# LAB 1

# BUILDING CIRCUITS

In Lab 1, you design and build combinational circuits of logic functions. Before the lab, you prepare by drawing schematics of logic functions using Logisim Evolution. During the lab, you build the circuits using chips and wires. Although circuits are no longer built this way in industry, it is useful to show how discrete gates are connected together to form a logic function.

This document describes what you need to prepare and demonstrate for Lab 1. Section 1.3 introduces the digital lab space and describes the different physical parts and tools you need for your lab demonstration. Section 1.4 describes the tasks you must complete *before* your lab session. Section 1.5 describes the tasks you complete *during* your lab session. The next section describes lab logistics in more detail.

## 1.1 Logistics

Even though you work in pairs during your lab session, you are assessed individually on your Lab Preparation ("pre-lab") and Lab Demonstration ("demo"). All pre-lab exercises are submitted electronically before your lab (see the course website for exact due dates, times, and the submission process). So, **before** each lab, you must read through this document and complete all the pre-lab exercises. During the lab, use your pre-lab designs to help you complete all the required in-lab actions. The more care you put into your pre-lab designs, the faster you will complete your lab.

The Lab Preparation must be completed individually and submitted online by the due date. Before beginning the pre-lab, read Section 1.3. Then follow the steps in Section 1.4 for the pre-lab. Remember to **download the starter files**.

You must upload *every required file* for your pre-lab submission to be complete. But you do *not* need to include images that are not on the list of required files (even if those images are in your lab report). If you have questions about the submission process, please ask ahead of time. The required files for Lab 1's pre-lab (Section 1.4) are:

- Your lab report: `lab1_report.tex`, `lab1_report.pdf` (as generated from the tex file)
- Your digital designs: `lab1_part1.circ`, `lab1_part2.circ`, `lab1_part3.circ`

The Lab Demonstration must be completed during the lab session that you are enrolled in. During a lab demonstration, your TA may ask you to: go through parts of your pre-lab, run and simulate your designs in Logisim, and answer questions related to the lab. You may not receive outside help (e.g., from your partner) when asked a question.

## 1.2 Marking Scheme

Each lab is worth 4% of your final grade, where you will be graded out of 4 marks for this lab, as follows.

- Prelab: 1 mark
- Part I (in-lab): 1 mark
- Part II (in-lab): 1 mark
- Part III (in-lab): 1 mark

## 1.3 Introduction to the digital lab

This section describes the different pieces of equipment you use in this lab:

- The protoboard
- The digital switch/light board
- The logic probe
- The wire strippers
- The chip puller
- The 7400-series chip packages

### 1.3.1 The boards

The protoboard (also called the breadboard) is for holding and connecting chips. As illustrated in Figure 1.1, chips are inserted across the middle *valley* in the protoboard. The set of holes in a vertical line above the valley are connected electrically, as are the vertically aligned holes below the valley. Therefore, each pin of the chip in the board is connected to the holes above (or below) the pin as directed by the location of the pin with respect to the valley. To make a connection to a specific pin, you need only make connections between the holes, by plugging the bare end of a wire into any of the holes above (or below) that pin.

