XILINX®

XAPP785 (v1.0) June 22, 2005

Level Translation Using Xilinx CoolRunner-II CPLDs

Summary

As electronic design has advanced over the years, more and more I/O standards have been created. Since the days when the 5V CMOS and TTL standards were the prevalent standards with which to design logic circuits, there have been a lot of changes. Today, there are many voltage standards operating at different voltages with different thresholds. This application note explains how to use a Xilinx CoolRunner-II CPLD to translate between different voltage levels.

Introduction

A typical electronic system will no longer operate at only one voltage. The most popular voltages used to interface between components on a board are 3.3V, 2.5V and 1.8V. However, more frequently, devices need to interface to 'unusual' voltages.

Since the introduction of the CoolRunner-IIA parts (XC2C32A and XC2C64A), all Xilinx CoolRunner-II devices have multiple I/O banks. The XC2C32A, XC2C64A, XC2C128, and XC2C256 have two banks each, and the XC2C384 and XC2C512 have four banks each. This means that the V_{CCIO} rail (the power supply for the device I/O) is split, enabling I/O in different banks to be powered at different voltage levels. Each I/O bank can support one V_{CCIO} voltage at a time. The supported I/O standards for the CoolRunner-II device can be seen in Table 1 below.

Table 1: **Supported I/O Standards in CoolRunner-II Family**

IOSTANDARD Attribute	Output V_{CCIO}	Input V_{CCIO}	Input[1] V_{REF}	Board TerminationVoltage V_{TT}
LVTTL	3.3	3.3	N/A	N/A
LVCMOS33	3.3	3.3	N/A	N/A
LVCMOS25	2.5	2.5	N/A	N/A
LVCMOS18	1.8	1.8	N/A	N/A
LVCMOS15[2]	1.5	1.5	N/A	N/A
HSTL_1[3]	1.5	1.5	0.75	0.75
SSTL2_1[3]	2.5	2.5	1.25	1.25
SSTL3_1[3]	3.3	3.3	1.5	1.5

1. For information on assigning Vref pins, see XAPP399
2. LVCMOS15 requires use of Schmitt-trigger inputs.
3. HSTL_1, SSTL2_1 and SSTL3_1 are supported on XC2C128 and larger

The I/O characteristics of each standard can be found in the device specific CoolRunner-II Datasheets, for example XC2C128. More information about each of the I/O standards can be found in XAPP382. For information about interfacing to 5V, see XAPP429.

Configuring I/O to Use I/O Standards

The designer specifies which I/O standard to use at the time of design entry. I/Os can be configured to operate at different I/O standards in a number of ways.

Default

The default I/O Standard can be set in the process properties for the Fit process in the ISE software. The use of the default I/O Standard will set all I/O used in the design to the I/O standard specified. This is useful if all the I/O are powered at the same voltage. However, this application note is discussing interfacing between different voltages, so another I/O standard assignment method is required.

PACE

The Xilinx Pinout Area and Constraints Editor (PACE) tool can be used to assign a variety of constraints, including pin locations, slew rate, Schmitt trigger and I/O standard. The package diagram distinguishes between I/O banks by using different colors.

The Design Object List then enables you to assign I/O standards to the I/O pins in the design. If the banking rules are broken, then PACE will issue a message that the I/O standard is not V_{CCIO} compatible.

UCF

The final method of assigning I/O standards in the software is to enter them directly into the User Constraint File (UCF). The syntax for entering I/O standard constraints into the UCF is as follows:

```
NET "user_net" IOSTANDARD = xx ;
```

Where ***xx*** is one of the permissible values: LVCMOS33, LVCMOS25, LVCMOS18, LVCMOS15, LVTTL, and for devices of 128 macrocells and above, HSTL_1, SSTL2_1 and SSTL3_1.

Xilinx recommends the use of the Default assignment in co-operation with PACE to get the optimum results, since PACE has the design rules check that the UCF entry method lacks.

Using the CPLD as a Level Shifter

In addition to knowing of how to configure the I/O of the CPLD at different voltages, it is also necessary to have an understanding of how to use the device as a level shifter. It is possible to have this device act as a route-through to simply shift, for example, 1.8V inputs to 3.3V outputs. This would simply require a logical assignment of the input bus to the output bus, and would use one pin per input/output, and one product term and one macrocell per output.

