XAPP335 (v1.0) April 17, 2000

Macrocell Configurations in CoolRunner XPLA3 CPLDs

Summary

This document describes the macrocell configurations of Xilinx CoolRunner XPLA™ CPLDs .

Introduction

Xilinx CoolRunner XPLA3 CPLDs provide designers with several useful configuration options for each macrocell. These options allow greater flexibility when creating complex designs. Some of the configurations available are: data register (D, T, and Latch), input register, buried combinatorial or registered node, and I/O port. Combinations of these configurations can be used to increase macrocell utilization.

In addition to better macrocell utilization, the XPLA3 input registers have a very short setup time. This feature is beneficial when data transfers between devices on a board must occur within one clock cycle.

A brief overview of the XPLA3 architecture as it pertains to macrocell configuration will be described in the following sections. The XPLA3 architecture will not be fully described in this application note. For more details on the XPLA3 architecture, please refer to white paper "WP105, XPLA3 CoolRunner CPLD Architecture Overview."

CoolRunner XPLA3 Architecture

From a high-level, the architecture of CoolRunner XPLA3 CPLDs appear to be similar to many other CPLD architectures. As shown in Figure 1, the XPLA3 architecture consists of logic blocks containing macrocells interconnected by a routing matrix. The routing matrix is called the ZIA (Zero-power Interconnect Array) and provides 36 true and complement signals to each logic block. Each XPLA3 logic block contains 16 macrocells.

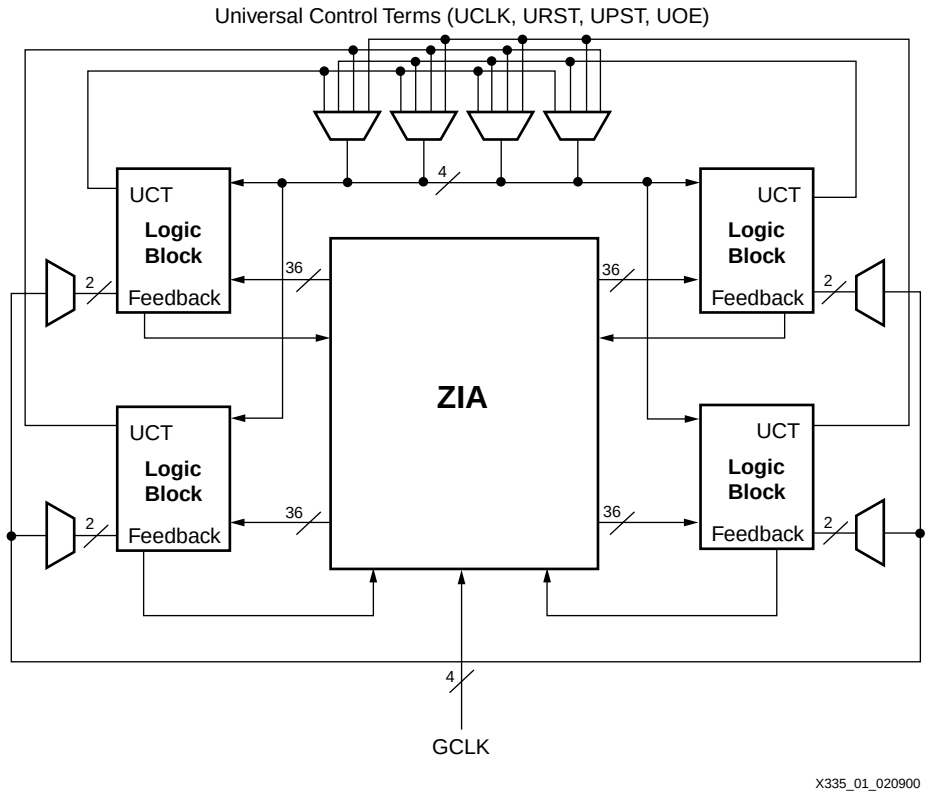


Figure 1: **XPLA3 High-level Architecture (64 macrocell device shown)**

The XPLA3 architecture is unique, however, in the fact that each logic block contains a pure PLA array (programmable AND, programmable OR). Each logic block can be further broken down to an I/O logic block (or Macrocell) and PLA logic block, as shown in Figure 2. The PLA array provides a pool of 48 product terms that can be used as macrocell clocks, to generate control terms (reset, preset, clock-enables, or output-enables), or as needed by the 16 macrocells in the logic block.

