ELEX 7660 Project FAQ - Part 1

This lecture answers some questions that students have frequently asked while working on their projects.

Why isn't my circuit working?

When troubleshooting electronic equipment you should *always* follow the following sequence of steps[1]:

1. check power, ground and clock: your circuit will not work without these. If it's a new design, check at each IC.
2. set up an input that results in incorrect output and:
3. compare measured against predicted values starting at the input (correct) and working towards the output (incorrect). At some point you will find a difference between the predicted and measured values. The fault lies between this and the previous measurement.
4. repeat until all valid inputs produce the correct output

When most of your design is within an FPGA it will be difficult to probe intermediate points (e.g. with a 'scope). In many cases, particulary for unit testing, simulation is the most effective tool because of the ease with which you can change the design, the input, and display any signal.

When troubleshooting in real-time and in-circuit you can connect intermediate points to unused output pins and monitor these with a 'scope or use an embedded logic analyzer (Altera's SignalTap or Xilinx's ChipScope).

Other useful hints to minimize troubleshooting effort:

1. double-check everything as you go: It's much easier to avoid an error than to find it after you've made it.
2. test as you go (unit testing): it's much easier to find an error in a simple design or sub-circuit than in a complex one
3. neatness counts: our brains are good at spotting patterns and exceptions. Use regular patterns as much as possible (regular indentation, signal names, code structures, ...)

[1] This could be the most useful thing will learn at BCIT!

What's a design review?

A design review is a process that allows other engineers to review a design and point out potential problems that the design team may have missed. Companies conduct design reviews at critical stages of a design, for example, before proceeding to production. Depending on the complexity of the design, a review could range from an informal exchange of e-mails to multi-day meetings. Design reviews for software are often called "code reviews."

The review involves the designers presenting their design documents to the reviewers and answering questions. Any issues are noted and followed up.

Design reviews can be stressful because they implicitly involve criticism of work. They need to be carefully managed to ensure they remain constructive.

Does my project need a design review?

You are welcome to conduct design reviews with your fellow students but in this course your instructor will only review aspects of your design that have the potential to damage the FPGA. This includes:

- external power supplies: the design must make provisions to ensure the voltage levels cannot be accidentally changed and that they are switched on and off (sequenced) correctly,
- interfaces at levels other than 3.3V: these can damage the FPGA (whether on the inputs or outputs), and
- interfaces to inductive components (motors, relays, solenoids,...): these can create voltages much higher than the supply voltages when they are switched.

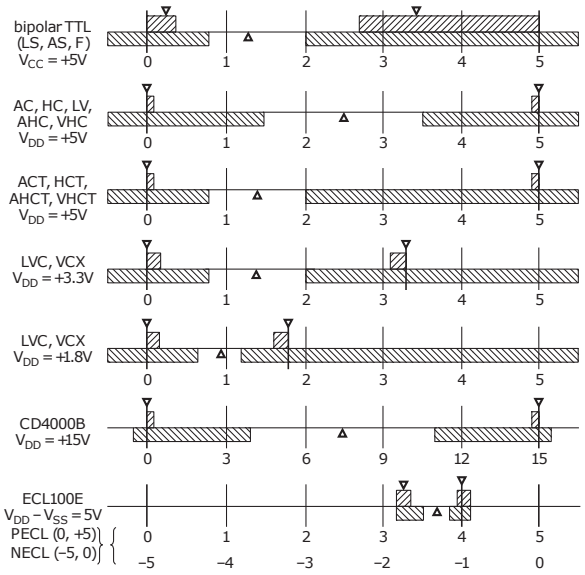
Before the review you should supply schematics, component datasheets and any calculations you've used to verify that the recommended operating conditions of the FPGA will not be exceeded under any reasonably likely conditions (including user errors).

How can I interface the FPGA to other devices?

You need to check datasheets of the FPGA (on the course web site) and the other device to make sure the interface specifications of the two devices are compatible. Datasheets often include simplified schematics of the input and output circuits to describe the behaviour of the interface.

Interface Voltages and Currents

The following diagram from *The Art of Electronics*[2] shows the guaranteed output levels (above the line), required input levels (below the line) and typical outputs and thresholds for different logic families.



For an interface to work correctly:

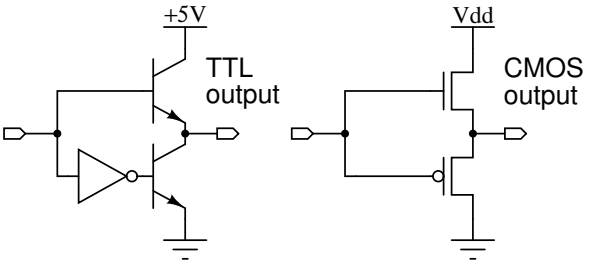
1. the maximum output low level must be less than the minimum input low level, and

2. the minimum output high level must be greater than the maximum input high level, and

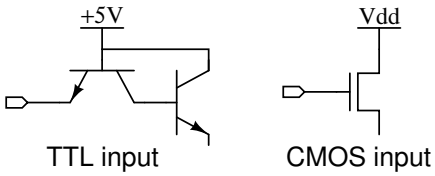
3. the high-level minimum output curent must be greater than the low-level maximum input current.

