CPLDs vs. FPGAs
Comparing High-Capacity Programmable Logic

February 1995, ver. 1 Product Information Bulletin 18

Introduction

The high-capacity programmable logic device (HCPLD) market has expanded dramatically in recent years with the introduction of new devices and architectures. While these new devices offer greater design flexibility, design engineers must sift through all available devices to determine the best device for an application. Choosing the right device can lead to market success, while choosing the wrong device can result in major project setbacks.

To make an informed decision, designers need to understand the strengths and limitations of different HCPLDs, specifically complex programmable logic devices (CPLDs) and field-programmable gate arrays (FPGAs). This product information bulletin provides guidelines on choosing the correct devices for design applications and discusses the following topics.

- ❏ PLD market overview
- ❏ CPLD vs. FPGA architecture
- ❏ CPLD vs. FPGA interconnect structures
- ❏ CPLD vs. FPGA process alternatives
- ❏ CPLD vs. FPGA development software

PLD Market Overview

The PLD market consists of low- and high-capacity devices. Low-capacity devices, called simple PLDs, typically contain fewer than 600 usable gates and include products such as PALs, GALs, and 22V10s. Simple PLDs are manufactured using CMOS technology offering EPROM, EEPROM, and FLASH memory elements.

HCPLDs typically contain more than 600 usable gates, and include both CPLDs and FPGAs. HCPLDs are manufactured using CMOS technology with EPROM, EEPROM, FLASH, SRAM, and antifuse options. HCPLDs can be differentiated by their interconnect structure: CPLDs use continuous interconnect structures, while FPGAs use segmented interconnect structures. See Figure 1.

A-PIB-018-01

Figure 1. Programmable Logic Market

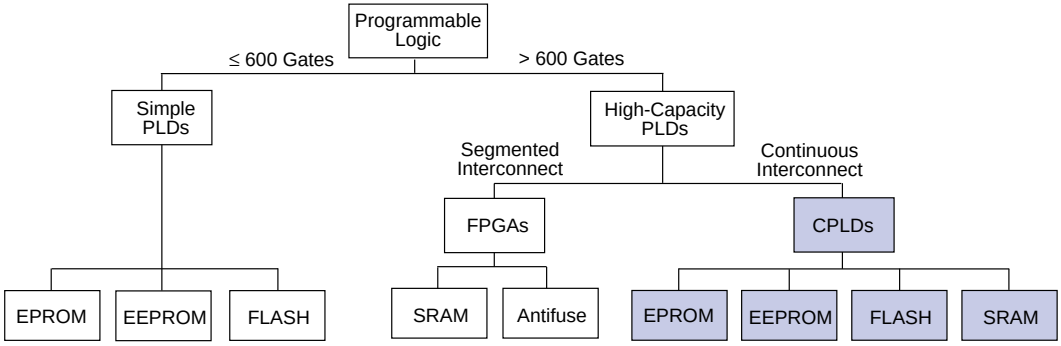


Table 1 describes CPLD and FPGA features.

Table 1. CPLD vs. FPGA Features

Feature	FPGAs	CPLDs
Leading vendor	Xilinx	Altera
Density	Medium to high	Low to high
Interconnect structure	Segmented	Continuous
Timing	Variable/unpredictable	Fixed/predictable
CMOS options	SRAM, antifuse	EPROM, EEPROM, FLASH, SRAM
Device performance	Moderate	High
Device utilization	Moderate	High
Hand-routing required	Yes	No
Reprogrammability	Yes (SRAM only)	Yes
In-circuit reconfigurability	Yes (SRAM only)	Yes (SRAM only)
In-system programmability	No	Yes (FLASH, EEPROM only)
Compilation times	Slow	Fast
Logic synthesis	Yes (third-party only)	Yes

