Laser Range Finder Using Actel's Axcelerator FPGA

Introduction

In modern warfare, laser-guided weapons play a significant role in ensuring each warhead deployed will only strike its intended target. Each laser-guided missile or bomb has a laser seeker that consists of an array of photodiodes, and these photodiodes are sensitive to a predefined laser's optical wavelength. A high-intensity laser designator must acquire and lock onto the target—either from the air or from the ground. This is necessary to allow the missile or bomb to identify the target. Once the laser-guided weapon is launched, the laser seeker senses the laser beam reflected from the target, and the seeker's control system will then guide the missile straight to the target (Figure 1).

In general, the laser pulse width presented to the control system is very short. The control system must be fast enough to reliably capture this laser pulse pattern to calculate the range to the target. This application note describes the details of implementing such a laser range finder design using Actel's Axcelerator family. However, this reference design could be applied to any application where precise timing and the ability to capture a narrow pulse width are required.

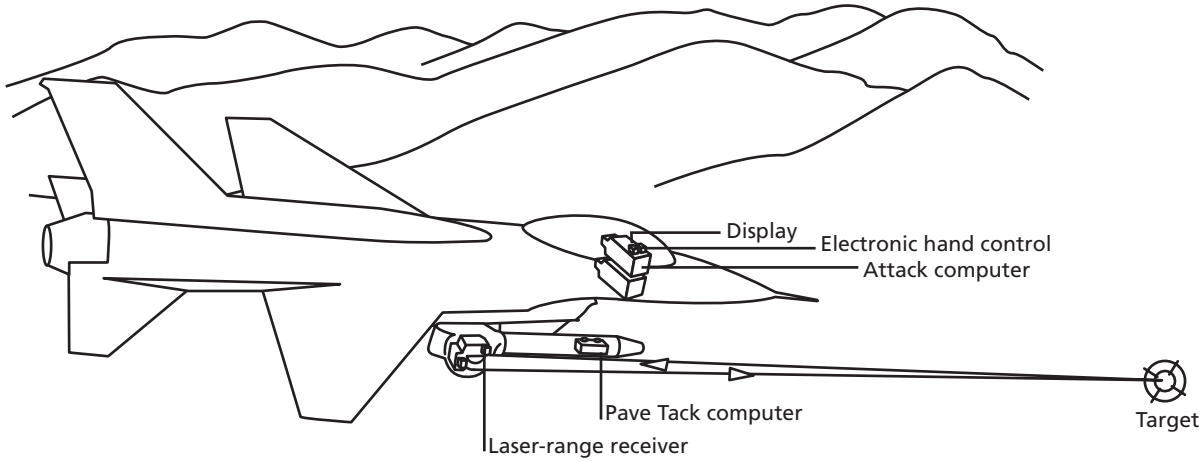

Figure 1 • **Laser-Guided Weapon System Overview**

Design Description

This particular design application requires capturing a pulse of 800 ps in width. Capturing an 800 ps pulse reliably using a single clock would require a 4 GHz clock and a flip-flop toggle rate above 2 GHz. Instead, multiple clocks can be created and phase-shifted in order to capture this narrow pulse width. The system must be able to sample every 250 ps so that the 800 ps pulse can be captured reliably in at least three consecutive clock cycles.

To sample the 800 ps pulse every 250 ps, four clocks (CLKA, CLKB, CLKC, CLKD) running at 500 MHz are created and phase shifted by 250 ps using the Axcelerator PLLs and the programmable delay lines. As shown in Figure 2 on page 2, the original 100 MHz input reference clock is multiplied by the PLLs to 500 MHz and each successive clock is shifted in phase by an additional 250 ps from the base clock, CLKA.

The Axcelerator family provides register cells (R-cell) with a 700 MHz toggle rate and combinatorial cells (C-cell) to implement various logic functions. The R-cell contains a flip-flop featuring asynchronous clear, asynchronous preset, and active-low enable control logic. The clock source for each R-cell can be the hardwired clock, routed clock, or internal logic. Moreover, the R-cell registers feature selectable clock

polarity on a register-by-register basis with no timing penalty. With four hardwired clocks, four routed clocks, and eight PLLs with a maximum output frequency of 1 GHz, Axcelerator can be used to meet the most challenging design specifications. In this particular design, four PLLs, programmable delay lines, and the selectable clock polarity for the R-cell features were used. Using the selectable clock polarity feature of the R-cell, clock edges 5 to 8 in Figure 3 on page 3 can also be used to sample the short data pulse. As a result, eight samples can be taken in 2 ns. Figure 3 on page 3 shows the phase relationship of the four PLL output clocks.

