



## Pin Locking in CoolRunner XPLA3 CPLDs

XAPP332 (v1.0) January 7, 2000

Author: Anita Schreiber

### Summary

This document highlights the architectural features provided with CoolRunner CPLDs that enable pin assignments to be maintained through many design iterations.

### Introduction

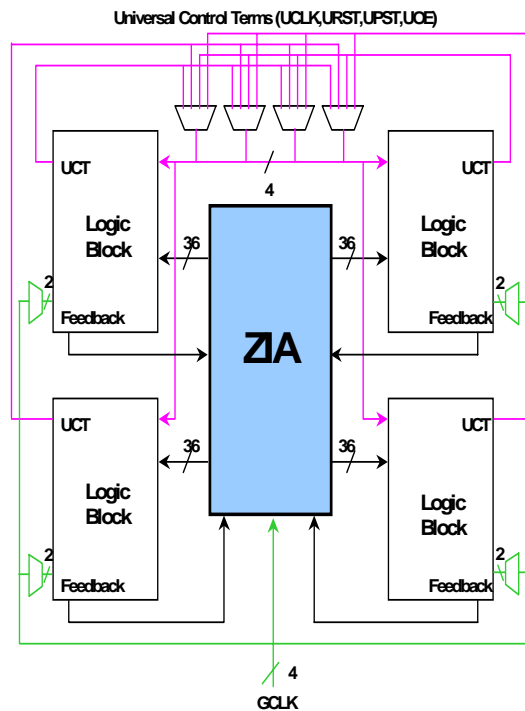
Today's In-System Programmable (ISP) CPLDs provide designers two main advantages. One of the greatest advantages is the ability to make design changes without removing the device from the board. These design changes may be necessary to fix bugs in the design, add additional features to the design, or to download a test design to exercise and debug the system. This advantage, however, requires that the architecture of the CPLD supports maintaining the pinout of the device if internal logic changes.

Another advantage of ISP CPLDs is the ability to achieve faster time to market. It is now becoming necessary to get an initial pin definition of the CPLD so that the board layout can progress in parallel with the detailed CPLD design. To achieve this benefit, the architecture of the CPLD must support internal design changes while maintaining the user-specified pinout.

Xilinx CoolRunner XPLA3 CPLDs have superior architectural features insuring designers that changes can be made while maintaining the device pinout. These features will be discussed in the following sections.

### CoolRunner XPLA3 Architecture

From a high-level, the architecture of CoolRunner XPLA3 CPLDs appears 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. Each XPLA3 logic block contains 16 macrocells. The routing matrix is called the ZIA (Zero-power Interconnect Array) and provides 36 true and complement signals to each logic block.



**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) 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. In addition, each XPLA3 logic block provides eight Fold-back NANDs to provide efficient optimization of logic.

