DocuSign Envelope ID: E62D3E7D-FEFF-4D42-9969-492E19C32F1E

IEEE
Advancing Technology for Humanity

DECLARATION OF GORDON MACPHERSON

I, Gordon MacPherson, am over twenty-one (21) years of age. I have never been convicted of a felony, and I am fully competent to make this declaration. I declare the following to be true to the best of my knowledge, information and belief:

1. I am Director Board Governance & IP Operations of The Institute of Electrical and Electronics Engineers, Incorporated ("IEEE").

2. IEEE is a neutral third party in this dispute.

3. I am not being compensated for this declaration and IEEE is only being reimbursed for the cost of the article I am certifying.

4. Among my responsibilities as Director Board Governance & IP Operations, I act as a custodian of certain records for IEEE.

5. I make this declaration based on my personal knowledge and information contained in the business records of IEEE.

6. As part of its ordinary course of business, IEEE publishes and makes available technical articles and standards. These publications are made available for public download through the IEEE digital library, IEEE Xplore.

7. It is the regular practice of IEEE to publish articles and other writings including article abstracts and make them available to the public through IEEE Xplore. IEEE maintains copies of publications in the ordinary course of its regularly conducted activities.

8. The article below has been attached as Exhibit A to this declaration:

A.	S. Brown; J. Rose; "FPGA and CPLD architectures: a tutorial", published in IEEE Design & Test of Computers (Volume: 13, Issue: 2, Page(s) 42 - 57), date of publication Summer 1996.

9. I obtained a copy of Exhibit A through IEEE Xplore, where it is maintained in the ordinary course of IEEE's business. Exhibit A is a true and correct copy of the Exhibit, as it existed on or about February 28, 2023.

10. The article and abstract from IEEE Xplore show the date of publication. IEEE Xplore populates this information using the metadata associated with the publication.

Intel Exhibit 1014
Intel v. Iida

DocuSign Envelope ID: E62D3E7D-FEFF-4D42-9969-492E19C32F1E

11. S. Brown; J. Rose; “FPGA and CPLD architectures: a tutorial”, published in IEEE Design & Test of Computers (Volume: 13, Issue: 2, Page(s) 42 - 57), date of publication Summer 1996. Copies of this publication was made available no later than the last day of the publication month. The article is currently available for public download from the IEEE digital library, IEEE Xplore.

12. I hereby declare that all statements made herein of my own knowledge are true and that all statements made on information and belief are believed to be true, and further that these statements were made with the knowledge that willful false statements and the like are punishable by fine or imprisonment, or both, under 18 U.S.C. § 1001.

I declare under penalty of perjury that the foregoing statements are true and correct.

Executed on: 2/28/2023

DocuSigned by:
Gordon MacPherson
E768DB210F4E4EF...

DocuSign Envelope ID: E62D3E7D-FEFF-4D42-9969-492E19C32F1E

EXHIBIT A

FPGA and CPLD Architectures: A Tutorial

STEPHEN BROWN
JONATHAN ROSE
University of Toronto

This tutorial surveys commercially available, high-capacity field-programmable devices. The authors describe the three main categories of FPDs: simple and complex programmable logic devices, and field-programmable gate arrays. They then give architectural details of the most important chips and example applications of each type of device.

RECENTLY, the development of new types of sophisticated field-programmable devices (FPDs) has dramatically changed the process of designing digital hardware. Unlike previous generations of hardware technology in which board level designs included large numbers of SSI (small-scale integration) chips containing basic gates, virtually every digital design produced today consists mostly of high-density devices. This is true not only of custom devices such as processors and memory but also of logic circuits such as state machine controllers, counters, registers, and decoders. When such circuits are destined for high-volume systems, designers integrate them into high-density gate arrays. However, the high nonrecurring engineering costs and long manufacturing time of gate arrays make them unsuitable for prototyping or other low-volume scenarios. Therefore, most prototypes and many production designs now use FPDs. The most compelling advantages of FPDs are low startup cost, low financial risk, and, because the end user programs the device, quick manufacturing turnaround and easy design changes.

The FPD market has grown over the past decade to the point where there is now a wide assortment of devices to choose from. To choose a product, designers face the daunting task of researching the best uses of the various chips and learning the intricacies of vendor-specific software. Adding to the difficulty is the complexity of the more sophisticated devices. To help sort out the confusion, we provide an overview of the various FPD architectures and discuss the most important commercial products, emphasizing devices with relatively high logic capacity.

Evolution of FPDs

The first user-programmable chip that could implement logic circuits was the programmable read-only memory (PROM), in which address lines serve as logic circuit inputs and data lines as outputs. Logic functions, however, rarely require more than a few product terms, and a PROM contains a full decoder for its address inputs. PROMs are thus inefficient for realizing logic circuits, so designers rarely use them for that purpose.

The first device developed specifically for implementing logic circuits was the field-programmable logic array, or simply PLA for short. A PLA consists of two levels of logic gates: a programmable, wired-AND plane followed by a programmable, wired OR plane. A PLA's structure allows any of its inputs (or their complements) to be ANDed together in the AND plane; each AND plane output can thus correspond to any product term of the inputs. Similarly, users can configure each OR

0740-7475/96/$05.00 © 1996 IEEE

Authorized licensed use limited to: IEEE Publications Operations Staff. Downloaded on February 28,2023 at 16:58:07 UTC from IEEE Xplore. Restrictions apply.

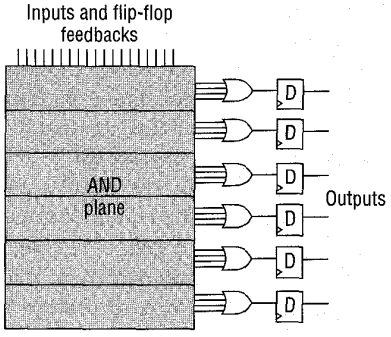


Figure 1. *PAL structure.*

plane output to produce the logical sum of any AND plane output. With this structure, PLAs are well-suited for implementing logic functions in sum-of-products form. They are also quite versatile, since both the AND and OR terms can have many inputs (product literature often calls this feature "wide AND and OR gates").

When Philips introduced PLAs in the early 1970s, their main drawbacks were expense of manufacturing and somewhat poor speed performance. Both disadvantages arose from the two levels of configurable logic; programmable logic planes were difficult to manufacture and introduced significant propagation delays. To overcome these weaknesses, Monolithic Memories (MMI, later merged with Advanced Micro Devices) developed PAL devices. As Figure 1 shows, PALs feature only a single level of programmability—a programmable, wired-AND plane that feeds fixed-OR gates. To compensate for the lack of generality incurred by the fixed-OR plane, PALs come in variants with different numbers of inputs and outputs and various sizes of OR gates. To implement sequential circuits, PALs usually contain flip-flops connected to the OR gate outputs.

The introduction of PAL devices profoundly affected digital hardware design, and they are the basis of some of the newer, more sophisticated architectures that we will describe shortly. Variants of the basic PAL architecture appear in several products known by various acronyms. We group all small FPDs, including PLAs, PALs, and PAL-like devices, into the single category of simple programmable-logic devices (SPLDs), whose most important char-

Terminology

CPLD (complex PLD): an arrangement of multiple SPLD-like blocks on a single chip. Alternative names are enhanced PLD (EPLD), superPAL, and megaPAL.

FPD (field-programmable device): any integrated circuit used for implementing digital hardware that allows the end user to configure the chip to realize different designs. Programming such a device often involves placing the chip into a special programming unit, but some chips can also be configured "in system." Another name for FPDs is programmable logic devices (PLDs), although PLDs are the same type of chips as FPDs, we prefer the term FPD because historically PLD denoted relatively simple devices.

FPGA (field-programmable gate array): an FPD featuring a general structure that allows very high logic capacity. Whereas CPLDs feature logic resources with a wide number of inputs (AND planes), FPGAs offer narrower logic resources. FPGAs also offer a higher ratio of flip-flops to logic resources than do CPLDs.

HCPLD (high-capacity PLD): term coined in trade literature refers to both CPLDs and FPGAs. We do not use this term here.

Interconnect: the wiring resources in an FPD.

Logic block: a relatively small circuit block replicated in an FPD array. A circuit implemented in an FPD is first decomposed into smaller subcircuits that can each be mapped into a logic block. The term occurs mostly in the context of FPGAs but can also refer to a block of circuitry in a CPLD.

Logic capacity: the amount of digital logic that we can map into a single FPD, usually measured in units of the equivalent number of gates in a traditional gate array. In other words, we measure an FPD's capacity as its comparable gate array size. Thus, we can refer to logic capacity as the number of two-input NAND gates.

Logic density: the amount of logic per unit area in an FPD.

PAL (programmable array logic): a relatively small FPD containing a programmable AND plane followed by a fixed-OR plane.

PLA (programmable logic array): a relatively small FPD that contains two levels of programmable logic—an AND plane and an OR plane. (Although PLA structures are sometimes embedded into full-custom chips, we refer here only to user-programmable PLAs provided as separate integrated circuits.)

Programmable switch: a user-programmable switch that can connect a logic element to an interconnect wire or one interconnect wire to another.

Speed performance: the maximum operable speed of a circuit implemented in an FPD. For combinational circuits, it is set by the longest delay through any path, and for sequential circuits, it is the maximum clock frequency at which the circuit functions properly

SPLD (simple PLD): usually a PLA or a PAL.

Authorized licensed use limited to: IEEE Publications Operations Staff. Downloaded on February 28,2023 at 16:58:07 UTC from IEEE Xplore. Restrictions apply.