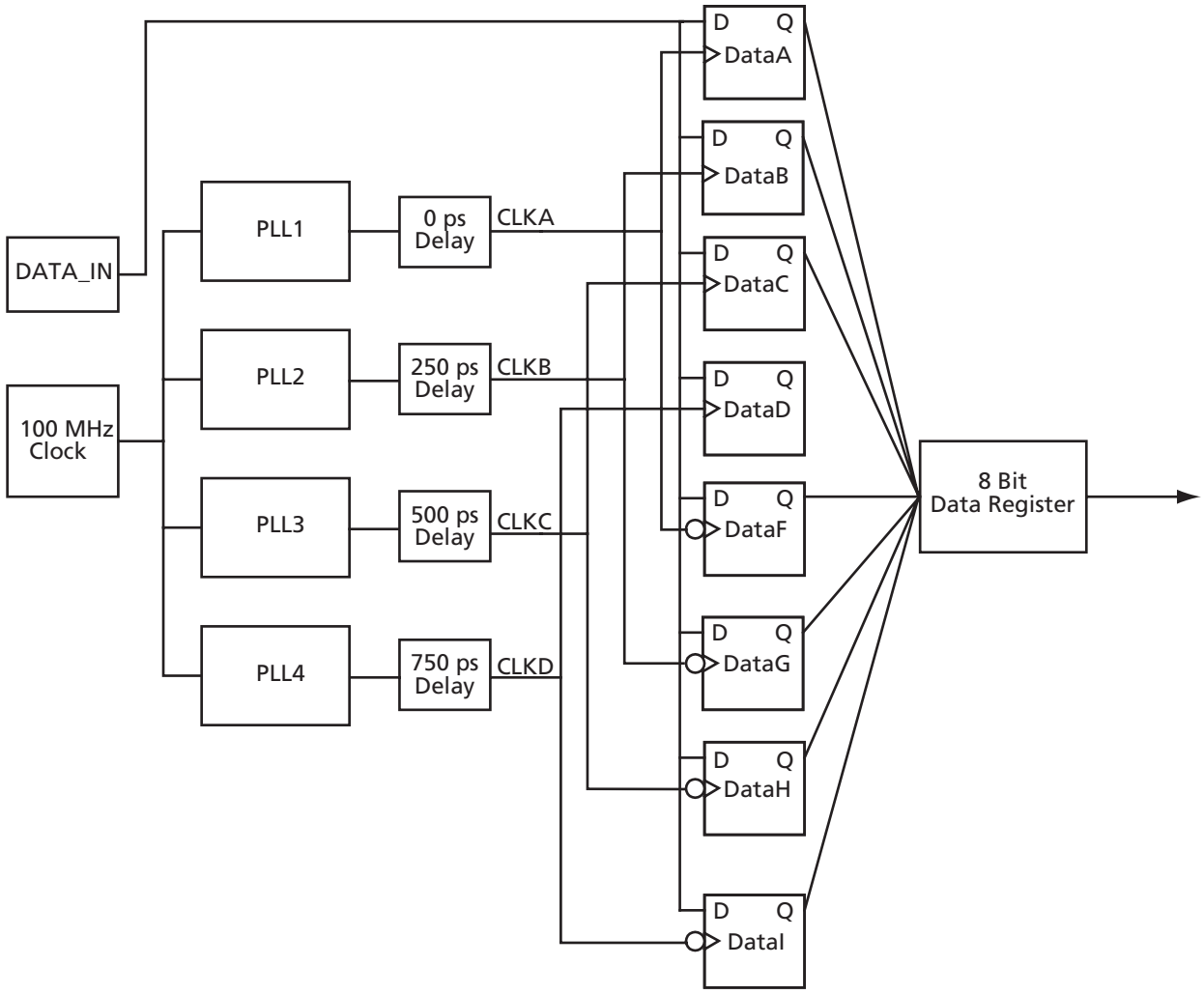

Figure 2 • **Simplified Block Diagram**

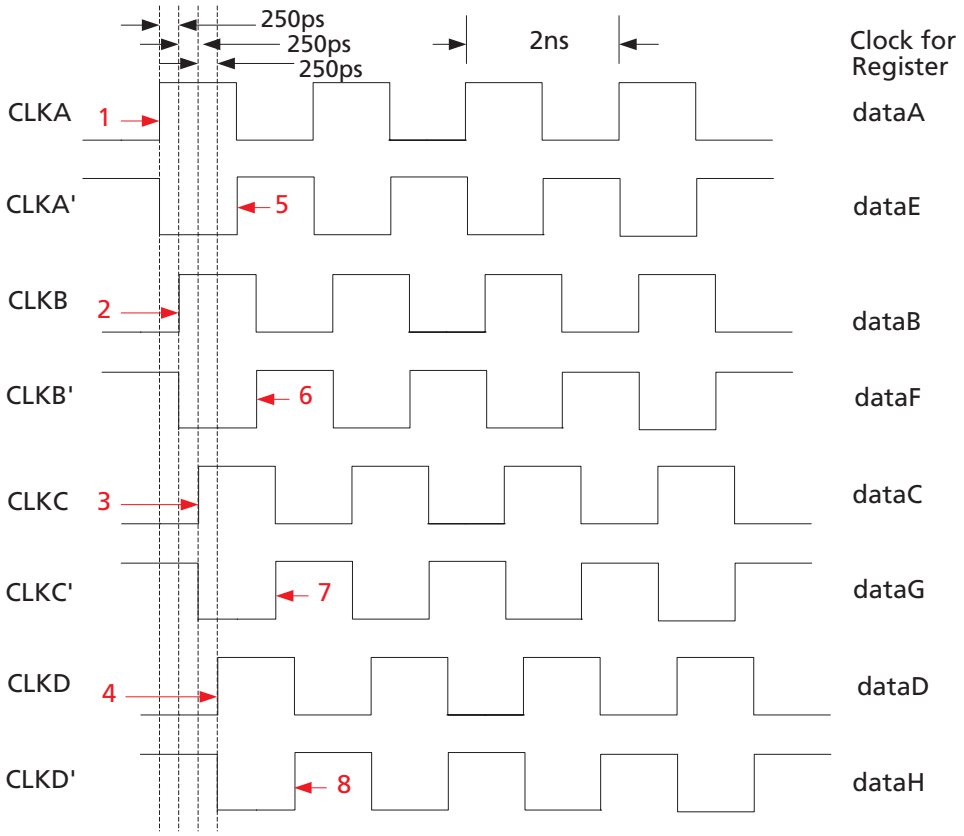

Figure 3 • **PLL Output Clock Phase Relationship (250 ps Sampling Interval)**

Design Implementation

This reference design was developed using the Actel Libero™ Integrated Design Environment (IDE), which provides everything required from schematic capture or HDL entry to place-and-route tools. Table 1 summarizes the Libero IDE features used in this design. For more information, visit http://www.actel.com/download/software/libero/default.aspx.

When the design is completed and verified through simulation and static timing analysis, a device can be programmed using the Actel Silicon Sculptor II programmer. For more information regarding programming Actel's devices using Silicon Sculptor II, visit http://www.actel.com/products/hardware/program_debug/ss/default.aspx.

Note that Actel provides design files for this reference design. Please refer to the "Appendix" on page 10 for detailed information on obtaining these files.

Table 1 • **Key Libero IDE Features**

Function	Tool	Company
Project Manager, HDL editor	Libero™ IDE	Actel
Synthesis	Synplify® AE	Synplicity®
Simulation	Model*Sim*™ AE	Mentor Graphics®
Testbench Creation	WaveFormer Lite ™ AE	SynaptiCAD™
Timing/Constraints, Macro Generation, ChipPlanner, Place-and-Route, Programming	Designer	Actel
Programming Software	Silicon Sculptor II	Actel

Design Creation

Create a top-level VHDL design called Top_level.vhd, and use the ACTgen macro builder to generate the PLL macros. Set the **Clock Out** to be **Routed** (Figure 4). By setting **Routed** as **Clock Out** in ACTgen, PLL blocks on the south side of the chip will be used, and each output is capable of driving the CLK (routed clock) network. Four PLLs (and hence four generated CLKs) are used for this part of the design, leaving four PLLs and four HCLKs for the remaining design implementation. On the other hand, if the clock outputs were chosen to be hardwired, the PLL block would be located on the north side of the chip. Four PLLs with different delay settings are used in generating the clock system that can sample a narrow pulse width every 250 ps (Table 2). Refer to the *Axcelerator Family PLL and Clock Management* application note and *ACTgen User's Guide* for more information.