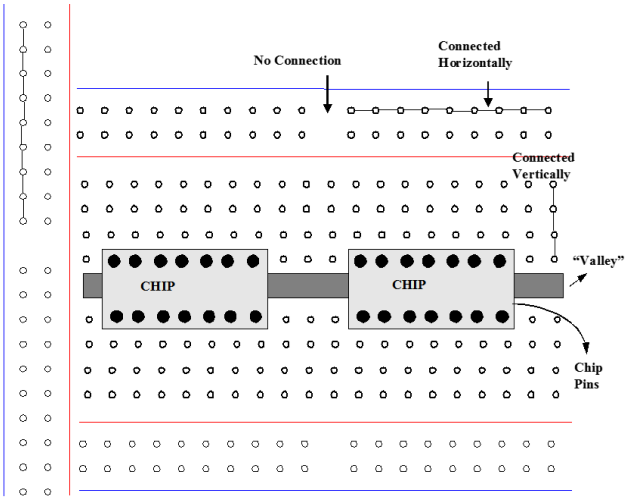


Figure 1.1: Protoboard

In Figure 1.1, the horizontal lines at the top and bottom of the board delineate holes that are connected horizontally; note that the space in the middle indicates a disconnection. The horizontally-connected holes at the top and the vertically connected holes at the side are usually connected to the power and ground provided by the external connector. The power and ground of the chips are then connected to these strips of holes. **The first thing you should do in the lab is connect power and ground to these horizontal and vertical strips.** However, do not turn the power supply on until you have verified that all the connections in your circuit are as intended.

The digital switch/light board provides switches that have digital output **(5V = logic 1, 0V = logic 0)** and lights that can be driven by logic signals (i.e., logic 1 turns a light on, logic 0 turns it off). The pin out connections for the header on the digital switch board can be seen in Figure 1.7. Test the board by connecting the switches to the lights. The board also provides a clock, which can have its frequency varied by inserting different capacitors into the holes next to it, and a seven-segment display.

### 1.3.2 The tools

The logic probe is used for measuring the logic values of signals on the board. Ensure that it has power attached to the correct terminals. To test the probe, touch it to the +5V and ground on the protoboard, to verify that it correctly indicates the values high (1) and low (0) respectively.

A wire stripper is used to (more) easily remove the electrical insulation from electric wires. You strip a wire so that you can connect it to the breadboard. The wire strippers are attached to each workstation to make sure they don't get lost. If you have never stripped a wire before, try it!

The chip puller should always be used to remove chips from the protoboard. Doing it with your fingers bends the pins and ultimately breaks them, so **only remove chips with chip puller**!

### 1.3.3 The chips

The chips that you will use in this lab are Small Scale Integration (SSI - meaning there's not much logic on a single chip) 7400 series (see Figure 1.2). Depending on exactly which chip you end up using in the lab you may have to set the logic probe to one of two settings: TTL or CMOS. This setting depends on the type of technology used for the transistors in the chips.

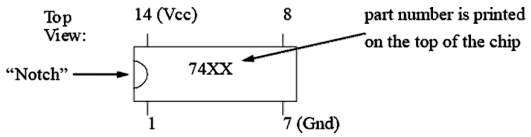


Figure 1.2: Pin Numbering of TTL Chips

All of the chips you will use are *Dual In-line Packages* or DIPs. Most of the packages have 14 pins, and the pins are numbered from looking at the chip from the top (note the notch in Figure 1.2). Below the *notch* is pin 1 to pin 7 and above the notch is pin 14 down to 8. Figure 1.8 shows the Pin-out numbers and schematics for all of the 7400-series chips that we have access to in our labs. For this lab, you should take a close look at 74LS04, 74LS08, and 74LS32.

**CAUTION**

You should always connect the power (Vcc; Pin 14 in 14-pin chips) and ground (Pin 7 in 14-pin chips) of chips to VCC and ground. Leaving the ground pin (Pin 7) unconnected is **NOT** the same as connecting it to ground, and your chip will not work correctly.

### 1.3.4 Bringing it all together

The components described above are used to implement the physical circuits that you will be designing below. In the first 15 minutes of the first lab session, the Teaching Assistants will give a short demo of how to use the protoboard and chips and wires and tools. Before you start working on the lab exercises below though, you should view the module on Quercus called "Lab Breadboard Demo".

## 1.4 Lab Preparation

*Make sure you have read Section 1.3 before beginning your pre-lab. You may also want to refer to the figures in Section 1.6.*

The pre-lab for Lab 1 consists of three parts. In Part 1, you design a 2-to-1 multiplexer. In Part 2, you design an unnamed logic function. In Part 3, you design a 2-to-1 multiplexer a little differently (peek at Section 1.5 for why). Make sure that, in Logisim Evolution, `Enable Vcc and Gnd ports` is set to `Yes` for each `TTL` chip in your schematic.

