Application Note
January 2003

ORCA® Series 3 Programmable Clock Manager (PCM)

Introduction

As FPGA designs continue to increase in size, speed, and complexity, the need for system-level functions becomes extremely important to maintain the time-to-market advantage that is inherent with FPGAs. With the introduction of such items as embedded microprocessor interfaces, programmable clock managers, and the field-programmable system chip (FPSC), Lattice is prepared to offer system-level performance while maintaining the advantages of traditional SRAM based FPGAs.

The *ORCA* Series 3 programmable clock manager (PCM) is a special function block that is used to modify or condition clock signals for optimum system performance. To accommodate various applications, each PCM contains both a PLL (phased-locked loop) and a DLL (delay-locked loop). Some of the functions that can be performed with the PCM are clock skew reduction (both internal and board level), duty-cycle adjustment, clock phase adjustment, and clock frequency multiplication/division from 1/8x to 64x. By using *ORCA* Series 3 FPGA logic resources in conjunction with the PCM, many other functions, such as frequency synthesis, are possible.

This application note describes the modes of operation for the programmable clock managers. It will also provide the details necessary to incorporate the PCM into a design. Additionally, it will describe how the *ORCA* Foundry software will interpret these elements and generate the timing preferences required to perform a timing-driven place and route on the design.

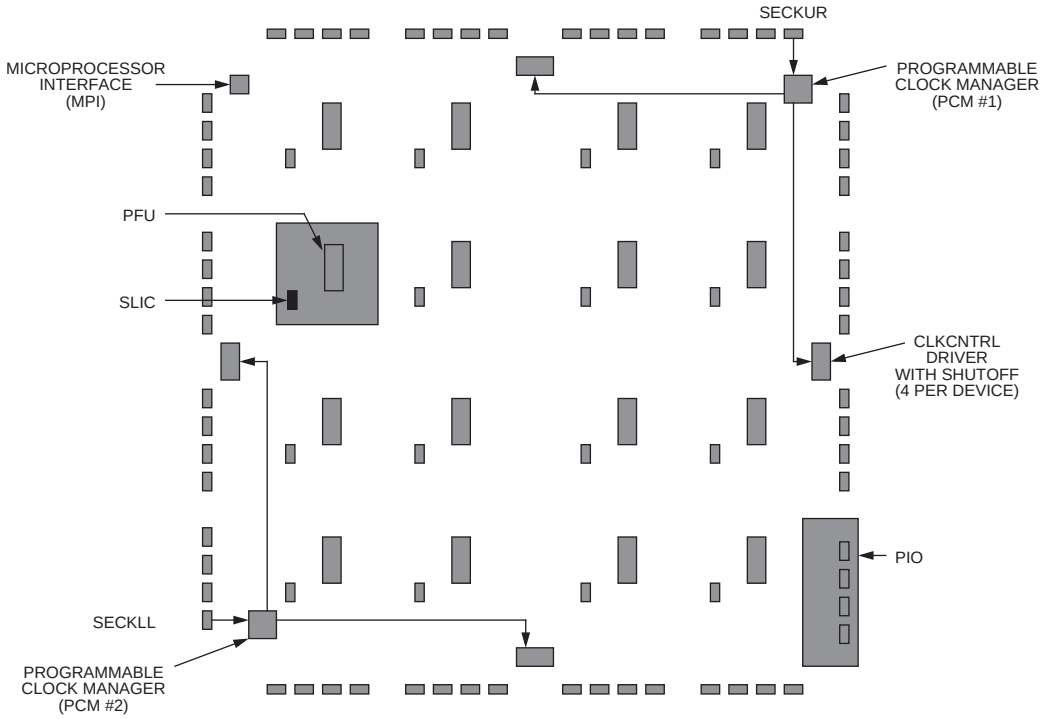


Figure 1. *ORCA* Series 3: Device Overview

Programmable Clock Manager

Every *ORCA* Series 3 FPGA and Series 3+ FPSC contains two programmable clock managers. Each PCM can handle one clock input and generate two different but interrelated clock signals.

Input Clock (CLKIN)

The PCM input can be driven by any internal or external signal. For fastest operation, use the appropriate ECLK input pin (SECKLL or SECKUR). These input pins (in addition to functioning as general-purpose I/O) have a direct connection to the CLKIN pin of the nearest PCM.