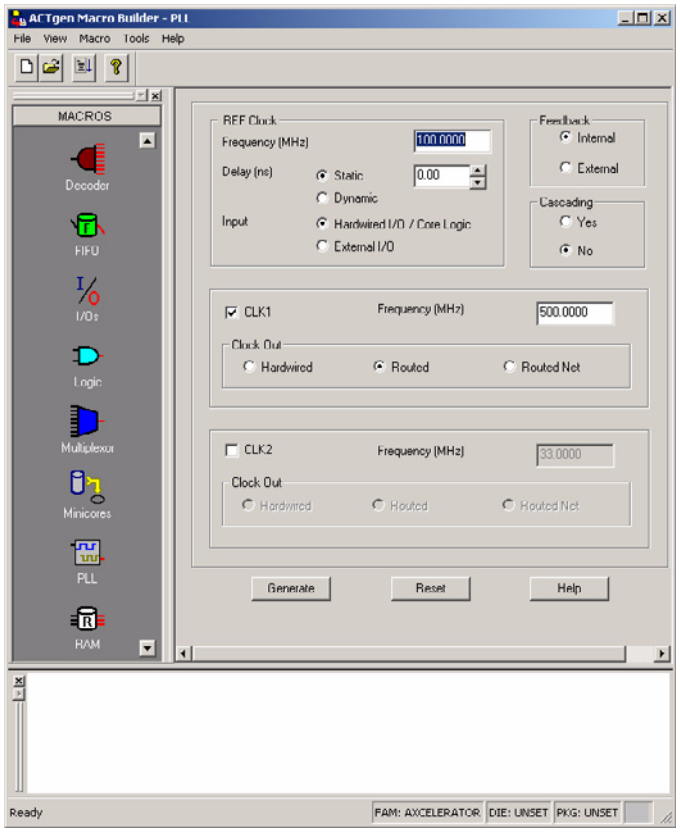

Figure 4 • **PLL Macro Generation Using ACTgen**

Table 2 • **Clock System PLL Settings**

File	Input Frequency (MHz)	Output Frequency (MHz)	Static Delay (ps)
pll_a.gen	100	500	0
pll_b.gen	100	500	250
pll_c.gen	100	500	500
pll_d.gen	100	500	750

Design Verification—Functional Simulation

After describing the design, you must verify functionality. After creating a testbench, Top_Level_tbench.vhd, using SynaptiCAD WaveFormer Lite Actel Edition (AE), use the Mentor Graphics Model*Sim* AE simulator to perform functional simulation on your HDL design.

Synthesis

Since this is an HDL-based design, it must be synthesized to generate an EDIF netlist. Use the Synplicity Synplify AE to generate your EDIF netlist and you may want to reverify your design by performing a post-synthesis simulation using the Model*Sim* AE simulator.

Design Implementation

This reference design was implemented in an AX125-3 FGA256 device. Four PLLs were used and only 24 out of 672 of the R-cells (3.57%) were needed, leaving most of the remaining logic resources free. The Actel Designer software automatically places and routes the design and returns timing information. Since this application requires capturing a very narrow pulse width, precise timing is important. With the unique routing structures offered by the Axcelerator family, skew for each register pair using the same PLL clock source can be virtually eliminated by manually placing the registers. The Designer software Timer tool can then be used to verify that the register pair using the same PLL clock source has a uniform timing delay.

Figure 5 illustrates the routing structures in Axcelerator devices. FastConnects provide high-performance, horizontal routing inside the SuperCluster, and vertical routing to the SuperCluster immediately below. This particular design takes advantage of the FastConnect used for vertical routing. The register pairs, using the same PLL output clock, are manually placed vertically and immediately with each other. For example, register dataA is placed directly below register dataE (Figure 2 on page 2). Use ChipPlanner to manually place the register (Figure 6 on page 6). When the registers are manually placed and committed in the software, the place-and-route tool will not change the placement of those registers.