### 1.4.1 Part I

The multiplexer is a combinational circuit that chooses one of multiple inputs to be the output. The number of inputs to a multiplexer is typically a power of 2. Multiplexers are an extremely useful circuit with many applications and are used, for example, in the datapath of a CPU.

The Boolean function $f = x\overline{s} + ys$ is a 2-to-1 multiplexer. The $s$ is called the control signal or "select". As we can see, when select is 0, then the output is $x$. However, when $s$ is a 1, then the output is $y$. Perform the following steps:

1. Write out the truth table for the design and include it in your report.

2. In Logisim Evolution, draw a schematic that implements the function, using only 74LS04 (NOT), 74LS08 (AND) and 74LS32 (OR) chips. Save your design with the name: `lab1_part1.circ`. Export the schematic as an image and include it in your report. Also remember to upload the schematic (i.e., the `.circ` file) as part of your required files.

3. Is there a *cheaper* implementation for your design, assuming you are still limited to using the same three chip types? Include your analysis in the report.

### 1.4.2 Part II

Consider the Boolean function: $f = \overline{a + b} + c\overline{b}$. Perform the following steps:

1. Write out the truth table for the design and include it in your report.

2. In Logisim Evolution, draw a schematic that implements the function, using only 74LS04 (NOT), 74LS08 (AND) and 74LS32 (OR) chips. Save your design with the name: `lab1_part2.circ`. Export the schematic as an image and include it in your report. Also remember to upload the schematic (i.e., the `.circ` file) as part of your required files.

3. Is there a *cheaper* implementation for your design, assuming you are still limited to using the same three chip types? Include your analysis in the report.

### 1.4.3 Part III

Re-create the 2-to-1 mux from Section 1.4.1. Except, instead of using the `TTL` chips, use the NOT, AND, and OR gates directly (i.e., under `Gates`). Save your design with the name: `lab1_part3.circ`. Export the schematic as an image and include it in your report. Also remember to upload the schematic (i.e., the `.circ` file) as part of your required files.

## 1.5 Lab Demonstration

The lab demonstration consists of three parts. The first two parts involve building a physical circuit of your designs from the pre-lab. The third part involves *synthesising* a circuit so that it runs on the DE1-SoC board. When demonstrating a part to your TA, be ready to answer questions **individually**.

### 1.5.1 Part I

Wire your design from Section 1.4.1 on the protoboard. Compare the output of your circuit to your truth table from the pre-lab and make sure they match. When you are ready to demonstrate this part to your TA, make sure you have your Logisim circuit open.

### 1.5.2 Part II

Wire your design from Section 1.4.2 on the protoboard. Compare the output of your circuit to your truth table from the pre-lab and make sure they match. When you are ready to demonstrate this part to your TA, make sure you have your Logisim circuit open.

### 1.5.3 Part III

Synthesise your design from Section 1.4.3 and download it onto the DE1-SoC board. You will need to perform the steps below. When you are done, demonstrate your working circuit to your TA.

**CAUTION**

Make sure you are using a lab computer (i.e., a desktop computer in BA3135, BA3145, BA3155, or BA3165) and have 'installed' Logisim 3.6.1. Also make sure you have `DE1_SOC.xml` downloaded on the lab computer. Finally, make sure the DE1-SoC board is turned on.

1. Navigate to `FPGA > Synthesize & Download`. A small window should open (Figure 1.3).
2. In the `Target board` panel of the window, select `Other` from the pull-down menu. An open file dialogue should open.
3. Navigate to, and `Open`, the `DE1_SOC.xml` file that you downloaded.
4. Click on the `Settings` button just above the `Target board` panel. A Preferences window should open (Figure 1.4). You should see `DE1_SOC` under `External FPGA Board list:`.
5. Click on the `Browse` button beside `Workspace location`. An open file dialogue should open.
6. Create a new subdirectory in your home directory and select it as your workplace location.