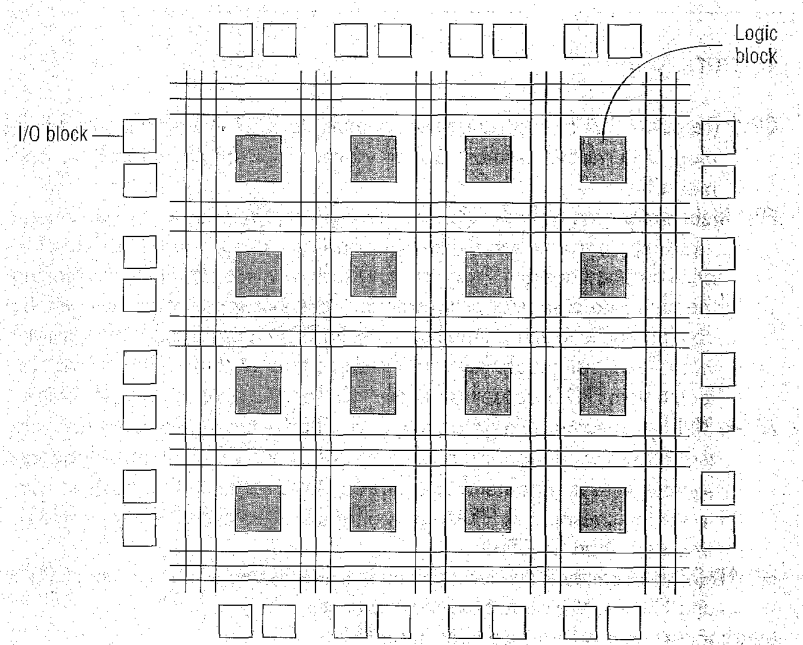


Figure 2. *FPGA structure.*

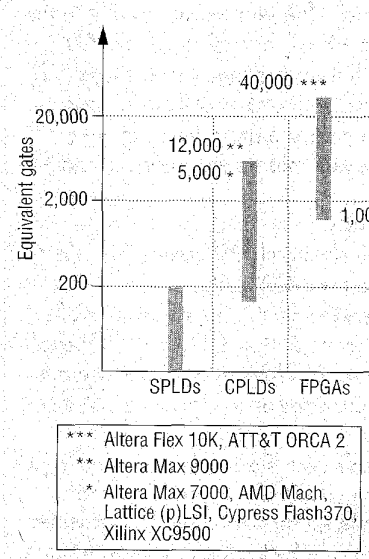


Figure 3. *FPD logic capacities.*

acteristics are low cost and very high pin-to-pin speed performance.

Advances in technology have produced devices with higher capacities than SPLDs. The difficulty with increasing a strict SPLD architecture's capacity is that the programmable-logic plane structure grows too quickly as the number of inputs increases. The only feasible way to provide large-capacity devices based on SPLD architectures is to programmably interconnnect multiple SPLDs on a single chip. Many FPD products on the market today have this basic structure and are known as complex programmable-logic devices.

Altera pioneered CPLDs, first in their Classic EPLD chips, and then in the Max 5000, 7000, and 9000 series. Because of a rapidly growing market for large FPDs, other manufacturers developed CPLD devices, and many choices are now available. CPLDs provide logic capacity up to the equivalent of about 50 typical SPLD devices, but extending these architectures to higher densities is difficult. Building FPDs with very high logic capacity requires a different approach.

The highest capacity general-purpose logic chips available today are the traditional gate arrays sometimes referred to as mask-programmable gate arrays. An MPGA consists of an array of prefabricated transistors customized for the user's logic circuit by means of wire connections. Because the silicon foundry performs customization during chip fabrication, the manufacturing time is long, and the user's setup cost is high.

Although MPGAs are clearly not FPDs, we mention them here because they motivated the design of the field-programmable equivalent, FPGAs. Like MPGAs, an FPGA consists of an array of uncommitted circuit elements (logic blocks) and interconnect resources, but the end user configures the FPGA through programming. Figure 2 shows a typical FPGA architecture. As the only type of FPD that supports very high logic capacity, FPGAs have engendered a major shift in digital-circuit design.

Figure 3 illustrates the logic capacities available in each FPD category. "Equivalent gates" refers loosely to the number of two-input NAND gates. The chart serves as a guide for selecting a device for an application according to the logic capacity needed. However, as we explain later, each type of FPD is inherently better suited for some applications than for others. There are also special-purpose devices optimized for specific applications (for example, state machines, analog gate arrays, large interconnection problems). Since such devices have limited use, we do not describe them here.

User-programmable switch technologies

User-programmable switches are the key to user customization of FPDs. The first user-programmable switch developed was the fuse used in PLAs. Although some smaller devices still use fuses, we will not discuss them here because newer technology is quickly replacing them. For higher density devices, CMOS dominates the IC industry, and different approaches to implementing programmable switches are necessary. For CPLDs, the main switch technologies (in commercial products)

Authorized licensed use limited to: IEEE Publications Operations Staff. Downloaded on February 28,2023 at 16:58:07 UTC from IEEE Xplore. Restrictions apply.

Table 1. *Summary of FPD programming technologies.*

Switch type	Reprogrammable?	Volatile?	Technology
Fuse	No	No	Bipolar
EPROM	Yes (out of circuit)	No	UVCMOS
EEPROM	Yes (in circuit)	No	EECMOS
SRAM	Yes (in circuit)	Yes	CMOS
Antifuse	No	No	CMOS+

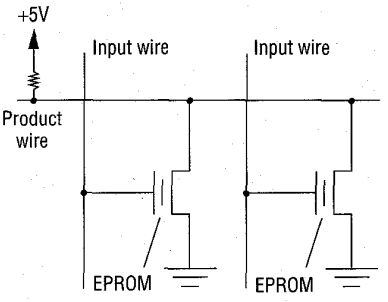


Figure 4. *EPROM programmable switches.*

are floating gate transistors like those used in EPROM (erasable programmable read-only memory) and EEPROM (electrically erasable PROM). For FPGAs, they are SRAM (static RAM) and antifuse. Table 1 lists the most important characteristics of these programming technologies.

To use an EPROM or EEPROM transistor as a programmable switch for CPLDs (and many SPLDs), the manufacturer places the transistor between two wires to facilitate implementation of wired-AND functions. Figure 4 shows EPROM transistors connected in a CPLD's AND plane. An input to the AND plane can drive a product wire to logic level 0 through an EPROM transistor, if that input is part of the corresponding product term. For inputs not involved in a product term, the appropriate EPROM transistors are programmed as permanently turned off. The diagram of an EEPROM-based device would look similar to the one in Figure 4.

Although no technical reason prevents application of EPROM or EEPROM to FPGAs, current commercial FPGA products use either SRAM or antifuse technologies. The example of SRAM-controlled switches in Figure 5 illustrates two applications, one to control the gate nodes of pass-transistor switches and the other, the select lines of multiplexers that drive logic block inputs. The figure shows the connection of one logic block (represented by the AND gate in the upper left corner) to another through two pass-transistor switches and then a multiplexer, all controlled by SRAM cells. Whether an FPGA uses pass transistors, multiplexers, or both depends on the particular product.

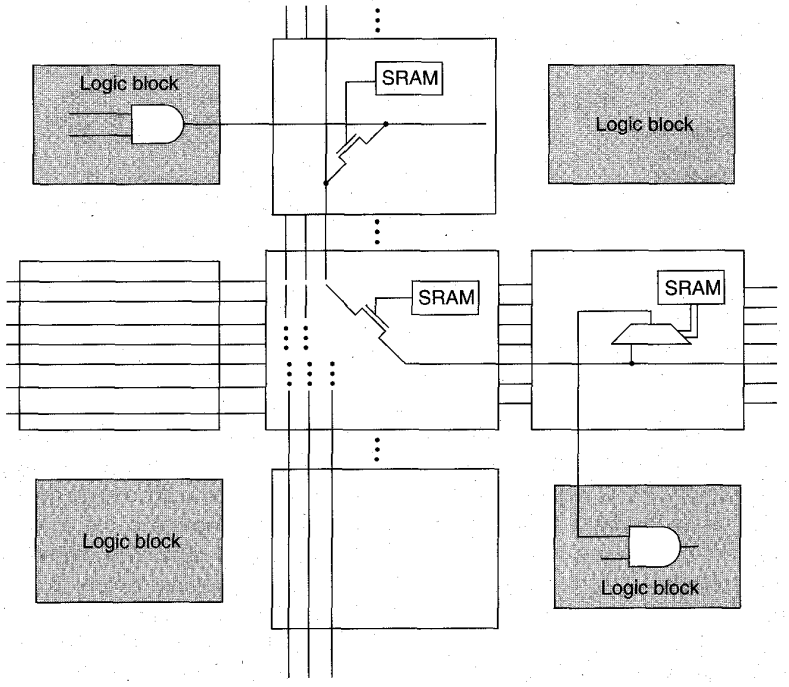


Figure 5. *SRAM-controlled programmable switches.*

Antifuses are originally open circuits that take on low resistance only when programmed. Antifuses are manufactured using modified CMOS technology. As an example, Figure 6 (next page) depicts Actel's PLICE (programmable-logic interconnect circuit element), an tifuse structure.[1] The antifuse, positioned between two interconnect wires, consists of three sandwiched layers: conductors at top and bottom and an insulator in the middle. Unprogrammed, the insulator isolates the top and bottom layers; programmed, the insulator becomes a low-resistance link.

Authorized licensed use limited to: IEEE Publications Operations Staff. Downloaded on February 28,2023 at 16:58:07 UTC from IEEE Xplore. Restrictions apply.

