EE241 Digital Design: FPGA Wiring Issues
April 18, 2017

Background:

The final EE241 Laboratory Project is to design and operate a Sequential Machine that is to direct traffic at an intersection. The preferred way to do this is using an Altera (Intel) "Max II" Field Programmable Gate Array (FPGA). Other documentation distributed separately describe the process of writing a Verilog program, or drawing a schematic, and then programming the device to do what is required. (The most difficult part of the design process is coming up with a state diagram and state table. Once that is done, converting it into Verilog or a schematic is relatively straightforward.) However, once programmed, the FPGA needs to operate in a circuit that includes a clock, timers, LED's and inputs that represent inductive sensors, treadles, or other stimuli. The purpose of this document is to address these circuit issues.

The critical issue:

The FPGA is a 3.3 Volt device. Most of the TTL or other devices we use (like the LM555 timer IC) usually run on 5 Volts. In general, it is highly desirable that any input or output connection to a device not be higher than the power supply (usually designated V_{DD} or V_{CC}) or lower than the reference / negative supply (Ground or V_{SS}). The FPGA board actually operates from a 5 Volt supply (the ones from the lab kit), but has an on-board regulator to reduce that to 3.3 Volts. The 3.3 Volts is available to power off-board circuitry, but care should be taken not to draw very much current. The 3.3 Volt supply maximum current or dissipation is unknown. The designer's primary concern becomes making sure that inputs don't exceed (at least by much) the 3.3 Volt V_{DD} supply.

Inputs:

There are two primary cases of inputs that need protection. One is signals taken from 5 Volt devices, typically TTL devices or things like the LM555 timer. Figure 1 shows reasonable precautions which should be taken. The diode reduces the maximum output Voltage by .7 Volts. The resistor to ground (which could be larger than 1K Ohms) ensures that enough current flows to reduce the output by about the .7 Volts expected from a silicon diode. The 10K Ohm resistor to the FPGA input ensures that any current flowing (presumably into the FPGA input protection diode circuitry) is minimal, and will not damage the device. In theory the input could exceed 3.3 Volts but will not be damaging because of the current limitations of the 10K Ohm resistor.)

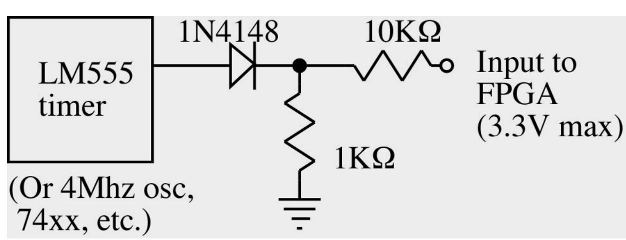

Figure 1 FPAA inputs from TTL or linear circuits

This circuit will be needed for the clock input, which is probably most conveniently taken from your 4MHZ TTL oscillator (unless you want some particular other frequency). A timer clock, for example to set Yellow and Green phases, is probably most conveniently provided by a 555 timer. If input is taken from the output of an op-amp (say, used as a comparator) extra precautions may be needed to prevent the output from swinging below ground. (The LM3900 is a nice op-amp for 5V power situations since it needs only the single supply.)

The other input case is from a switch or pushbutton as seen in Figure 2. The best solution, as shown, is to power the input from a 3.3 Volt source as shown in the figure. If a 5 Volt source is used, the protection circuit shown for TTL in Figure 1 should be used instead.

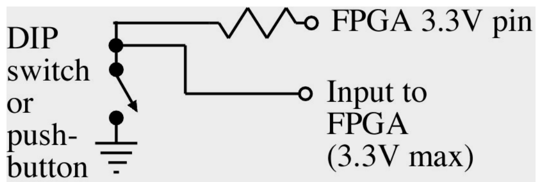

Figure 2 FPGA input from a switch or pushbutton

Outputs:

For this project, outputs will generally drive LEDs. That can be done by either sourcing or sinking current. (Be aware of current limits on pins, and design to avoid drawing too much current. If you need more current, use a buffer/amplifier or gang pins together.) For driving pins that connect to an LED to ground (so that a “1” drives the LED on), the issue is to expect about 3.3 Volts for a “1” output. For sinking current to turn an LED on, if the LED draws current from a 5Volt supply, one might be concerned that the current through the LED might put a Voltage above 3.3 Volts at the output when the output is high, or in three state. That really shouldn’t be a problem, because even for a Red LED the current is minimal at about 1.5 V drop, and the output protection should be able to handle it. (I’d be more cautious with infrared, though, and optoisolators, which generally use infrared LEDs. Those devices have about a 1.2 V drop.) Figure 3 shows acceptable output circuits.

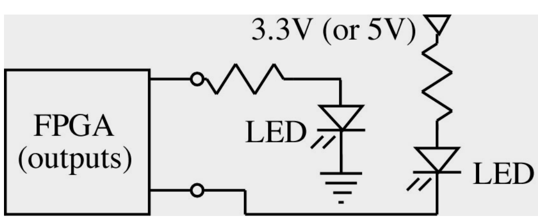

Figure 3 FPGA Output circuits

Conclusion:

The circuitry for the Traffic Light Controller (Lab 9) should be simple and easily designed and wired: It should just consist of LED displays and inputs including a clock or two and sensors represented by pushbuttons. The main point is to protect the FPGA from Voltages above 3.3 Volts. In developing a demonstration, the problem I had was that I tried to protect the FPGA by using two diode drops, which left the clock inputs too low in amplitude. Once the circuits above were adopted, it worked. Do be careful to trace the wires from the FPGA board to the solderless breadboard correctly; it’s easy to get turned around and not wire to the correct pins. Use of color coding can help a lot.