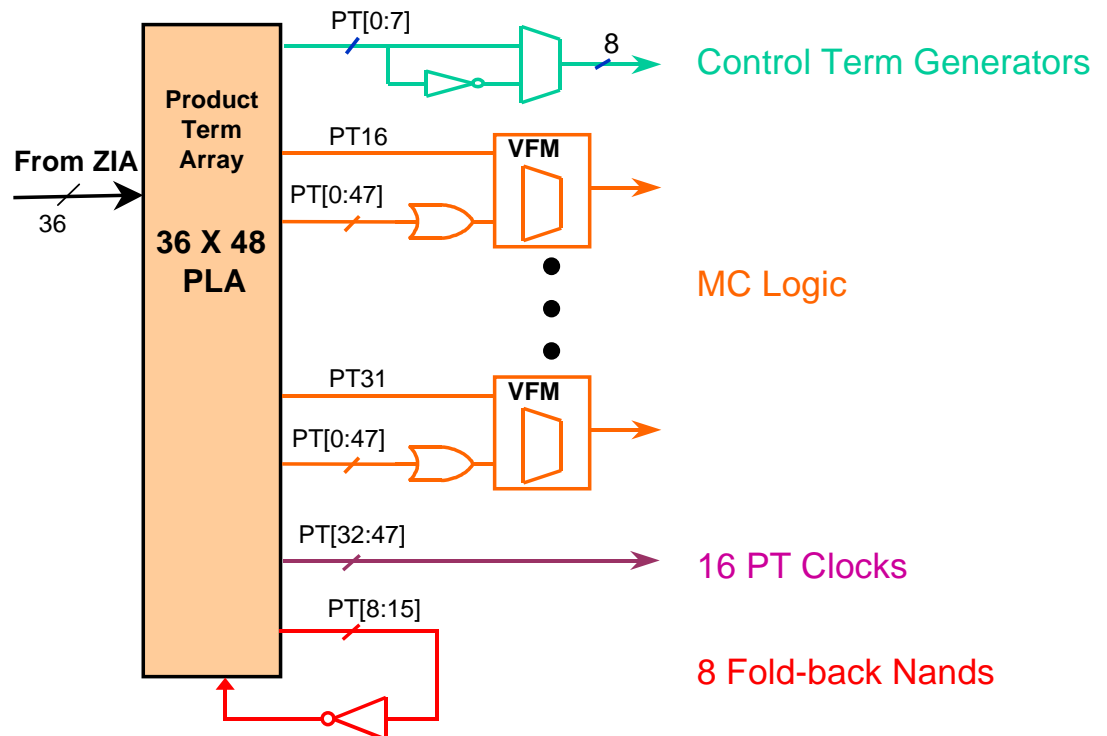


Figure 2: XPLA3 Logic Block Architecture

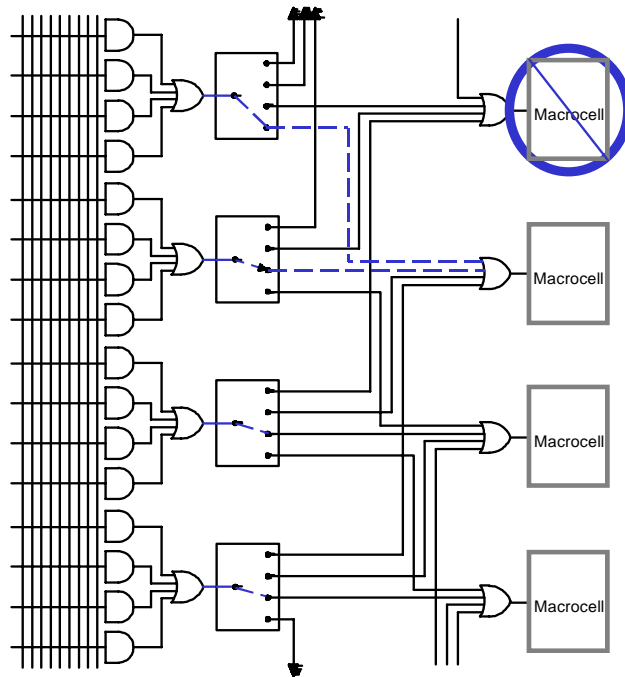
More detailed information about the CoolRunner XPLA3 architecture can be found in the Xilinx white paper, [WP105, titled "CoolRunner XPLA3 CPLD Architecture Overview"](#).

## Full PLA vs. Product-term Steering

Each logic block in a CoolRunner XPLA3 CPLD contains a PLA array which provides a pool of 48 product terms that can be used as needed by one or all 16 of the macrocells in the logic block. None of the product terms are dedicated, therefore, product terms that are not used for control terms or macrocell clocks are available for use by the macrocell logic.

The PLA array provides several advantages over the product-term steering approach used by many other CPLDs. Product-term steering dedicates a certain number of product terms to each macrocell in the logic block. Steering mechanisms are used that allow a macrocell's product terms to be steered to adjacent macrocells when needed. The PLA array is a programmable AND, programmable OR structure, therefore all product terms in the array are available to all of the macrocells in the logic block; there are no dedicated product terms. An additional advantage is that product terms that are common to multiple macrocells in the logic block can be implemented once and shared by all macrocells in the logic block.

Figure 3 illustrates a product-term steering type of logic-block architecture. Note that this logic-block architecture is not used in XPLA3 devices. In this figure, each macrocell has four dedicated product terms. When a macrocell needs additional product terms, the product terms from an adjacent macrocell are steered to this macrocell. This macrocell whose product terms were re-directed may now be stranded and un-usable.