Implementation issue. There is no special buffer required for the SECKLL or SECKUR pin to function as the PCM source (only the normal IBT or IBM type buffers which are necessary for all inputs). The only requirement is to use the attribute LOC to assign the port to the proper input pin. (Refer to the *ORCA* series 3 and Series 3+ data sheets for the specific pin number and refer to the synthesis or schematics vendor documentation for details regarding the assignment of attributes.)

Output Clocks (SCLK and ECLK)

The *ORCA* Series 3 PCM has two clock outputs that can be driven simultaneously. Their outputs can be identical or different waveforms depending on the mode of operation.

The SCLK output pin drives to local routing and to SCLK clock routing.

Implementation Issue. The *ORCA* Foundry place and route tool will route this signal as if it were any other PFU output. This means that par will use local routing resources for the net attached to this output unless a high priority is assigned to it in the .prf file or there are at least six PFU clock loads attached to the net. In those cases, the tool will utilize the SCLK routing resources to reduce the delay and skew of this net.

The ECLK output pin drives both of the adjacent CLKCNTRL blocks (CLKCNTLT/CLKCNTLR or CLKCNTLL/CLKCNTLB). By using the CLKCNTL blocks, the PCM output clock has access to ECLK routing (for high-speed I/O clocking along the side adjacent to this PCM) and FCLK routing (for high-speed, low-skew, chipwide clock distribution).

Implementation Issue. When using the ECLK output of the PCM, it is not required to instantiate the adjacent CLKCNTL library elements. *ORCA* Foundry automatically activates these blocks and makes the proper placement and routing connections. If the clock shutoff feature of the CLKCNTL block is required, then the components must be instantiated into the design.

The ECLK output of the upper right PCM connects to the CLKCNTLR and CLKCNTLT blocks. The lower left PCM connects to the CLKCNTLL and CLKCNTLB blocks. (See Figure 1.)

Programmable Clock Manager

Lock Output

The LOCK signal is a filtered, active-high signal indicating when the PCM input clock and feedback clock are aligned.

Implementation Issue. Although the LOCK signal is filtered, it must be integrated over time to prevent a false lock indication during acquisition. This integrator circuit is shown in Figure 2 and should be used if the designer is required to implement LOCK functionality.

