CoolRunner XPLA3 CPLD Architecture Overview

WP105 (v1.0) January 6, 2000

Author: Reno Sanchez

Summary

This document describes the CoolRunner™ XPLA3 CPLD architecture.

Introduction

This document describes the CoolRunner XPLA3 (eXtended Programmable Logic Array—third generation) CPLD architecture and compares it to existing CPLD architectures. Before the XPLA3 architecture is examined, a brief description of CoolRunner CPLD families is given.

CoolRunner Family Description

The CoolRunner CPLD families are the world's only CPLDs using the patented Fast Zero Power (FZP™) design technique to simultaneously deliver high performance and low power consumption. These devices offer pin-to-pin (T_{PD}) delays of 5 ns (greater than 200 MHz system operation), less than 100 µA of standby current and approximately 1/3 of the total power consumed by all non-CoolRunner CPLDs at maximum frequency (f_{MAX}).

These characteristics make CoolRunner devices ideal for low power applications including portable, handheld, and power-sensitive applications. These devices are also ideal for systems that have a strict power or thermal budget, a need for increased system reliability (less power means less activation energy and lower FIT rates), a need for lower costs (by eliminating or reducing the system cooling requirements), or a need for reduced power supply requirements or extended battery life. Figure 1 illustrates the CoolRunner CPLD familes.

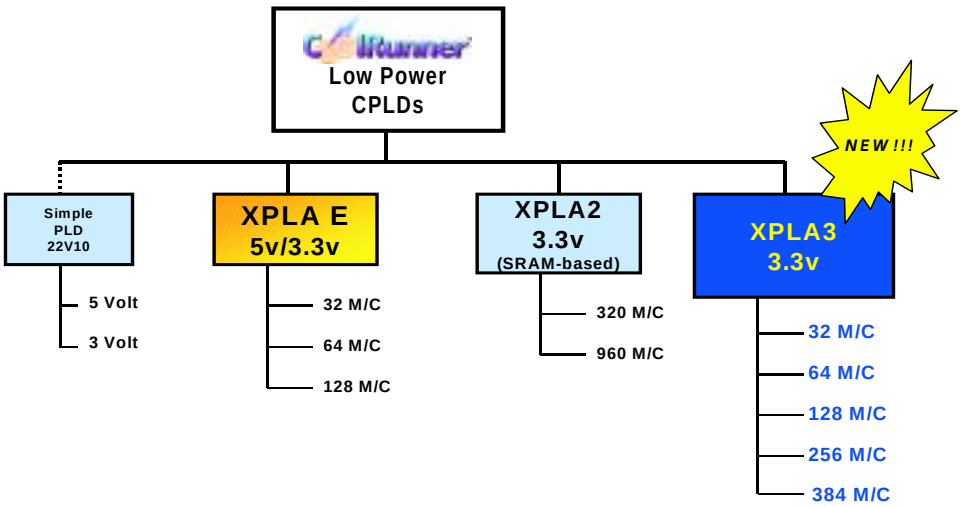

Figure 1: CoolRunner CPLD Families

CoolRunner XPLA3 Family

XPLA3 is the newest CoolRunner CPLD family. The XPLA3 family includes devices ranging from 32 to 384 macrocells. XPLA3 was created to maintain the same competitive advantages as the existing CoolRunner families, add additional features, increase performance, and offer a substantially lower cost. The CoolRunner XPLA3 is a non-volatile (Flash based), 0.35µ CMOS CPLD which offers ultra low power consumption, a flexible architecture, and high-speed capabilities.

CoolRunner XPLA3 Architecture

Designers want CPLD devices that offer high speed, high density, and the flexibility to make changes to their design at any stage of the design process. A particular device's ability to meet all of these critical needs efficiently is often constrained by the basic architecture of the CPLD. The basic components of CPLD architecture that affect the device's speed, density, and design flexibility can be broken into four distinct areas. These four areas are the basic interconnect methodology, logic block architecture, logic allocation method, and the timing model of the device. The CoolRunner XPLA3 architecture is the result of extensive research into the effect architecture has on these critical system needs and delivers a third generation solution that is superior to previous architectures.

From a high-level, the architecture of CoolRunner XPLA3 CPLDs appears similar to many other CPLD architectures. As shown in Figure 2, 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. A 4-bit Universal Bus is used to provide an individual asynchronous clock (UCLK), reset (URST), preset (UPST), and output enable (UOE). These bus lines are driven by four multiplexers (muxes), with the mux inputs consisting of a single control p-term from each Logic Block. The 32 macrocell version of XPLA3 will have two logic blocks, which means a maximum of two universal control functions can be implemented. All other family members can have up to four universal functions. Four external clock signals (Global Clocks) are muxed down to two, selectable at each logic block.