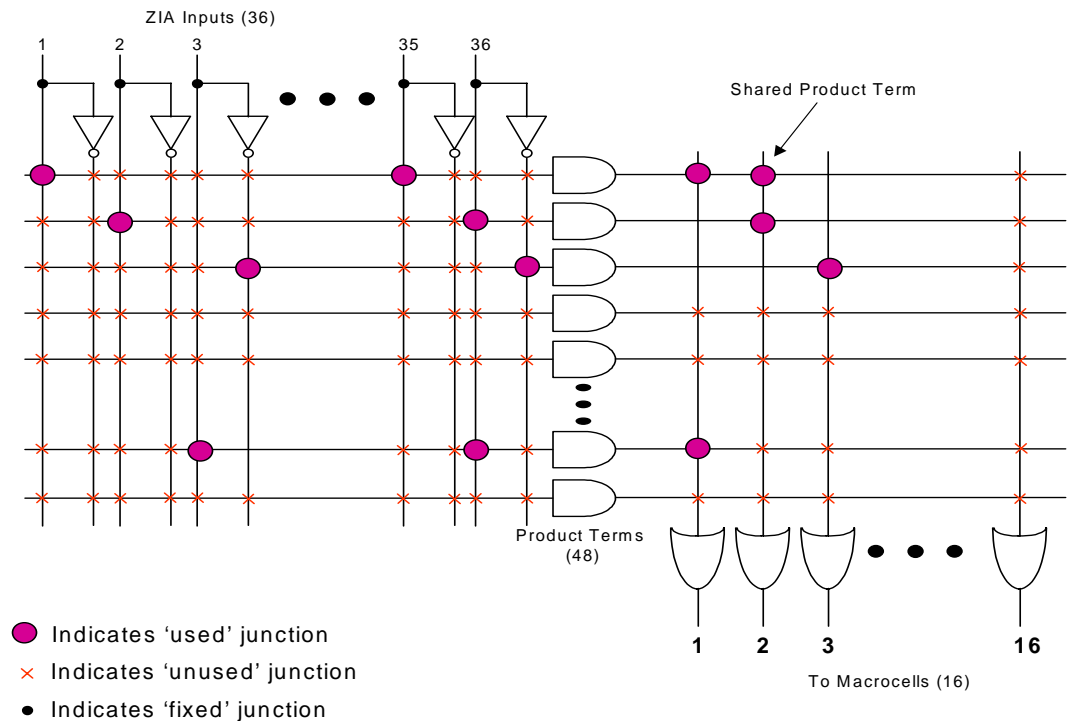


**Figure 3: Logic Block Architecture with Product-term Steering**

Consider the case where a design has been completed and the board-level debugging has begun. It is determined that a design change is necessary in the CPLD logic and this design change requires that a certain macrocell now requires seven product terms. If the CPLD logic block architecture is similar to that shown in [Figure 3](#) and the surrounding macrocells are currently utilized as outputs, this design change can not be implemented.

Taking this example further, if a design change is such that the pinout can be maintained, there is a high probability that the timing of the design will change as additional product terms are steered to implement the new logic.

If, however, this design is targeted to a CoolRunner XPLA3 CPLD, unused product terms in the PLA can be used to implement this design change **WITHOUT** affecting any of the macrocells in the logic block or the device pinout as shown in [Figure 4](#). All product terms in the PLA are available to each macrocell in the logic block, therefore, the pinout can be maintained. An additional advantage of the PLA structure is that product terms that are common to multiple macrocells in the logic block are implemented once and shared.



**Figure 4: XPLA3 Logic Block with Pure PLA Array**

CPLD manufacturers who implement product-term steering will advise designers to move to the next larger macrocell count device if their current device utilization is greater than 80% so that pinout can be maintained if design changes are necessary. Note that this recommendation means that a design using 102 macrocells of a 128 macrocell device should be implemented in the next larger CPLD (typically a 256-macrocell device) if there is a possibility of needing to implement design changes! Because of the high probability of implementing design changes due to debug, feature additions, or system test, this recommendation can significantly increase the cost of a system and is a waste of silicon that has already been purchased.

Note that the key to maintaining a fixed pinout is not only the efficiency of the routing matrix but how the product terms are allocated within the logic block. With the PLA structure implemented in CoolRunner XPLA3 CPLDs, fixed pinouts are maintained after logic changes even at device densities of > 99%. The CoolRunner XPLA3 CPLD architecture is optimized for ISP!

## Logic Block Fan-in

The fan-in to each logic block in a CoolRunner XPLA3 CPLD from the ZIA is advertised as 36. However, the architecture of the XPLA3 CPLD actually provides 40 routing channels to each logic block. The software defaults to using a logic block fan-in of 36 and can utilize any 36 of the 40 fan-in to the logic block, i.e., the 36 routing channels utilized by the software are not dedicated. These four extra fan-in signals are reserved and can be enabled in software when necessary.

## Conclusion

Many CPLDs claim to support pin-locking and ISP, however, few provide the architecture to really support these necessary features. CoolRunner XPLA3 CPLDs truly support ISP by providing flexible allocation of both logic and routing resources that allow for design changes with a fixed pinout.

---

---

## Revision History

Date	Version #	Revision
01/07/00	1.0	Initial Xilinx release.

© 1999 Xilinx, Inc. All rights reserved. All Xilinx trademarks, registered trademarks, patents, and disclaimers are as listed at <http://www.xilinx.com/legal.htm>. All other trademarks and registered trademarks are the property of their respective owners.