Compute a histogram in an FPGA with one clock

Mohit Kumar, Texas Instruments, Bangalore, India; Edited by Martin Rowe and Fran Granville - February 03, 2011

Histograms are often useful tools for analyzing digital data. To get reliable results from a histogram, though, you must collect large amounts of data, often with 100,000 to 1 million points. If you need to collect an ADC’s digital outputs for analysis, you can use an FPGA (Figure 1).

The figure shows the histogram, RAM, and pulse-generator blocks, which let you capture and display the histogram computation based on 14-bit data. The RAM block is the FPGA’s built-in RAM, and the histogram block is the VHDL (very-high-level-design-language) code to compute the histogram. You can also download the VHDL code for this application here.

The 14-bit parallel data, Device_Data[13..0], from an ADC goes to the histogram block and to the RAM Rd.Addr input. The RAM provides the data at its address location, RAMDataOut[15..0]. This data loops back to the histogram block, which increments it by one and sends it to output pin DataOut[15..0], a 16-bit data output. When the WREN (write-enable) pin is at logic level one, the data is written at the address at pin Wr.Addr[13..0]. That approach is the same as if the data were coming from Device_Data[13..0].
The RAM has a fixed delay from its input to its output. That is, when the input is at Rd_Addr, data becomes available at its output pin, RAMDataOut, after a fixed delay. This delay might vary across the FPGA. To take care of this delay, allow a two-clock delay to Device_Data before calculating the histogram. The delay in RAM should be less than two clock periods; otherwise, there could be data loss. This constraint limits the maximum frequency of Device_Clk.

The Cntr_Value gives the number of input data for which the histogram block computes the histogram. To reset the counter, the Pulse_Gen block generates a pulse, which enters at input Rst_Cntr. At this point, the histogram again computes the histogram for the next set of input data from Cntr_Value. The Cntr_Value is 15 bits, but you can increase it to collect more histogram data.

The signals Sel_Data and Rst_RAM reset the data stored in the FPGA’s RAM. Whenever the high signal is at the Rst_RAM pin, the DataOut pin of the histogram block gives all bits as 0. When the high signal is at the Sel_Data input pin of the histogram block, the output from RAM_Wr_Addr is not the Device_Data but an internally generated ramp that ramps up from 0 to 16,384. The histogram block does no computation because doing so would reset the address of the RAM.

When the FPGA completes the histogram computation, the RAM can read the histogram data by selecting Sel_Data as logic high and keeping Rst_RAM as logic low. The data in the RAM address sequentially exits the output pin, and you can transfer the data to a PC. Because all the blocks run on a single clock, Device_Clk, the design is simple and helps you meet timing constraints. You can easily modify the design to accommodate 16- or 12-bit data histograms.