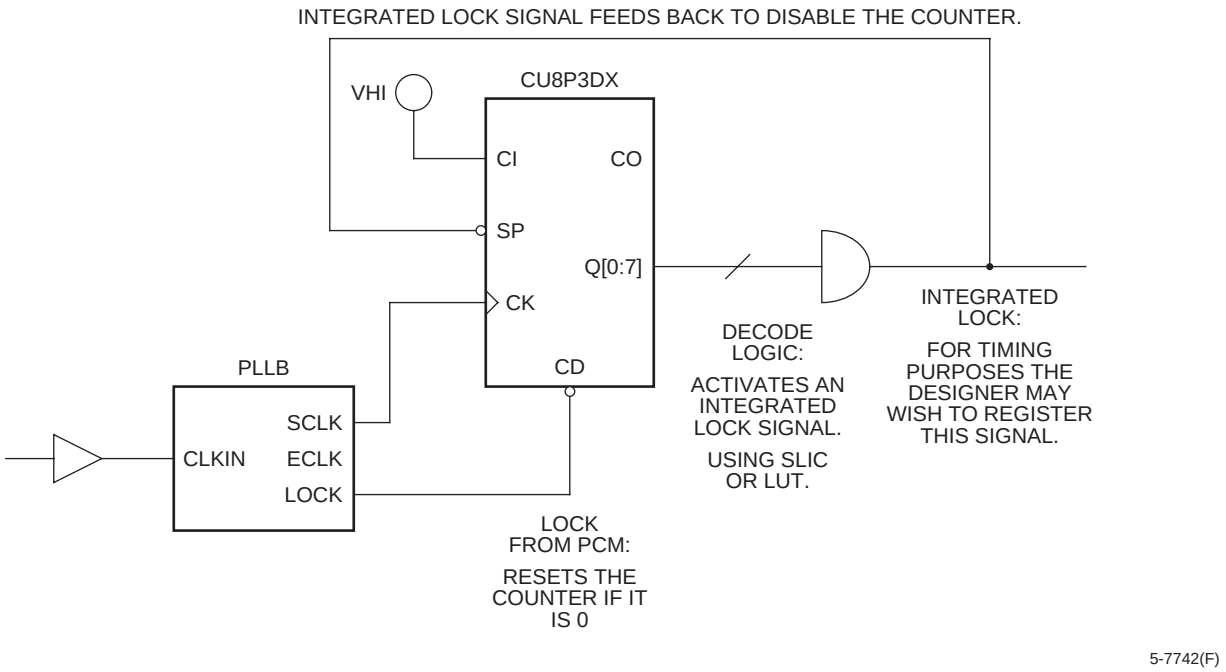


Figure 2. LOCK Integrator Circuit

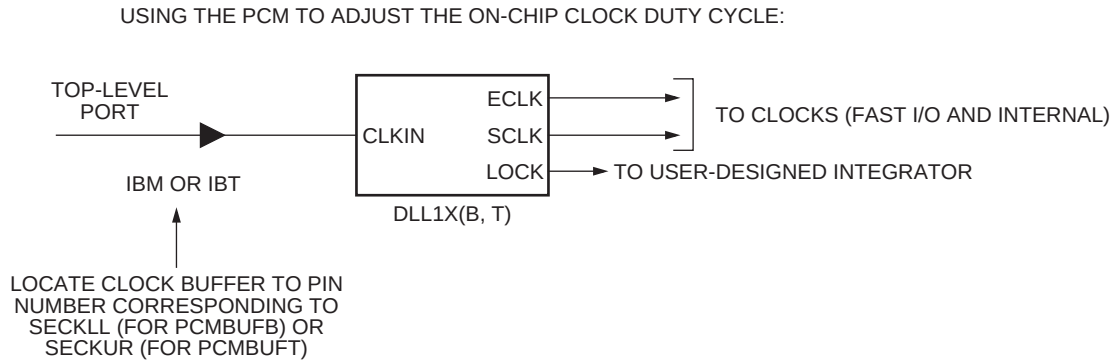
Feedback

When the PCM is configured as a DLL, the feedback clock is routed internally within the PCM. When the PCM is configured as a PLL, the feedback path is a dedicated connection from the adjacent CLKCNTL to the PCM. The PCM does have the ability to use a user-defined feedback source. As of *ORCA* Foundry 9.35, this feature is not implemented in the *ORCA* Foundry software and is planned for a future release.

1X Clock Duty-Cycle Adjustment (DLL1X Mode)

A duty-cycle adjusted replica of the input clock can be constructed in DLL1X mode. The duty cycle can be adjusted in 1/32 increments of the input clock period (3.125% increments of the output frequency).

Implementation via *ORCA* Foundry



5-7530(F)

Figure 3. DLL1X Library Cell

Library Element: DLL1XT or DLL1XB (T indicates upper right PCM, B indicates lower left PCM.)

Pins

(CLKIN, ECLK, SCLK, LOCK) This information can be found on pages 2 and 3.

Attributes (FREQUECNY, DUTY)

There are two attributes that must be assigned to the instance of this element, they are FREQUENCY and DUTY.

FREQUENCY

An attribute of type string that specifies the output frequency of the SCLK and ECLK pins (in MHz). This attribute is required.

DUTY

An attribute of type string ranging from 1—100. It specifies the duration (in terms of % of the input FREQUENCY) that the output clock signals (ECLK and SCLK) are to remain high during a full period. The default value for DUTY is 50.

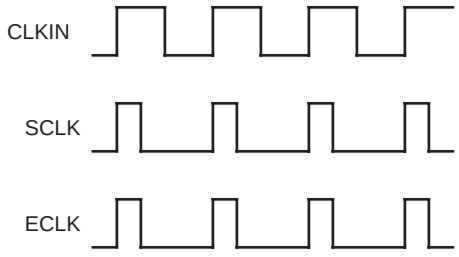
Example: (DLL1X)

DLL1XT
FREQUENCY = 20
DUTY = 25

The top PCM will be configured in DLL1X mode. The user will provide a 20 MHz clock signal as input. It will output a clock signal on ECLK and SCLK with a 20 MHz frequency and 25% duty cycle (see Figure 4).

