Axcelerator Family PLL and Clock Management

Introduction

Clock management is very important in communication and networking applications where low noise and accurate clocks are vital for the design performance. Multiple clock domains with a wide range of operating frequencies are another characteristic of networking applications. The Actel Axcelerator FPGA family provides plenty of resources to meet the requirements for high speed, multiple frequency clock applications. Axcelerator devices contain eight PLLs with two clock outputs for each PLL; the devices are capable of generating sixteen independent clocks with different frequencies. PLLs can be used in modulators, demodulators, tracking generators, or clock/data recovery circuits.

A typical block of Phase Locked Loop (PLL) architecture consists of a phase detector, filter, voltage-controlled oscillator (VCO), and divider (if necessary) as indicated in Figure 1.

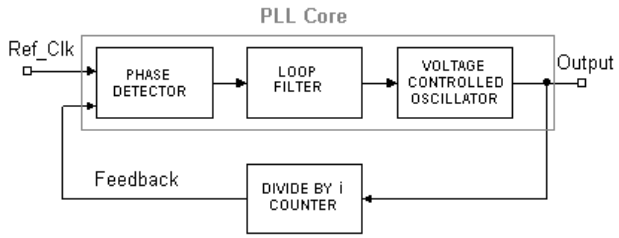

Figure 1 • Basic PLL Building Blocks

A PLL is a negative feedback control system that controls the phase of a VCO. The output of the phase detector is a voltage proportional to the phase difference between the input reference and the feedback signal. This voltage drives the VCO, and it changes the output frequency. The feedback loop will cause the output of the PLL to be locked at the same phase of the input. Note that the output frequency is n times greater than the feedback signal and the reference clock. Inserting time delays at the feedback loop (after divider) or output (after VCO) will generate desired phase lead or lag at the output (programmable time delay). A series of different dividers at the feedback loop or output will enable the PLL to generate different frequencies from the reference frequency, based on Equation 1:

$$f_{out} = f_{REF} \times \frac{i}{j} \quad (1)$$

Where:

f_{out} = the output frequency of the PLL
f_{REF} = the frequency of the reference input to the PLL
i = the divider in the feedback loop
j = the divider on the output

There are important parameters that define the performance of a PLL. The basic parameters of a PLL are acquisition range, acquisition time, and jitter. Acquisition range is the range of input frequency that a PLL can lock in and is proportional to the loop bandwidth. The acquisition time is the time that the PLL takes to converge within a certain phase error of the input signal or simply lock in. The acquisition time is inversely proportional to the loop bandwidth, and hence, there is always a compromise between acquisition time and range. Jitter is a time domain parameter, which is equal to the phase noise in the frequency domain. Internal jitter is mostly generated by the VCO and is lowered by a low pass filter in the PLL.

The Axcelerator family provides eight PLLs, four on the north side and four on the south side of the device. The outputs of the north-side PLLs can be connected to either hard-wired clock networks or regular nets. The outputs of the south-side PLLs can be connected to either routed clock networks or regular nets. The Axcelerator family PLLs have many outstanding features, including the following:

- PLLs can multiply and/or divide the reference clock frequency by factors ranging from1 to 64. In other words, each of i and j in Equation 1 can have an integer value between 1 and 64. As a result, there are many available output frequencies for each PLL, based on the input frequency.
- PLLs are capable of inserting programmable delays on the output from –3.75ns to +3.75ns with the steps of 250ps. The delay is programmed either statically or dynamically. Dynamic programming means that the user can change the delay value during the operation when the device is functional. This benefits users who want to shift the clock frequency back and forth to compensate for external or internal delays and synchronization issues. The delay blocks are valuable for data-recovery applications, where DC phase is critical.
- The input frequency of PLLs can vary from 14 MHz to 200 MHz and can generate a range of 20 MHz to 1 GHz

output frequency. The maximum acquisition time of the PLLs in Axcelerator family devices is 20µs.

- The maximum long-term jitter is 1% or 100ps (whichever is greater), and the maximum short-term jitter is 50ps plus 1% of output frequency.

PLL Architecture in Axcelerator Family

Actel Axcelerator family devices incorporate eight PLLs with similar architecture. Figure 2 describes the general architecture of Axcelerator device PLLs.