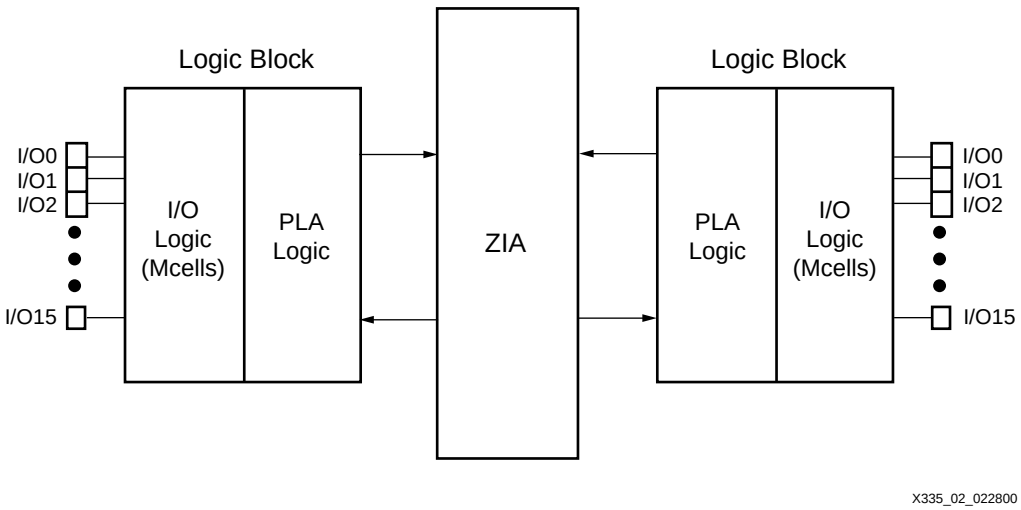


Figure 2: **XPLA3 Logic Block Architecture**

As shown in Figure 3, the macrocell provides a Variable Function Multiplexer (VFM), data register, I/O port and other multiplexers (muxes) for routing control terms within the macrocell. By using the control muxes within the macrocell, the macrocell can be used in several different configurations.