After the design, compile, and layout stages, the Designer Timer tool can be used to ensure the PLL output clock arrives at both halves of its register pair at the same time. As shown in Figure 7, each register pair has the same timing delay from its corresponding PLL reference clock.

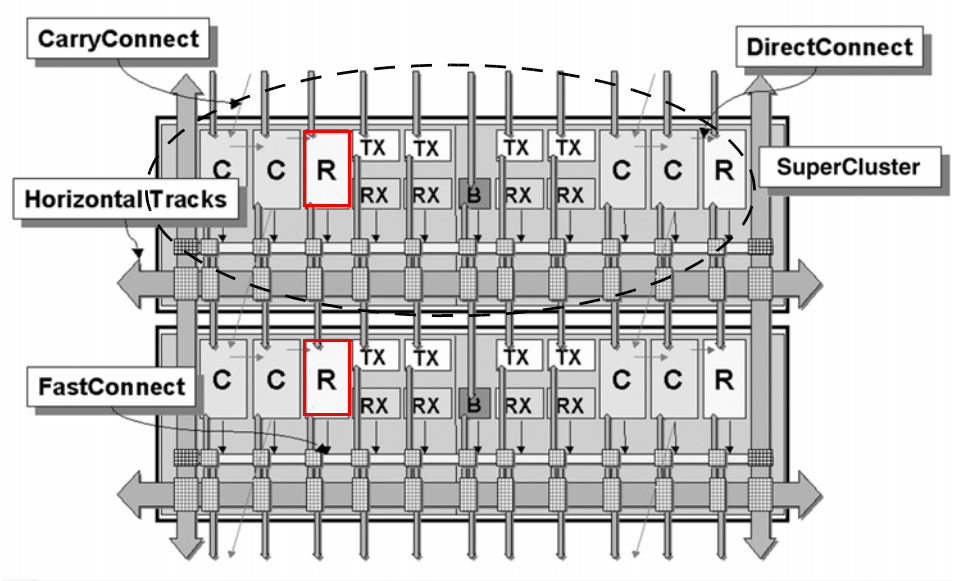

Figure 5 • **Axcelerator Routing Structure**

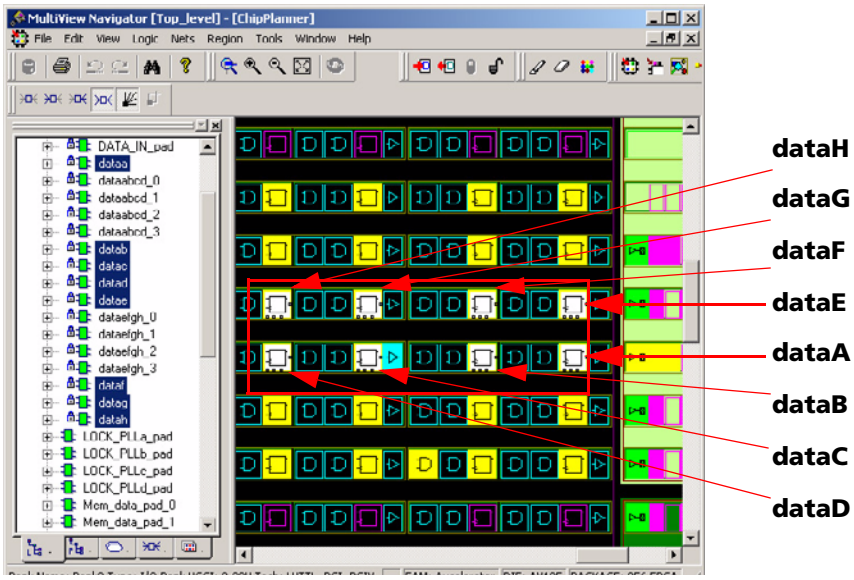

Figure 6 • **Designer ChipPlanner – Manual Placement of Data Registers**

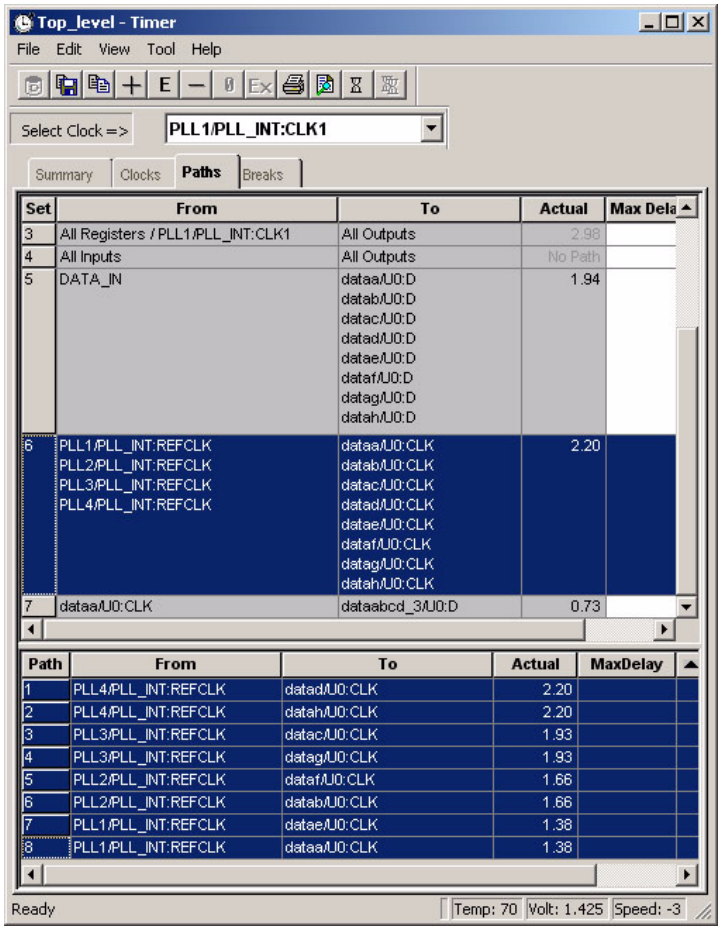

Figure 7 • **Timing Delay for Each Register Pair**

Post-Layout Simulation

After design implementation is completed, verify that your design meets timing specifications by performing the post-layout simulation. Figure 8 illustrates best-case conditions, and Figure 9 on page 8 shows the same simulation under worst-case conditions. In Figure 8 (best-case), five samples were captured successfully at registers dataC, dataD, dataE, dataF, and dataG. In Figure 9 on page 8 (worst-case), six samples were captured successfully at registers dataB, dataC, dataD, dataE, dataF, and dataG. Changing from best-case to worst-case conditions involves changing the setting in Libero IDE under the Options > Project Settings > Simulation. For the best-case condition, the vsim command should be set to the minimum and maximum for the worst-case condition (Figure 10 on page 8).