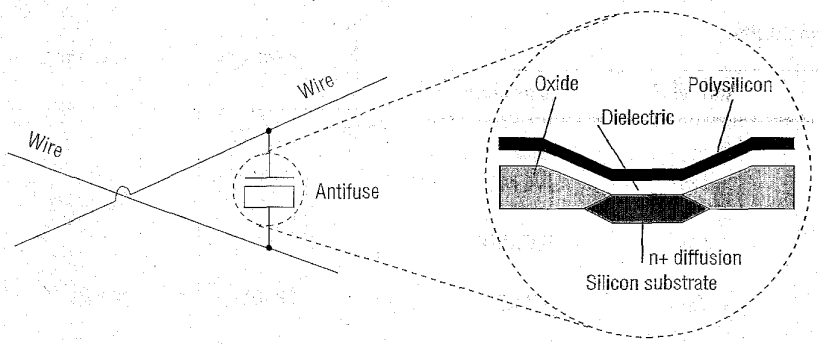


Figure 6. *Actel's PLICE antifuse structure.*

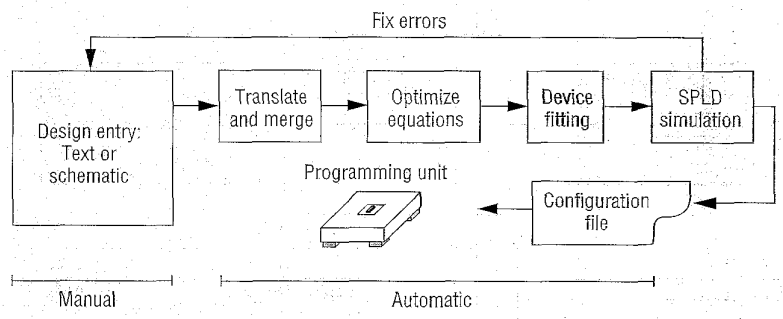


Figure 7. *CAD process for SPLDs.*

PLICE uses polysilicon and n+ diffusion as conductors and a custom-developed compound, ONO (oxide-nitride-oxide),[1] as an insulator. Other antifuses rely on metal for conductors, with amorphous silicon as the middle layer.[2,3]

CAD for FPDs

Computer-aided design programs are essential in designing circuits for implementation in FPDs. Such software tools are important not only for CPLDs and FPGAs, but also for SPLDs. A typical CAD system for SPLDs includes software for the following tasks: initial design entry, logic optimization, device fitting, simulation, and configuration. Figure 7 illustrates the SPLD design process. To enter a design, the designer creates a schematic diagram with a graphical CAD tool, describes the design in a simple hardware description language, or combines these methods. Since initial logic entry is not usually in an optimized form, the system applies algorithms to optimize the circuits. Then additional algorithms analyze the resulting logic equations and fit them into the SPLD. Simulation verifies correct operation, and the designer returns to the design entry step to fix errors. When a design simulates correctly, the designer loads it into a programming unit to configure an SPLD. In most CAD systems, the designer performs the original design entry step manually, and all other steps are automatic.

The steps involved in CPLD design are similar to those for SPLDs, but the CAD tools are more sophisticated. Because the devices are complex and can accommodate large designs, it is more common to use different design entry methods for different modules of a circuit. For instance, the designer might use a small hardware description language such as ABEL for some modules, a symbolic schematic capture tool for others, and a full-featured hardware description language such as VHDL for still others. Also, the device-fitting process may require steps similar to those described next for FPGAs, depending on the CPLD's sophistication. Either the CPLD manufacturer or a third party supplies the necessary software for these tasks.

The FPGA design process is similar to that of CPLDs but requires additional tools to support increased chip complexity. The major difference is in device fitting, for which FPGAs need at least three tools: a technology mapper to transform basic logic gates into the FPGA's logic blocks, a placement tool to choose the specific logic blocks, and a router to allocate wire segments to interconnect the logic blocks. With this added complexity, the CAD tools take a fairly long time (often more than an hour or even several hours) to complete their tasks.

Commercially available FPDs

This overview provides examples of commercial FPD products and their applications. We encourage readers interested in more details to contact the manufacturers or distributors for the latest data sheets. Most FPD manufacturers provide data sheets on the World Wide Web at http://www.company-name.com.

SPLDs. As a staple of digital hardware designers for the past two decades, SPLDs are very important devices. They have the highest speed performance of all FPDs and are inexpensive. Because they are straightforward and well understood, we discuss them only briefly here.

Two of the most popular SPLDs are the AMD (Advanced Micro Devices) 16R8 and 22V10 PALs. Both of these devices are industry standards, widely sec-

Authorized licensed use limited to: IEEE Publications Operations Staff. Downloaded on February 28,2023 at 16:58:07 UTC from IEEE Xplore. Restrictions apply.

ond-sourced by other companies. The designation 16R8 means that the PAL has a maximum of 16 inputs (eight dedicated inputs and eight input/outputs) and a maximum of eight outputs, and that each output is registered (R) by a D flip-flop. Similarly, the 22V10 has a maximum of 22 inputs and ten outputs. The V means versatile—that is, each output can be registered or combinational.

Another widely used and second-sourced SPLD is the Altera Classic EP610. This device is similar in complexity to PALs, but offers more flexibility in the production of outputs and has larger AND and OR planes. The EP610's outputs can be registered, and the flip-flops are configurable as D, T, JK, or SR.

Many other SPLD products are available from a wide array of companies. All share common characteristics such as logic planes (AND, OR, NOR, or NAND), but each offers unique features suitable for particular applications. A partial list of companies that offer SPLDs includes AMD, Altera, ICT, Lattice, Cypress, and Philips-Signetics. The complexity of some of these SPLDs approaches that of CPLDs.

CPLDs. As we said earlier, CPLDs consist of multiple SPLD-like blocks on a single chip. However, CPLD products are much more sophisticated than SPLDs, even at the level of their basic SPLD-like blocks. In the following descriptions, we present sufficient details to compare competing products, emphasizing the most widely used devices.

Altera Max. Altera has developed three families of CPLD chips: Max 5000, 7000, and 9000. We focus on the 7000 series because of its wide use and state-of-the-art logic capacity and speed performance. Max 5000 represents an older technology that offers a cost-effective solution; Max 9000 is similar to Max 7000 but offers higher logic capacity (the industry's highest for CPLDs).

Figure 8 depicts the general architecture of the Altera Max 7000 series. It consists of an array of logic array blocks and a set of interconnect wires called a programmable interconnect array (PIA). The PIA can connect any logic array block input or output to any other logic array block. The chip's inputs and outputs connect directly to the PIA and to logic array blocks. A logic array block is a complex, SPLD-like structure, and so we can consider the entire chip an array of SPLDs.

Figure 9 shows the structure of a logic array block. Each logic array block

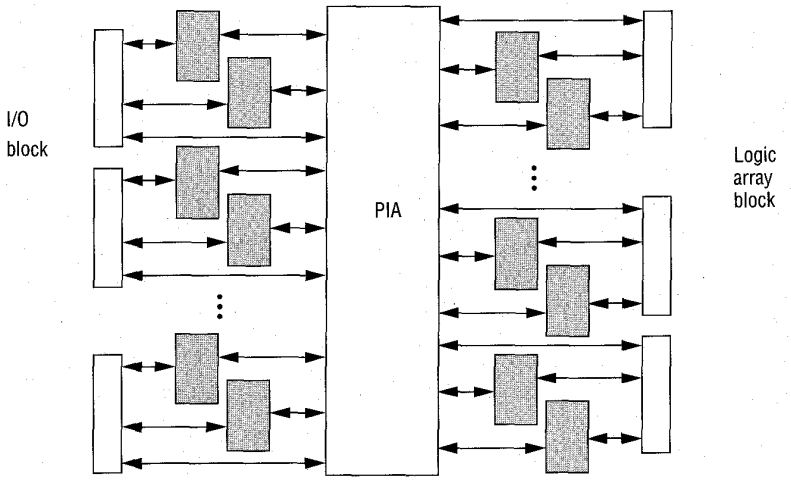


Figure 8. *Altera Max 7000 series architecture.*

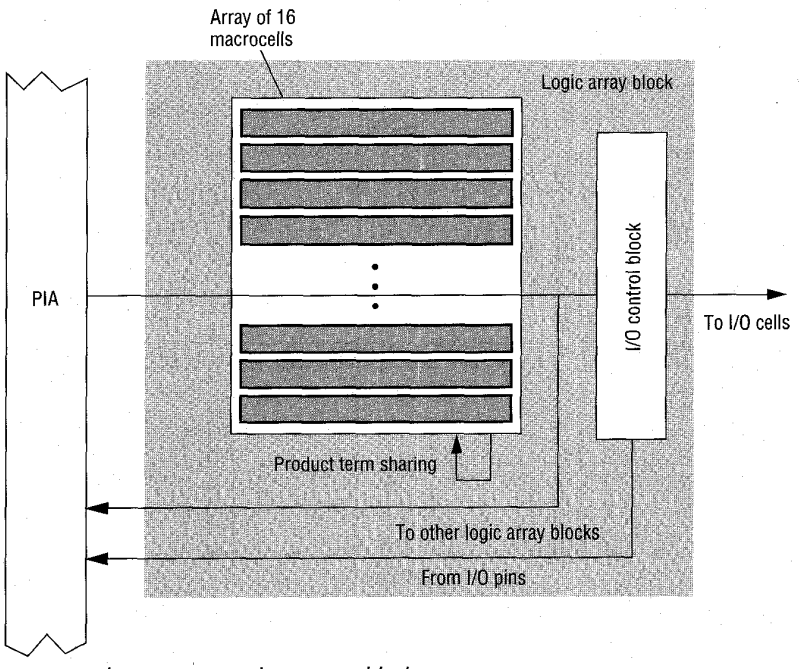


Figure 9. *Altera Max 7000 logic array block.*

Authorized licensed use limited to: IEEE Publications Operations Staff. Downloaded on February 28,2023 at 16:58:07 UTC from IEEE Xplore. Restrictions apply.