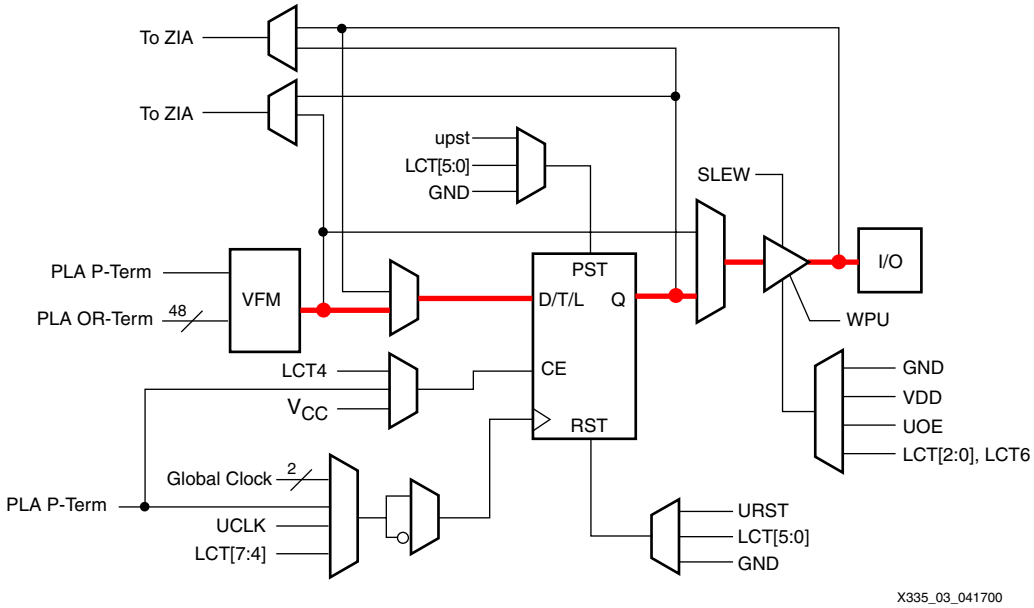


Figure 3: **XPLA3 Macrocell Structure**

Basic Register Implementation

The data input to each macrocell register is derived from the output of the VFM. Each macrocell register can be configured as a D, T, or Latch type flip-flop and output to the I/O port as outlined in Figure 3. The type of register to be used can be specified either in a schematic capture design or Hardware Description Language (HDL) file. These registers have a data input, a clock or latch enable input, and register output. Each register is also capable of using both asynchronous reset and preset control terms. The output of each macrocell register can either be fed back into the ZIA for use within the CPLD or sent to an I/O port.

D-Type Registers

The D-type register is the most commonly used type of register and is used to clock data through a design or to temporarily store data. D-type registers are edge triggered registers, or only accept new data on the rising edge of a clock signal. In addition, this register may also be configured to accept data with the falling edge of a clock signal instead of the rising edge. At all other times, the output of the register remains unchanged by the value of the input signal, as seen in Figure 4. In addition, CoolRunner XPLA3 D-type registers, as well as T-type and Latch, are capable of using both asynchronous reset and preset signals. With the D-type registers, the order of precedence with respect to reset and preset signals is listed below. Note that the reset and preset signals are independent of a clock signal.

- If Reset = 1 and Preset = 0 then Output = 0.
- If Reset = 1 and Preset = 1 then Output = 0.
- If Reset = 0 and Preset = 1 then Output = 1.
- If Reset = 0 and Preset = 0 and rising/falling edge of clock then Output = Input.
- If Reset = 0 and Preset = 0 and not rising/falling edge of clock then Output = Output.

In designs specified by an HDL file where the type of register to be used is not explicitly specified, the D-type register is most commonly used by default.

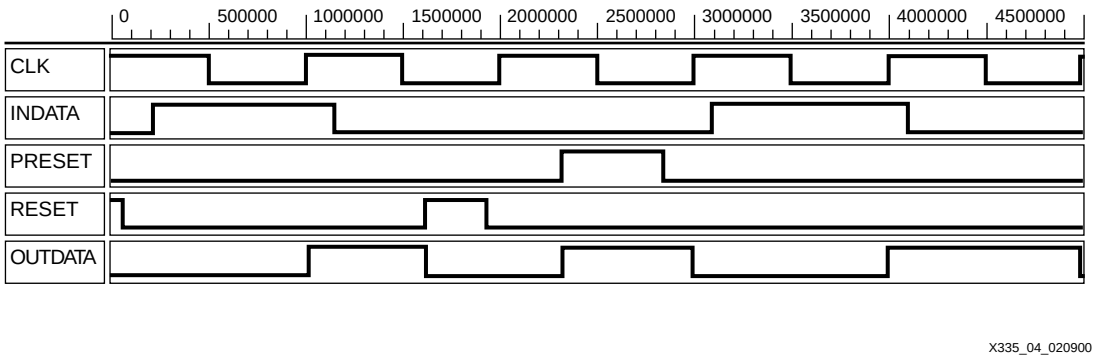


Figure 4: **D-type Register Characteristics**

T-Type Registers

The T-type register, or Toggle register is commonly used when generating counters and clock dividers in a design. As with the D-type register, the output signal may only change with either the rising or falling edge of a clock signal, depending on configuration. The output of a T-type register, however, is not only dependent on the input value but also dependent on the current value stored in the register. When the input is high, or asserted, the output will toggle between a high and low state with each rising/falling clock edge. When the input is low, or unasserted, the output will not toggle and therefore remain in its current state as seen in Figure 5. The T-type register is also capable of using the asynchronous reset and preset control signals. The reset and preset signals function in the same order as in the D-type register.

