: Basic > Family: RESISTOR, and ground under Place > Component > Group: Sources > Family: POWER_SOURCES > Component: GROUND.

Figure 4.16: Completed Circuit using Current Limited Resistors and Output Indicators
When the output of the logic gate is 0 V, there is no voltage difference between the anode (positive end) and cathode (negative end) of the LED. With no voltage difference, no current flows through the device, and the light remains off. Since typical Red Si LEDs have a turn on voltage of approximately 1.7 V, when the output of the logic gate is 5 V, about 3.3 V is dropped across the 330 Ω resistor (5.0 V – 1.7 V = 3.3 V). Ohm’s Law dictates that the current flowing through the resistor (and hence the LED) is therefore $I = \frac{V}{R} = \frac{3.3 \text{ V}}{330 \Omega} = 10 \text{ mA}$. Since typical TTL logic gates can only source around 20 mA of current, and most LEDs can only handle 20 mA of current before burning out, the 330 Ω resistor ensures that the output current is limited through the TTL logic gate as well as through the LED.

After completing your circuit, simulate the result. Toggle the A and B inputs to determine the Truth Table of the circuit. Describe the circuit in detail. Remember that the input is 1 (5 V) when the SPST switch is open, and 0 (0 V) when the SPST switch is closed.

As a final note, once the simulation of the completed circuit is working, the Schematic Capture may be exported to Ultiboard by clicking on Transfer > Transfer to Ultiboard, as indicated by Figure 4.17. This exports the circuit to an initial Printed Circuit Board (PCB) design using standardized circuit components, which can be used to generate a Bill of Materials, Netlist, etc.

Although PCB design is beyond the scope of this lab, it is the next stage in the fabrication process for a “real” circuit. The file that is generated by Ultiboard can be exported to a Computer Controlled Milling Machine, or sent to a PCB Company for fabrication. Once the PCB has been fabricated, it can be populated with standardized circuit components (7404N, 7408N, etc.), allowing for a rapid prototyping environment.

Figure 4.17: Export of Schematic Capture to Ultiboard for initial PCB Design.
4.3 Extra Credit

The HEX_DISPLAY Family contains a variety of displays that show multi-bit values as hexadecimal numbers. This is useful for circuits such as binary counters, as illustrated in Figure 4.18. In this example, the **DCD_HEX_GREEN** Indicator is a “Decoded” Hex display digit from a **74LS163D** binary counter, which means that the indicator displays the Hex number corresponding to the states of its 4 binary inputs.

Create the circuit shown in Figure 4.18 in Multisim. Explain the functionality of the circuit, along with all inputs and outputs, to the best of your knowledge. Verify the operation of the circuit with the Logic Analyzer with an external trigger from the 10 Hz clock source. Provide and analyze the timing waveforms from the Logic Analyzer (see Figure 4.19 for an example) to support your explanation. Can you modify the circuit to preload a specific hex digit into the counter?

Sequential circuits, including counters, will be covered in detail toward the end of the semester.

![Figure 4.18: DCD_HEX_GREEN Decoded Hex Indicator Displays a Counter Output](image)
4.4 Review Questions

1. How are Truth Tables in Binary Logic represented using TTL Logic Gates?

2. How can you avoid shorting an input to a TTL Logic Gate?

3. How is current limited in an LED?

4. What other important factors must be considered when creating a digital simulation of a physical circuit?

5. ACKNOWLEDGEMENT

Portions of this lab have been adapted from Dr. Victor Nelson and Dr. Guofu Niu, Electrical and Computer Engineering Department, Auburn University.