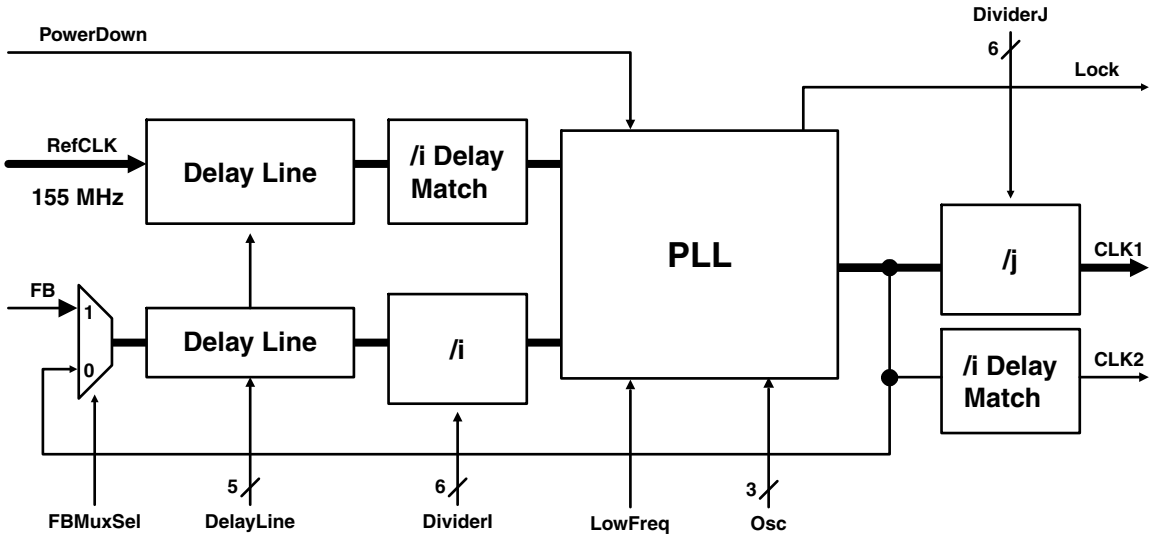

Figure 2 • *Basic PLL Architecture in Axcelerator Devices*

Figure 2 indicates that there are peripheral blocks around the PLL core to increase its functionality and performance. This section of the document discusses the inputs, outputs, internal blocks, and their functionality.

Refclk is the reference input to the PLL. The frequency of Refclk can vary from 14 MHz to 200 MHz. The reference can be supplied from a dedicated pad or an internal net. The ACTgen macro builder will give the user an option to select the source of Refclk (See the "ACTgen Macro Builder" section on page 3 for more information). If the reference input of the PLL is a core net from internal logic, a PLLINT buffer is required before the reference input of the PLL. The core net can be connected to the input of the buffer and the buffer output will load the Refclk port of the PLL. If PLLINT is not used, then Refclk can only be connected to the dedicated clock pins on two sides of the device.

If the user needs to bring the reference clock from a regular I/O, an instantiation of the PLLINT buffer is required before the Refclk input of the PLL. The output of the I/O pad should be connected to the input of the PLLINT buffer, and the buffer output supplies the Refclk input of the PLL.

The *power-down* input acts like an active-high enable signal, and when low, the PLL will shutdown.

FB is the external feedback signal. As shown in Figure 2, there is also an internal feedback path from the output of the PLL. Both internal and external feedback signals are inputs to a MUX, which is controlled by the FBMuxSel signal. The FBMuxSel signal selects external or internal feedback. The user can select the type of the feedback in the ACTgen macro builder or by setting the FBMuxSel value, if the PLL macro is instantiated manually. The external feedback configuration can be used for compensating the clock skew and tuning the clock-to-out timing performance. This will be further discussed in the "Clock to Out Timing" section on page 5.

LowFreq is an active-high input that indicates the operating mode of the PLL core based on the input reference frequency. If the input frequency is between 14 MHz and 50 MHz, the PLL operates in the low-frequency mode and therefore, the LowFreq should be activated (connected to V_{CC}). For input frequencies between 50 MHz and 200 MHz, the PLL is in high-frequency operating mode, and the LowFreq input should be inactive (grounded). If the PLL macro is created by ACTgen, the LowFreq input will be set by the software and is transparent to the user.