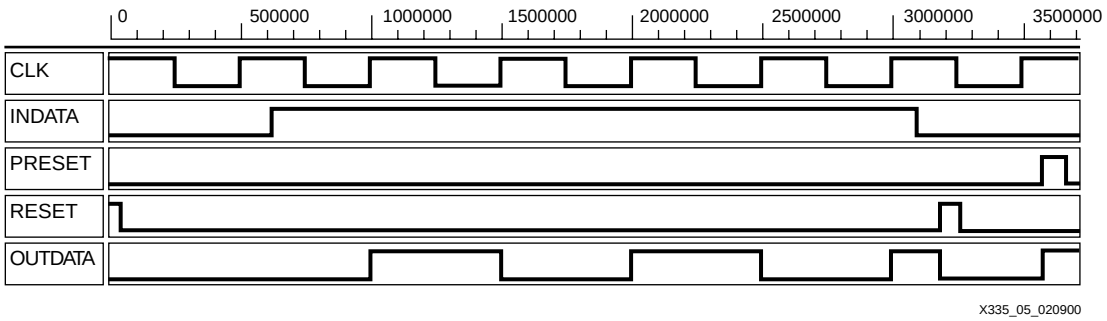


Figure 5: **T-type Register Characteristics**

One of the most common uses of T-type registers is with counters. T-type registers allow for significant product term savings when generating counters. For example, the code in Figure 6 demonstrates the implementation of a 32-bit counter in ABEL, a hardware description language. If the counter were implemented with D-type registers, the compiled design equations would use a total of 41 macrocells and 92 product terms. If however, this counter was implemented using T-type registers, the design would only use 32 macrocells and 32 product

terms. As one can see, there is a significant savings when using T-type registers rather than D-type registers in counters.

```
MODULE counter

DECLARATIONS
clk              pin;
data31..data0    pin istype 'reg';
data = [data31..data0];

EQUATIONS
data.clk = clk;
data.d = data.q + 1;
END
```

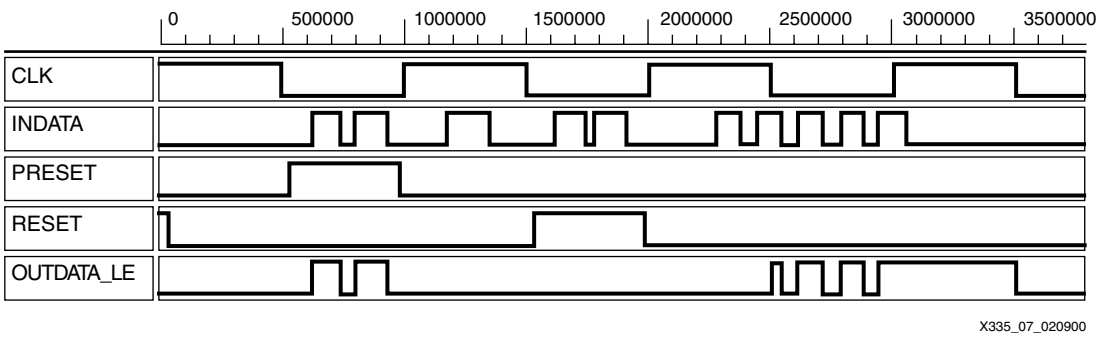
X335_06_020900

Figure 6: **32-bit Counter Implementation with ABEL**

Latch Registers

The Latch-type register used in XPLA3 devices is a true latch and is not emulated combinatorially. This is beneficial since an emulated latch requires two macrocells for implementation whereas the latch in XPLA3 devices only requires one. This allows designers the save macrocells when creating designs that use latch registers, such as an address encoder/decoder that interfaces with a bus. Each Latch register typically has a data input, latch enable, and an output. Note that when the macrocell register is configured as a Latch, the Latch Enable signal is input to the register via the clock signal port in Figure 3. However, the Latch register does not require a clock signal explicitly, although the Latch Enable can be assigned to a clock signal if necessary. The output of a Latch is controlled by the Latch Enable signal. While the Latch signal is Low, any changes in the input are directly sent to the output signal. When the Latch Enable is High, the output reflects the last state that input was in, as seen in Figure 7. Note, however, that the Latch Enable function can be reversed if desired. In other words, the latch can be set to accept data when the Latch Enable signal is High and remain locked when the signal is Low. The order of precedence with Latch registers, listed below, is slightly different than with D and T-type registers.

