Designing a SuperClock with an Axcelerator Device

Introduction

Many board designs today require complex clocking schemes involving multiple frequencies and phases. Semiconductor manufacturers have developed a multitude of products to address these situations, from simple low-skew clock buffers to multi-clock frequency synthesizers. In an effort to provide more system-level capabilities, field programmable gate array (FPGA) manufacturers have also introduced clock-conditioning elements in their devices. The Axcelerator family, with its eight segmentable clock buffers, eight 1 gigahertz (GHz) phase-locked loops (PLLs), and extremely fast routing resources, is ideally suited for system-level applications. This application note is intended as a template to help designers create complex, high-performance clock-conditioning circuits quickly and easily.

What is SuperClock?

SuperClock is a dynamic clocking system that enables users to generate multiple clock frequencies from a single source. An example block diagram of a SuperClock design is shown in Figure 1. The blocks shaded in red are implemented in the SuperClock reference design (Figure 4 on page 6). The areas in yellow are suggested blocks for SuperClock enhancements, and are not implemented in the reference design.

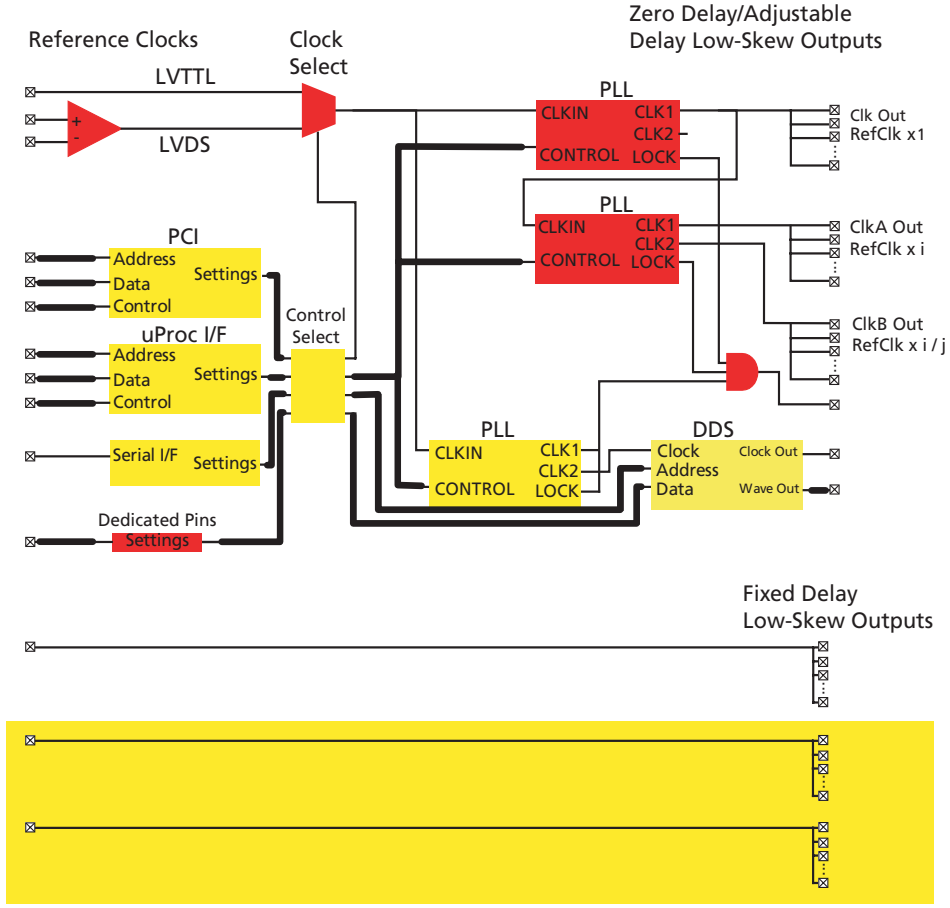

Figure 1 • **SuperClock Example**

With Printed Circuit Board (PCB) real estate ever more valuable, single-chip solutions offer many benefits. SuperClock is a single-chip solution implemented in an Axcelerator device. Here are some SuperClock highlights:

