APPLICATION NOTE

Xilinx has acquired the entire Philips CoolRunner Low Power CPLD Product Family. For more technical or sales information, please see: www.xilinx.com

XAPP 305

Understanding CoolRunner^{TM} clocking options

1998 Jul 16

Philips
Semiconductors

PHILIPS

Understanding CoolRunnerTM clocking options XAPP 305

UNDERSTANDING COOLRUNNERTM CLOCKING OPTIONS

The CoolRunnerTM family of CPLDs includes versatile clocking options that include both synchronous (external) and asynchronous (internal, equation-based) clocking and selectable clock polarity at every macrocell. This application brief describes in detail these clocking options, and shows how to access these features using Philips XPLA Designer. We also detail how to synthesize 'soft' flip-flops and latches for those instances where these devices can be useful.

XPLA Clocking Architecture

All CoolRunner devices provide multiple clock sources to each register of the device. These sources support both synchronous and asynchronous clocking. Each type of clock source has well defined capabilities, depending upon whether the clock is generated from a dedicated input, a multi-purpose I/O pin, or a logic block control term. The number of each type of clock varies with device density. Table 1 indicates the number of each type of clock found in various CoolRunner devices.

Dedicated Input Pin Clocks

The first type of clock source (clk0) is associated with a dedicated input pin. As Figure 1 shows, this input is attached directly to a low-skew, dedicated clock network. These sources can generate only synchronous (external) clocks to the associated clock network, and polarity of this clock is selectable at every macrocell. All of the CoolRunnerTM devices have one or more of these clock sources. It is worth noting that the input may be used as both the input to the associated clock network and as an input to the logic array (via the ZIA interconnect) at the same time. Thus this input can be used as both a clock and as a signal in the logic simultaneously. With regard to timing, the Tsu and Tco specs in the datasheet refer to synchronous clocks.

Table 1. Clock resources by device type

Part Number	Dedicated Input Pin Clocks	I/O Pin Clocks	Control Term Clocks	Total Clock Resources
PZx032–/I	clk0	clk1	0	2
PZx032C/N	clk0	clk1	2 x 2 logic blocks = 4	6
PZx064–/I	clk0	clk1, clk2, clk3	0	4
PZ3064A/D	clk0	clk1, clk2, clk3	2 x 4 logic blocks = 8	12
PZ5064C/N	clk0	clk1, clk2, clk3	2 x 4 logic blocks = 8	12
PZx128–/I	clk0	clk1, clk2, clk3	0	4
PZ3128A/D	clk0	clk1, clk2, clk3	2 x 8 logic blocks = 16	20
PZ5128C/N	clk0	clk1, clk2, clk3	2 x 8 logic blocks = 16	20
PZ3960C/N	clk0–clk7	0	2 x 48 logic blocks = 96	104

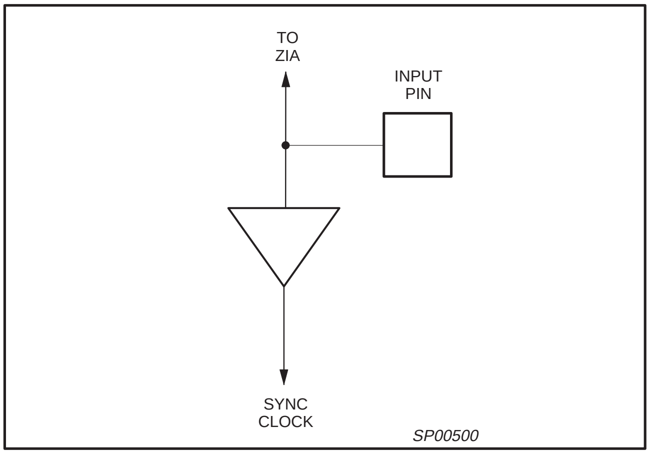


Figure 1.

I/O Pin Clocks

The second type of clock source comes from clock pins that are associated with I/O macrocells. There is one of these input types on the CoolRunner™32, and three of these on the CoolRunner™64 and CoolRunner™128 devices. These clock inputs have more versatility than the dedicated input type. In the synchronous clocking configuration (Figure 2), the output buffer is set to the High-Z state, and the I/O pin is propagated to the associated clock network and the logic array (via the ZIA). This behaves identically to the dedicated input clock in all respects. The Macrocell is still usable for internal 'buried' logic in this configuration. Configuration of the macrocell for buried logic and disabling the output buffer is automatically done by the design software.