- If Reset = 1 and Preset = 0 then Output = 0.
- If Reset = 1 and Preset = 1 then Output = 0.
- If Reset = 0 and Latch Enable = 1 then Output = Input.
- If Latch Enable = 0 and Preset = 1 then Output = 1.
- If Latch Enable = 0 and Preset = 0 then Output = Output.



X335_07_020900

Figure 7: **Latch-type Register Characteristics**

Input Register Implementation and Benefits

CoolRunner XPLA3 macrocells can also be configured as input registers. As seen in Figure 8, the input from the I/O port is routed directly to the register instead of through the ZIA and corresponding PLA block. When implementing an input register, this type of configuration has two significant benefits. First, this register configuration has a very fast setup time of 2 ns. This is useful when performing data transfers between chips on a board in which only one clock cycle is available for completing the transfer. In most circuits with a setup time of 2 ns, data being transferred has enough time to stabilize before the clock cycle has ended. Furthermore, by reducing the setup time of the register, one can also reduce the clock period used by the design and therefore increase the clock frequency as seen in Figure 9.

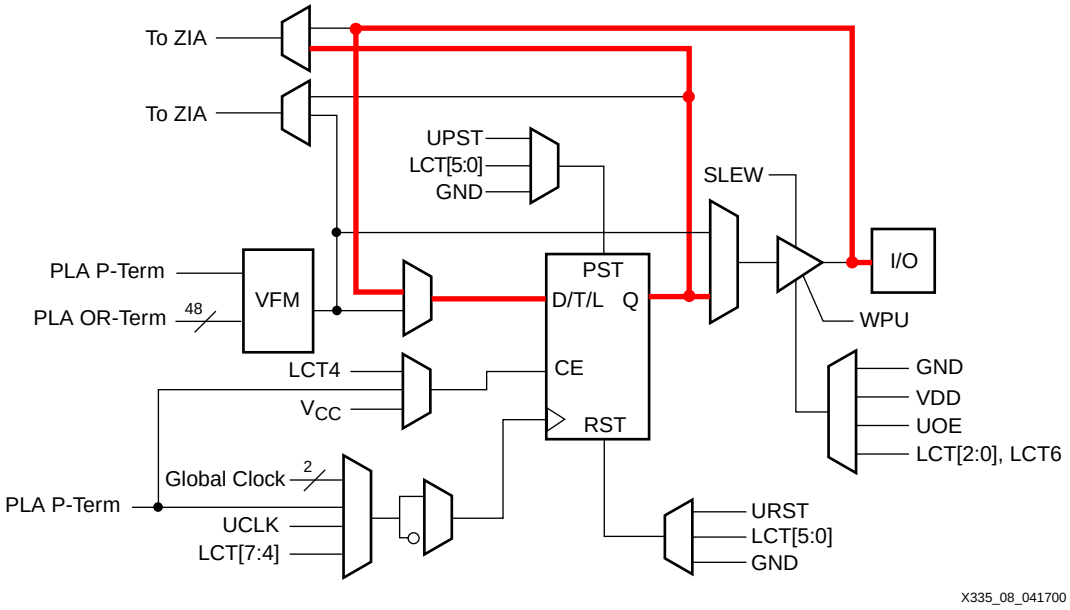


Figure 8: **XPLA3 Input Register Path**

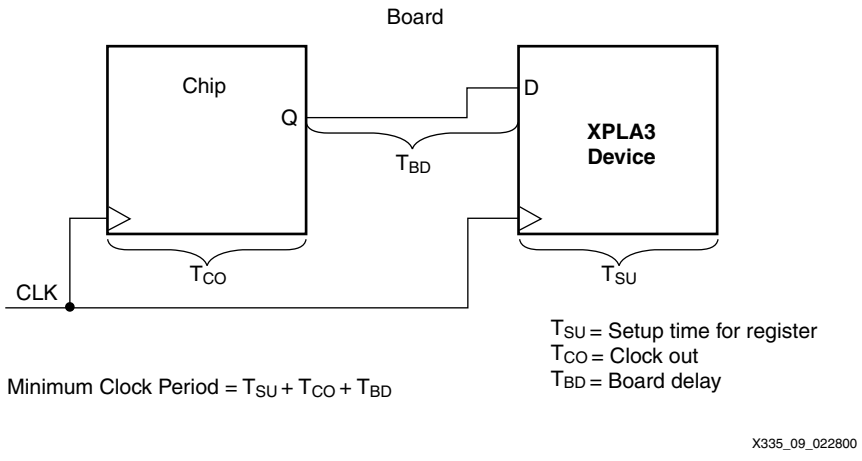


Figure 9: **Input Register Setup Time Example**

The second benefit of this input register configuration allows the preceding PLA logic and VFM to still be used, if needed, by the design. As highlighted in Figure 10, a usable data path still exists to the ZIA, thus allowing the available combinatorial logic in the preceding PLA and VFM