- 20 megahertz (MHz) to 1 GHz output frequency range with unlimited fanout buffers
- Up to sixteen clock frequency domains
  - Dynamic control of each frequency domain
  - Nearly zero effective skew between two clock domains
  - Low output jitter
- Adjustable clock delay, 32 steps of 250 picoseconds (ps)
- Low-skew clock buffers without using any global resources
- Support for a wide range of input/output (I/O) standards for all inputs and outputs
  - Low-Voltage Differential Signaling (LVDS), 350 MHz
  - Low-Voltage Positive Emitter Coupled Logic (LVPECL)
  - Low-Voltage Transistor-Transistor Logic (LVTTL)/Low-Voltage Complementary Metal Oxide Semiconductor (LVCMOS)
  - Gunning Transceiver Logic plus (GTL+)
  - Stub Series Terminated Logic (SSTL) 2/3
  - High-Speed Transceiver Logic (HSTL)
  - Peripheral Component Interconnect (PCI)
  - Peripheral Component Interface eXtended (PCI-X)
  - Input/reference clock selector (multiplexed from a wide variety of I/O standards)
- All adjustable parameters are dynamically reconfigurable
- Programmable logic resources for implementing additional functionality

Refer to the "Design Implementation" section on page 3 for a functional block description.

Using SuperClock

SuperClock can be used in telecommunications, optical transceivers, data and storage area networks, and wireless products where low skew and accurate clocks are essential for performance. Figure 2 is an example application of SuperClock circuitry.

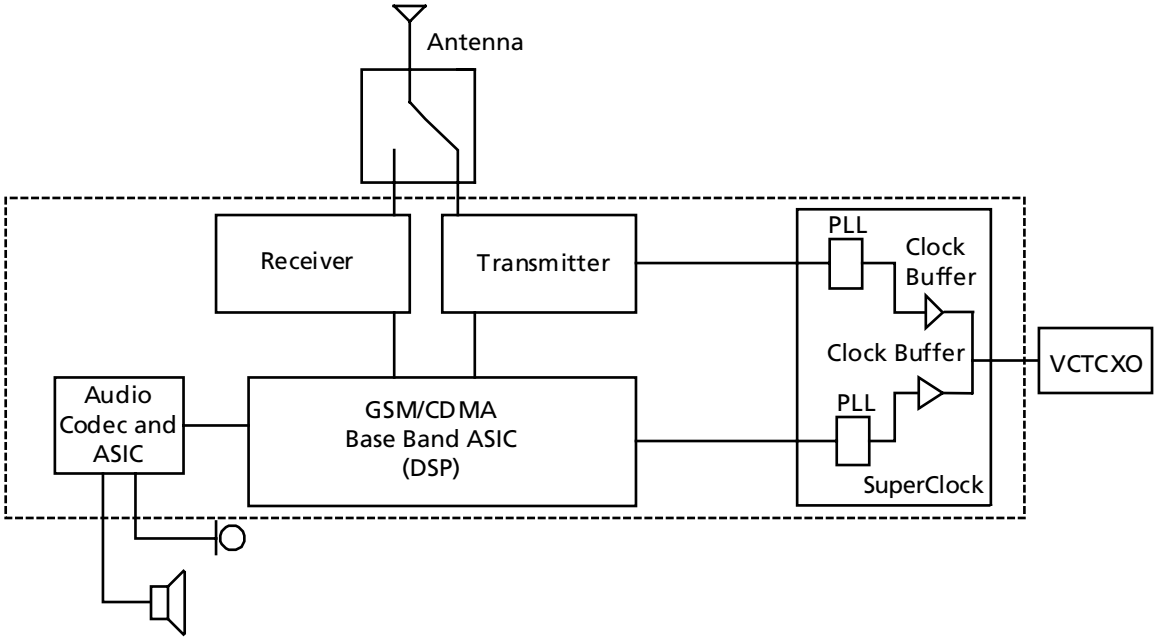

Figure 2 • **SuperClock in a Cell Phone Application**

Cell phones require clock buffering of the system reference voltage-controlled temperature-controlled crystal oscillator (VCTCXO). The transmitter and baseband application-specific integrated circuits (ASICs) need phase adjustment of the frequency modulation (FM) audio signal. As depicted in Figure 2, SuperClock's low-skew clock buffers and dynamically reconfigurable PLLs can provide a single-chip solution for the cell phone system.