Osc, which defines the output frequency range, is an input to the PLL core. Osc is 3 bits wide and along with LowFreq input, specifies the operating range of the PLL. The possible values for Osc are shown in Table 1 on page 3. Osc and LowFreq values tune the bandwidth of the internal filters of the PLL core for a proper operation and accurate output frequency. If the PLL macro is generated by ACTgen, the software will set the Osc value based on the desired output frequency.

Table 1 • *Osc Values for PLL Output Frequency Range*

Osc[2:0]	Output Frequency Range (MHz)
XX0	400 - 1000
001	200 - 400
011	100 - 200
101	50 - 100
111	20 - 50

Delay Line is another input to PLL, which defines the delay value (phase shift) on the outputs. The Delay Line is 5 bits wide and can be programmed dynamically or statically. The static delay means that the amount of the delay is fixed and cannot be changed once the device is programmed. If generated by ACTgen, the Delay-Line bits will be assigned automatically (This will be further discussed in the ACTgen Macro Builder section.). If instantiated manually, the user should configure these bits (by connecting to V_{CC} or GND) to implement the desired delay. The first 4 bits of the Delay Line behave like a counter with "0000" corresponding to 0ns, and each increment counting for a 250ps step increase in the amount of the delay (e.g. 0011 corresponds to 0.75ns delay). Hence, a value of "1111" on the first 4 bits of the Delay Line corresponds to 3.75ns delay (16 steps from 0ns). The MSB of the Delay Line (Delay Line[5]) indicates the sign of the delay value with "0" for positive and "1" for negative delays. The Delay-Line value can also vary during the operation (dynamic delay). In this case, the 5-bit Delay Line will be presented as a port to the PLL and can be driven by any internal or external signal. The variable Delay Line is useful in synchronization and data-recovery applications. There are two Delay-Line blocks in the PLL. One is located in the feedback loop, and the other one is placed at the input of the PLL.

The PLL block contains two ÷i or ÷j Delay Match blocks. The delay that these two blocks insert on the path is fixed and compensates for the delay of the divider blocks. In other words, the ÷i Delay Match block at the input compensates for the delay of the ÷i block to synchronize the feedback and reference inputs to the PLL core. The other ÷j Delay Match block synchronizes CLK1 and CLK2 outputs by inserting a delay in the CLK2 path equal to the ÷j block delay in the CLK1 path. The delay amount of these blocks cannot be changed by the user.

Two divider blocks exist in the Axcelerator family PLLs: ÷j and ÷i. These blocks are responsible for generating the desired output frequencies from the reference clock. The ÷i block is located in the feedback path of the PLL core, which causes the output frequency to be i times the reference frequency (Notice that the frequencies of the feedback and reference at the input to the PLL core are similar). The ÷j block, located in the CLK1 output path, divides the output frequency of the PLL core by a factor of j. In general, CLK1 and CLK2 output frequencies can be defined as the following:

$$f_{CLK1} = \frac{f_{REF} \times i}{j} \quad (2)$$

$$f_{CLK2} = f_{REF} \times i \quad (3)$$

Equations 2 and 3 show that CLK2 is more restricted in terms of the available frequencies based on the reference input. Values of i and j are programmed statically.

Another output of the PLL core is Lock. This signal becomes active once the PLL locks in the reference frequency. This signal can be used to determine the acquisition time of the PLL.

If the PLL block is located on the north side of the chip, the CLK1 and CLK2 can be connected to either an HCLK network or a regular net. If the PLL output is required to be connected to the HCLK network, it should be connected a to PLLHCLK buffer input. The output of this buffer will be connected to the HCLK network; otherwise, a PLLOUT buffer is required to connect the PLL output to regular nets. If the PLL block is in the south side of the chip, the outputs are capable of driving the RCLK network or a regular net. The PLLRCLK buffer is required at the output of the PLL if the output clock needs to be connected to the routed clock network. Similarly, the PLLOUT buffer connects the output clocks to the regular nets.

