XAPP776 (v1.0) April 4, 2005

AC Coupling Bypass for High-Speed Digitizing on Virtex-II Pro X FPGAs

Author: Tim Hagen

Summary

This application note describes a method for bypassing the AC coupling in Virtex™-II Pro X devices. Doing so allows use of the 10 Gb/s RocketIO™ Multi-Gigabit Transceiver (MGT) in DC-coupled over-sampling applications. The Verilog source code for a lab-tested reference design is available on the Xilinx web site.

Introduction

The RocketIO MGT requires some form of DC coupling when used as a high-speed oversampling or digitizing engine in the presence of unfavorable signaling conditions such as bad DC balance in the input data stream, long run lengths, or other completely unknown characteristics. Because the Virtex-II Pro X FPGA has built-in AC coupling, some method of simulating DC coupling must be used.

Externally mixing in a known low-frequency signal, and then digitally canceling out that mix signal inside the FPGA fabric, simulates DC coupling. Figure 1 shows the basic function of the circuit.

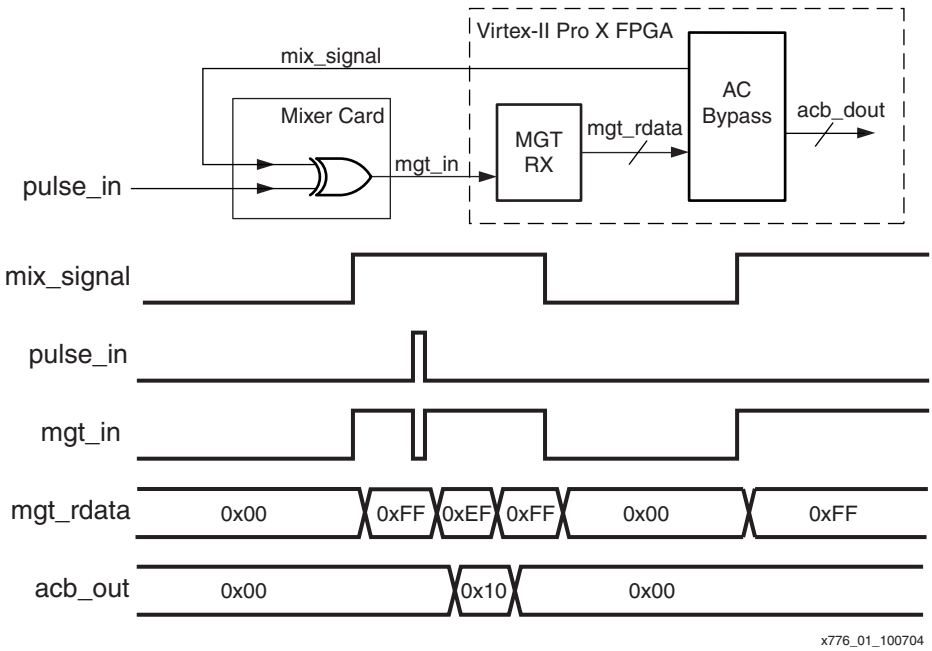

Figure 1: **Basic Circuit Function**

Mix_signal is a low-frequency mix signal generated by the bypass circuit. It is combined with the input signal *pulse_in* using an external XOR gate. The resulting signal has enough DC balance, as well as high enough a frequency content, to be properly received by the 10 Gb/s RocketIO MGT, even in AC-coupled mode.

This produces the *mgt_rdata* signal where the combined mix signal and input signal can be seen. *Mix_signal* is then digitally removed by the AC bypass circuit, and the resulting output *acb_dout* contains a digitized version of just the input signal.

Sampling Error

There can be some sampling errors around the *mix_signal* transitions, especially at high sampling rates (> 8 Gb/s). This results in single-bit error pulses that correspond to the rising and falling edge locations in *mix_signal.* See Figure 2.