As indicated by the dotted lines in Figure 2, the entire system can be implemented in a single Axcelerator chip with a chip-scale package (CS180) to meet the space-saving requirements of cell phone applications.

Design Implementation

The Axcelerator family features low power, high speed, and faster routing than other Actel families. It also has on-chip low-jitter, flexible PLLs capable of generating multiple frequency clocking resources. The SuperClock reference design was implemented on the AX250 device but can be used in any member of the Axcelerator family. The low logic utilization of this design enables the user to implement other functionality in the chip (refer to the "Utilization" section on page 8). The design takes advantage of the following features of the Axcelerator device:

- Support for a variety of I/O standards
- Cascaded PLLs
- Dynamic reconfigurability of PLLs
- Low-skew clock drivers
- Clock spines

Functional Block Description

This section describes the blocks shown in Figure 1 on page 1.

Clock Select

Implemented by a 2:1 multiplexer (MUX), clock select provides the option to dynamically switch between two different reference clocks. The reference design selects either an LVTTL or an LVDS clock input. Axcelerator devices support multiple I/O standards, so the clock sources can be any of these standards.

Control Select

Although it is likely that only a single control source would be used, any number of interfaces can be supported in cases where users want a single chip that handles different applications. The microprocessor interface is a simple general-purpose synchronous eight-bit data bus used to load settings into control registers. Dedicated pins on the FPGA are also provided with internal pull-ups to allow users to set parameters by using jumpers. The microprocessor interface (µProc I/F) can override these settings as well as load the direct digital synthesis (DDS) table.

Other control sources can be I^2C, Joint Test Action Group (JTAG), PCI, etc. Each of these sources may also have their own interfaces. For simplicity, none of the control source interfaces in this section are implemented in the SuperClock reference design shown in Figure 4 on page 6.

DDS

The DDS is an accumulator-based frequency synthesis look-up table (LUT) capable of creating any cyclical waveform with very fine frequency resolution. It uses random access memory (RAM) blocks to create the LUT. Additional RAM blocks can be used to increase the word width and/or frequency resolution. A single PLL is used for phase adjustment and frequency synthesis for the wave table clock; a second PLL might add even more accuracy to the final clock frequency. Advanced Encryption Standard (AES) decryption can also be used to update the DDS table or other parameters securely for security sensitive applications such as frequency-hopping radio. For simplicity, this block was not implemented in the SuperClock reference design (Figure 4 on page 6).

PLLs

The PLLs are used to provide phase compensation and frequency synthesis. The design uses the first PLL in the cascade to provide a 1x clock output, but could be easily modified to provide a second output at a different frequency or with extra phase compensation. Axcelerator devices contain eight PLLs, each with two clock outputs capable of generating sixteen independent clocks with different frequencies. There are two ways of configuring the PLL input and output frequencies. The Actel ACTgen core generator can be used to generate PLL macros, which are then instantiated in the design. The ACTgen graphical user interface (GUI) provides all possible configurations. The other way of configuring the PLL is to instantiate PLL primitives in the design and dynamically control the configuration bits through an I/O interface (refer to Figure 3 on page 5). This allows dynamic control of both the output frequency and feedback delays.