CPLD vs. FPGA Architecture

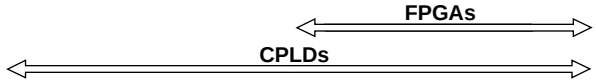
CPLDs use several different varieties of CMOS technology and architectural alternatives to address the wide range of logic design applications. EPROM-, EEPROM-, and FLASH-based devices—such as the Altera Classic, MAX 5000, MAX 7000, MAX9000, and FLASHlogic families—use a product-term architecture that is optimized for combinatorial-intensive logic designs. EPROM, EEPROM, and FLASH devices are reprogrammable and nonvolatile. SRAM-based CPLDs, such as the Altera FLEX8000 family, use a look-up table (LUT) architecture that is optimized for register-intensive designs. SRAM-based devices offer in-circuit reconfigurability for "on-the-fly" logic changes. Altera is the only CPLD vendor to offer all four major CMOS processes—EPROM, EEPROM, FLASH, and SRAM—to encompass the broadest range of logic design applications.

In contrast, most FPGAs use SRAM or one-time-programmable antifuse memory elements. The granular architectures of FPGAs can implement a wide range of applications, but tend to be less efficient than CPLD architectures at implementing combinatorial-intensive logic designs.

While antifuse-based devices typically implement the same register-intensive logic designs as SRAM-based FPGAs, they cannot be erased or reconfigured. Whereas low-cost, simple PLDs—such as 22V10s—may not require erasability and reconfigurability, these characteristics are essential for HCPLDs. Typically, large designs that contain several thousand gates require several design iterations. Computer simulation helps prevent design errors; however, designers often need to test hardware or incorporate unexpected changes to system specifications. With one-time-programmable antifuse elements, multiple design iterations can result in thousands of dollars of additional expenses in engineering time and device cost.

Figure 2 shows the logic design applications best suited for CPLD and FPGA architectures.

Figure 2. Logic Design Applications



Combinatorial-Intensive	Register-Intensive
Bus interfacing	Data paths
Comparators	Hardware emulation
High-speed wide decoders	Gate-array prototyping
Large/fast state machines	JTAG applications
High-speed glue logic	Battery-powered applications
System video controllers	Field-test equipment
PAL integration	Image controllers

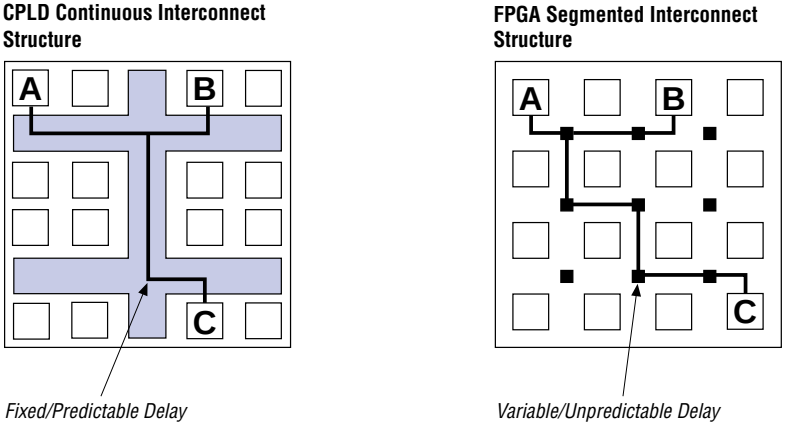
CPLD vs. FPGA Interconnect Structure

CPLDs and FPGAs use different interconnect structures: CPLDs use a continuous interconnect structure, while FPGAs use a segmented interconnect structure.

The continuous interconnect structure of CPLDs consists of metal lines of uniform length that traverse the entire length and width of the device. Since the resistance and capacitance of all interconnect paths is fixed, delays between any two logic cells in the device are predictable. This uniformity of performance across the device minimizes signal skew.