Even in the smallest device in the CoolRunner-II family, the XC2C32A, performing this translation on an 8-bit bus would utilize less than one quarter of the resources. As the input signals must go through the central interconnect switch, they can be assigned to any output pin required by user. This makes the CPLD perfect for performing additional functions such as bit-swapping of the input bus, or changing from little-endian to big-endian format. CoolRunner-II CPLDs have a very fast T_{PD} (propagation delay), as fast as 3.8 ns in the XC2C32A -4, so very little delay will be incurred by using a CPLD in this situation. It is also important to note that, due to the uniform nature of the architecture, all signals going through a similar path in the device will incur a similar delay and will, therefore, have minimal skew from each other.

While this is a perfectly legitimate use for a CPLD, there are a lot of other resources available that can be used to perform operations on the incoming data.

Incorporating the Level Shifter into a Common Interface

It is common for devices on a board to operate at different I/O voltages. The information above explained how to use the CPLD as a simple level shifter, but all too often, the data coming into the device will need some operation performed upon it. There are many communication interfaces for use in electronic systems (USB, SPI, I2C), several of which can fit into a Xilinx CPLD. Figure 1 shows a generic example that builds on the previous example. Here the 1.8V input bus goes into the device, has a few operations performed on it, such as shifting left or right, checksum calculation, parallelization, serialization, interrupt handling, and so on, and emerges from the device at 3.3V.

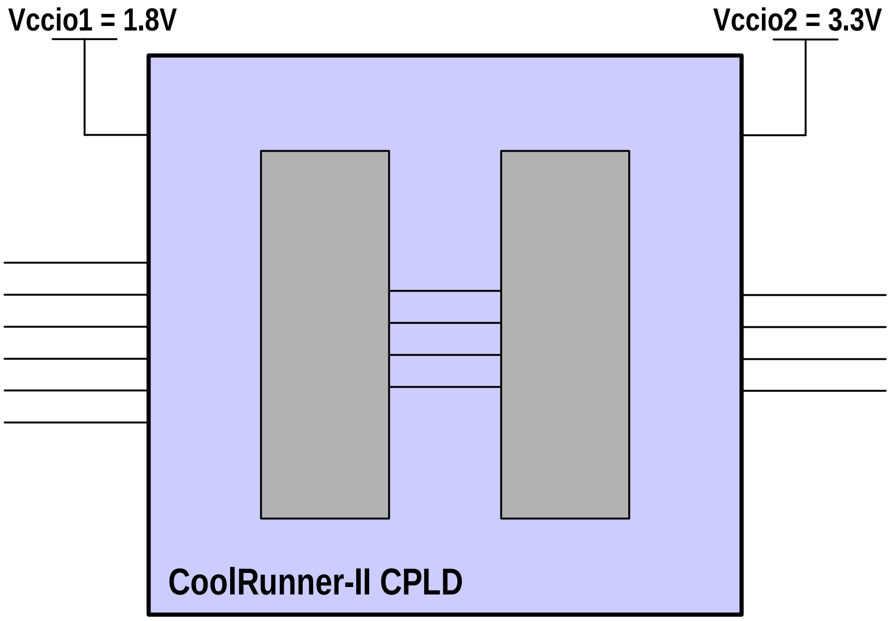

Figure 1: **Using CoolRunner-II as a Level Shifter Interface**

A few examples of common interfaces can be found in other application notes on the Xilinx website.

- XAPP341 UARTs in Xilinx CPLDs
- XAPP354 Using Xilinx CPLDs tp Interface to a NAND Flash Memory Device
- XAPP384 Interfacing to DDR SDRAM with Xilinx CoolRunner-II CPLDs

CPLDs Can be Used to Supply Current

An added advantage of using a CPLD in this situation is if the device providing the 1.8V signals to the CPLD cannot supply sufficient current. The CPLD can be used to supply the current required. To ascertain how much current the I/O of the CPLD can provide under typical conditions, the I-V plots for the I/O are needed. These can be seen in the device specific CoolRunner-II datasheets. Figure 2 is taken from the XC2C32A datasheet.