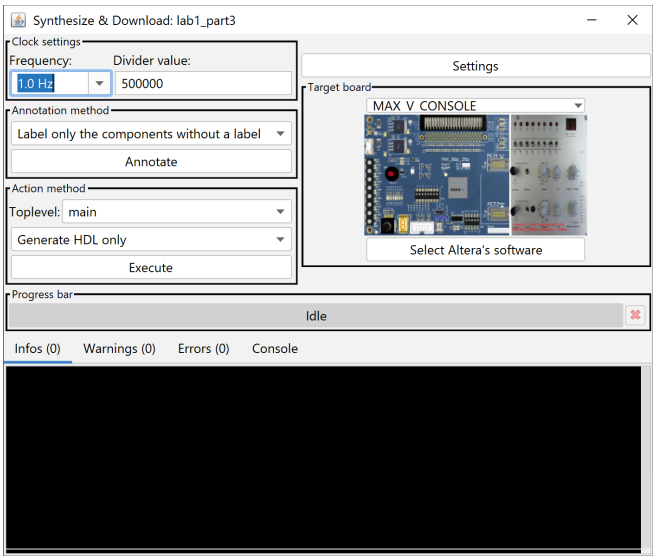


Figure 1.3: The 'Synthesize & Download' window

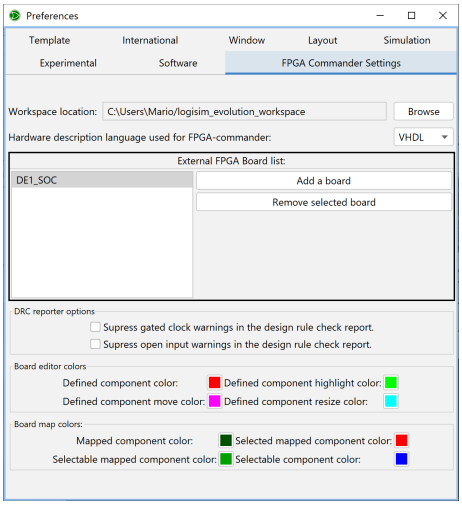


Figure 1.4: The Preferences: FPGA Commander tab

7. Navigate to the `Software` tab (Figure 1.5). Click on `Browse` next to the `Altera/Intel Quartus toolpath`. An open folder dialogue should open.

8. Navigate to, and `Open`, the `C:\DESL\Quartus18\quartus\bin64` folder.

9. Close the `Preference` window. Everything should now be configured.

10. In the `Action method` panel (Figure 1.3), select the appropriate Toplevel circuit (typically main). From the pull-down menu, make sure that `Synthesize & Download` (not `Generate HDL only`) is selected.

11. Click on `Execute` under `Action method`. A window should open (Figure 1.6).

    With this version of Logisim Evolution, your circuit's inputs can be mapped to the buttons and switches on the physical DE1-SoC board. Similarly, your circuit's outputs can be mapped to the LEDs.

12. Map the pins in the `Unmapped List:`. First, select one of the pins. The components that this pin can be mapped to are highlighted in translucent red on the image of the board. Click on one to