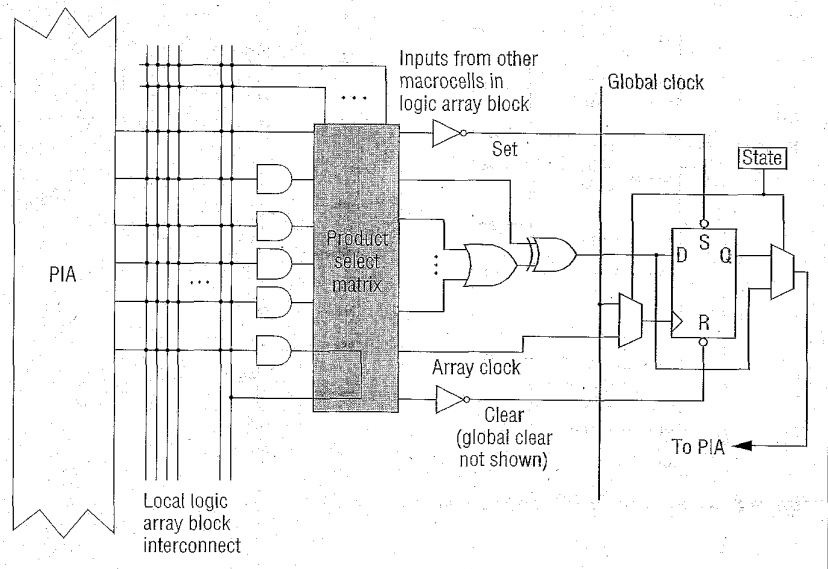


Figure 10. *Max 7000 macrocell.*

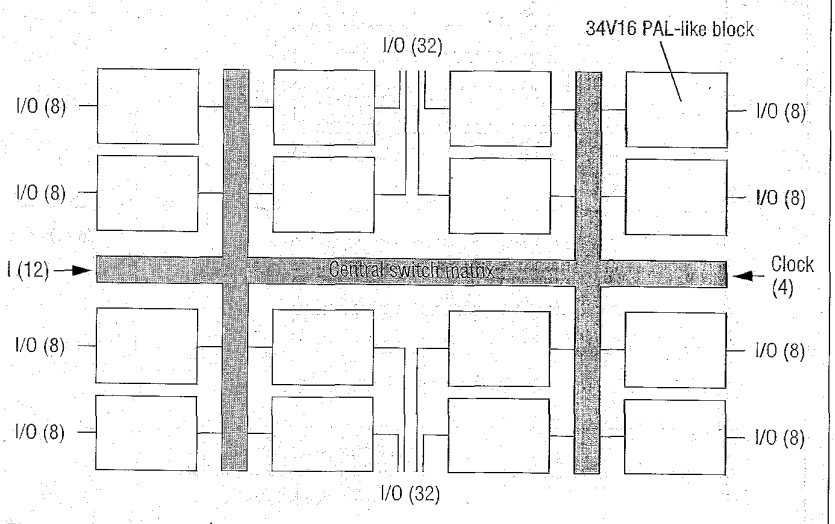


Figure 11. *AMD Mach 4 structure.*

consists of two sets of eight macrocells (shown in Figure 10). A macrocell is a set of programmable product terms (part of an AND plane) that feeds an OR gate and a flip-flop. The flip-flops can be D, JK, T, or SR, or can be transparent. As Figure 10 shows, the product select matrix allows a variable number of inputs to the OR gate in a macrocell. Any or all of the five product terms in the macrocell can feed the OR gate, which can have up to 15 extra product terms from macrocells in the same logic array block. This product term flexibility makes the Max 7000 series more efficient in chip area than classic SPLDs, because typical logic functions need no more than five product terms, and the architecture supports wider functions when necessary. Variable-size OR gates of this sort are not available in basic SPLDs (see Figure 1), but similar features exist in other CPLD architectures.

Max 7000 devices are available in both EPROM and EEPROM technologies. Until recently, even with EEPROM, Max 7000 chips were programmable only out of circuit in a special-purpose programming unit; in 1996, however, Altera released the 7000S series, which is in-circuit reprogrammable.

AMD Mach. AMD offers a CPLD family comprising five subfamilies called Mach. Each Mach device consists of multiple PAL-like blocks (or optimized PALs). Mach 1 and 2 consist of optimized 22V16 PALs, Mach 3 and 4 consist of several optimized 34V16 PALs, and Mach 5 is similar to Mach 3 and 4 but offers enhanced speed performance. All Mach chips use EEPROM technology, and together the five subfamilies provide a wide range of selection, from small, inexpensive chips to larger, state-of-the-art ones. We will focus on Mach 4 because it represents the most advanced currently available parts in the family.

Figure 11 depicts a Mach 4 chip, showing the multiple 34V16 PAL-like blocks and the interconnect, called the central switch matrix. The in-circuit programmable chips range in size from 6 to 16 PAL-like blocks, corresponding roughly to 2,000 to 5,000 equivalent gates. All connections between PAL-like blocks (even from a PAL-like block to itself) pass through the central switch matrix. Thus, the device is not merely a collection of PAL-like blocks but a single, large device. Since all connections travel through the same path, circuit timing delays are predictable.

Figure 12 illustrates a Mach 4 PAL-like block. It has 16 outputs and a total of 34 inputs (16 of which are the fed-back outputs), so it corresponds to a 34V16 PAL. However, there are two key differences

Authorized licensed use limited to: IEEE Publications Operations Staff. Downloaded on February 28,2023 at 16:58:07 UTC from IEEE Xplore. Restrictions apply.

between this block and a normal PAL: 1) a product term (PT) allocator between the AND plane and the macrocells (the macrocells comprise an OR gate, an EXOR gate, and a flip-flop), and 2) an output switch matrix between the OR gates and the I/O pins. These features make a Mach 4 chip easier to use because they decouple sections of the PAL-like block. More specifically, the product term allocator distributes and shares product terms from the AND plane to OR gates that require them, allowing much more flexibility than the fixed-size OR gates in regular PALs. The output switch matrix enables any macrocell output (OR gate or flip-flop) to drive any I/O pin connected to the PAL-like block, again providing greater flexibility than a PAL, in which each macrocell can drive only one specific I/O pin. Mach 4's combination of in-system programmability and high flexibility allow easy hardware design changes.

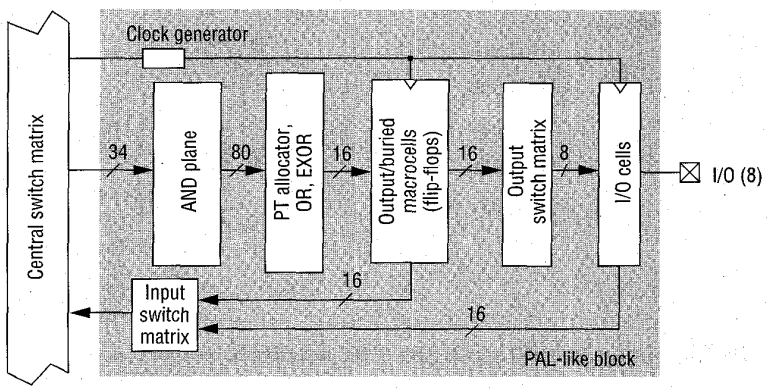


Figure 12. *Mach 4 34V16 PAL-like block.*

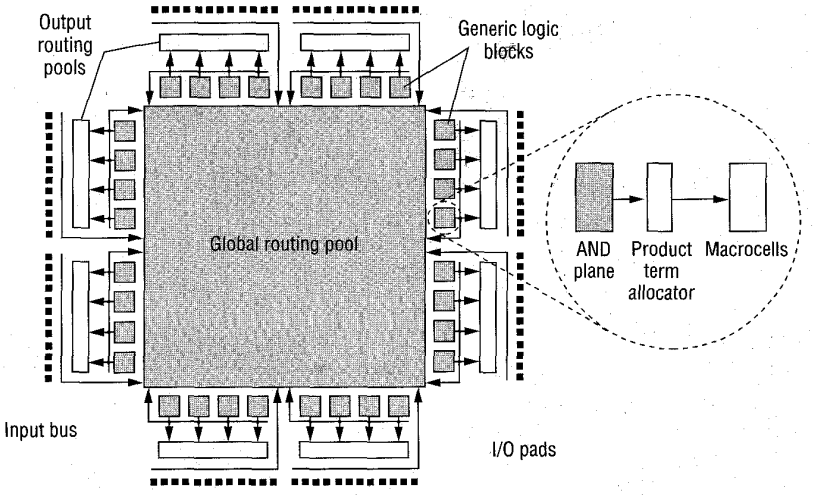


Figure 13. *Lattice pLSI and ispLSI architecture.*

Lattice pLSI and ispLSI. Lattice offers a complete range of CPLDs, with two main product lines: the pLSI and the ispLSI. Each consists of three families of EEPROM CPLDs with different logic capacities and speed performance. The ispLSI devices are in-system programmable.

Lattice's earliest generation of CPLDs is the pLSI and ispLSI 1000 series. Each chip consists of a collection of SPLD-like blocks and a global routing pool to connect the blocks. Logic capacity ranges from about 1,200 to 4,000 gates, and pin-to-pin delays are 10 ns. Lattice also offers the 2000 series—relatively small CPLDs with between 600 and 2,000 gates. The 2000 series features a higher ratio of macrocells to I/O pins and higher speed performance than the 1000 series. At 5.5-ns pin-to-pin delays, the 2000 series provides state-of-the-art speed.

Lattice's 3000 series consists of the company's largest CPLDs, with up to 5,000 gates and 10- to 15-ns pin-to-pin delays. Compared with the chips discussed so far, the functionality of the 3000 series is most similar to that of the Mach 4. Unlike the other Lattice CPLDs, the 3000 series offers enhancements to support more recent design styles, such as IEEE Std 1149.1 boundary scan.

Figure 13 shows the general structure of a Lattice pLSI or ispLSI device. Around the chip's outside edges are bidirectional I/Os, which connect to both the generic logic blocks and the global routing pool. As the magnified view on the right side of the figure shows, the generic logic blocks are small PAL-like blocks consisting of an AND plane, a product term allocator, and macrocells. The global routing pool is a set of wires that span the chip to connect generic logic block inputs and outputs. All interconnects pass through the global routing pool, so timing between logic levels is fully predictable, as it is for the AMD Mach devices.

Cypress Flash370. Cypress has recently developed CPLD products similar to the AMD and Lattice devices in several ways. Cypress Flash370 CPLDs

Authorized licensed use limited to: IEEE Publications Operations Staff. Downloaded on February 28,2023 at 16:58:07 UTC from IEEE Xplore. Restrictions apply.