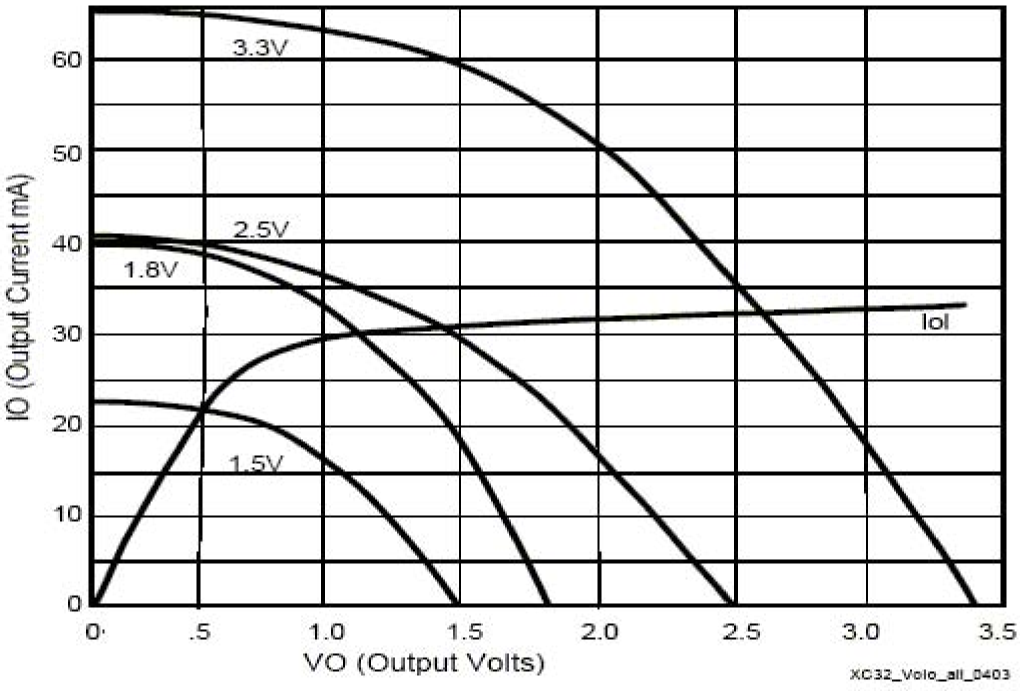

Figure 2: **IV Curves for the CoolRunner-II I/O Standards**

Powering V_{CCIO} Between Specifications

The CoolRunner-II architecture supports 1.5V, 1.8V, 2.5V and 3.3V I/O. However, 2.85V, 2.8V and 2.7V are also commonly used I/O voltages in certain applications. Can CoolRunner-II support 2.85V?

The input buffer and output buffer are the same when using a 2.5V I/O standard (LVCMOS25) as when using a 3.3V I/O standard (LVCMOS33 or LVTTL). The permissible V_{CCIO} range for LVCMOS25 is 2.3V to 2.7V, and the permissible V_{CCIO} range for LVCMOS33 is 3.0V to 3.6V. So, there is a clear gap from 2.7V up to 3.0V that is "out of spec." Due to the linear scaling nature of the input and output buffers, it is safe to assume that the V_{IH} min and V_{OH} min specifications for a pseudo-2.85V I/O standard can be interpolated from these specifications. V_{IH} will be in the region of 1.85V and V_{OH} will be V_{CCIO} minus 0.4V at (maximum) 8 mA loading.

Conclusion

Xilinx CoolRunner-II CPLDs are perfectly suited to perform level translation operations on signals. They offer split V_{CCIO} rails on all devices providing at least two I/O banks. Signals can enter the device at one voltage and be output from the device at another voltage without introducing any skew on the signals. This application, which previously required a dedicated IC, can now be incorporated into the CoolRunner-II, which can also be performing other system functions.

Additional Information

CoolRunner-II Data Sheets, Application Notes, and White Papers

Access to all Xilinx Data Sheets, Application Notes, and White Papers

Device Packages

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
06/22/05	1.0	Initial Xilinx release.