The *ORCA* Foundry map tool will generate a period preference in the file <design>.prf that describes this signal:

PERIOD NET sclk 50.000000 ns HIGH 12.500000 ns;
PERIOD NET eclk 50.000000 ns HIGH 12.500000 ns.



5-7526(F)

Figure 4. DLL1X

Delayed Clock (DLLPD Mode)

A delayed version of the input clock can be constructed in DLLPD mode. The output clock can be delayed by increments of 1/32 of the input clock period. In this mode, the ECLK output can also be divided by an integer from 1 to 8. (The SCLK frequency will not change.)

The PCM should not be used in DLLPD mode to compensate for delay caused by internal logic. It is commonly known that logic delays for CMOS devices can vary as a result of changes to voltage, temperature, and process. The delay used in DLLPD mode is fixed as a percentage of the input clock and therefore would not properly track logic delay as environmental conditions change. This could cause unanticipated race conditions in a design, potentially changing functionality without the designer's knowledge. The preferred method of delay compensation is the PLL mode, which will be discussed later.

Implementation via *ORCA* Foundry

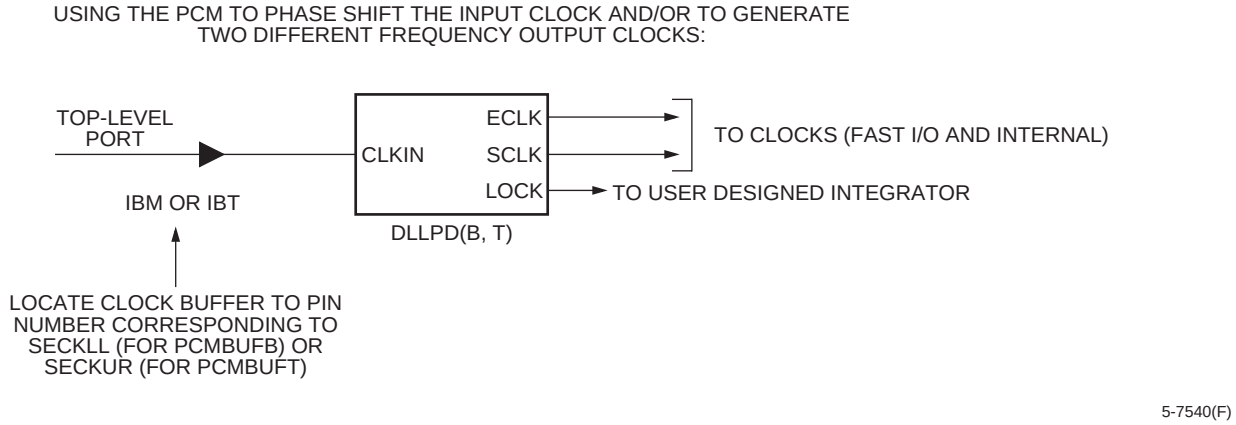


Figure 5. DLLPD Library Cell

Library Element: DLLPDT or DLLPDB

Pins

(CLKIN, ECLK, SCLK, LOCK) This information can be found on pages 2 and 3.

Attributes (FREQUENCY, DIV2, PDELAY)

There are 3 attributes that can be assigned to this element in the design, they are FREQUENCY and DUTY.

FREQUENCY

An attribute of type string that specifies the input frequency of CLKIN pin (in MHz). This attribute is required.

DIV2

An attribute of type string ranging in integer values from 1—8. It specifies the amount that the ECLK FREQUENCY will be divided down from the original CLKIN FREQUENCY. The SCLK frequency will be the same as the CLKIN frequency and will not be affected by this attribute. The default value for DIV2 is 1.

ECLK frequency = CLKIN FREQUENCY/DIV2

PDELAY is an attribute of type string ranging in integer values from 1—32. It specifies the amount of delay that is added to the output clock signals relative to the CLKIN period. The actual delay that is introduced is based on the following calculation:

delay in ns = CLKIN PERIOD x (PDELAY/32)

The delay is added to both the SCLK and ECLK signals, which insures that they remain in phase with each other.