4. the low-level maximum output (typically negative) current must exceed the maximum low-level input (often negative) curent.

"TTL" interface levels are based on logic circuits using bipolar transistors which CMOS interface levels are based on those using MOSFETS. The following schematics show the two output circuits:



And the input circuits (note that TTL inputs *source* current so the current is negative):



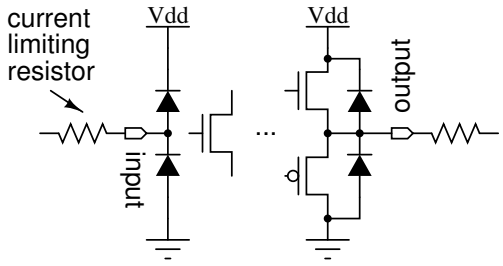
TLL output levels are thus significantly above ground and below the 5 V supply voltage. CMOS levels typically swing from just above ground to just below the supply voltage. However many interface devices combine MOSFETs and BJTs and so it's important to consult the datasheet for the interface specifications.

The FPGA can be configured for different logic families but this requires supplying the I/O supply pins with the appropriate voltages (this is not possible with the DE0-Nano board).

Clamp Diodes

Many (but not all) inputs and outputs use "clamp" diodes that connect the I/O pin to the supply voltage and ground and prevent the pin from going more negative than a diode drop below ground or a diode drop above the supply voltage:

[2] An excellent reference work on many aspects of analog electronics. A copy of Chapter 12, "Logic Interfacing", is available on the course web site.



External clamp diodes can be used if the device does not include them or if they cannot handle the expected voltage or current levels. Schottky diodes can be used if a lower forward voltage drop (about 0.3 V) across the clamp diode is required.

If clamp diodes are used to limit the voltage at the input or output, a current-limiting resistor may need to be used to avoid exceeding the maximum clamp diode current as shown above.

Open-Collector/ Open-Drain Outputs

One way to drive logic that requires higher voltages is to use an open-drain (or open-collector) output. When the output transistor is on turned on the output goes low, when it's off, an external pull-up resistor pulls the output up to the desired high logic voltage. At low speeds outputs can be tri-stated to achieve the same result.

Some buses (e.g. I^2C are bidirectional and outputs pull bus signals low rather than driving them.

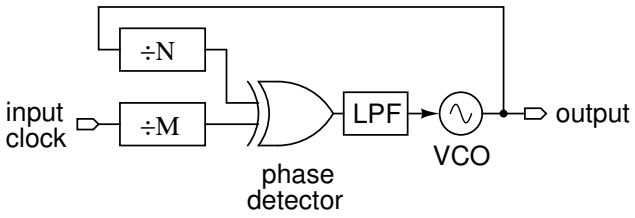
Rise Times

Current-limiting resistors increase the time required to charge pin input and interconnect capacitances (typically from a few pF to tens of pF). This limits the speed of such interfaces. For example, a 10 k resistor and a 100 pF capacitance has a time constant of 1 us and such a circuit will be limited to frequencies of 100 kHz or so.

High-speed interface require the use of interface logic rather than the current-limiting techniques above. For example the HCT family (e.g. 74HCT244) has wide-swing CMOS outputs that can operate from a wide range of voltages and it's input levels that are compatible with many logic families.

How Do I Generate an x kHz/MHz Clock?

The Cyclone IV FPGA has four Phase-Locked Loops (PLLs), whose block diagram is (approximately):



The inputs to the xor phase comparator are voltage-controlled oscillator (VCO) frequency divided by N, f_{VCO}/N, and the reference frequency diviced by M, f_{ref}/M. The VCO feedback loop[3] forces these frequencies to be equal. Thus $f_{VCO} = f_{ref}\frac{N}{M}$.

There are limits to the clock dividers that set the ratios N and M so there are limits to the PLL output maximum and minimum frequency and to the resolution. A divider can be used to obtain lower frequencies but should be used as an enable signal, not as the clock input to flip-flops.

It's also possible to maintain a phase relationship (or equivalently, a delay) between the PLL outputs.

There is a "wizard" accessible from the Quartus IP Catalog menu item that can be used to generate the Verilog source for a module with the appropriate parameters values. This module can then be included in your project and instantiated in your code.

[3] The output of the xor gate is 0 when the input phase difference is zero and 1 when the phase difference is 180 degrees.