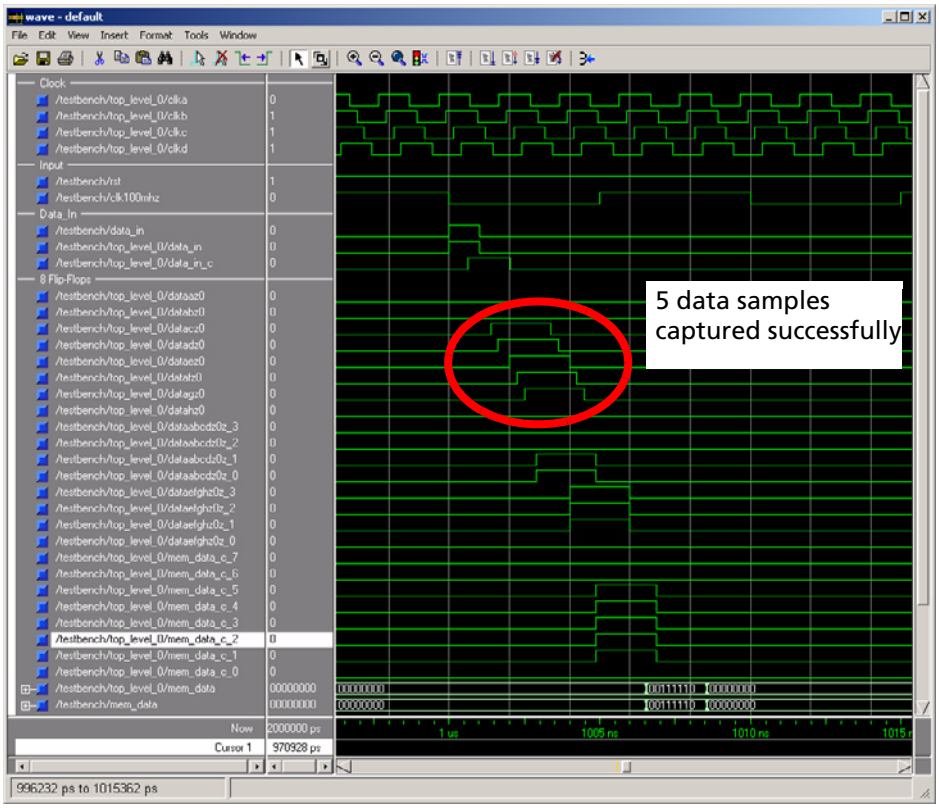

Figure 8 • **Simulation Under Best-Case Condition**

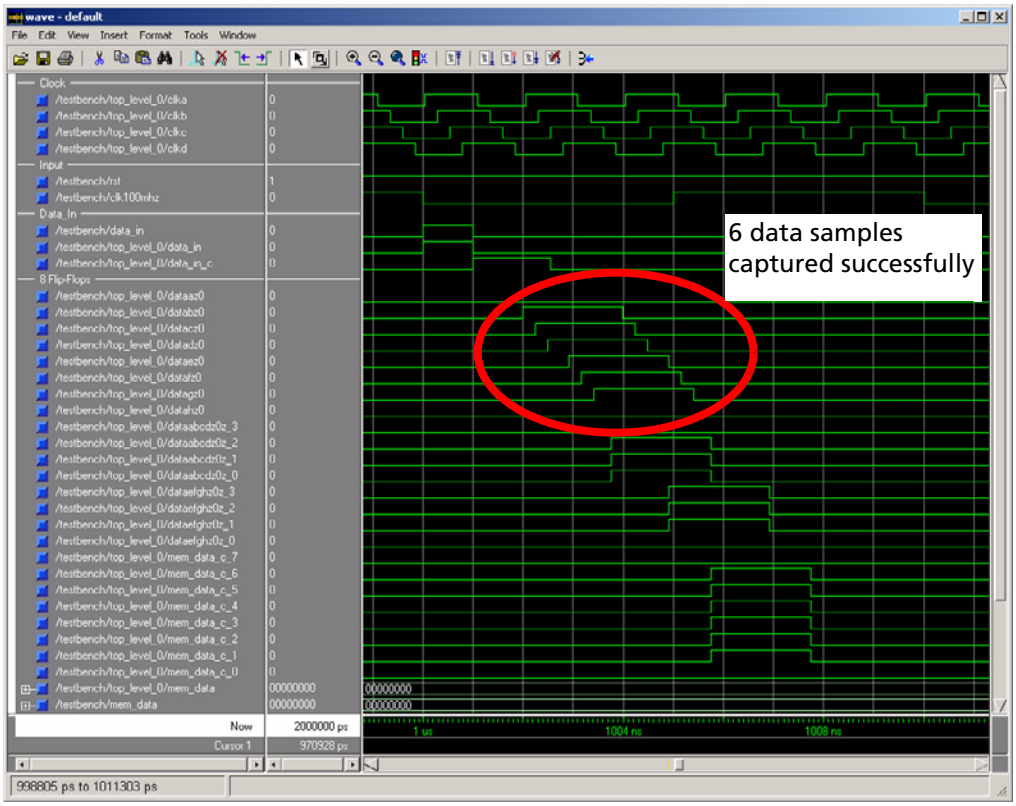

Figure 9 • **Simulation Under Worst-Case Condition**

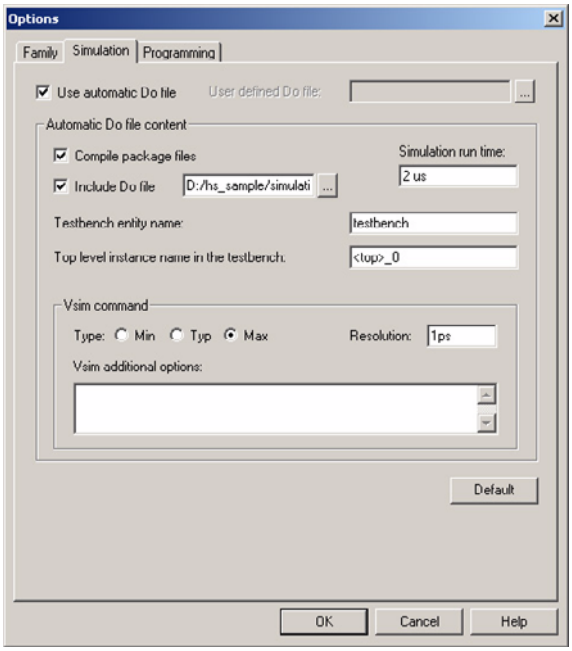

Figure 10 • **Changing the Simulation Condition**

Device Programming

Finally, create the .AFM programming file. Use the Silicon Sculptor II to perform devices programming.