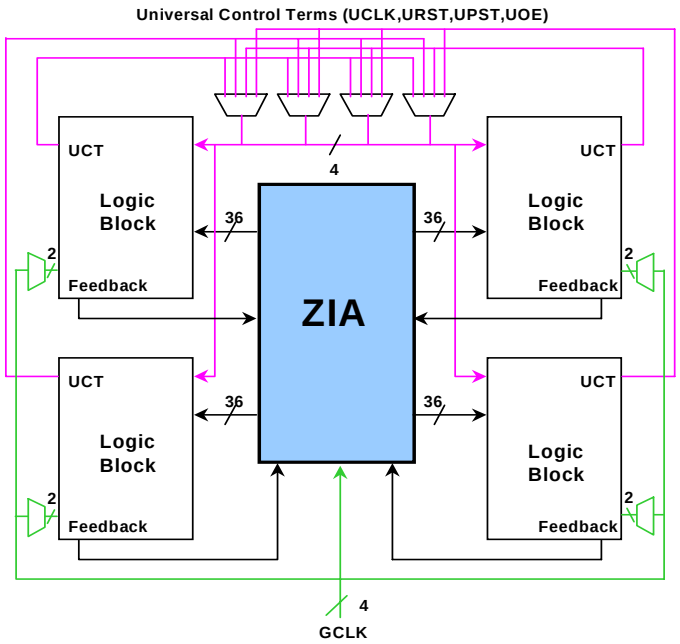

Figure 2: XPLA3 High-level Architecture (64 Macrocell device shown)

ZIA

The basic premise of CPLD architecture is the construction of large devices that are built upon multiple PLD blocks that are connected via an interconnect matrix (see Figure 3). In CPLDs, this interconnect resource is supposed to act like a crosspoint switch to route signals from the Inputs, I/Os, and macrocell feedbacks to the logic blocks where these signals are needed. The interconnect must also be very fast to support the high speeds that designers expect in today's CPLDs. The ability to lock pins is problematic when interconnect fails in its ability to route signals under worst case conditions.

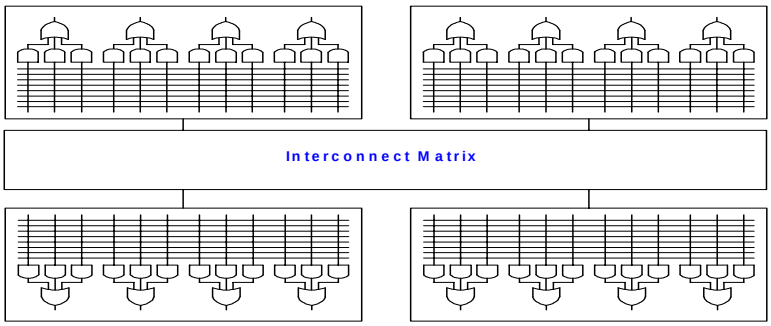

Figure 3: Basic CPLD Architecture

The ideal performance of an interconnect is to fully emulate a crosspoint switch, where every input to the array can be connected to every output of the array under fixed pinouts. Some first generation devices used full crosspoint switch arrays, and as a result offered 100% routability, at a significant price. As shown in Figure 4, building a crosspoint switch with these devices required a fuse at every intersection of the input and output line in the array. For a 128 macrocell device, this would translate into more than 65,000 connections. More significantly, these fully populated crosspoint switches were relatively slow, accounting for an 8 ns to 15 ns delay.

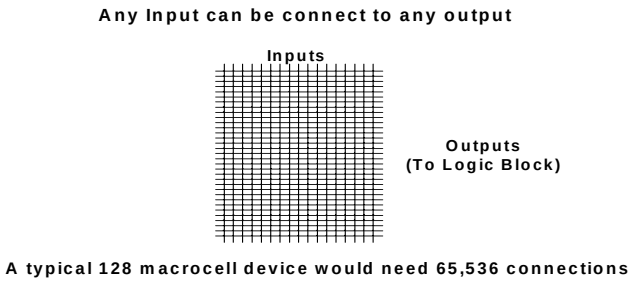

Figure 4: Cross Point Switch

The next step in interconnect evolution was the use of multiplexers to emulate crosspoint switches, a technique that all contemporary devices employ. Figure 5 shows a set of 16 muxes that are two bits wide which form an interconnect that has 32 inputs and 16 outputs. The use of muxes has two immediate effects. The first is that the delay through the interconnect is typically equivalent to a single mux delay, which is typically well under 0.5 ns. The second effect is the reduction of connections required to implement the interconnect. As a result, the number of connections required for a 128 macrocell device can be reduced from approximately 65,000 to less than 2,000.

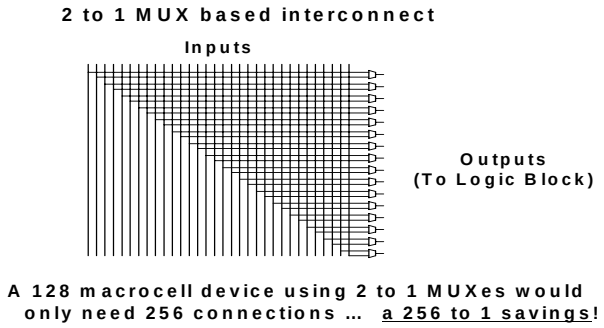

Figure 5: 2:1 Mux Based Interconnect

Unfortunately, if the architecture of this interconnect is not well engineered, signal blocking can occur. The issue with this 2:1 Mux approach is that only half of the inputs can enter the logic block. The key to building an efficient non-blocking muxed based interconnect is to include many overlapping, wide muxes which give each input multiple chances to get into the logic block. The main trade-off is routability (non-blocking) and costs (silicon area and performance). Enough resources (wide Muxes) need to be included to ensure not only that the device will be able to route the design but must also be architected to facilitate last minutes design changes once the pins have been locked.