to still be used by the designer. This configuration, referred to as a buried combinatorial node, can significantly increase the utilization efficiency of each logic block.

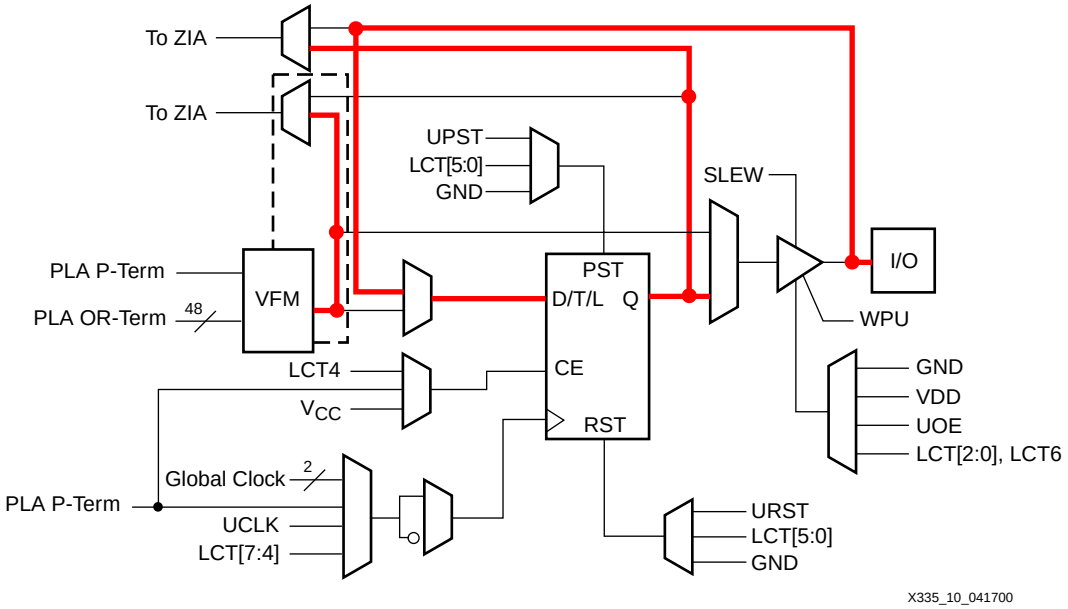


Figure 10: **XPLA3 Input Register Path with Buried Combinatorial Node**

Other Macrocell Configurations

The architecture of the XPLA3 macrocell is very flexible and thus can be used in several configurations. For example, the output from a D-type register can be routed back to the ZIA rather than output to an I/O port. This is referred to as a buried registered node. In this configuration, the I/O port can still be used, if desired, as outlined in Figure 11.