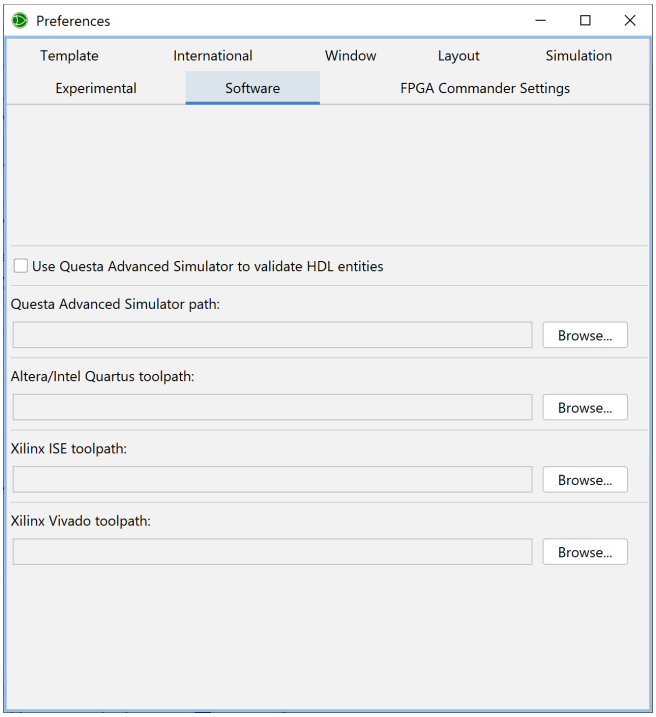


Figure 1.5: The Preferences: Software tab

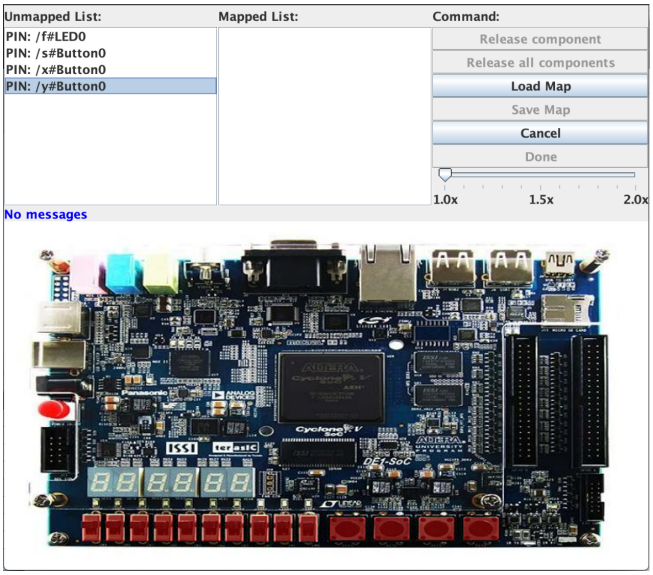


Figure 1.6: The board mapping window

map the pin. Repeat until all pins have been mapped to a physical component on the DE1-SoC board.

13. Click on `Done` to start synthesising your circuit. This may take some time.

14. Select `Yes, download` when presented with the option.

15. Test your circuit by toggling switches and observing the LEDs.

## 1.6 Reference figures

| Digital Board Header Pin Assignment | | | | | |
|---|---|---|---|---|---|
| **Pin#** | **Description** | | | **Description** | **Pin#** |
| 1 | Switch #1 | o | o | Switch #2 | 2 |
| 3 | Switch #3 | o | o | Switch #4 | 4 |
| 5 | Switch #5 | o | o | Switch #6 | 6 |
| 7 | Switch #7 | o | o | Switch #8 | 8 |
| 9 | Ground | o | o | NC | 10 |
| 11 | Ground | o | o | NC | 12 |
| 13 | Ground | o | o | NC | 14 |
| 15 | Ground | o | o | NC | 16 |
| 17 | LED #1 | o | o | LED #2 | 18 |
| 19 | LED #3 | o | o | LED #4 | 20 |
| 21 | LED #5 | o | o | LED #6 | 22 |
| 23 | LED #7 | o | o | LED #8 | 24 |
| 25 | Ground | o | o | NC | 26 |
| 27 | Ground | o | o | NC | 28 |
| 29 | Ground | o | o | NC | 30 |
| 31 | Ground | o | o | NC | 32 |
| 33 | Clock | o | o | NC | 34 |
| 35 | NC | o | o | NC | 36 |
| 37 | NC | o | o | Pulse Button | 38 |
| 39 | NC | o | o | NC | 40 |