The XPLA3 interconnect (see Figure 6) was architected by the software group that was responsible for ensuring that it would be capable of re-routing fixed pinout designs. By extensively simulating the width of the muxes and the number deployed, a mux based interconnect can be designed such that the probability of signal blocking is statistically very low. The XPLA3 interconnect employs a sufficiently large number of input muxes, of sufficient width, to guarantee routability under worst case conditions. The final interconnect architecture was subjected to over 16 million iterations of worst case fixed pinout routing. This resulted in worst case signal routing of 99.997% when every I/O, input pin, and macrocell is in use and has a fixed pinout. If only 35 of the 36 logic block inputs are used, 100% of the 16 million fixed signal routings completed successfully. It is believed that this type of solution allows designers total freedom to make design iterations without the fear of having to re-layout the PCB. Not every CPLD architecture is able to support interconnect routing that is this robust.

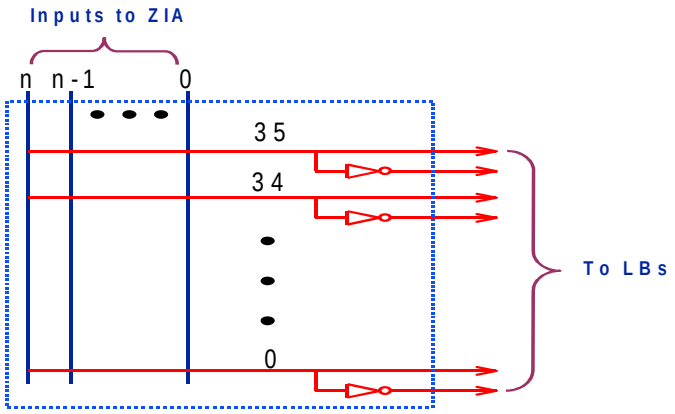

Figure 6: XPLA3 Zero Power Interconnect Array (ZIA)

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.

Logic Block

Figure 7 illustrates the logic block architecture. There are 36 pairs of true and complement inputs from the ZIA that feed 48 product terms (PTs) in the array. Each logic block contains a PLA (Programmable Logic Array) which provides a pool of 48 PTs that can be used as macrocell clocks, control terms (reset, preset, clock-enables, or output-enables), or as needed by the 16 macrocells in the logic block. These 48 product terms can be used 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 may be used for macrocell logic.

Within the 48 product terms there are:

- Eight product terms, PT[0-7], can be used to generate eight local control terms (LCT[0:7]) that are available for use, by each macrocell, as asynchronous clocks, resets, presets, and output enables. One of the control product terms in each Logic Block is made available to the Universal Control Term Bus. It can drive any one of the four Universal Control Terms through the Universal Control Term Mux.
- Sixteen product terms, PT[16:31], are coupled with the associated programmable OR gate into the VFM (Variable Function Multiplexer). The VFM increases logic optimization by implementing any two input logic function before entering the macrocell.
- Sixteen product terms, PT[32:47], can be used as asynchronous clocks or as clock enables.
- Eight fold-back NAND product terms, PT[8:15], are available for ease of fitting and pin locking. These fold-back NAND structures increase the virtual width of the product term and allow more logic to be placed in less silicon area.

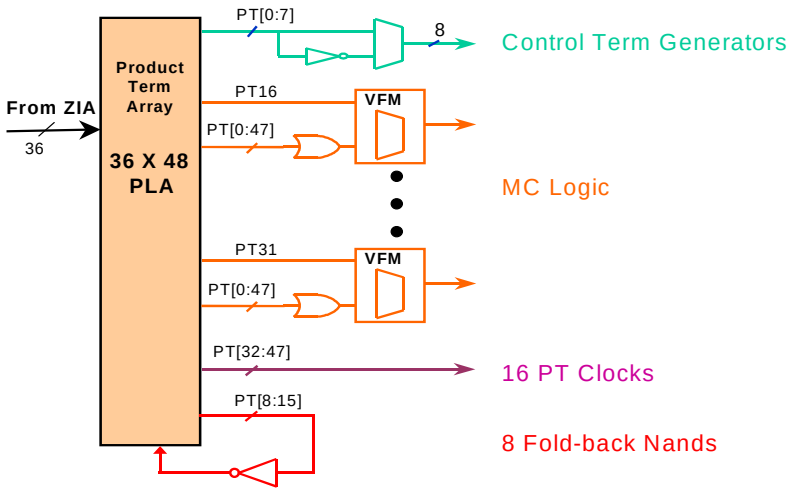

Figure 7: XPLA3 Logic Block Architecture

A "Power Up" input to the ZIA, active during the initialization of the PLD, allows individual macrocells to be reset or preset depending on the customer configuration pattern.

Variable Function Mux (VFM)

A Variable Function Mux is a small part of the PLA logic that feeds directly into each macrocell. Every macrocell has its own VFM. As shown in Figure 8, this VFM can be thought of as a flexible programmable logic element that can synthesize any two input logic element, such as XOR, XNOR, etc. Each VFM has two inputs, a single product term input and a sum of products input. Both of these inputs come directly from the PLA; the sum of products input may consist of up to 48 product terms.