The I/O pin clocks can also be used to generate asynchronous 'equation-based' clocks. Figure 3 shows that in this configuration the output buffer is enabled. Therefore, the logic that is generated in the macrocell is propagated to the associated clock network, the I/O pin, and is also fed back into the logic array. Since macrocells in the XPLA architecture can deploy as many as 37 Sum of Product equations, the resulting clocking equation in this configuration may be much more complex than in competing devices that have only a single product term available for asynchronous clocking. It also significant to note that the asynchronous clock that is generated is observable on the associated I/O pin. For this reason, the associated pin should not be terminated by tying to ground or V_{DD}. The timing for asynchronous clocks is different in that the Tco time is extended by the amount of time that it takes for the signal to propagate through the array and reach the clock network, and the Tsu time is reduced. This time is dependent on whether the PAL or PAL+PLA paths are used to generate the equation. For clock equations that use only the PAL path, Tco is extended by Tpdf_pal. Using the PAL+PLA path extends Tco by Tpdf_pla.

Control Term Clocks

The third type of clock source is provided by the control terms in each logic block. These types of clocks available only in the XPLA Enhanced and XPLA2 device families. Figure 4 shows the macrocell architecture for each of these families. In the XPLA Enhanced device family (Figure 4a), there are six control terms in each logic block, and in the XPLA2 device family (Figure 4b), there are eight control terms in each logic block. As shown for the XPLA Enhanced family, two of the six control terms are shared by the output enable multiplexer and the clock source multiplexer. These control terms can be used as either an output enable, a clock, or both. The XPLA2 family has two extra control terms that are dedicated to only the clock source multiplexer. These clocks can be individually configured as any PRODUCT term or SUM term equation created from the 36 signals available inside the logic block. If a SUM-OF-PRODUCTS equation is required, it must be implemented in a macrocell and then fed-back into a control term through the ZIA (see the application note Using Sum of Products Control Terms for more information).

Each control term clock is available to all the macrocells within a logic block, but it must be duplicated on another control term if the same clock is used in different logic blocks. These clocks are not attached to a low-skew clock network, and they must pass through the interconnect array and a single product term before reaching the flip-flop. Therefore, Tco time is extended by the amount of time that it takes for the signal to propagate through the appropriate array, and the Tsu time is reduced. Unlike the other two types of clock sources, control term clocks are not associated with specific pins and may be assigned to any I/O or dedicated input.

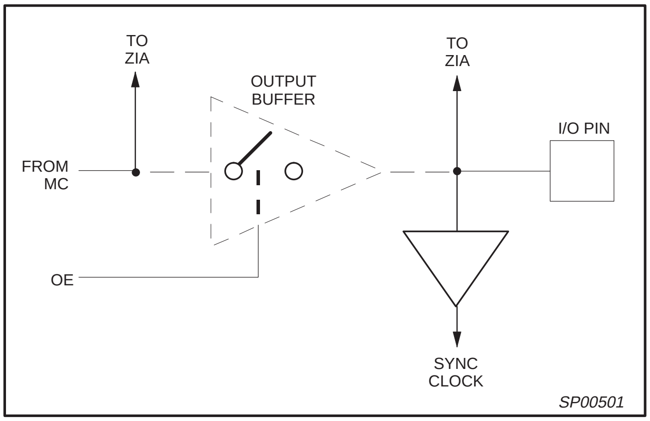


Figure 2.