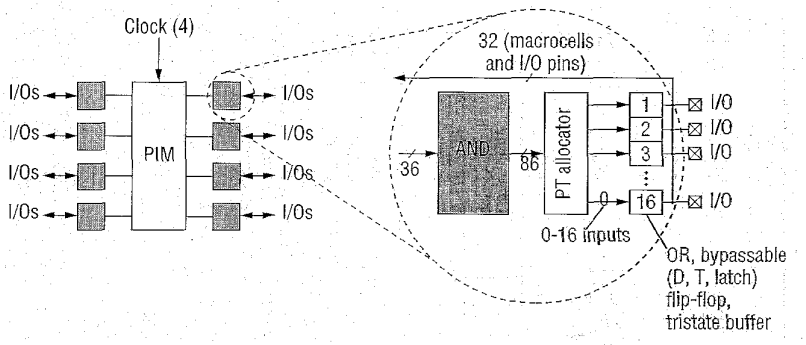


Figure 14. *Cypress Flash370 architecture. (PIM: programmable interconnect matrix.)*

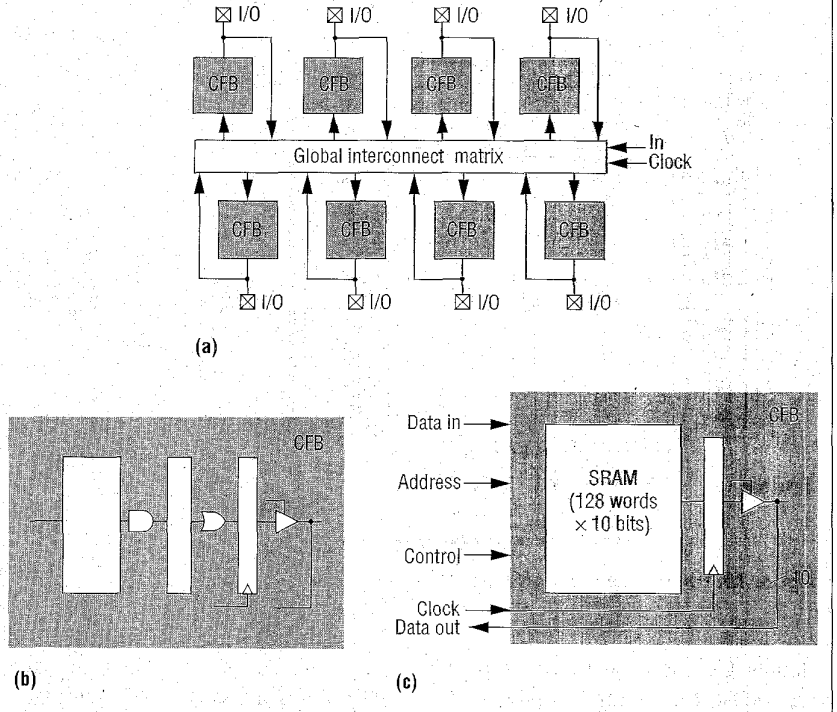


Figure 15. *Altera Flashlogic CPLD: general architecture* ***(a)****; CFB in PAL mode* ***(b)****; CFB in SRAM mode* ***(c)****.*

use flash EEPROM technology and offer speed performance of 8.5 to 15 ns pin-to-pin delays. The Flash370s are not in-system programmable. To meet the needs of larger chips, the devices provide more I/O pins than competing products, with a linear relationship between the number of macrocells and the number of bidirectional I/O pins. The smallest parts have 32 macrocells and 32 I/O pins; the largest have 256 macrocells and 256 pins.

Figure 14 shows that Flash370s have a typical CPLD architecture with multiple PAL-like blocks connected by a programmable interconnect matrix. Each PAL-like block contains an AND plane that feeds a product term allocator that directs from 0 to 16 product terms to each of 32 OR gates. The feedback path from the macrocell outputs to the programmable interconnect matrix contains 32 wires. This means that a macrocell can be buried (not drive an I/O pin), and yet the I/O pin that the macrocell would have driven can still serve as an input. This capability is another type of flexibility available in PAL-like blocks but not in normal PALs.

Xilinx XC7000. Although primarily a manufacturer of FPGAs, Xilinx also offers the XC7000 series of CPLDs. The two main XC7000 families are the 7200 series (originally marketed by Plus Logic as Hiper EPLDs) and the 7300 series developed by Xilinx. The 7200s are moderately small devices with about a 600 to 1,500 gate capacity, and they offer speed performance of about 25-ns pin-to-pin delays. Each chip consists of a collection of SPLD-like blocks containing nine macrocells each. Unlike those in other CPLDs, a macrocell includes two OR gates, each of which becomes an input for a 2-bit arithmetic logic unit. The ALU can produce any functions of its two inputs, and its output feeds a configurable flip-flop. The 7300 series is an enhanced version of the 7200 with greater capacity (up to 3,000 gates) and higher speed performance. Xilinx also has announced a new CPLD family, the XC9500, which will offer in-circuit programmability with 5-ns pin-to-pin delays and up to 6,200 logic gates.

Altera Flashlogic. Previously known as Intel's Flexlogic, these devices feature in-system programmability and on-chip SRAM blocks, a unique feature among CPLD products. Figure 15a illustrates the Flashlogic architecture, a collection of PAL-like blocks called configurable function blocks (CFBs), each of which represents an optimized 24V10 PAL.

Flashlogic's basic structure is similar to other products already discussed. However, one feature sets it apart from

Authorized licensed use limited to: IEEE Publications Operations Staff. Downloaded on February 28,2023 at 16:58:07 UTC from IEEE Xplore. Restrictions apply.

all other CPLDs: Instead of containing AND/OR logic, a CFB can serve as a 10-ns SRAM block. Figure 15b shows a CFB configured as a PAL, and Figure 15c shows another configured as an SRAM. In the SRAM configuration, the PAL block becomes a 128-word by 10-bit read/write memory. Inputs that would normally feed the AND plane in the PAL become address lines, data lines, or control signals for the memory. Flip-flops and tristate buffers are still available in the SRAM configuration.

In the Flashlogic device, the AND/OR logic plane's configuration bits are SRAM cells connected to EPROM or EEPROM cells. Applying power loads the SRAM cells with a copy of the nonvolatile EPROM or EEPROM, but the SRAM cells control the chip's configuration. The user can reconfigure the chips in system by downloading new information into the SRAM cells. The user can make the SRAM cell reprogramming nonvolatile by writing the SRAM cell contents back to the EPROM cells.

ICT PEEL Arrays. ICT PEEL (programmable, electrically-erasable logic) Arrays are large PLAs that include logic macrocells with flop-flops and feedback to the logic planes. Figure 16 illustrates this structure, which consists of a programmable AND plane that feeds a programmable OR plane. The OR plane's outputs are partitioned into groups of four, and each group can be input to any of the logic cells. The logic cells provide registers for the sum terms and can feed back the sum terms to the AND plane. Also, the logic cells connect sum terms to I/O pins.

Because they have a PLA-like structure, the logic capacity of PEEL Arrays is difficult to measure compared to the CPLDs discussed so far, but we estimate a capacity of 1,600 to 2,800 equivalent gates. Containing relatively few I/O pins, the largest PEEL Array comes in a 40-pin package. Since they do not consist of SPLD-like blocks, PEEL Arrays do not fit

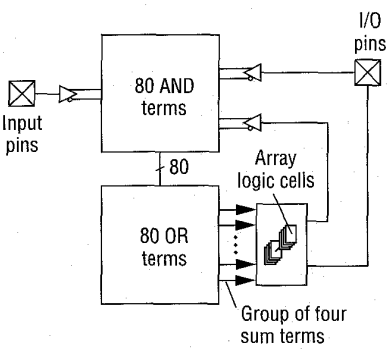


Figure 16. *ICT PEEL Array architecture.*

well into the CPLD category. Nevertheless, we include them here because they exemplify PLA-based (rather than PAL-based) devices and offer larger capacity than a typical SPLD.

The PEEL Array logic cell, shown in Figure 17, includes a flip-flop, configurable as D, T, or JK, and two multiplexers. Each multiplexer produces a logic cell output, either registered or combinational. One logic cell output can connect to an I/O pin, and the other output is buried. An interesting feature of the logic cell is that the flip-flop clock, preset, and clear are full sum-of-product logic functions. Distinguishing PEEL Arrays from all other CPLDs, which simply provide product terms for these signals, this feature is attractive for some applications. Because of their PLA-like OR plane, PEEL Arrays are especially well suited to applications that require very wide sum terms.

CPLD applications. Their high speeds and wide range of capacities make CPLDs useful for many applications, from implementing random glue logic to prototyping small gate arrays. An important reason for the growth of the CPLD market is the conversion of designs that consist of multiple SPLDs into a smaller number of CPLDs.

CPLDs can realize complex designs such as graphics, LAN, and cache controllers. As a rule of thumb, circuits that

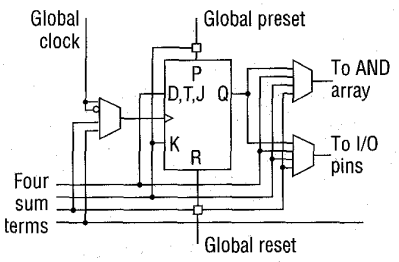


Figure 17. *ICT PEEL Array logic cell structure.*

can exploit wide AND/OR gates and do not need a large number of flip-flops are good candidates for CPLD implementation. Finite state machines are an excellent example of this class of circuits. A significant advantage of CPLDs is that they allow simple design changes through reprogramming (all commercial CPLD products are reprogrammable). In-system programmable CPLDs even make it possible to reconfigure hardware (for example, change a protocol for a communications circuit) without powering down.

Designs often partition naturally into the SPLD-like blocks in a CPLD, producing more predictable speed performance than a design split into many small pieces mapped into different areas of the chip. Predictability of circuit implementation is one of the strongest advantages of CPLD architectures.

FPGAs. As one of the fastest growing segments of the semiconductor industry, the FPGA marketplace is volatile. The pool of companies involved changes rapidly, and it is difficult to say which products will be most significant when the industry reaches a stable state. We focus here on products currently in widespread use. In describing each device, we list its capacity in two-input NAND gates as given by the vendor. Gate count is an especially contentious issue in the FPGA industry, and so the numbers given should not be taken too seriously. In fact, wags

Authorized licensed use limited to: IEEE Publications Operations Staff. Downloaded on February 28,2023 at 16:58:07 UTC from IEEE Xplore. Restrictions apply.