Example: (DLLPD)

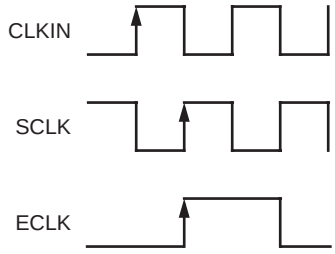
DLLPDB
FREQUENCY = 20
PDELAY = 16
DIV2 = 2

The bottom PCM will be configured in delayed clock mode. The user will be provided an input frequency of 20 MHz. The output frequency for ECLK will be 10 MHz and SCLK will be 20 MHz. Both of these signals will be delayed by 16/32 of the CLKIN frequency (see Figure 6).

Implementation via *ORCA* Foundry (continued)

The *ORCA* Foundry map tool will generate period preferences in file <design>.prf that describe these signals:

PERIOD NET sclk 50.000000 ns LOW 25.000000 ns HIGH 25.000000 ns;
PERIOD NET eclk 100.000000 ns LOW 25.000000 ns HIGH 50.000000 ns;



5-7528(f)

Figure 6. DLLPDB

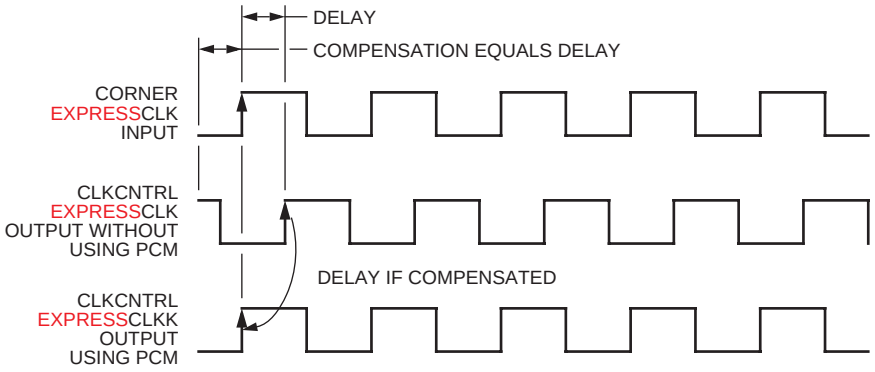
Phase-Locked Loop Mode (PLL Mode)

Clock Multiplication/Division

Output clocks that are multiples of the input clocks can be generated by the PCM in PLL mode. The multiplication is based on three attributes and can range from 1/8x to 64x the input clock frequency. When clock multiplication is performed using PLL mode, the phase of the output clocks are controlled by the clock delay minimization circuitry. This ensures that the output clocks are phase-aligned with the input clocks.

Clock Delay Minimization

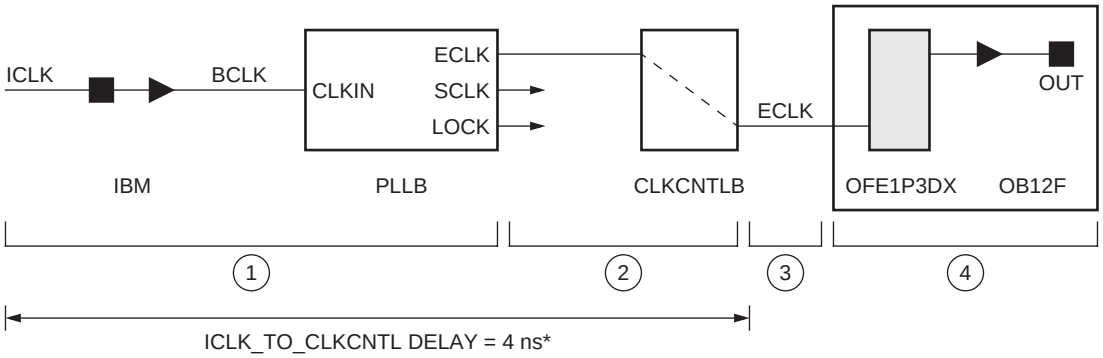
There is a dedicated feedback path from an adjacent CLKCNTRL block to the PCM. For the top PCM, the feedback path comes from the CLKCNTRL block on the top edge of the device, and for the bottom PCM, this feedback path comes from the CLKCNTRL block on the bottom edge of the device. Using the corner Express CLK pad (SECKUR or SECKLL) as the input to the PCM and using this dedicated feedback path, the clock from the ECLK output of the PCM, as viewed at the CLKCNTRL block, will be phase-aligned with the input of the SECKUR or SECKLL I/O buffer, effectively removing the I/O buffer and most of the clock routing delays from the clock network. These relationships are diagrammed below and an example is shown in Figure 8.