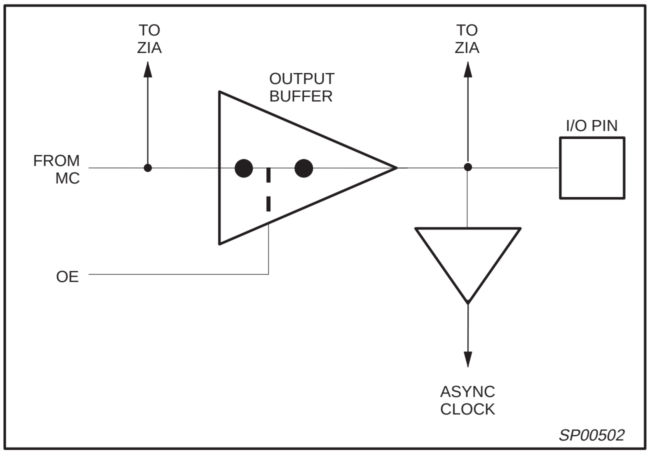
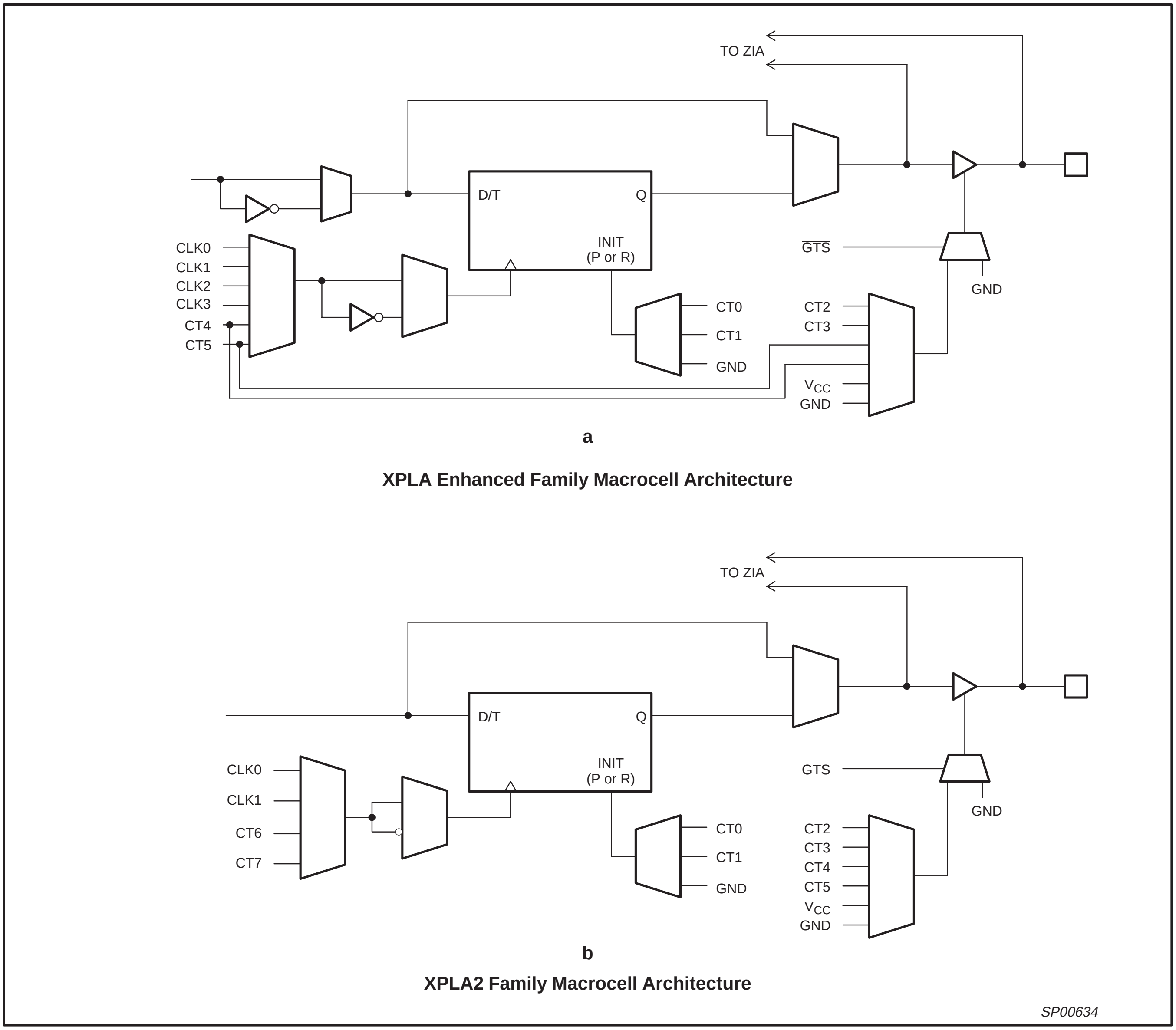


Figure 3.