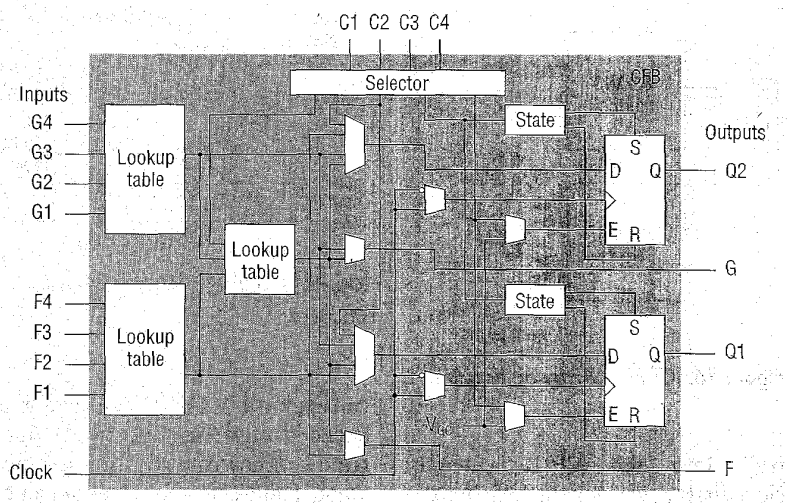


Figure 18. *Xilinx XC4000 CLB.*

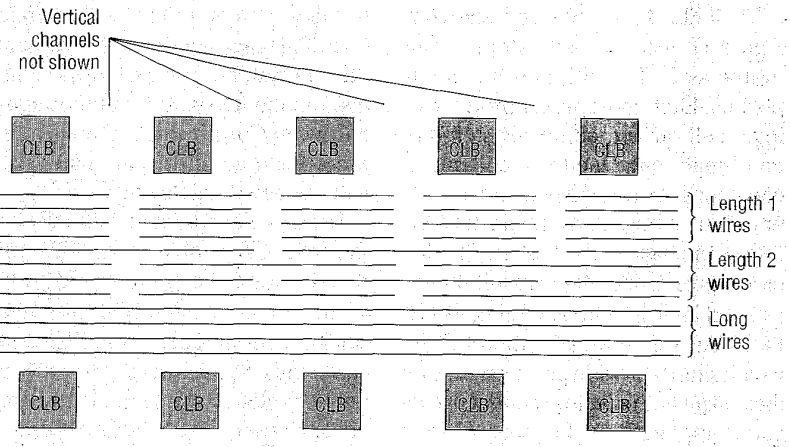


Figure 19. *Xilinx XC4000 wire segments.*

have coined the term "dog gates," a reference to the often-cited ratio between human and dog years, to indicate the dubiousness of vendors' figures.

The two basic categories of FPGAs on the market today are SRAM- and antifuse-based FPGAs. In the first category, Xilinx and Altera lead in number of users, their major competitor being AT&T. For antifuse-based products, Actel, Quicklogic, and Cypress are the leading manufacturers.

Xilinx FPGAs. Xilinx FPGAs have an array-based structure, each chip comprising a two-dimensional array of logic blocks interconnected by horizontal and vertical routing channels (see Figure 2). Xilinx introduced the first FPGA series, the XC2000, in about 1985 and now offers three more generations: XC3000, XC4000, and XC5000. Although the XC3000 devices are still widely used, we focus on the more recent and more popular XC4000 family. The XC4000 devices range in capacity from about 2,000 to more than 15,000 equivalent gates. The XC5000 family provides similar features at a more attractive price with some penalty in speed. Xilinx has recently announced an antifuse-based FPGA family, the XC8100. The XC8100 has many interesting features, but since it is not yet in widespread use, we do not discuss it here.

The XC4000 features a configurable logic block (CLB) based on lookup tables. A lookup table is a 1-bit-wide memory array; the memory address lines are logic block inputs, and the 1-bit memory output is the lookup table output. A lookup table with K inputs corresponds to a $2^K \times 1$-bit memory, and the user can realize any K-input logic function by programming the logic function's truth table directly into the memory. In the configuration shown in Figure 18, an XC4000 CLB contains two four-input lookup tables fed by CLB inputs, and a third lookup table fed by the other two. This arrangement allows the CLB to implement a wide range of logic functions of up to nine inputs, two separate four-input functions, or other possibilities. Each CLB also contains two flip-flops.

The XC4000 chips have features designed to support the integration of entire systems. For instance, each CLB contains circuitry that allows it to efficiently perform arithmetic (that is, a circuit that implements a fast carry operation for adder-like circuits). Also, users can configure the lookup tables as read/write RAM cells. A new addition, the 4000E allows configuration as a dual-port RAM with a single write and two read ports, and RAM blocks can be synchronous RAM. Each XC4000 chip includes very wide AND planes around the periphery of the logic block array to facilitate implementation of circuit blocks such as wide decoders.

Besides its logic, the other key feature that distinguishes an FPGA is its interconnect structure. Horizontal and vertical channels characterize the XC4000 interconnect. Each channel contains

Authorized licensed use limited to: IEEE Publications Operations Staff. Downloaded on February 28,2023 at 16:58:07 UTC from IEEE Xplore. Restrictions apply.

short wire segments that span a single CLB (the number of segments in each channel varies for each member of the XC4000 family), longer segments that span two CLBs, and very long segments that span the chip's entire length or width. Programmable switches are available (see Figure 5) to connect CLB inputs and outputs to the wire segments or to connect one wire segment to another. A small section of an XC4000 routing channel appears in Figure 19. The figure shows only the wire segments in a horizontal channel—not the vertical routing channels, CLB inputs and outputs, and the routing switches. An important point about the Xilinx interconnect is that signals must pass through switches to reach one CLB from another, and the total number of switches traversed depends on the particular set of wire segments used. Thus, an implemented circuit's speed performance depends in part on how CAD tools allocate the wire segments to individual signals.

Altera Flex 8000 and Flex 10K. Altera's Flex 8000 series combines FPGA and CPLD technologies. The devices consist of a three-level hierarchy much like that of CPLDs. However, the lowest level of the hierarchy is a set of lookup tables, rather than an SPLD-like block, and so we categorize the Flex 8000 as an FPGA. The SRAM-based Flex 8000 features a four-input lookup table as its basic logic block. Logic capacity of the 8000 series ranges from about 4,000 to more than 15,000 gates.

Figure 20 illustrates the overall Flex 8000 architecture. The basic logic block, called a logic element, contains a four-input lookup table, a flip-flop, and special-purpose carry circuitry for arithmetic circuits (similar to the Xilinx XC4000). The logic element also includes cascade circuitry that allows efficient implementation of wide AND functions. Figure 21 shows details of the logic element.

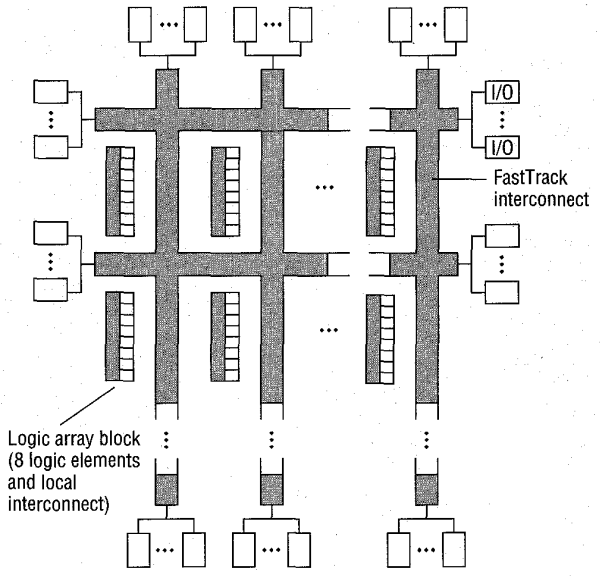


Figure 20. *Altera Flex 8000 architecture.*

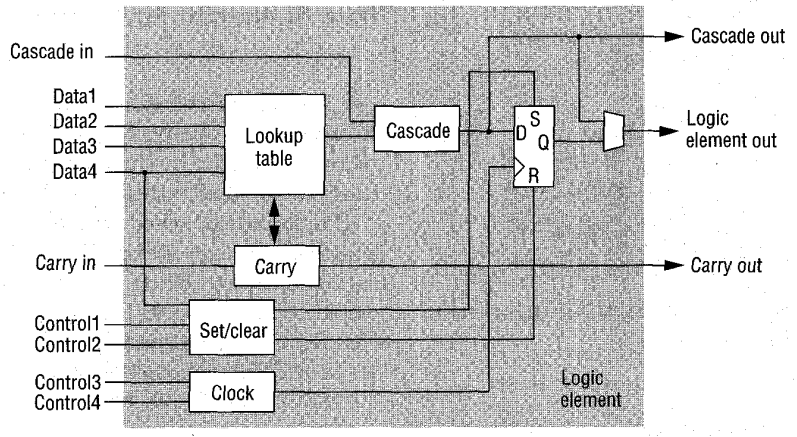


Figure 21. *Flex 8000 logic element.*

This design groups logic elements into sets of eight, called logic array blocks (a term borrowed from Altera's CPLDs). As shown in Figure 22 on the next page, each logic array block contains local interconnection, and each local wire can connect any logic element to any other logic element within the same logic array block. The local interconnect also connects to the Flex 8000's FastTrack global interconnect. Like the long wires in the Xilinx XC4000, each FastTrack wire extends the full width or height of the device. However, a major difference between Flex 8000 and Xilinx chips is that FastTrack consists only of long lines, making the Flex 8000 easy for CAD tools to configure automatically. All FastTrack horizontal wires are identical. Therefore, interconnect delays in the Flex 8000 are more predictable than in FPGAs that employ many shorter segments because

Authorized licensed use limited to: IEEE Publications Operations Staff. Downloaded on February 28,2023 at 16:58:07 UTC from IEEE Xplore. Restrictions apply.

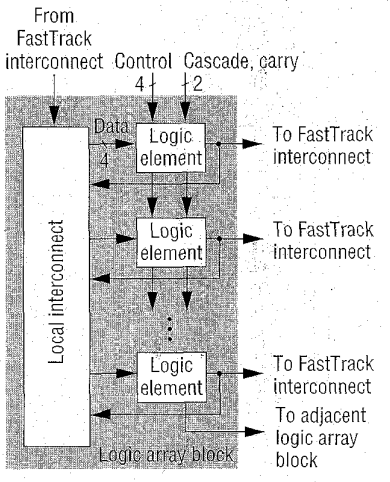


Figure 22. *Flex 8000 logic array block.*

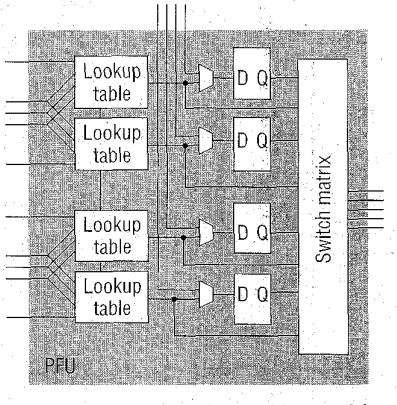


Figure 24. *AT&T ORCA programmable-function unit.*

the longer paths contain fewer programmable switches. Moreover, connections between horizontal and vertical lines pass through active buffers, further enhancing predictability.