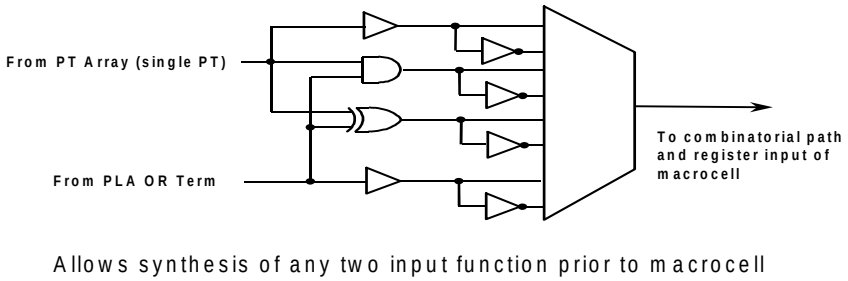

Figure 8: Variable Function Mux

Fold-back NANDs

Fold-back NANDs are powerful architectural additions for synthesis capable tools and provide increased density. Figure 9 illustrates how a Fold-back NAND is constructed. The key operation of a Fold-back NAND is to allow the software synthesis to use De Morgan's Theorem to create "virtual" PTs. In other words, equations can be re-written using De Morgan's Theorem to reduce the number of PTs needed to implement the equivalent functionality.

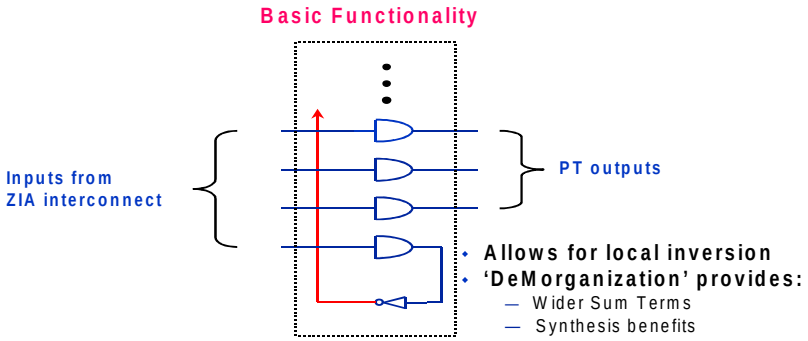

Figure 9: Fold-back NAND

Figure 10 shows an example of how Fold-back NANDs can be used to reduce the number of required product terms. Fold-back NANDs are especially effective in state machines and decoder designs. XPLA3 devices use Fold-back NANDs to cost effectively implement higher density logic (Fold-back NANDs require much less silicon area than additional product terms).

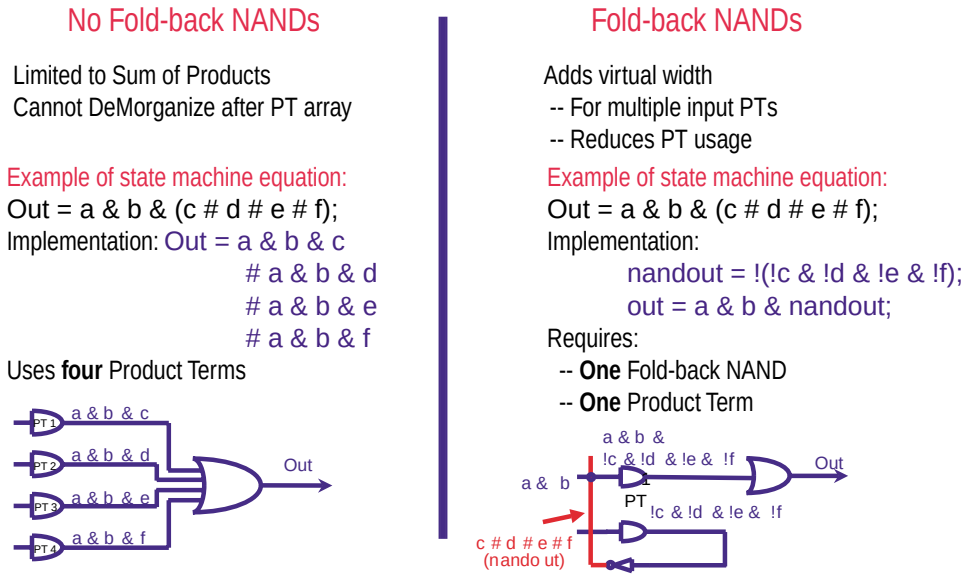

Figure 10: Fold-back NAND Design Example

PAL versus PLA

There are two basic types of product term generators: a PAL (programmable AND, fixed OR) and a PLA (programmable AND, programmable OR). Both types of logic use an AND array to build up a product term; the difference between the two is how the product terms are summed. The PAL is comprised of a programmable AND array followed by a fixed number input OR element. Each OR element has a certain number of dedicated PTs and sharing of these PTs is not allowed. The PLA array differs from the PAL in that the AND array is followed by a programmable width OR array (see Figure 11). Having a programmable OR Array allows PTs to be shared between macrocells (effectively increasing device density), to have excellent pin locking (every PT is available to every macrocell), and provides a very simple timing model. The XPLA3 architecture is based on a PLA; all non-CoolRunner CPLDs employ a PAL.