Figure 1.7: Digital board header pin assignment

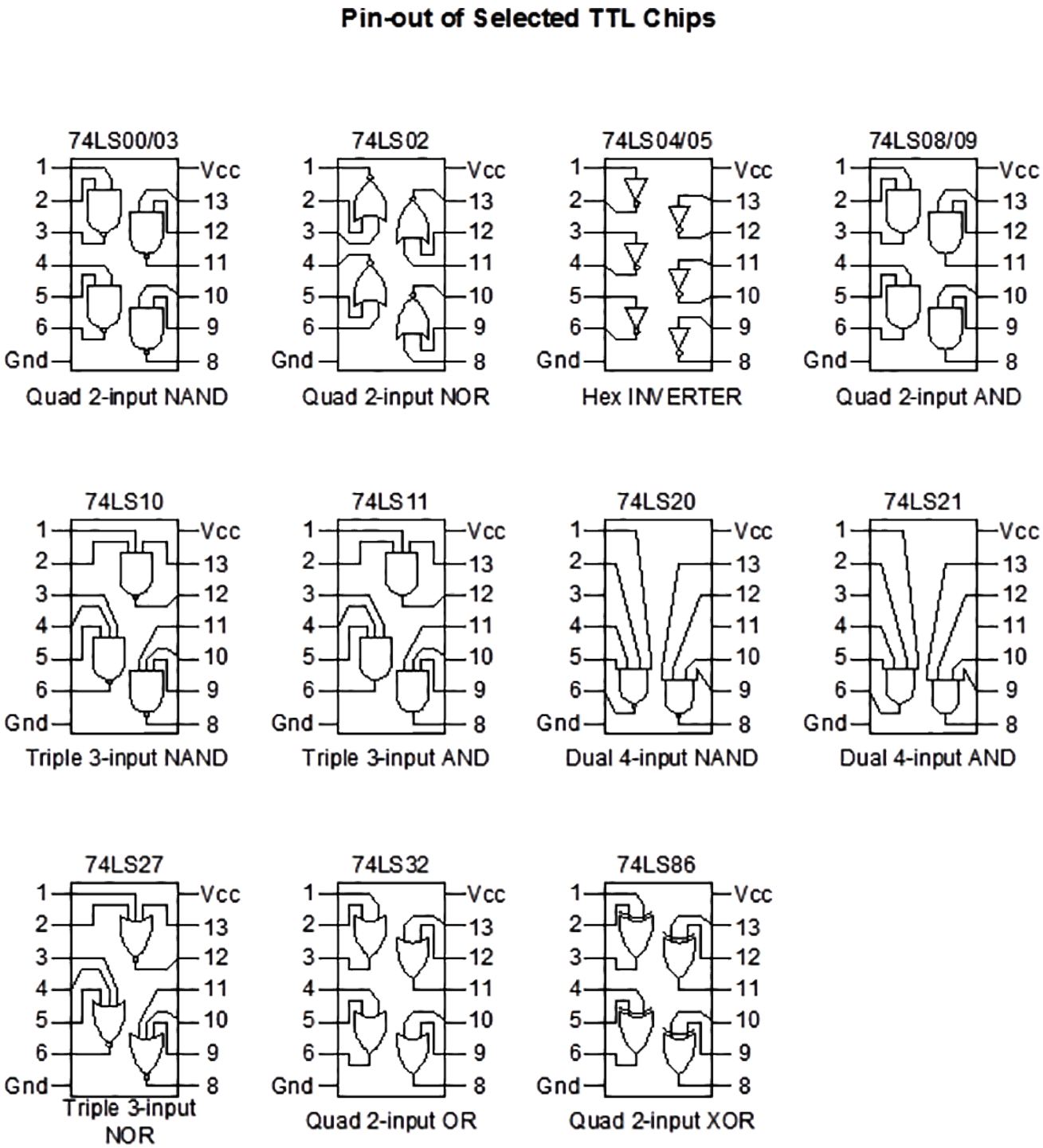


Figure 1.8: Schematics of 7400-series TTL Chips