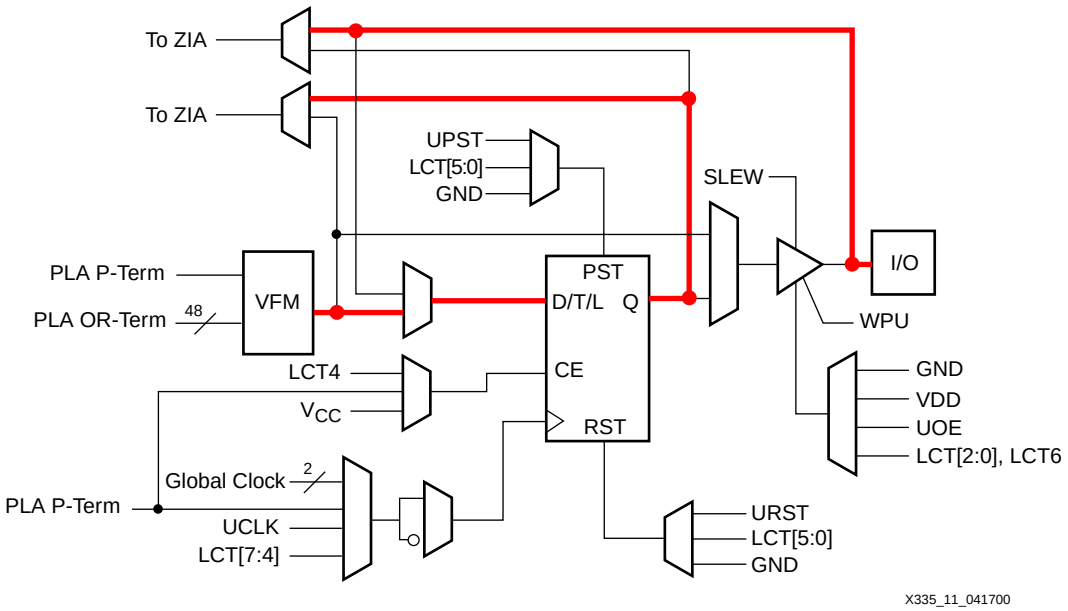


Figure 11: **Buried Registered Node with I/O Port Used**

The macrocell may also be used to output the results of combinatorial logic. In this configuration, the macrocell register is bypassed using a mux and output directly to an I/O, port as seen in Figure 12.

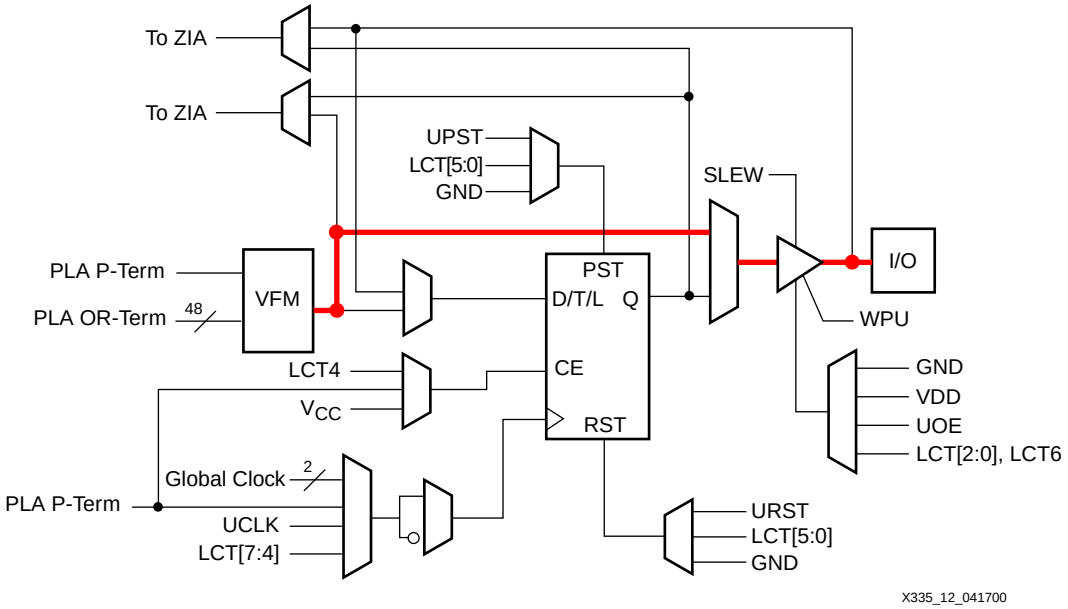


Figure 12: **Combinatorial Output**

Conclusion

The highly configurable macrocell architecture found in CoolRunner XPLA3 CPLDs greatly increases the efficiency of resource utilization within the device. This flexible architecture allows designers to create increasingly more complex designs with CoolRunner XPLA3 CPLDs without necessarily increasing the number of logic blocks used. With the short setup time of macrocells configured as input registers, CoolRunner XPLA3 CPLDs can also easily be used for data transfers between devices.

Revision History

The following table shows the revision history for this document.

Date	Version	Revision
04/17/00	1.0	Initial Xilinx release.