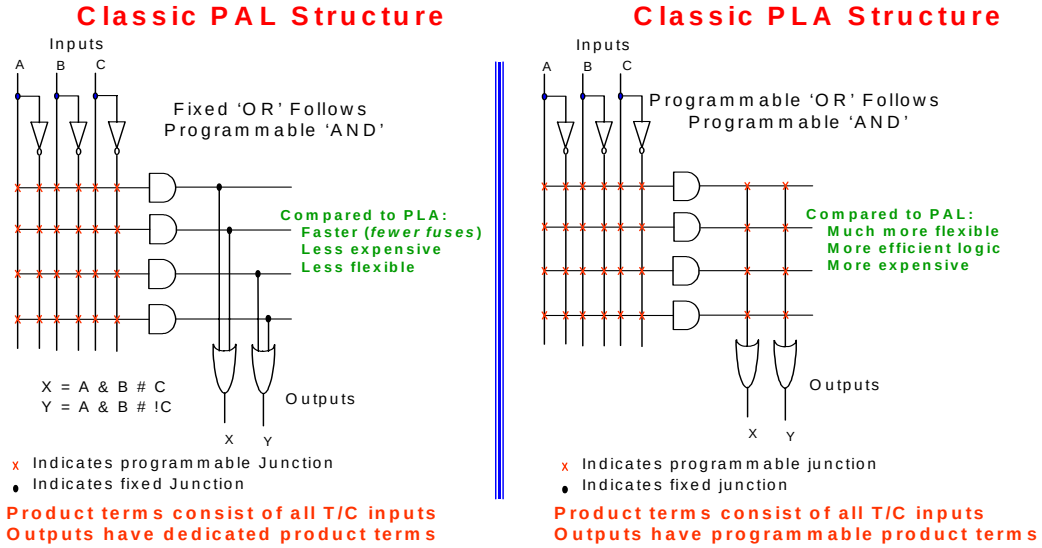

Figure 11: Basic PAL / PLA Structures

Product Term Sharing

Many logic designs, such as decoders and state machines, have common logic components. If an architecture allows for sharing of resources (e.g., PLA), this common logic can be built up one time and provided to all higher expressions that require this common sub-expression. This sharing of logic means that common logic does not have to be duplicated. Figure 12 is intended to help illustrate the PAL / PLA product term sharing differences.

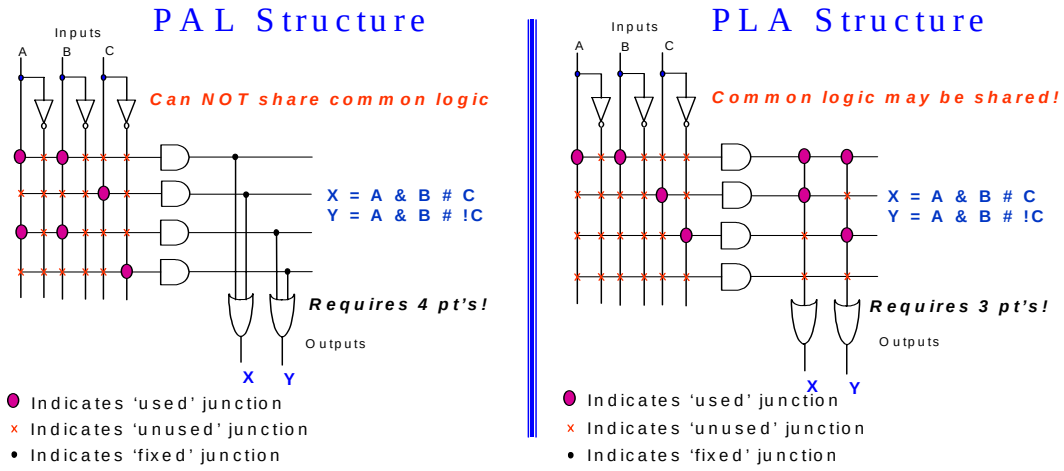

Figure 12: Product Term Sharing Design Example

Product Term Allocation Method

There are two different approaches used by CPLD manufactures to allocate logic: product terms steering and PLA implementation. Product-term steering dedicates a certain number of product terms to each macrocell in the logic block. Steering mechanisms are used that allow a macrocel'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. In addition, all 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 13 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.

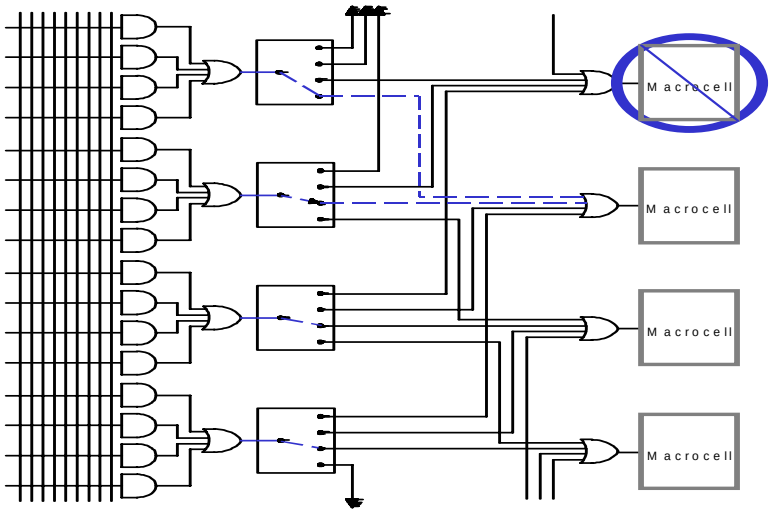

Figure 13: Logic Block Architecture with Product-term Steering

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 13 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 thpinoutis design change without affecting any of the macrocells in the logic block or the device pinout as shown in Figure 14. 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.