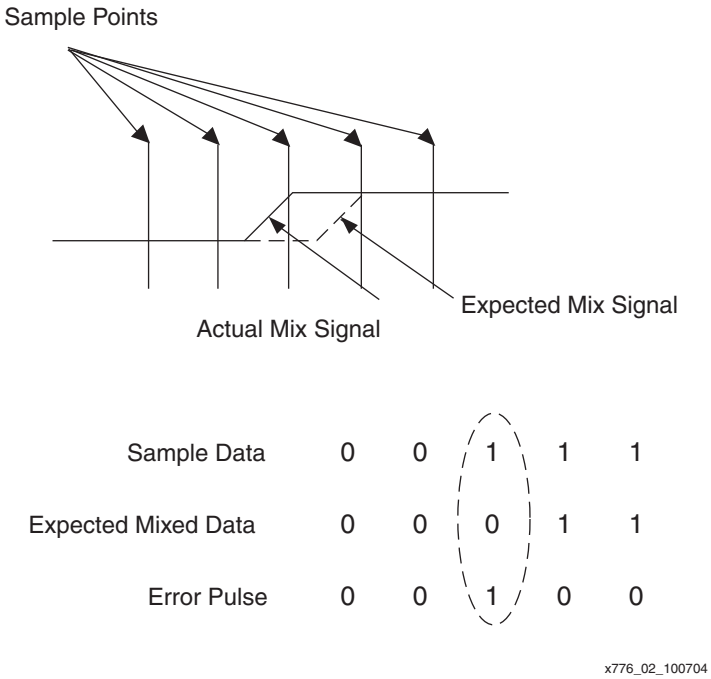

Figure 2: **Sample Errors**

If the presence of error pulses resulting from sampling errors affects the operation of the downstream logic, the error pulses can be removed by the bypass circuitry. This results, however, in a small window of opportunity during which single bit-wide input pulses could be missed, since they are indistinguishable from sampling error pulses. The statistical possibility of this occurring is given by the formula:

$$Probability = \frac{4 \times MixFrequency}{SampleFrequency}$$

Mix_signal Selection

For many applications with known input characteristics, as well as for the operation of the bypass circuit itself, the frequency of *mix_signal* is not important, and can be chosen rather easily. Picking a *mix_signal* frequency with a period longer than the maximum active run length of the input should provide good results.

If the input signal is completely unknown, or could have a period close to the chosen *mix_signal* frequency, an end case exists that could cause problems. If the input signal to the 10 Gb/s RocketIO MGT closely matches *mix_signal* for any significant length of time, the input signal will have unacceptable DC balance, or could disappear altogether in the case of a perfect match. In this case, a *mix_signal* with a random component should be used.

Reference Design

Design Overview

The reference design implements the AC coupling bypass circuit described in the preceding sections. It includes the following features:

- Programmable *mix_signal* frequency for standard operations
- Pseudo-Random Bit Sequence (PRBS)-based random mix frequency modes
- Clean function to eliminate sampling errors

The reference design files can be downloaded from the Xilinx web site at:

http://www.xilinx.com/bvdocs/appnotes/xapp776.zip

Port List

The reference design inputs and outputs are shown in Table 1.

Table 1: **Port List for the AC Bypass Module**

Port Name	I/O	Description
clean_enable	Input	Enables clean function. This compensates for sampling errors around *mix_signal* edges.
clk	Input	Clock.
din [39:0]	Input	Input parallel data from the 10 Gb/s RocketIO MGT.
dout [39:0]	Output	Output data.
enable	Input	Circuit enabled.
error_enable	Input	If asserted, a synchronization error will cause the circuit to drop out of operation and attempt to re-acquire synchronization. If not asserted, the user must cycle *enable_signal* to get the circuit to re-acquire synchronization.
mix_delay [4:0]	Input	Sets the period of *mix_signal*. The period of the signal is determined by $2M(N + 3)$ where N = binary weight of *mix_signal[4:0]* (range 0-31) M = the period of the parallel clock If set to 0, a random PRBS-based *mix_signal* is used.
mix_out	Output	The mixed-signal output.
rst_n	Input	Low TRUE reset.
rxslide_out	Output	Connects to the RXSLIDE port of the 10 Gb/s RocketIO MGT, and is used to synchronize *mix_signal* edges to the AC bypass circuit.
sync	Output	Asserted High if the circuit has successfully synchronized with *mix_signal*.
sync_error	Output	Asserted High if the circuit has lost synchronization with *mix_signal* for more than 30 *mix_signal* periods.