If necessary, the CLK1 output of the PLL can also drive the global network (i.e. RCLK or HCLK network) of its adjacent PLL (e.g. if both CLK1 and CLK2 are selected to drive hard-wired global network). Note that there are only eight global networks (four HCLK and four RCLK) on the device.

The PLLs can be cascaded in order to achieve more frequencies inside the acquisition range. This will be further discussed in the "PLL Cascading" section on page 5.

ACTgen Macro Builder

Actel's ACTgen macro builder is the recommended tool to generate and instantiate PLLs in the design. The ACTgen GUI allows all the possible features of the Axcelerator family to be configured. Based on the user's selection, ACTgen creates the macro in the desired format (Edif, VHDL, or Verilog) with required ports. These macros can be instantiated and connected to the rest of the design. This section discusses the PLL generation of the ACTgen macro builder and its features.

Figure 3 on page 4 shows the ACTgen GUI for a PLL macro generation. All the different PLL macros are built from a basic PLL macro.

The GUI has been segmented into five categories: REF Clock, Feedback, Cascading, CLK1, and CLK2. Under REF Clock, the user selects the frequency of the reference input, the amount of delay and its type, and the source of the input

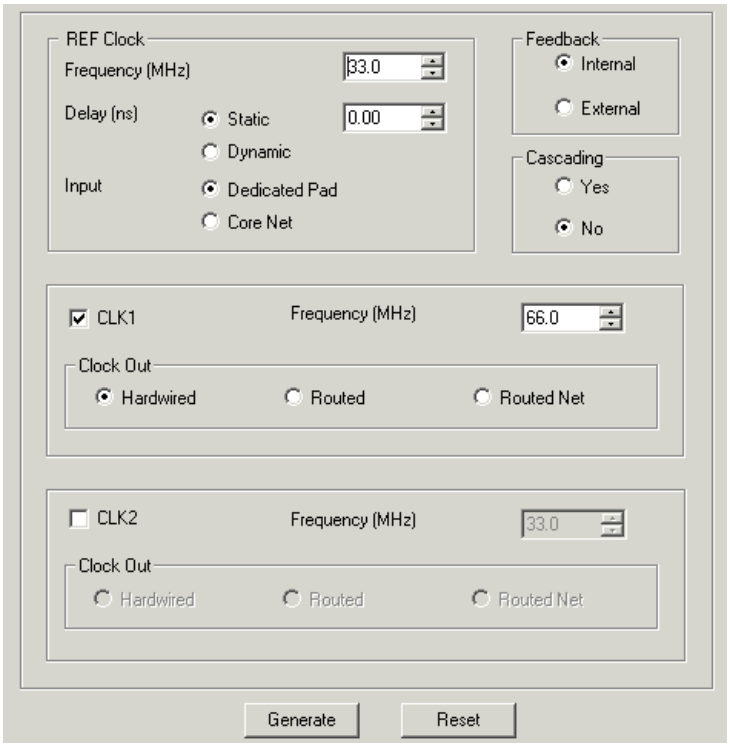

Figure 3 • *ACTgen GUI for Generating a PLL*

reference. The input frequency can be anywhere between 14 MHz and 200 MHz. The delay amount should be selected in the range of –3.75ns to +3.75ns. If the user selects the dynamic delay, then the 5-bit Delay Line port will be added to the generated code and is accessible to the user. The PLL input can be selected to be either a dedicated pad or a core (PLLINT) net. If input is selected to be a core net, an internal buffer will be added to the PLL before the reference clock input.

Under the "Clock to Out Timing" section on page 5, the user can select to have an internal or external feedback. Selecting an external feedback will add a port (named FB) to the PLL block, and FBMuxSel will be set to pass the external feedback through the MUX into the PLL and block the internal feedback.