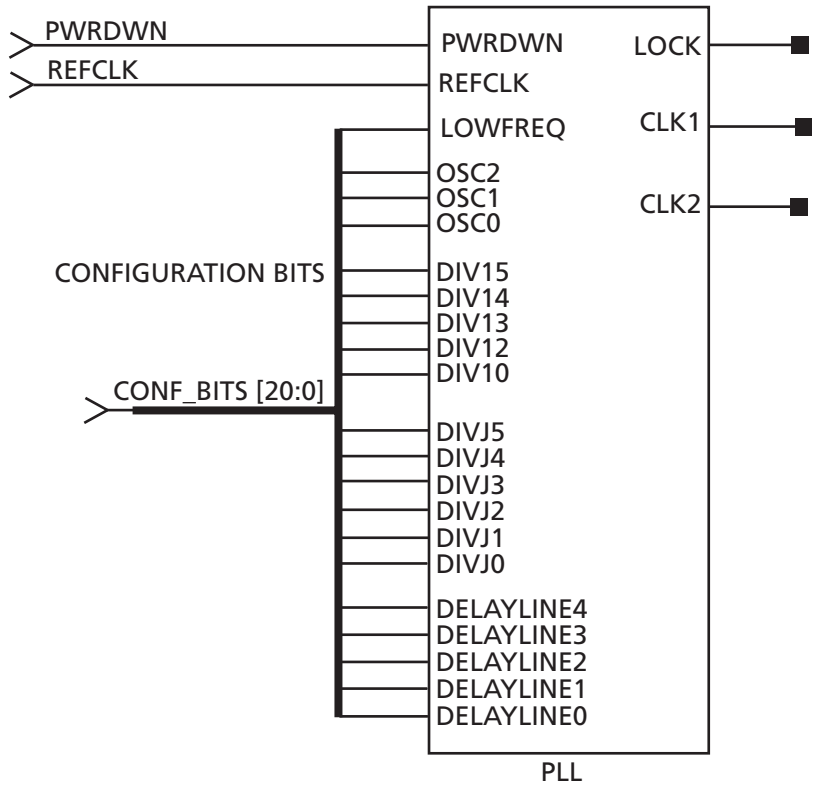

Figure 3 • **PLL Primitive**

Table 1 illustrates the allowable values for PLL configuration.

Table 1 • **Configuration Bits**

Signal Range	Description	Value	Parameter Range (MHz)
LowFreq	Input frequency range selector	0	50-200
		1	14-50
Osc[2:0]	Output frequency range selector[1]	xx0	400-1000
		001	200-400
		011	100-200
		101	50-100
		111	20-50
DIVI[5:0][2]	Feedback divider (multiplier)	1 to 64[3]	000000 to 111111
DIVJ[5:0][2]	CLK1 divider	1 to 64[3]	000000 to 111111
DelayLine[4:0]	Nominal clock delay (positive/negative) in 250 ps increments. MSB is sign bit (0 is positive, 1 is negative)	-3.75 to +3.75 ns	Positive delays = 00000 to 01111; Negative delays = 10000 to 11111 Example: 10011 is –750 ps nominal delay

Notes:

1. Select ranges based on the higher frequency of CLK1 and CLK2.
2. $f_{CLK1} = f_{REFCLK} * (DIVI)/(DIVJ)$; $f_{CLK2} = f_{REFCLK} * (DIVI)$.
3. Integer value; note that 000000 corresponds to a divider of 1 and 111111 is a divider of 64.

The following is an example illustrating how Table 1 can be used to select the configuration bits:

Assume the input frequency is set as f_{REFCLK} = 20 MHz. The two output frequencies are f_{CLK1} = 60 MHz and f_{CLK2} = 120 MHz. Both clocks have a negative delay of 2.5 ns

This results in the following settings:

DIVI = f_{CLK2} / f_{REFCLK} = 120/20 = 6 (setting is 000101)

DIVJ = (f_{REFCLK} / f_{CLK1})* DIVI = 2 (setting is 000001)

DelayLine = -2.5 ns (setting is 11010)

Osc = Based on value of f_{CLK2} (setting is 011)

As shown in Figure 4, CLK1 of the primary PLL drives the REFCLK signal of the secondary PLL and CLK2 is brought out to external pins through a PLLRCLK buffer. The output of the PLL must be connected to the PLLHCLK, PLLRCLK, or PLLOUT. PLLRCLK can feed high-fanout nets. The output of PLLRCLK can be output to any number of pins and used as a clock source for multiple sockets. Similarly, both output clocks of the secondary PLL are used as two clock groups, each driving four pins.

In this reference design (Figure 4), the configuration bits are controlled directly from the I/Os. Table 2 on page 7 illustrates the mapping between the 21-bit control bus and the configuration bits of the primary and secondary PLLs.