The segmented interconnect structure of FPGAs consists of a matrix of metal segments that run throughout the device. Depending on the type of FPGA, either switch matrices or antifuses join the ends of these segments, allowing signals to travel between logic cells. The number of segments required to interconnect signals is neither constant nor predictable; therefore, delays cannot be quantified until placement and routing has been completed. See Figure 3.

Figure 3. CPLD vs. FPGA Routing Scheme



Interconnect structures affect the following device characteristics:

- ❏ Performance predictability
- ❏ In-system performance
- ❏ Logic utilization

Performance Predictability

The constant delay between any two logic cells in a continuous interconnect structure allows designers to predict the performance of a CPLD design. The timing model for Altera CPLDs contains all parameters necessary for a designer to predict the overall performance of a design, including interconnect delays.

The total interconnect delay in a segmented interconnect structure is directly proportional to the number of segments necessary to route a signal. It is impossible to know this number until the logic design has been completely placed and routed by the software. Even minor logic changes can require major routing changes, affecting the performance characteristics of the overall device. Timing models and interconnect delay parameters do not exist for FPGAs because delays cannot be predicted.

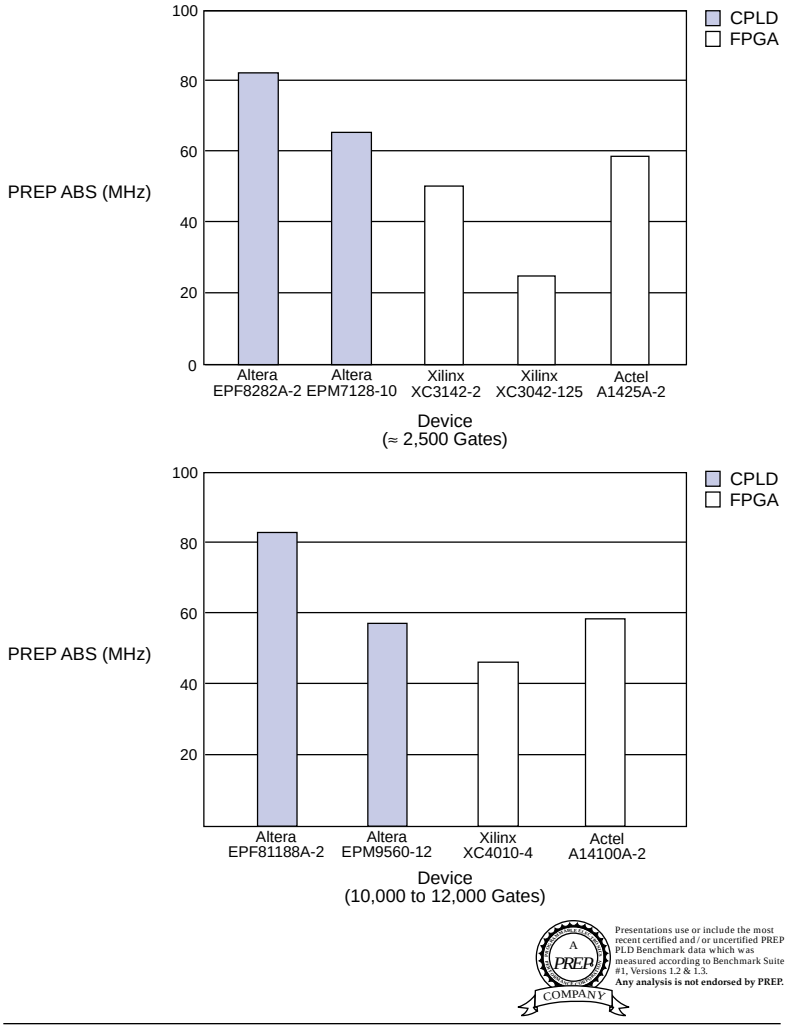
In-System Performance

CPLDs are known for high in-system performance, which is a direct result of their continuous interconnect structure and efficient signal routing. CPLD delays are not cumulative; the delay the signal incurs is independent of the path the signal takes. Signals that must travel to different locations in the device arrive at their destinations with negligible signal skew.