Cascading is an option that helps users to generate a wider range of output frequencies. If the cascading is set to "No," and the output frequency is chosen as a value that cannot be achieved by f_{REF} * i/j, then the PLL will try to set i and j in order to reach to the closest vicinity of the desired frequency. If cascading is set to be "Yes," then for the conditions in which the desired frequency is unattainable by a single PLL, another PLL will be cascaded to the first PLL and then the final output frequency will be:

$$f_{out} = f_{REF} \times \left(\frac{i1}{j1}\right) \times \left(\frac{i2}{j2}\right) \quad (4)$$

Where each of i1, i2, j1, and j2 can be anywhere in the range from 1 to 64. Note that for CLK2, Equation 4 will be in the form of:

$$f_{out} = f_{REF} \times \left(\frac{i1}{j1}\right) \times (i2) \quad (5)$$

Moreover, if the desired frequency is still not achievable, then the cascaded PLLs will lock in the closest possible frequency to the desired one. While cascading, all the outputs of both PLLs (CLK1 and CLK2) are accessible and can be used. Please note that ACTgen will not cascade more than 2 PLLs.

The last two segments of the ACTgen GUI are the CLK1 and CLK2 outputs. In these sections, the user is required to specify the desired output frequencies and the networks that the outputs should drive. Note that if the cascading is disabled, the CLK2 frequency can only be a multiple of reference frequency. As mentioned earlier, if the selected values for output frequencies cannot be achieved, they will be set to the closest possible frequency. For each output, there are three routing resources. Hard-wired is the HCLK network which reaches to the clock input of R-cells. Selecting a hard-wired output for the PLL implies that the PLL should be located at the north side of the device. If one of the outputs is connected to hard-wired global network, the routed clock network cannot be chosen as the second output because the routed clock network is only accessible by the PLLs on the south side. ACTgen helps users select the

output type by keeping the possible outputs active and disabling the illegal combinations.

After the desired options have been set by clicking the "Generate" icon, ACTgen will generate the PLL macro with specified characteristics in the user defined format: VHDL, Verilog, or Edif.

PLL Cascading

Cascading the PLLs might be required for different purposes. One of them is to acquire more possible output frequencies. This kind of cascading can be done automatically via ACTgen by selecting the cascading option as "yes." ACTgen will try to extract the required frequency by one PLL. If failed to do so, the second PLL will be instantiated and cascaded to the first PLL to acquire an output frequency as close as possible to the desired value.

In some cases, the user needs to cascade PLLs for different purposes such as providing more programmable delay on the clock signal. The user can cascade as many as eight PLLs. Figure 4 shows a simple block diagram of two cascaded PLLs.

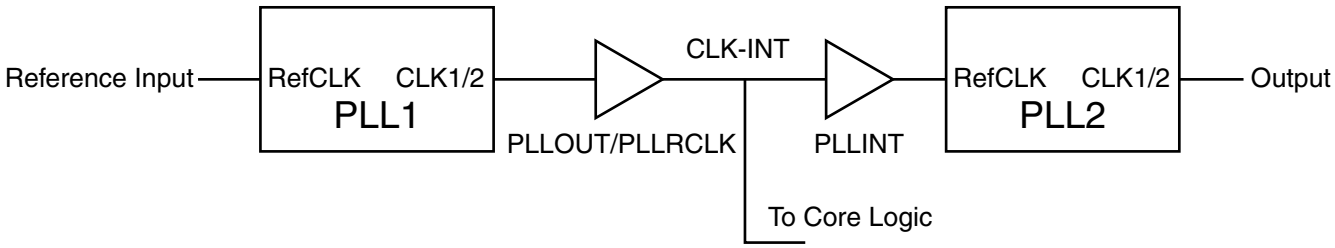

Figure 4 • *Block Diagram of Cascaded PLLs*

PLL1 and PLL2 can be from both sides of the device. However, there are few conditions to successfully cascade the PLLs. The Refclk input of the PLL can only be connected to regular nets (PLLINT buffer is required); therefore, the intermediate net between two PLLs (designated as CLK_INT in Figure 4) cannot be a hard-wired clock because the HCLK network cannot be connected to a regular net. However, if the output of PLL1 (i.e. CLK_INT in Figure 4) is selected to be a routed clock (RCLK network), it can drive the Refclk input of the next PLL through a buffer. If not inserted manually, Designer software will add this buffer before the Refclk input of the next PLL automatically. In the case that the output clock of the first PLL is selected to be a core net, there are no restrictions to drive all logics, including the Refclk input of PLLs. However, in this case, the user should be concerned about the long routing delay if the PLLs are in the opposite sides of the device. Being an RCLK or regular net, the clock signal between the cascaded PLLs (CLK_INT in Figure 4) can be used to drive the core logic.