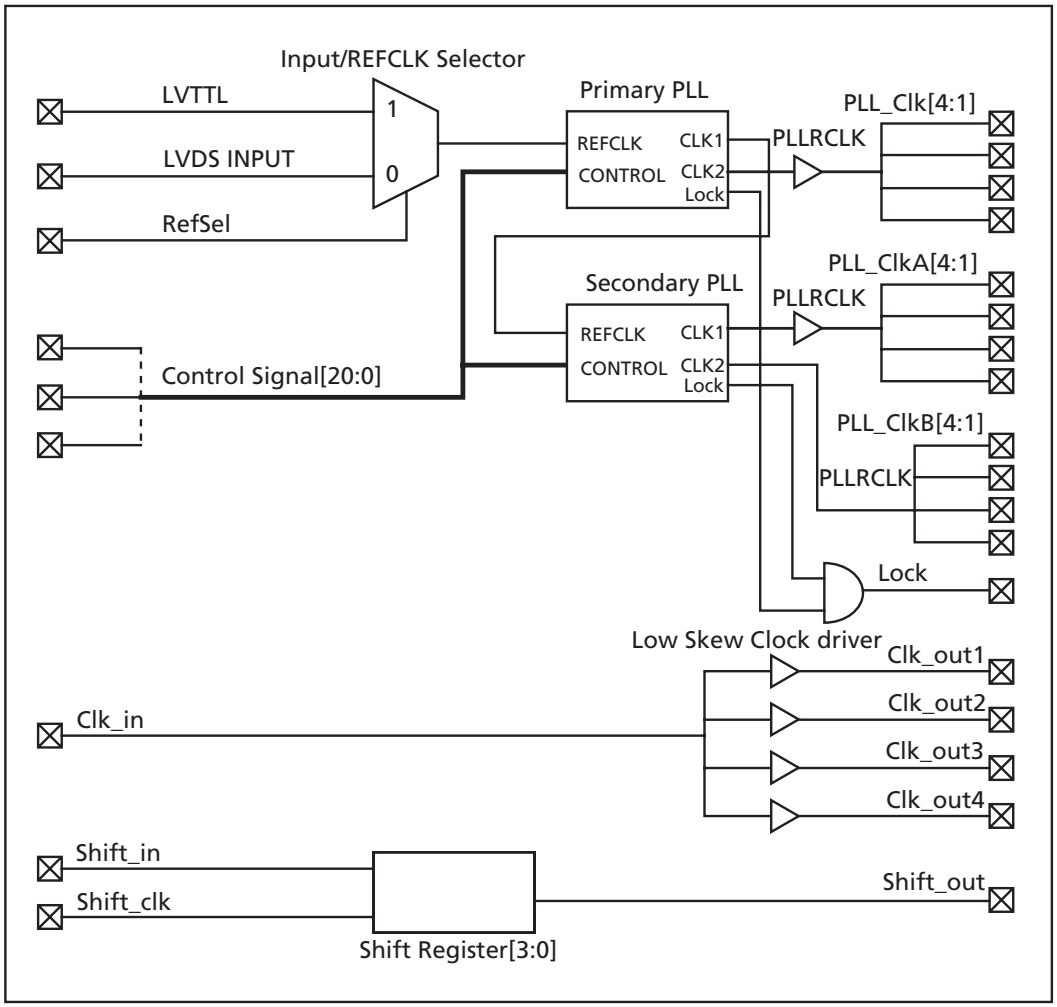

Figure 4 • **SuperClock Reference Design**

Table 2 • **SuperClock Configuration Bit Mapping**

Signals	Primary PLL	Secondary PLL
LowFreq	Control[0]	Control[0]
Osc[2:0]	Control[3:1]	Control[3:1]
DIVI[5:0]	Tied to GND	Control[9:4]
DIVJ[5:0]	Tied to GND	Control[15:10]
DelayLine[4:0]	Tied to GND	Control[20:16]

Low-Skew Fixed Delay Buffers

Any number of buffers can be added for simple clock buffering requirements. The extremely fast routing of the Axcelerator family means that the outputs can have very low skew without using any global buffer resources.

Implementation Details

The Actel Libero® Integrated Design Environment (IDE) tool was used to design the SuperClock. The design was implemented in Verilog code. Two cascaded PLLs were instantiated at the top level. Regular output buffers were used for input-to-output clock buffering.

The Axcelerator architecture gives the user the flexibility of partitioning the clock resources on the die. The user can constrain the routing to a particular portion of the die to achieve better control and subsequently enhanced performance. There are two levels of clock segmentation, which can be used separately or in combination for any clock. The first is the Tile level where the user constrains the clock to use the Tile local clock routing (either HCLK or RCLK). The second level of control allows use of specific column(s) (HCLK) or row(s) (RCLK) within Tiles. Figure 5 shows some examples of clock segmentation.