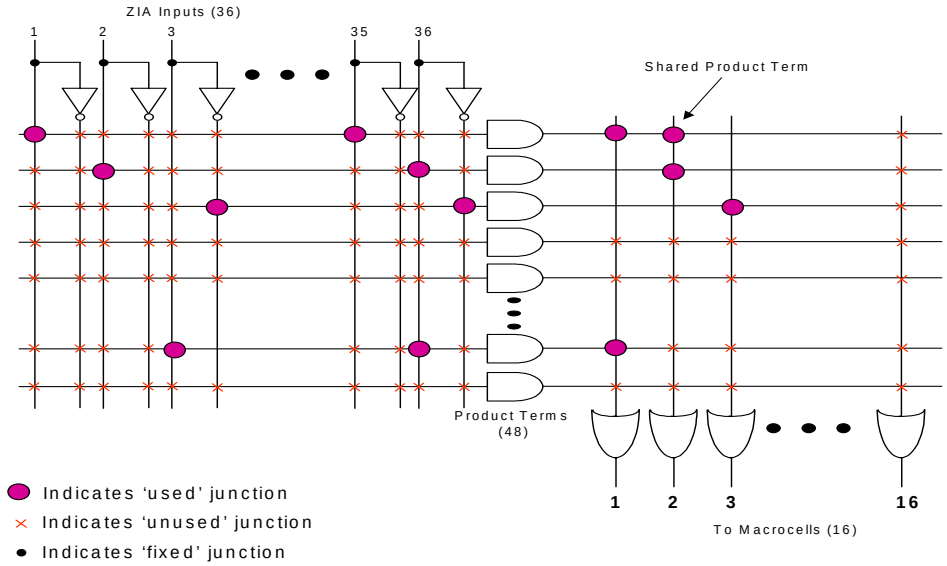

Figure 14: XPLA3 Logic Block with Pure PLA Array

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 > 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 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. CPLDs that utilize product-term steering do not allow designers to maximize the benefits of In-System Programmability (ISP).

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 a CoolRunner XPLA3 CPLD, fixed pinouts are maintained after logic changes even at device utilizations of > 99%, making the XPLA3 architecture optimal for ISP.

Macrocell

Each macrocell, as shown in Figure 15, can support combinatorial or registered inputs, a universal preset and reset for each macrocell and configurable D, T, or L registers with maximum clocking flexibility. There are two feedback paths to the ZIA: one from the macrocell and one from the I/O pin. When the I/O is used as an output, the output buffer is enabled, and the macrocell feedback path can be used to feed back the logic implemented in the macrocell. When the I/O pin is used as an input, the output buffer will be tri-stated and the input signal will be fed into the ZIA via the I/O feedback path. The logic implemented in the buried macrocell can be fed back to the ZIA via the macrocell feedback path. macrocells which are buried within a Logic Block and not connected to an I/O are identical to the non-buried macrocells. Each macrocell can be used to implement either register or combinatorial functions.

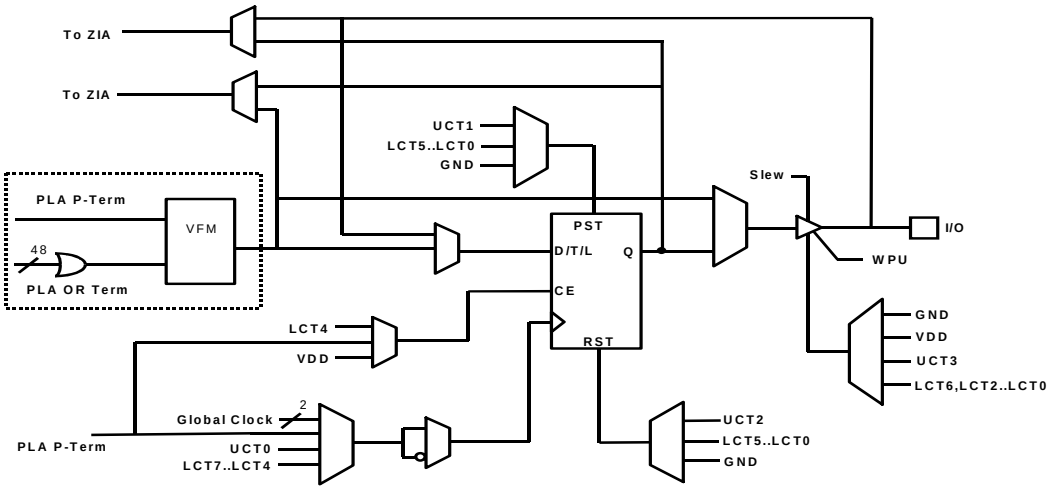

Figure 15: XPLA3 Macrocell

Register Functionality

The data input to each macrocell register is derived from the output of Variable Function Multiplexer. Each macrocell register can be configured as a D-, T-, or Latch-type flip-flop; this flip-flop may also be configured to be an input register (see the section entitled, "Input Register Configuration" for more details). Each flip-flop has both asynchronous preset and reset capabilities. There are seven different preset and reset sources: one universal control term (one for preset [UCT1] and one for reset [UCT2]) and six shared local control terms (LCT[0-5]).