The Flex 10K family offers all the Flex 8000 features with the addition of variable-size blocks of SRAM called embedded array blocks. As Figure 23 shows, each row of a Flex 10K chip has an embedded array block on one end. Users can configure each embedded array block to serve as an SRAM block with a variable aspect ratio: 256×8, 512×4, 1K×2, or 2K×1. Alternatively, CAD tools

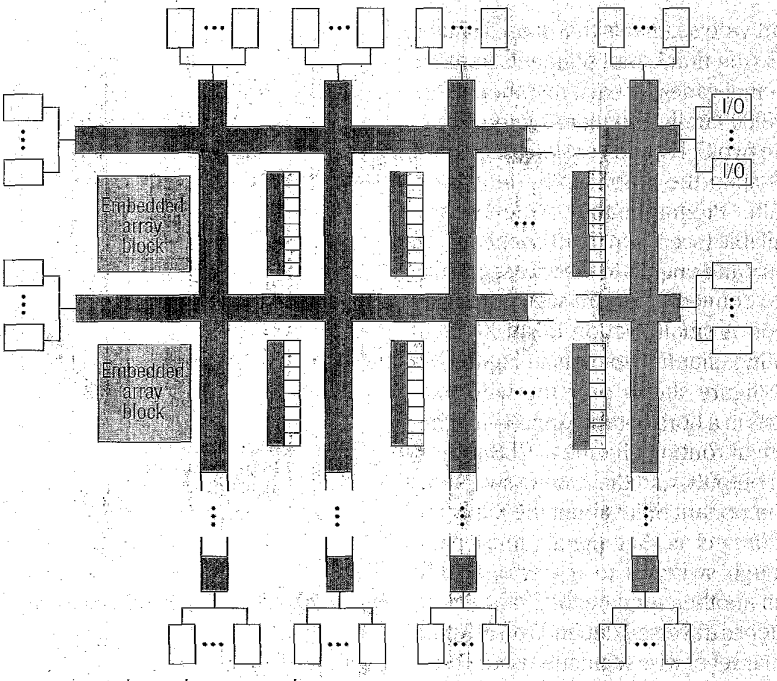


Figure 23. *Altera Flex 10K architecture.*

can configure an embedded array block to implement a complex logic circuit, such as a multiplier, by employing it as a large, multioutput lookup table. Altera CAD tools provide several macrofunctions that implement useful logic circuits in embedded array blocks. Counting the embedded array blocks as logic gates, Flex 10K offers the highest logic capacity of any FPGA, although obtaining an accurate number is difficult.

AT&T ORCA. AT&T's SRAM-based FPGAs, called Optimized Reconfigurable Cell Arrays (ORCAs), feature an overall structure similar to that of Xilinx FPGAs. The ORCA logic block contains an array of programmable-function units (Figure 24) based on lookup tables. A programmable-function unit is unique among lookup-table-based logic blocks. It is configurable as four 4-input lookup tables, two 5-input lookup tables, or one 6-input lookup table. A key element of this architecture is that when the programmable-function unit serves as four 4-input lookup tables, several of the lookup tables' inputs must come from the same programmable-function unit input. While this restraint reduces the programmable-function unit's flexibility, it also significantly reduces the chip's wiring cost. The programmable-function unit includes arithmetic circuitry, as do the Xilinx XC4000 and the Altera Flex 8000, and like the XC4000, is configurable as a RAM block. A recently announced version of the ORCA chip also allows dual-port and synchronous RAM.

ORCA's interconnect structure is also different from other SRAM-based FPGAs. Each programmable-function unit connects to an interconnect configured in four-bit buses. This structure supports system level designs more efficiently, since buses are common in such applications.

The ORCA2 series extends the family, offering a capacity of up to 40,000 logic gates. ORCA2 features a two-level hierarchy of programmable-function

Authorized licensed use limited to: IEEE Publications Operations Staff. Downloaded on February 28,2023 at 16:58:07 UTC from IEEE Xplore. Restrictions apply.

units based on the original ORCA architecture.

Actel FPGAs. Actel offers three main FPGA families: Act 1, Act 2, and Act 3. Although the three generations have similar features, we focus on the most recent devices. Unlike the FPGAs described so far, Actel's devices use antifuse technology and a structure similar to traditional gate arrays. Their design arranges logic blocks in rows with horizontal routing channels between adjacent rows (Figure 25). Actel logic blocks, based on multiplexers, are small compared to those based on lookup tables. Figure 26 illustrates the Act 3 logic block, which consists of an AND and an OR gate connected to a multiplexer-based circuit block. In combination with the two logic gates, the arrangement of the multiplexer circuit enables a single logic block to realize a wide range of functions. About half the logic blocks in an Act 3 device also contain a flip-flop.

Actel's horizontal routing channels consist of various-length wire segments with antifuses to connect logic blocks to wire segments or one wire to another. Although not shown in Figure 25, vertical wires also overlie the logic blocks, forming signal paths that span multiple rows. The speed performance of Actel chips is not fully predictable because the number of antifuses traversed by a signal depends on how CAD tools allocate the wire segments during circuit implementation. However, a rich selection of wire segment lengths in each channel and algorithms that guarantee strict limits on the number of antifuses traversed by any two-point connection improve speed performance significantly.

Quicklogic pASIC. Actel's main competitor in antifuse-based FPGAs is Quicklogic, which has two device families, pASIC and pASIC2. The pASIC, illustrated in Figure 27a, has similarities

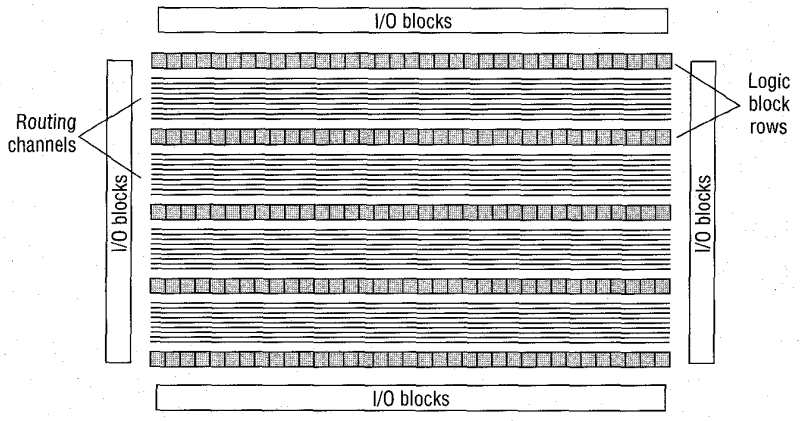


Figure 25. *Actel FPGA structure.*

to several other FPGAs: Like Xilinx FPGAs, it has an array-based structure; like Actel FPGAs, its logic blocks use multiplexers; and like Altera Flex 8000s, its interconnect consists only of long lines. The pASIC2 is a recently introduced enhanced version, which we will not discuss here. Cypress also offers devices using the pASIC architecture, but we discuss only Quicklogic's version.

Quicklogic's ViaLink antifuse structure (see Figure 27b) consists of a metal top layer, an amorphous-silicon insulat-

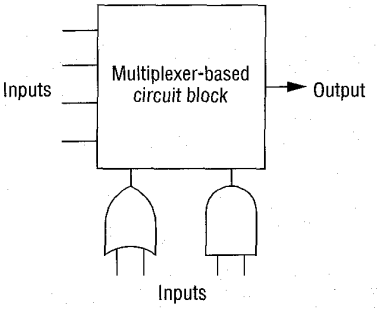


Figure 26. *Actel Act 3 logic module.*

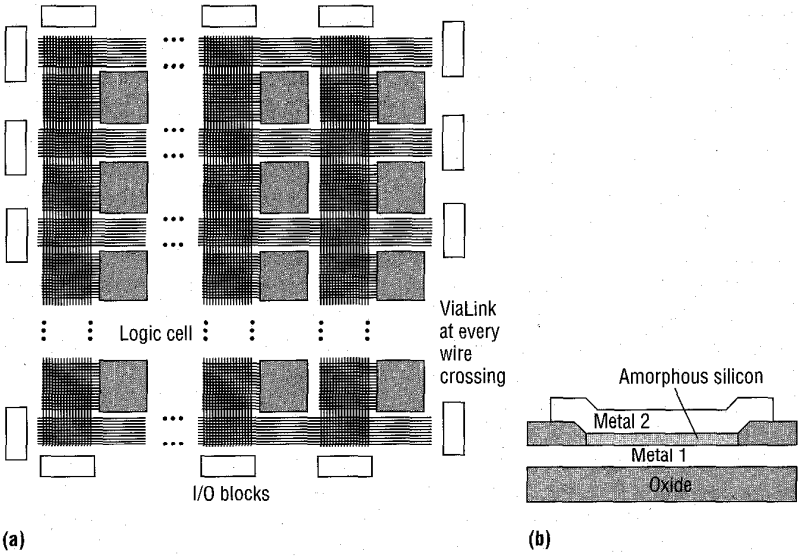


Figure 27. *Quicklogic pASIC structure* ***(a)*** *and ViaLink* ***(b)****.*

Authorized licensed use limited to: IEEE Publications Operations Staff. Downloaded on February 28,2023 at 16:58:07 UTC from IEEE Xplore. Restrictions apply.

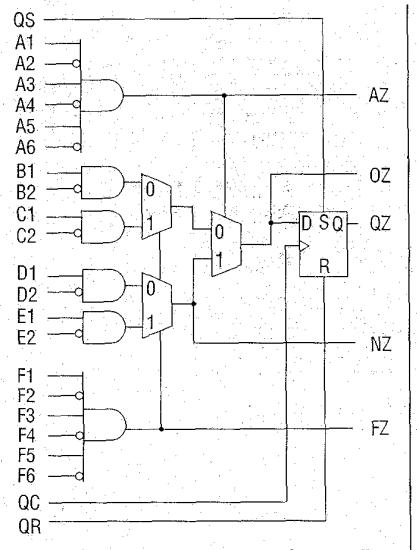


Figure 28. *Quicklogic pASIC logic cell.*

ing layer, and a metal bottom layer. Compared to Actel's PLICE antifuse, ViaLink offers very low on-resistance—about 50 ohms (PLICE's is about 300 ohms)—and a low parasitic capacitance. ViaLink antifuses are present at every crossing of logic block pins and interconnect wires, providing generous connectivity. Figure 28 shows the pASIC multiplexer-based logic block. It is more complex than Actel's logic module, with more inputs and wide (six-input) AND gates on the multiplexer select lines. Every logic block also contains a flip-flop.