In contrast, FPGA segmented interconnect delays are cumulative. As the number of interconnect segments increases, the interconnect delay also increases. If interconnect segments for a direct path are already used, a signal must pass through a different path, i.e. more segments, to reach its destination. Therefore, no two signals can be guaranteed to arrive at the same time. Typically, one signal arrives after the other, which limits the performance of the design. Signal skew and performance degradation become more prevalent as more interconnect segments are used, inefficiently routing signals through the device.

CPLDs offer higher performance than FPGAs of equivalent densities. Device performance is verified by the Programmable Electronics Performance Corporation (PREP), a consortium of PLD companies that has established standard benchmarks for measuring the performance and logic capacity of PLDs. Figure 4 shows the average benchmark speed (ABS) of CPLDs and FPGAs with similar gate counts. ABS is calculated by averaging the mean internal and external benchmark results for all PREP benchmarks.

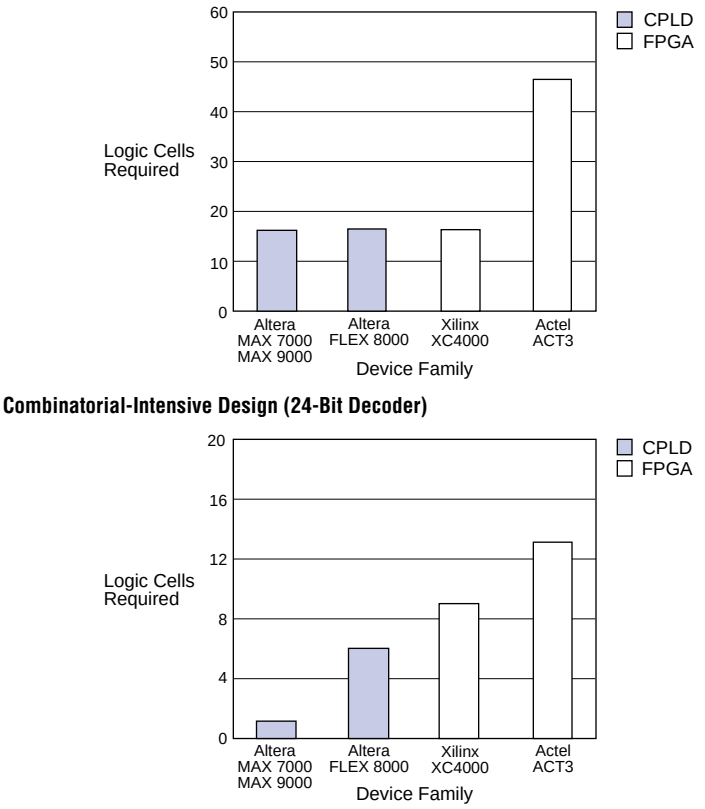
Figure 4. CPLD vs. FPGA Performance



Logic Utilization

The logic cells in most FPGA architectures have fine granularity; therefore, more logic cells are required to implement a function in an FPGA than in a CPLD. Figure 5 shows the number of logic cells necessary to implement a register-intensive design (16-bit up/down binary counter) and a combinatorial-intensive design (24-bit decoder) in each architecture.