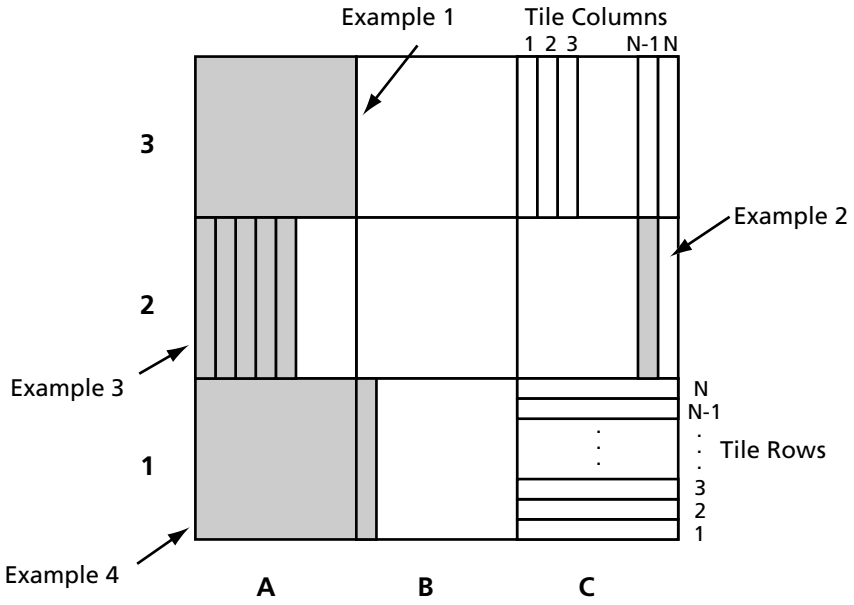

Figure 5 • **Clock Segmentation by Rows and Columns**

Example 1 – Full Tile Assignment

Here the entire tile 3A is assigned to a local clock associated with net a1, using an RCLK resource.

```
assign_local_clock -type rclk -net a1 tile3A
```

Example 2 – Single Column Assignment within a Tile

This command assigns a single column in tile 2C to a local clock associated with net a2. The routing resource is an HCLK. The column is the next-to-last in an AX100 device – the 11th of 12.

```
assign_local_clock -type hclk -net a2 tile2C.col11
```

Example 3 – Multiple Column Assignment within a Tile

Columns 1 through 5 of tile 2A are associated with a local clock on net a3.

```
assign_local_clock -type hclk -net a3 tile2A.col1 tile2A.col2 tile2A.col3 tile2A.col4
tile2A.col5
```

Example 4 – Assigning a Full Tile Plus a Partial Tile

This example mixes the local_clock_region assignment types. A complete tile, 1A, is assigned to a local HCLK associated with net a4. In addition, a single column of neighboring tile 1B is associated with the same local clock.

```
assign_local_clock -type hclk -net a4 tile1A tile1B.col1
```

At the top level, four-bit shift register logic was created to show the usage of this clock segmentation. AX250 devices have four core tiles. Tile 1b (full tile assignment) was used to create a spine region for the shift register; its clock is assigned as a local clock in the region.

Synthesis

Specific names were given to each instantiated BUFD buffer. The ALSPRESERVE property ensures synthesis will not change the instance names and Designer will not optimize away the logic. This way, the manual locations can be preserved even if the design is re-synthesized.

A SuperClock.sdc constraint file was used so that Synplify® would not insert any HCLK for shift register inputs. This logic was only implemented to demonstrate the spine usage.

Utilization

The SuperClock design was targeted to an AX250-PQ208 device.

The SuperClock.pdc and SuperClock_spine.pdc constraint files were imported along with the netlist to perform compile and layout. The first physical design constraint (PDC) file assigns V_{REF} to an I/O bank and I/O pins for different clock output groups, while the later PDC file creates a spine region for the shift register logic. This can also be done manually in the Designer MultiView Navigator (MVN). ChipPlanner in MVN allows the user to select the V_{REF} for I/O banks and graphically create spine regions for any logic macros. The I/O attribute editor in MVN can be used for pin assignments. Please refer to the appropriate *User's Guides* on the Actel website for details.

The device utilization is shown in Table 3.