Clocking

Each macrocell register can be clocked from any one of ten sources.

- There are two global clocks that are derived from the four external clock pins via a 4:2 multiplexer.
- There is one universal clock signal (UCT0) sourced by a universal control term.
- There are four Local Control Terms (LCT4-LCT7) which can be used as clock signal and can be individually configured as either a product term or sum term equation created from the 36 signals available inside the logic block.
- There is one dedicated product term clock per macrocell.

In addition to having ten possible clock sources, polarity (rising or falling edge) is also selectable at each macrocell. Hardware clock enables are also available for added clock control.

Input Register Configuration

The XPLA3 device macrocells may have their registers configured as input registers; this means that signals from a pin may be directly latched by the register without having to pass through the interconnect array. The setup time for this is 2 ns and is accompanied by a 0 ns hold time. As shown in Figure 16, when implementing an input register the preceding macrocell logic is still available for use as a buried combinatorial node. This logic may be fed back to the interconnect array for distribution elsewhere in the device.

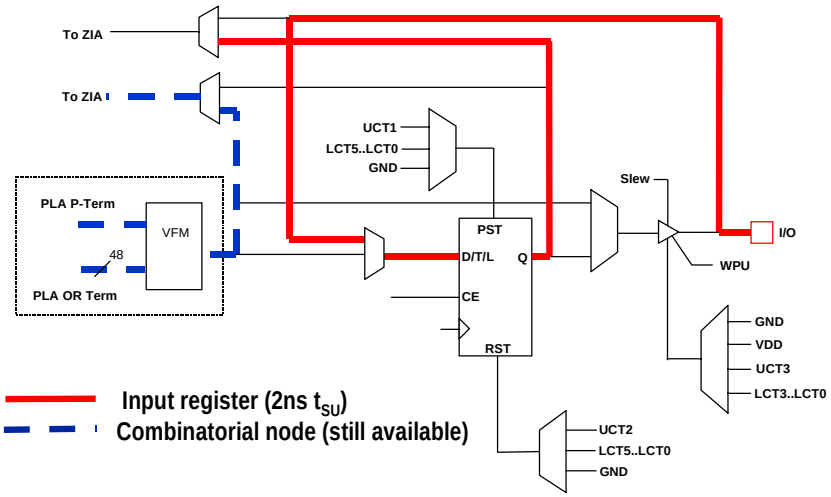

Figure 16: XPLA3 Input Register

I/O Cell

The XPLA3 devices are implemented on a 0.35µ process and run off of a single supply VDD of 3.3V. All I/Os are 5V tolerant and provide timing, voltage, and current characteristics required by the PCI specification. XPLA3 devices are PCI compatible but not compliant. CoolRunner CPLDs are not compliant because the specification requires overshoot signal conditioning diodes that the XPLA3 devices do not provide. If a designer wishes to implement a PCI driver/receiver, the designer will be required to provide external diode clamps. With the exception of the clamp diode, XPLA3 devices meet the stringent requirements of the PCI specification.

The Output Enable (OE) has eight possible states as shown in Table 1.

Table 1: XPLA3 Output Enable Selection

OE Decode	I/O Pin State
0	3-state
1	Function CT0
2	Function CT1
3	Function CT2
4	Function CT6
5	Universal OE (UCT3)
6	Enable
7	3-state with Weak Pull-up

The XPLA3 output buffers incorporate weak pull up resistors (option # 7) to provide internal termination when I/Os are unused. These pull-ups are not available for used I/Os. Please note that dedicated inputs (global clocks) do not have a pull-up resistor and therefore should be properly terminated (pulled Hgh or Low) if left unconnected.

XPLA3 devices also provide slew rate control for each macrocell output pin. The user has the option to enable the slew rate control to reduce EMI. The nominal delay for using this is 3 ns.

Timing Model

The final consideration when selecting a CPLD is the timing model. CPLDs should offer fast, deterministic timing that remains invariant as design changes are made. More specifically, since late changes often involve adding additional logic, the ability of the device to maintain

predictable timing as the logic width increases is important. Unfortunately, many devices have speeds that are attainable only under a limited set of conditions. As additional logic complexity is introduced, the timing may suffer and be significantly different from the 'peak' speed promised. It is a good idea to review the timing information and/or model for the device being considered. Sometimes it can be very difficult to determine what the timing will be if 16 (or more) product terms are used.

In non-CoolRunner architectures, the user may be able to fit the design, but may not be sure whether system timing requirements can be met until after the design has been fit into the device. This is because timing models of these architectures are very complex (dependent on many variables such as: the number of parallel expanders borrowed, sharable expanders, different routing channels, etc).