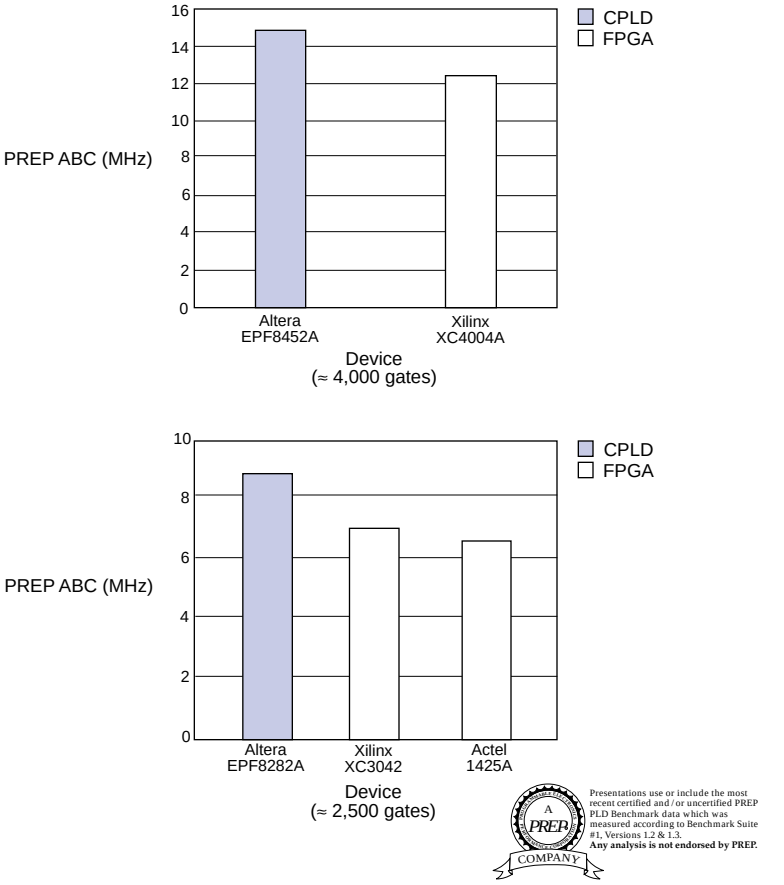
Figure 5. Logic Utilization



Because the logic cell of an FPGA can contain only small portions of a design, a heavy burden is placed on its segmented interconnect structure. Therefore, any routing bottleneck adversely affects device utilization, as well as device performance, as shown in Figure 4 earlier in the product information bulletin. As design complexity increases, the probability of routing conflicts also increases, ultimately leading to lower FPGA device utilization.

Figure 6 shows the average benchmark capacity (ABC)—the average number of benchmark instances implemented in a device across all PREP benchmarks—of CPLDs and FPGAs with similar gate counts. These results illustrate the lower device utilization in devices with segmented interconnect structures.

Figure 6. CPLD vs. FPGA Capacity



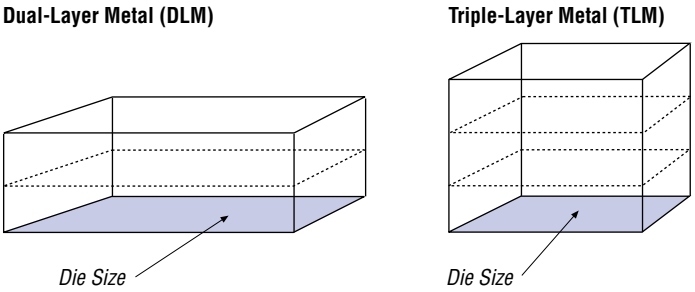
CPLD vs. FPGA Process Alternatives

PLD technology is measured in microns and metal layers. Micron size represents the smallest dimension of a transistor within the device; metal layers represent the number of levels in which metal is deposited in the device. Current standard programmable logic process technologies are 0.8-micron CMOS dual-layer metal (DLM) and 1.0-micron CMOS DLM technologies. Advances in both technologies provide higher performance and lower costs, but depend on device architecture. Because of their continuous interconnect structure, CPLDs benefit more from process migrations than FPGAs.