5-7529(F)

Figure 7. Clock Delay Minimization

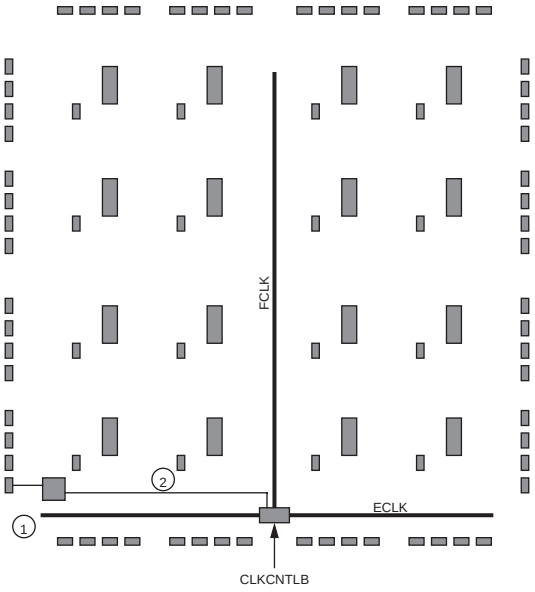
Phase-Locked Loop Mode (PLL Mode) (continued)



* This delay varies with changes to the temperature and voltage; the dedicated feedback paths allow this delay to adjust itself.

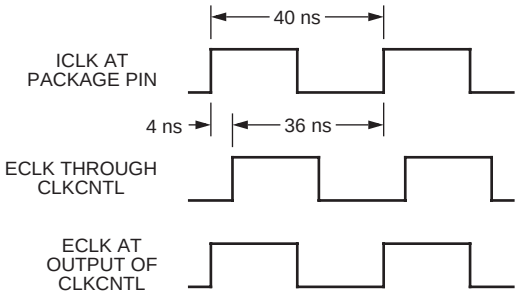
5-7537(F)

A. Schematic



5-7536(F)

B. PLL Clock Input Delay Compensation



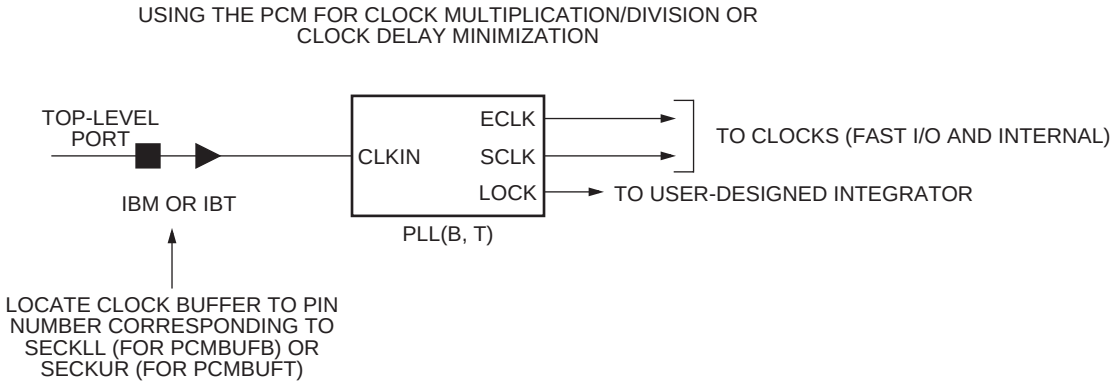
5-7538(F)

C. Timing Diagram

Figure 8. Clock Delay Minimization Example

Phase-Locked Loop Mode (PLL Mode) (continued)

Implementation via *ORCA* Foundry



5-7534(F)

Figure 9. PLL Library Cell

Library Element: PLLT or PLLB

Pins

(CLKIN, ECLK, SCLK, LOCK) This information can be found on pages 2 and 3.

Attributes (FREQUENCY, DIV[0:2])

There are four attributes that can be assigned to this element in the design.