Figure 17 shows the XPLA3 timing model which has three main parameters, including T_{PD}, T_{SU}, and T_{CO}. As a result of the simplicity of this timing model, designers can make reasonably accurate estimations of the performance of their design before they began using the device. It is important to note that the PLA timing is deterministic regardless of the number of PLA terms that are used or the number that are shared by multiple outputs. For more detail, please refer to "Understanding CoolRunner XPLA3 Timing Models and Clocking Options" (WP108).

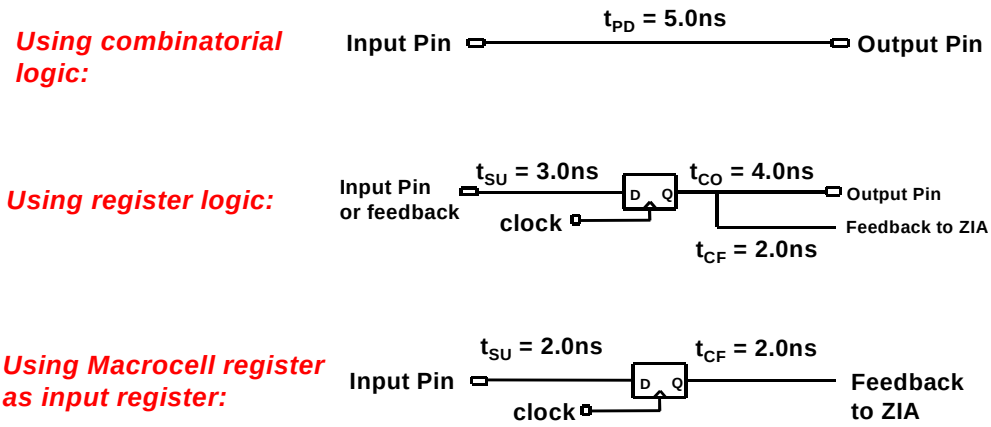

Figure 17: XPLA3 Timing Model

JTAG and ISP

JTAG is the commonly used acronym for the Boundary Scan Test (BST) feature defined for integrated circuits by IEEE Standard 1149.1. This standard defines input/output pins, logic control functions, and commands that facilitate both board and device level testing without the use of specialized test equipment. XPLA3 devices support the following JTAG commands: BYPASS, SAMPLE/PRELOAD, EXTEST, IDCODE, HIGH-Z, and INTEST.

XPLA3 devices also use the JTAG Interface for In-System Programming/Reprogramming. ISP is the ability to reconfigure the logic and functionality of a device, printed circuit board, or complete electronic system before, during, and after its manufacture and shipment to the end customer. XPLA3 CPLDs support and ISP commands (ENABLE, DISABLE, ERASE, PROGRAM, and VERIFY).

The XPLA3 JTAG port includes four of the five pins described in the JTAG specification: TCK, TMS, TDI, and TDO. The fifth signal defined by the JTAG specification is TRST (Test Reset) is optional and is used to reset the JTAG state machine and all BST registers. XPLA3 devices support the test reset functionality through the use of its power-up reset circuit. It should be noted that the input pins associated with the JTAG Port should connect to an external pull-up resistor typically to keep the JTAG signals from floating when they are not being used. Please refer to the "CoolRunner XPLA3 Family Data Sheet" for complete JTAG / ISP description.

XPLA3 devices are shipped with the JTAG port enabled. These pins can be left as dedicated JTAG functions (i.e., not general purpose I/O pins). If needed, at the time of ISP programming,

the port pins can be reconfigured as normal I/O pins. The XPLA3 JTAG interface includes a port enable signal (refer to section entitle "Port Enable Pin" for more detail).

Port-Enable Pin

The Port Enable pin is used to reclaim TMS, TDO, TDI, and TCK for JTAG purposes if the user has defined these pins as general purpose I/O during device programming. For ease of use, XPLA3 devices are shipped with the JTAG port pins enabled. Please note that the Port Enable pin must be held at a logic low level during the power-up sequence for the device to operate properly.

During device programming, the JTAG pins can be left as is or re-configured as user specific I/O pins. If the JTAG pins have been used for I/O pins, simply applying a high logic level to the Port Enable pin converts the JTAG pins back to their respective programming function and the device can be reprogrammed. After completing the desired JTAG ISP programming function, simply return Port Enable to a logic Low. This will re-establish the JTAG pins to their respective I/O function.

Conclusion

The XPLA3 architecture is a unique approach to interconnect methodology and logic allocation that provides the ability to refit designs that use 100% of the pins, macrocells, and logic in the device. Xilinx CoolRunner CPLDs have long been a pioneer in developing and patenting structures used in logic arrays such as the PLAs, VFMs, and fold-back NANDs. The unique XPLA3 combination of PLA arrays, VFMs, and fold-back NANDs allows logic to be allocated on an as needed basis and delivers high logic density at reasonable costs. XPLA3 logic is allocated at a granularity of one product term, across all macrocells, and can be shared resulting in the highest level of efficiency possible. Finally, the timing model for the device is simple and deterministic, allowing designers the ability to accurately predict design performance with tools no more complex than a common pencil and paper napkin. The combination of these features provides designers with high speed, high density, and ultimate flexible devices to make last minute changes to their designs.

Revision History

Date	Version #	Revision
01/05/00	1.0	Initial Xilinx release.