The continuous interconnect structure of CPLDs has many metal lines that traverse the entire device. These metal lines require few transistors to connect them to logic cells (see Figure 3). CPLD architectures are metal-optimized and benefit more from process migrations to triple-layer metal

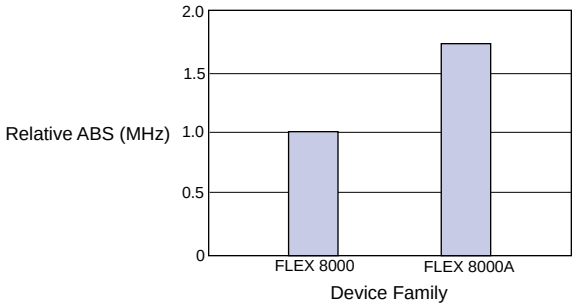
(TLM). For example, Altera's FLEX 8000 family is built on a 0.6-micron, TLM technology. Its three-dimensional FastTrack Interconnect structure maps efficiently to three-layers of metal. Figure 7 illustrates the FLEX 8000 TLM advantage using an analogy. Assume that a 30,000 square foot building needs to be built. This building can be two stories high and use 15,000 square feet of land, or it can be three stories high and use 10,000 square feet of land. Each floor in this analogy represents a layer of metal. With the TLM process, the usable density (i.e., the total size of the building) remains the same, while the die size (i.e., the required amount of land) has been dramatically reduced.

Figure 7. DLM vs. TLM Technology



A reduced die size—50% reduction for FLEX 8000A devices—translates to higher performance and lower costs. FLEX 8000A devices, which use a 0.6-micron TLM process, dramatically increase performance over the 0.8-micron DLM process of FLEX 8000 devices. See Figure 8.

Figure 8. Performance Improvements

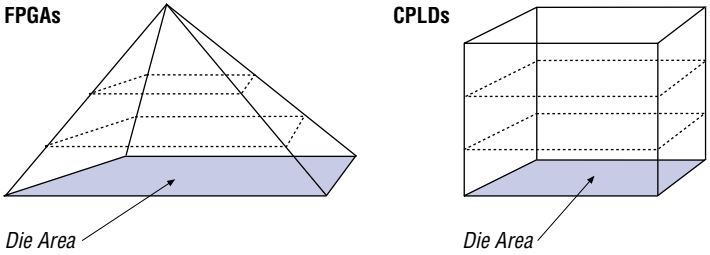


In contrast, a segmented interconnect structure contains relatively few metal segments that must pass through numerous transistors to connect logic cells. The large number of transistors in FPGA architectures limits die size reductions, so FPGAs receive a lesser benefit from TLM process migrations. Reducing the transistor size from 0.8-micron to 0.6-micron

shrinks FPGA die size. However, migrating from a DLM to TLM process does not create additional layers in which to place transistors; therefore, the die size of FPGAs remains relatively constant. Very few FPGA vendors use a 0.6-micron, TLM process.

Figure 9 illustrates the limited improvements that FPGA devices achieve from TLM process migration. FPGAs are limited by transistors, and can only create a pyramid-like, three-layer structure with most of the area still on the bottom layer. Without significant architectural changes, FPGA die size reductions are limited.

Figure 9. Triple-Layer Metal Process



As programmable logic vendors continue to push process technology to smaller dimensions and to increase the number of metal layers, the gap between CPLD and FPGA performance and cost will be magnified. Metal-optimized CPLD architectures will continually achieve higher performance and lower costs than transistor-limited FPGAs.

CPLD vs. FPGA Development Software

Fast, efficient development software, combined with an understandable device architecture, is an integral part of any programmable logic solution. Easy-to-use design entry, compilation, and simulation tools help increase design productivity, contributing to the time-to-market advantages of programmable logic. Altera's easy-to-use MAX+PLUSII development software provides designers with a fully integrated design tool that also interfaces with standard CAE tools. In addition, MAX+PLUS II provides the following features that increase the efficiency of CPLD designs.