Table 3 • **SuperClock Resource Utilization in an AX250**

Resources	Used	Available
Logic cells (R + C-cells)	20	4224
I/O with Clocks	47	115
CLK (Routed)	3	4
HCLK (Hardwired)	0	4
PLL	2	4

Design Verification

The testbench for the SuperClock design was created in SynaptiCAD WaveFormer Lite™ Actel Edition (AE) and the design was simulated in Mentor Graphics Model*Sim*® AE.

The control bus was configured to generate output frequencies of the primary and secondary PLLs as shown in Table 4. The REFCLK frequency of the primary PLL was set to 200 MHz.

Two configurations were used in Table 4. This shows the dynamic behavior of the PLL. For simplicity, the output frequency of the primary PLL was kept the same; only the output frequency of the secondary PLL was changed.

Figure 6 is a snapshot of the Model*Sim* AE wave window. This shows the output frequencies of the secondary PLL are changing dynamically (at 230 ns) when the control signal sets a new configuration.

Table 4 • **Frequency Combination Chart**

Control Signal [20:0]	**Primary PLL**			**Secondary PLL**		
	CLK1 (MHz)	**CLK2 (MHz)**	**Delay (ns)**	**CLK1 (MHz)**	**CLK2 (MHz)**	**Delay (ns)**
Configuration 1 21'b010000000010000110000	200	400	0	400	800	2
Configuration 2 21'b011000000010001000000	200	400	0	500	1000	3