FREQUENCY

An attribute of type string that specifies the input frequency of CLKIN pin (in MHz). This attribute is required.

DIV[0:2]

Attributes of type string ranging in integer values from 1—8. They specify the amount that the ECLK and SCLK FREQUENCIES will be changed from the original CLKIN FREQUENCY. The default value for each is 1.

$$\text{ECLK Frequency} = \frac{DIV1}{DIV0} \times \text{CLKIN Frequency}$$

SCLK Frequency = ECLK Frequency x DIV2

These formulas are based on the CLKCNTRL feedback clock path that is used by this mode of operation.

Example: (Clock Multiplication)

PLLB
FREQUENCY = 20
DIV0 = 2
DIV1 = 5
DIV2 = 2

The bottom PCM will be configured in PLL mode. The input frequency is 20 MHz. The output frequency for ECLK will be 50 MHz and SCLK will be 100 MHz (see Figure 10). Note that the *ORCA* PCM is extremely flexible and can generate output clocks that are not integer multiples of the input clocks.

The *ORCA* Foundry Map tool will generate frequency preferences in the file <design>.prf that describes these signals:

FREQUENCY NET sclk 100.000000 MHz;
FREQUENCY NET eclk 50.000000 MHz.

Phase-Locked Loop Mode (PLL Mode) (continued)

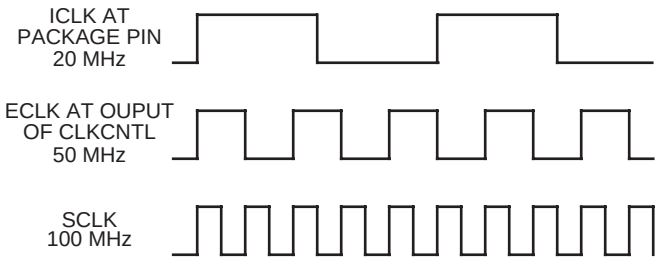


Figure 10. ICLK, ECLK, SCLK Timing Diagram

Example: (Clock Delay Compensation)

For clock delay compensation, we do not wish to perform any multiplication on the generated clock signals. The desired result, however, is to have an internal ECLK signal that is phase aligned with the clock signal on the input pin to the device. Because the feedback paths are automatically selected in PLL mode when the PLLT or PLLB library elements are used, the only requirements of the user are to instantiate the PLL element and locate the input clock buffer to the SECKLL or SECKUR pins of the device.

PLLB
Frequency = 20
DIV0 = 1
DIV1 = 1
DIV2 = 1

The bottom PCM will be configured in PLL mode. The input frequency and output frequencies are 20 MHz, and they will be phase aligned (see Figure 7).

Advanced Operation

The *ORCA* Series 3 PCM can implement a wide array of functions beyond what are implemented in the library elements. The PCM functionality is programmed using eight control registers. When the library elements are used as defined above, these registers are automatically programmed via the *ORCA* Foundry bit stream. This mode of operation in maintained until a new bit stream is downloaded to the *ORCA* Series 3 device.

The user also has the ability to read and write to these registers during the operation of the design. This allows the functionality of the PCM to be changed in real time. This functionality can also be easily controlled by connecting these registers to the MPI (microprocessor interface) inside the FPGAs. Access to this feature of the PCM, however, is only made available using the design tool EPIC (editor for programmable ICs). Future versions of *ORCA* Foundry will provide library elements for this purpose.

Please refer to the *ORCA* Series 3 data sheets for more information regarding the functionality of these registers. Also refer to the EPIC user guide in the *ORCA* Foundry documentation.

Device Reset

The PCM internal registers are not affected by the global set reset (GSR) or any other reset signal. Future support in *ORCA* Foundry will allow the user to enable or disable this feature.

Advanced Operation (continued)

Timing Information

More detailed information is found in the *ORCA* Series 3 data sheet (DS99-087FPGA).

Conclusion

This application note has discussed the details of implementing the *ORCA* programmable clock manager blocks into a design. It has covered the functionality of the different modes, as well as the library elements and attributes necessary to incorporate the functionality into a design. Detailed information on the PCM can be found in the *ORCA* Series 3 data sheet.

January 2003
AP99-015FPGA

Lattice® Semiconductor Corporation