- ❏ Efficient routing and logic utilization
- ❏ Quick compilation times
- ❏ Logic synthesis

Efficient Routing & Logic Utilization

The continuous routing structure of CPLDs allows the compiler to efficiently fit a design. Altera's MAX+PLUSII development software includes

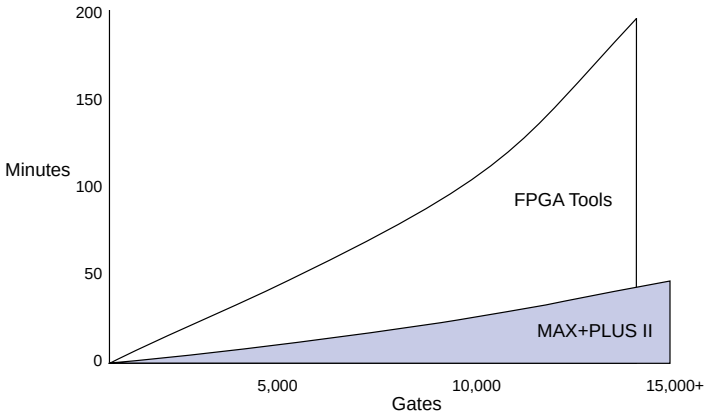
automatic logic synthesis, and provides 100% logic utilization with optimum speed performance. These results are achieved without hand-routing.

In contrast, the FPGA's segmented interconnect structure hampers the compiler's ability to fit a logic design. Designers must often hand-route FPGA designs to obtain initial fits or to optimize critical nets. This procedure requires designers to be intimately familiar with the device architecture and tools. Even after days or weeks of hand-routing iterations, the result may still be unsatisfactory due to unacceptable performance or poor device utilization.

Quick Compilation Times

Compilation times are usually a function of design complexity. However, a similar design may take hours to compile for FPGAs, while it takes only minutes for CPLDs. See Figure 10. The FPGA segmented interconnect structure, combined with fine-grained logic cells, requires the software to attempt several place-and-route iterations. If routing bottlenecks are encountered, the software must re-evaluate the entire design and use a new placement and routing scheme.

Figure 10. MAX+PLUS II vs. FPGA Tools Compile Times



Logic Synthesis

Logic synthesis minimizes a design by removing redundant logic and applying a set of optimization algorithms. Compilers from FPGA vendors such as Actel and Xilinx do not provide logic synthesis. Instead, these software packages perform technology mapping in which design descriptions are "mapped" into a specific architecture by applying simple

place-and-route techniques. While these techniques may be acceptable for simple PLDs, they are less efficient for HCPLDs.

As design size increases, high-level design descriptions become the most efficient design entry technique. Hardware description languages (HDLs) such as Verilog HDL, VHDL, and the Altera Hardware Description Language (AHDL) allow designers to focus on system level requirements rather than architectural details. Designers can rely on the software to synthesize and optimize a design to a specific PLD architecture. Since FPGA software does not synthesize designs, it is less efficient at implementing high-level design descriptions. Altera is the first and only programmable logic vendor with a development system that contains advanced, multilevel logic synthesis algorithms to efficiently implement high-level design descriptions.

Conclusion

Altera CPLDs use a continuous interconnect structure that offers predictable interconnect delays, high performance, and efficient logic utilization. Because this structure is metal-optimized, it enables CPLDs to take full advantage of advances in process technology, increasing performance and reducing customer costs. CPLD development software provides automatic logic synthesis and quick compile times.

In contrast, a segmented interconnect structure, found in Xilinx and Actel FPGAs, has many limitations, including unpredictable timing, slow performance, and poor device utilization. These devices are transistor-limited, restricting the effectiveness of TLM process technology to reduce die sizes. FPGA development software also requires extensive hand-routing and requires long compilation times.

Spanning all major PLD technologies and the entire breadth of logic applications, Altera CPLDs offer designers the most flexible and easy-to-use programmable logic devices and development software in the industry.

2610 Orchard Parkway
San Jose, CA 95134-2020
(408) 894-7000
Applications Hotline:
(800) 800-EPLD
Product Marketing:
(408) 894-7104
Literature Services:
(408) 894-7144