Note: *In cascading PLLs, the input frequency of each PLL should remain in the range of 14 MHz to 200 MHz.*

Clock to Out Timing

PLL blocks can be used to control the clock-to-out timing performance of the design, especially when the clock path suffers from large delay. As an example, consider a clock network with a delay driving a register. The corresponding waveforms are shown in Figure 5, where CLK_INPUT is the clock input to the FPGA, and CLK_FF is the input clock at the flip flop after a large delay.

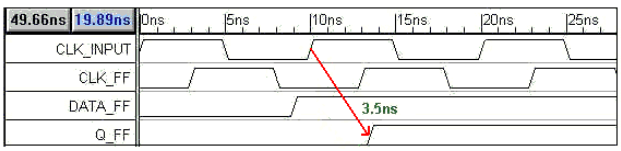

Figure 5 • *Example of a Timing Diagram with Large Clock-to-Out Performance*

As shown in Figure 5, the total clock-to-out time is 3.5ns, which only 0.25ns of it is due to the flip-flop, and the rest of the delay is a result of the clock delay.

PLLs can be used to compensate for the clock skew. If the clock input to the flip-flop is fed back to the PLL as shown in Figure 6 on page 6, the clock delay will be reflected on the feedback loop ($f_{RefClk} = f_{PLL\_OUTPUT}$).

Since the fanout of the feedback path is only 1, the delay on the feedback path is much less than the clock delay. Therefore, the total delay from the output of the PLL to the feedback input of the PLL is almost equal to the clock delay. As previously stated, a delay on the feedback will cause a phase lead on the PLL output. Therefore, by the time that clock signal gets to the flip-flop, it would be at the same phase of the input clock (CLK_INPUT in Figure 6). Figure 7 indicates the waveforms after using the PLL with worst-case skew point feedback.

As shown in Figure 7, the clock-to-out time has been reduced to 0.2ns, which is due to the clock-to-out delay of the flip-flop itself.

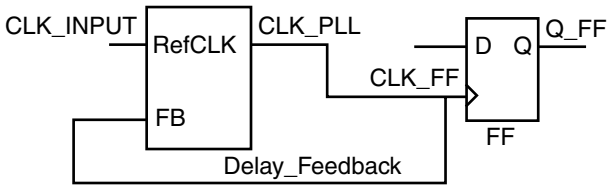

Figure 6 • *Feedback from Worst Skew Point to the PLL*

Figure 7 • *Example Timing Diagram after Worst-Case Skew Feedback*

Conclusion

The PLL macros in Actel's Axcelerator family of FPGAs make this antifuse family suitable for applications in which an accurate clock management is required. A large acquisition range along with a short acquisition time extends the use of PLLs into a wide range of applications in different levels of speed. Axcelerator family PLLs offer flexibility, including programmable delay, programmable input to output frequency ratio, various reference input sources, etc. A low jitter value extends the PLL use into the applications with high clock accuracy. PLLs can be cascaded to increase the number of available frequencies or any other applications. The ACTgen macro generator makes it easy to generate the PLL macro with the desired parameters.

The PLL blocks can also be used to tune the timing parameters of the design, such as clock-to-out timing. This can be done using feedback to the PLL from the worst clock skew point of the design.

Exploiting the clock management resources along with many other features, Actel's Axcelerator family is one of the best solutions for high-speed communication and networking applications.

http://www.actel.com

Actel Europe Ltd.
Maxfli Court, Riverside Way
Camberley, Surrey GU15 3YL
United Kingdom
Tel: +44 (0)1276 401450
Fax: +44 (0)1276 401590

Actel Corporation
955 East Arques Avenue
Sunnyvale, California 94086
USA
Tel: (408) 739-1010
Fax: (408) 739-1540

Actel Asia-Pacific
EXOS Ebisu Bldg. 4F
1-24-14 Ebisu Shibuya-ku
Tokyo 150 Japan
Tel: +81-(0)3-3445-7671
Fax: +81-(0)3-3445-7668

5192692-2/7.02