Conclusion

The Axcelerator family's unique architecture offers flexibility in designs where high performance and precise timing are important. In this reference design, PLLs in Axcelerator devices offer programmable delay and programmable output-to-input frequency ratios to enable a flexible clock system. Such a system can reliably capture a very narrow laser pulse width needed in military applications or other applications that require a high sampling rate.

Related Documents

Application Notes

Axcelerator Family PLL and Clock Management
http://www.actel.com/documents/AX_PLL_AN.pdf

User's Guides

ACTgen Cores Reference Guide
http://www.actel.com/documents/genguide.pdf
ACTgen User's Guide
http://www.actel.com/documents/genguide.pdf
MultiView Navigator User's Guide
http://www.actel.com/documents/mvn_ug.pdf
Designer User's Guide
http://www.actel.com/documents/designerUG.pdf
Libero IDE User's Guide
http://www.actel.com/documents/liberoUG.pdf
Timer User's Guide
http://www.actel.com/documents/timer.pdf

Appendix

Using this reference design requires that Actel Libero IDE be installed. Extract the hs_sample.zip file to your hard drive and open the project file hs_sample.prj from Libero IDE. Figure 11 shows the design hierarchy, and Figure 12 shows the design files for this reference design. Please refer to Table 3 for a description of the key design files.

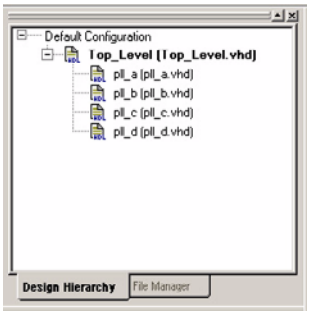

Figure 11 • **Design Hierarchy**

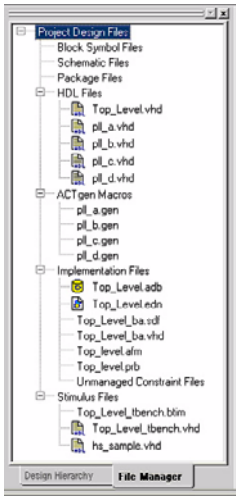

Figure 12 • **File Manager**

Table 3 • **Description of Design Files**

File name	Description
Top_Level.vhd	Top level of this reference design
Top_Level_ba.vhd	Structural netlist for post-layout simulation
Top_Level_ba.sdf	Standard delay file for post-layout simulation
Top_Level.edn	EDN netlist generate from Synplify
Top_Level_tbench.vhd	Testbench generated by Waveformer Lite for simulation
Top_Level_tbench.btim	Waveformer Lite project file for the testbench generation
pll_a.vhd	Multiply the input frequency by 5 with 0ps delay
pll_b.vhd	Multiply the input frequency by 5 with 250ps delay
pll_c.vhd	Multiply the input frequency by 5 with 500ps delay
pll_d.vhd	Multiply the input frequency by 5 with 750ps delay
*.gen	ACTgen macros for the PLLs

http://www.actel.com

Actel Corporation
2061 Stierlin Court
Mountain View, CA
94043-4655 USA
Phone 650.318.4200
Fax 650.318.4600

Actel Europe Ltd.
Dunlop House, Riverside Way
Camberley, Surrey GU15 3YL
United Kingdom
Phone +44 (0) 1276 401 450
Fax +44 (0) 1276 401 490

Actel Japan
EXOS Ebisu Bldg. 4F
1-24-14 Ebisu Shibuya-ku
Tokyo 150 Japan
Phone +81.03.3445.7671
Fax +81.03.3445.7668

Actel Hong Kong
39th Floor, One Pacific Place
88 Queensway, Admiralty
Hong Kong
Phone +852.227.35712
Fax +852.227.35999

51900075-0/07.04