a

XPLA Enhanced Family Macrocell Architecture

b

XPLA2 Family Macrocell Architecture

Figure 4. Macrocell Architectures

Using XPLA Designer to generate clocks

XPLA Designer will automatically assign clocks to the correct pins/macrocells based on the context of the clocking desired. Synchronous clocks are generated by declaring the name you want for the clock, and simply using this by itself in a **.clk** equation, as shown below for the clock signal we’ve created called CLOCK_1.

```
Module DEMO
Title ’A simple design:3-bit counter’

CLOCK_1        pin;
bit2..bit0     pin istype ’reg’;
count = [bit2..bit0];

equations
count.CLK = CLOCK_1;

count = count.q + 1;

end;
```

Asynchronous clocks are also easy to generate. Again, simply declare the variables that will make up the equation that the clock will be based upon. In the example below, we are generating a clock from the variables A, B, and C. Then in the **.clk** equation, we write any expression we want for the clock. The software will assign asynchronous clocks (product term or sum term) first to the control term clocks, until all available control term clocks are used. Then the design software will assign asynchronous clocks to a macrocell, enable the output buffer, and feed the clock equation to the register through the dedicated clock network as in Figure 3.

```
Module DEMO2
Title ’A simple design:3-bit asynch counter’

A,B,C          pin;
bit2..bit0     pin istype ’reg’;
count = [bit2..bit0];

equations

count.CLK = (A&B) # C;
count = count.q + 1;

end;
```

Soft Flip-Flops

For the rare cases where there are too few clocks in a CoolRunnerTM device to implement a large number of input registers (for example), soft D flip-flops or transparent latches may be useful. The following examples illustrate the generation of a transparent latch and D Flip-Flop using only the gates in the logic array. It is important to note that the ‘clock’ width **must** be longer than Tpd!

```
Module Soft_Latch
Title ’Soft Latch w/ Latch Enable –’

D              pin;
LE             pin;

/* This uses only one macrocell to implement a
transparent latch. It is level (not edge)
triggered – D hold must extend beyond LE’s
falling edge */

Q pin istype ’com,keep,retain’;

equations

Q = (D & LE) # (Q & !LE) # (D & Q);

end
```

```
Module Soft_D2
Title ’Soft D Flip Flop-Rising Edge triggered –’

/* This edge triggered soft flip-flop uses two
   macrocells. The Input latch opens when the
   clock is low, and closes when the clock is
   high. The output latch (Q) opens when the
   clock is high, and closes when the clock is
   low */

D              pin;
CLK            pin;
IL             node istype ’com,keep,retain’;
Q              pin istyp e’com,keep,retain’;

equations

IL = (D & !CLK) # (IL & CLK) # (D & IL);
Q = (IL & CLK) # (Q & !CLK) # (IL & Q);

end
```

Definitions

Short-form specification — The data in a short-form specification is extracted from a full data sheet with the same type number and title. For detailed information see the relevant data sheet or data handbook.

Limiting values definition — Limiting values given are in accordance with the Absolute Maximum Rating System (IEC 134). Stress above one or more of the limiting values may cause permanent damage to the device. These are stress ratings only and operation of the device at these or at any other conditions above those given in the Characteristics sections of the specification is not implied. Exposure to limiting values for extended periods may affect device reliability.

Application information — Applications that are described herein for any of these products are for illustrative purposes only. Philips Semiconductors make no representation or warranty that such applications will be suitable for the specified use without further testing or modification.

Disclaimers

Life support — These products are not designed for use in life support appliances, devices or systems where malfunction of these products can reasonably be expected to result in personal injury. Philips Semiconductors customers using or selling these products for use in such applications do so at their own risk and agree to fully indemnify Philips Semiconductors for any damages resulting from such application.

Right to make changes — Philips Semiconductors reserves the right to make changes, without notice, in the products, including circuits, standard cells, and/or software, described or contained herein in order to improve design and/or performance. Philips Semiconductors assumes no responsibility or liability for the use of any of these products, conveys no license or title under any patent, copyright, or mask work right to these products, and makes no representations or warranties that these products are free from patent, copyright, or mask work right infringement, unless otherwise specified.

Philips Semiconductors
811 East Arques Avenue
P.O. Box 3409
Sunnyvale, California 94088–3409
Telephone 800-234-7381

print code Date of release: 07-98

Document order number: 9397 750 04148