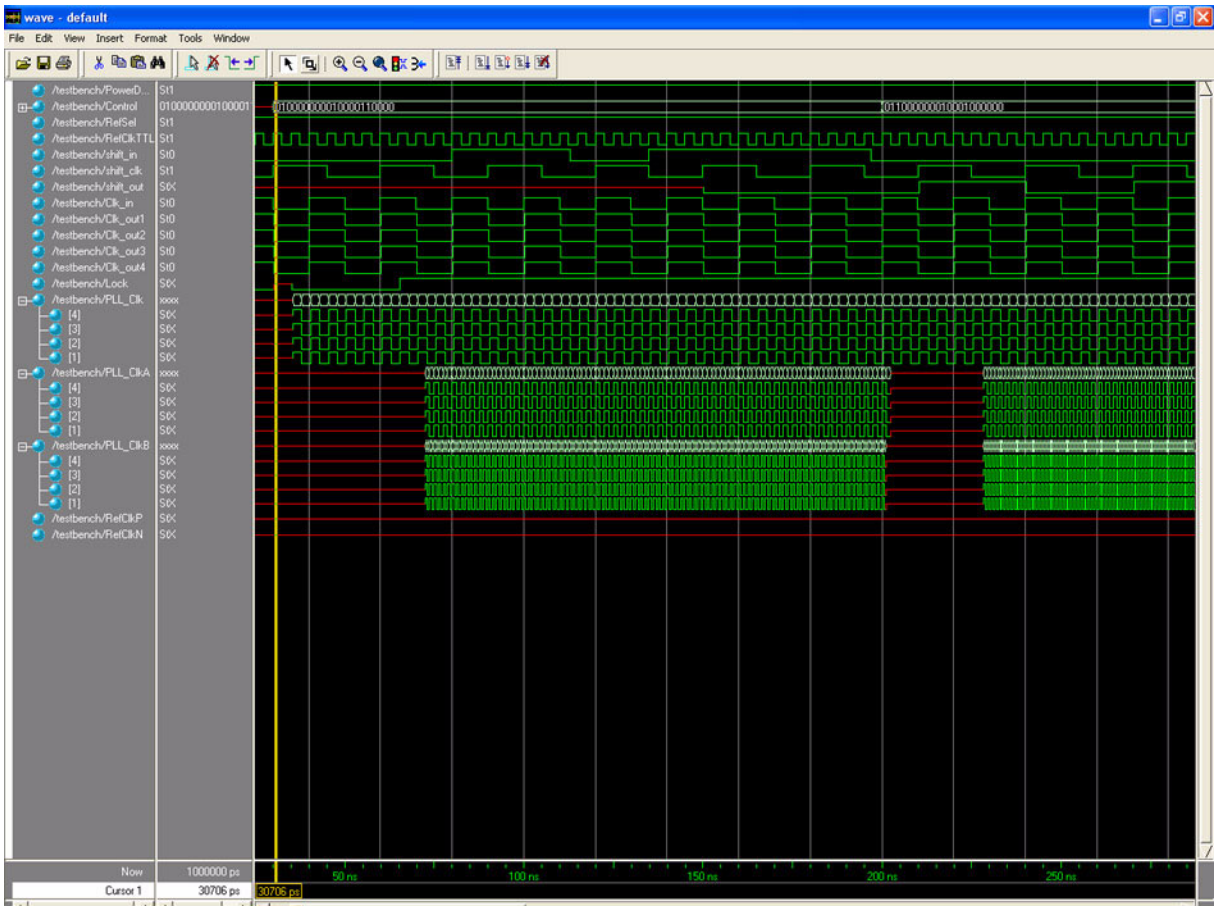

Figure 6 • **Simulation Waveform Window**

Conclusion

Designing the clock conditioning circuitry using an Axcelerator FPGA clearly has several advantages. Axcelerator provides flexible clock conditioning while allowing extremely high performance. The SuperClock reference design provides an excellent starting point for users to create complex and high-performance clock conditioning circuitry.

References

Datasheets

Axcelerator Family FPGAs
www.actel.com/documents/AX_DS.pdf

Application notes

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

User's Guides

Antifuse Macro Library Guide
www.actel.com/documents/libguide_UG.pdf
Libero IDE User's Guide
www.actel.com/documents/libero_UG.pdf
Designer User's Guide
www.actel.com/documents/designer_UG.pdf
MultiView Navigator v6.0 User's Guide Includes: NetlistViewer, PinEditor, I/O Attribute Editor, ChipPlanner
www.actel.com/documents/mvn_ug.pdf
ACTgen Cores Reference Guide
www.actel.com/documents/gen_refguide_UG.pdf

Appendix

Figure 7 shows the Libero IDE project directory structure.

Libero IDE v6.0 was used to create the project. You need version 6.0 or newer to open the project. Download the SuperClock Design Files to a hard disk. The directory structure will look like Figure 7 when the file is unzipped.

You can also use the standalone version of Designer (version 6.0 or newer) to open the Actel database (ADB) file.

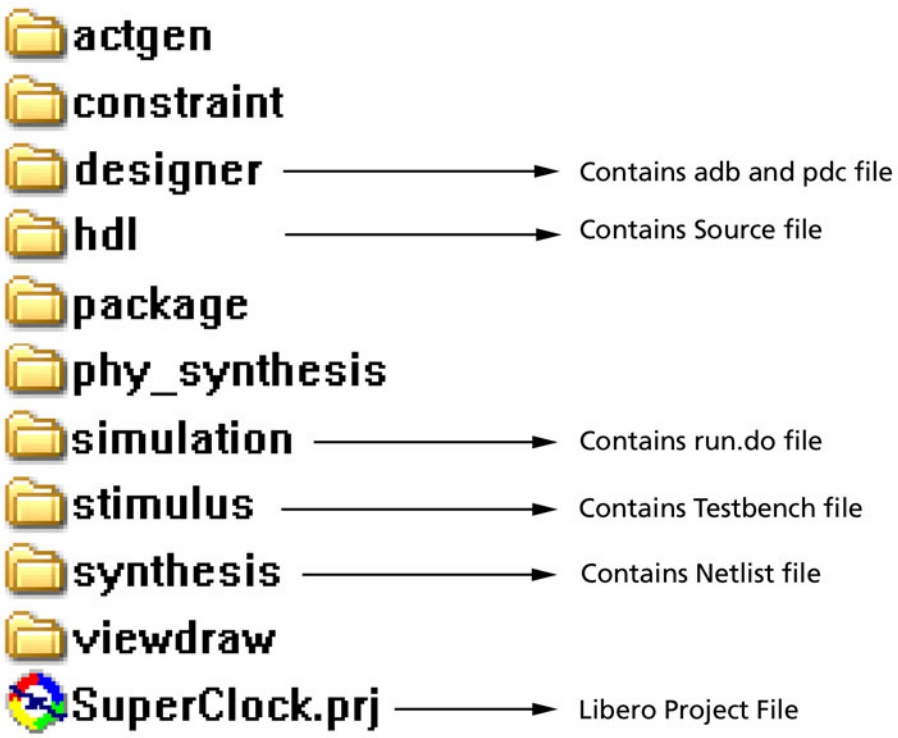

Figure 7 • **Libero IDE Project Directory Structure**

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

51900079-0/09.04