FPGA applications. FPGAs have gained rapid acceptance over the past decade because users can apply them to a wide range of applications: random logic, integrating multiple SPLDs, device controllers, communication encoding and filtering, small- to medium-size systems with SRAM blocks, and many more.

Another interesting FPGA application is prototyping designs to be implemented in gate arrays by using one or more large FPGAs. (A large FPGA corresponds to a small gate array in terms of capacity). Still another application is the emulation of entire large hardware systems via the use of many interconnected FPGAs. QuickTurn[4] and others have developed products consisting of the FPGAs and software necessary to partition and map circuits for hardware emulation.

An application only beginning development is the use of FPGAs as custom computing machines. This involves using the programmable parts to execute software, rather than compiling the software for execution on a regular CPU. For information, we refer readers to the proceedings of the IEEE Workshop on FPGAs for Custom Computing Machines, held for the last four years.[5]

As mentioned earlier, pieces of designs often map naturally to the SPLD-like blocks of CPLDs. However, designs mapped into an FPGA break up into logic-block-size pieces distributed through an area of the FPGA. Depending on the FPGA's interconnect structure, the logic block interconnections may produce delays. Thus, FPGA performance often depends more on how CAD tools map circuits into the chip than does CPLD performance.

THE LOW COST OF FPDs makes them attractive to small firms and large companies alike. Their fast manufacturing turnaround is an essential element of their market success. Although their large, slow programmable switches prevent FPDs from providing the speed performance and logic capacity of MPGAs, improvements in architecture and CAD tools will overcome these disadvantages. Over time FPDs will become the dominant technology for implementing digital circuits. D&T

Acknowledgments

We acknowledge students, colleagues, and acquaintances in industry who have contributed to our knowledge.

References

1. E. Hamdy et al., "Dielectric-Based Antifuse for Logic and Memory ICs," *Tech. Digest IEEE Int'l Electron Devices Meeting*, IEEE, Piscataway, N.J., 1988, pp. 786-789.
2. J. Birkner et al., "A Very-High-Speed Field-Programmable Gate Array Using Metal-to-Metal Antifuse Programmable Elements," *Microelectronics J.*, Vol. 23, 1992, pp. 561-568.
3. D. Marple and L. Cooke, "Programming Antifuses in CrossPoint's FPGA," *Proc. IEEE Int'l Custom Integrated Circuits Conf.*, IEEE, Piscataway, N.J., 1994, pp. 185-188.
4. H. Wolff, "How QuickTurn Is Filling the Gap," *Electronics*, Apr. 1990.
5. *Proc. IEEE Symp. FPGAs for Custom Computing Machines*, IEEE Computer Society Press, Los Alamitos, Calif., 1993-1996.

Suggested reading

S. Brown et al., *Field-Programmable Gate Arrays*, Kluwer Academic Publishers, Norwell, Mass., 1992. A general introduction to FPGAs.

J. Oldfield and R. Dorf, *Field-Programmable Gate Arrays*, John Wiley & Sons, New York, 1995. A textbook-like treatment, including digital logic design based on the Xilinx 3000 series and the Algotronix CAL chip.

J. Rose, A. El Gamal, and A. Sangiovanni-Vincentelli, "Architecture of Field-Programmable Gate Arrays," *Proc. IEEE*, Vol. 81, No. 7, July 1993, pp. 1013-1029. Detailed discussion of architectural trade-offs.

Field-Programmable Gate Array Technology, S. Trimberger, ed., Kluwer Academic Publishers, Norwell, Mass., 1994. Discussion of three FPGA/CPLD architectures.

Up-to-date FPD research appears in the published proceedings of several conferences:
Proc. IEEE Int'l Custom Integrated Circuits Conf., IEEE.
Proc. Int'l Conf. Computer-Aided Design (ICCAD), IEEE CS Press, Los Alamitos, Calif.

Authorized licensed use limited to: IEEE Publications Operations Staff. Downloaded on February 28,2023 at 16:58:07 UTC from IEEE Xplore. Restrictions apply.

Proc. Design Automation Conference (DAC), IEEE CS Press.

FPGA Symp. Series: Third Int'l ACM Symp. Field-Programmable Gate Arrays (FPGA 95) and *Fourth Int'l ACM Symp. Field-Programmable Gate Arrays (FPGA 96)*, Assoc. for Computing Machinery, New York.

Stephen Brown is an assistant professor of electrical and computer engineering at the University of Toronto. He holds a PhD in electrical engineering from that university; his dissertation (on architecture and CAD for FPGAs) won him the Canadian NSERC's 1992 prize for the best doctoral thesis in Canada. In 1990, the International Conference on Computer-Aided Design awarded him and coauthor Jonathan Rose a Best Paper award. A coauthor of the book *Field-Programmable Gate Arrays*, he has also won four awards for excellence in teaching electrical engineering, computer engineering, and computer science courses. Brown is the general and program chair for the Fourth Canadian Workshop on Field-Programmable Devices (FPD 96), and is on the Technical Program Committee for the Sixth International Workshop on Field-Programmable Logic (FPL 96). He is a member of the IEEE and the Computer Society.



Jonathan Rose is an associate professor of electrical and computer engineering at the University of Toronto. His research interests are in the area of architecture and CAD for field-programmable gate arrays and systems. He coauthored the book *Field-Programmable Gate Arrays*. Rose holds a PhD in electrical engineering from the University of Toronto. He is the general chair of the Fourth International Symposium on FPGAs (FPGA 96) and serves on the technical program committee for the Sixth International Workshop on Field-Programmable Logic. In 1990, ICCAD awarded him and coauthor Stephen Brown a Best Paper award. He is a member of the IEEE, the Computer Society, the Association for Computing Machinery, and SIGDA.

Direct questions concerning this article to Stephen Brown, Dept. of Electrical and Computer Engineering, Univ. of Toronto, 10 Kings College Rd., Toronto, ONT, Canada M5S 3G4; brown@eecg.toronto.edu.

CALL FOR ARTICLES

IEEE Design & Test of Computers

Special Issue on Microprocessors

Submission deadline: June 15, 1996 **Publication date: Spring 1997**

D&T focuses on practical articles of near-term interest to the professional engineering community. *D&T* seeks articles of significant contribution that address the design, test, debugging, manufacturability, and yield improvement of microprocessors and microcontrollers. The areas of interest include but are not limited to

- Circuit design and design methodologies
- Logic design and design methodologies
- CAD tools and methodologies
- Design-for-test techniques and applications
- Debugging experiences, tools, and methodologies
- Yield improvement experiences, tools, and methodologies
- Project management

Interested authors should submit four copies of a double-spaced manuscript no longer than 35 pages, in English, by **June 15, 1996**. Each copy must contain contact information (name, postal and e-mail addresses, and phone/fax numbers). Final articles will be due **October 15, 1996**. For author guidelines, see *D&T*'s Spring 1996 issue or Web page at http://www.computer.org/pubs/d&t/d&t.htm.

Submit manuscripts to:

Marc E. Levitt
Special Issue Guest Editor
Sun Microelectronics, USUN02-301
2550 Garcia Avenue, Mountain View, CA 94043
phone (408) 774-8268; fax (408) 774-2099
marc.levitt@eng.sun.com

Authorized licensed use limited to: IEEE Publications Operations Staff. Downloaded on February 28,2023 at 16:58:07 UTC from IEEE Xplore. Restrictions apply.

FPGA and CPLD architectures: a tutorial

Publisher: IEEE

Cite This

PDF

<< Results

S. Brown ; J. Rose **All Authors**

211 Paper Citations

32 Patent Citations

1948 Full Text Views

Alerts

Manage Content Alerts

Add to Citation Alerts

Abstract

Authors

References

Citations

Keywords

Metrics

More Like This

Downl
PDF

Abstract: This tutorial surveys commercially available, high-capacity field-programmable devices. The authors describe the three main categories of FPDs: simple and complex program... **View more**

Metadata

Abstract:
This tutorial surveys commercially available, high-capacity field-programmable devices. The authors describe the three main categories of FPDs: simple and complex programmable logic devices, and field-programmable gate arrays. They then give architectural details of the most important chips and example applications of each type of device.

Published in: IEEE Design & Test of Computers (Volume: 13 , Issue: 2, Summer 1996)

Page(s): 42 - 57

INSPEC Accession Number: 5293756

Date of Publication: Summer 1996

DOI: 10.1109/54.500200

ISSN Information:

Publisher: IEEE

Authors

References

Citations

Keywords

Metrics

Back to Results

More Like This

Effect of ESD on complex programmable logic device and field programmable gate arrays
IEEE Electromagnetic Compatibility Magazine
Published: 2013

NeuroFPGA-implementing artificial neural networks on programmable logic devices
Proceedings Design, Automation and Test in Europe Conference and Exhibition
Published: 2004

Show More

IEEE Personal Account

CHANGE USERNAME/PASSWORD

Purchase Details

PAYMENT OPTIONS

VIEW PURCHASED DOCUMENTS

Profile Information

COMMUNICATIONS PREFERENCES

PROFESSION AND EDUCATION

TECHNICAL INTERESTS

Need Help?

US & CANADA: +1 800 678 4333

WORLDWIDE: +1 732 981 0060

CONTACT & SUPPORT

Follow

About IEEE *Xplore* | Contact Us | Help | Accessibility | Terms of Use | Nondiscrimination Policy | IEEE Ethics Reporting | Sitemap | IEEE Privacy Policy

A not-for-profit organization, IEEE is the world's largest technical professional organization dedicated to advancing technology for the benefit of humanity.

© Copyright 2023 IEEE - All rights reserved.

IEEE Account

» Change Username/Password

» Update Address

Purchase Details

» Payment Options

» Order History

» View Purchased Documents

Profile Information

» Communications Preferences

» Profession and Education

» Technical Interests

Need Help?

» **US & Canada:** +1 800 678 4333

» **Worldwide:** +1 732 981 0060

» Contact & Support

A not-for-profit organization, IEEE is the world's largest technical professional organization dedicated to advancing technology for the benefit of humanity.
© Copyright 2023 IEEE - All rights reserved. Use of this web site signifies your agreement to the terms and conditions.