Design Startup

When the circuit is enabled, it goes through a synchronization cycle and tries to align *mix_signal* with its internal comparators. The input signal should be continuously Low while this is going on. Once the synchronized output is asserted, the circuit is ready to use. It continuously monitors and tracks the *mix_signal* edges through ±4 UI of drift without losing synchronization.

If *error_enable* (Table 1) is enabled and synchronization is lost, the *sync_error* signal is asserted, the circuit drops out of operation, and re-acquires synchronization in the new position. Note that for some applications, synchronization errors can be expected. If the input provides a constant input stream that spans more than 32 *mix_signal* periods, the *sync_error* signal is asserted even though the circuit is operating normally.

Clean Function

The "cleaner" circuit tries to eliminate the sampling error around the *mix_signal* edges in an effort to remove false pulses from the output stream. If enabled, the circuit cleans only single bits that are aligned exactly on the *mix_signal* edge transitions. If the clean circuit does not detect a `010` or `101` in exactly the right location relative to the *mix_signal* transition, no action is taken. Figure 3 shows the operation around one edge of *mix_signal*.

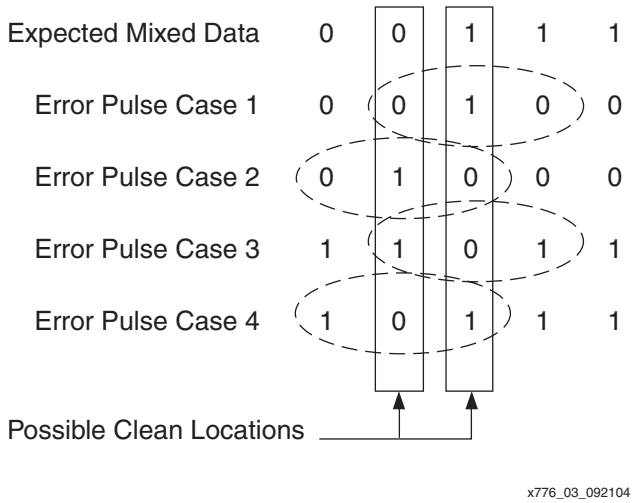

Figure 3: **Clean Function Operation**

The possibility of missing some one-bit-wide pulses exists. For example, in Error Pulse Case 2 in Figure 3 above, a one-bit-wide pulse that lined up exactly with the rising edge of *mix_signal* might have extended *mix_signal*. Again, it has no effect on wider pulses; it only removes solitary bits.

The likelihood of this occurring is a function of the *mix_signal* period, since there are four possible cleaned locations for each cycle of *mix_signal*. If using a 40-bit interface and a *mix_signal* period of 20 word clocks, there is a 4 in 800, or 0.5%, chance of missing a single bit pulse.

Lab Testing

The AC coupling bypass circuit has been run in the lab between 5 Gb/s and 10 Gb/s. At 5 Gb/s, the cleaner was not needed as the sampling error was essentially non-existent. At around 7 Gb/s, occasional (every 1 to 10 seconds) sampling error pulses showed up. At 10 Gb/s, sampling error pulses were fairly constant, and the cleaner needed to be enabled. The circuit was run overnight at 10 Gb/s with the cleaner enabled, and no false pulses were recorded.

To test the circuit, a 10 Gb/s RocketIO MGT-based pulse generator/check circuit was developed, along with an external mixer card. The pulse generator circuit put out widely spaced pulses of varying widths, down to a single UI in width, and the check circuit counted the incoming pulses in the digitized data stream. This represents a worst-case input data stream from a run-length and DC-balance perspective, and the data stream was captured without errors.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
04/04/05	1.0	Initial Xilinx release.