An Overview of Technology, Architecture and CAD Tools for Programmable Logic Devices

Stephen D. Brown

Department of Electrical and Computer Engineering
University of Toronto
E-mail: brown@eecg.toronto.edu

Abstract

Before the advent of programmable logic devices (PLDs), most digital hardware designs included significant numbers of small-scale integrated (SSI) circuits that comprised basic logic gates and flip-flops. However, modern designs contain virtually none of these low-density parts, but instead are built from more complex devices that consist of an uncommitted array of logic gates and memory elements that can be configured by the user to implement different circuits. The term *programmable logic device* is not easy to define because the assortment of chips that fall within this broad category, namely any integrated circuit that is programmable by the end user and intended for implementing hardware, has grown very large over the past few years. In fact, even the relevant terminology has become nebulous because of rapid technology changes and the introduction of new innovative architectures. The purpose of this paper is to provide an overview of programmable logic devices in order to present the reader with a clear view of what is available on the market today. The fundamental technologies employed to manufacture PLDs are presented, after which for each of the three categories of chips, namely Simple Programmable Logic Devices (SPLDs), Complex Programmable Logic Devices (CPLDs), and Field-Programmable Gate Arrays (FPGAs), the paper describes the most significant architectural features, examples of applications, and the CAD tool design flow typically used when implementing circuits.

1 Introduction

With the development of new types of sophisticated programmable logic devices (PLDs), the process of designing digital hardware has changed dramatically over the past few years. Unlike previous generations of technology, in which board-level designs included large numbers of SSI chips containing basic gates, virtually every new digital design produced today consists almost entirely of high-density devices. This applies not only to custom parts like processors and memory, but also for miscellaneous logic, state machines, decoders, control circuits, and the like. In very high-volume situations, it is feasible to integrate such circuits into high-density custom manufactured chips like Mask-Programmable Gate Arrays (MPGAs), but in most cases MPGAs are too expensive and take too long to manufacture to be viable for prototyping or other low-volume scenarios. For these reasons, most prototypes, and also many production designs, are now built using PLDs. The advantages of PLDs are instant manufacturing turnaround, low start-up costs, and ease of design changes (since programming is done by the end user). The number of applications for PLDs has grown so dramatically over the past decade that many companies have produced competing products and there is now a wide assortment of devices to choose from. A designer today who is not familiar with the various products faces a daunting task in order to research all of the different types of chips, try to understand what they can best be used for, choose a particular company's device, and then design the hardware. The purpose of this paper is to provide an introductory overview of the wide array of PLDs that are now on the market.

The paper first describes the fundamental technologies that are employed in programmable logic devices and then defines three main categories of chips: Simple Programmable Logic Devices (SPLDs), Complex Programmable Logic Devices (CPLDs), and Field-Programmable Gate Arrays (FPGAs). The paper describes the underlying technologies with which the devices are built, presents the main features of products in each category, gives an indication of typical applications, and describes the CAD design process for implementing circuits.

2 Background Information

Before describing the technology used to manufacture PLDs, it's important to ensure that the terminology used in this paper is unambiguous. This is necessary because the technical jargon for describing PLDs has become somewhat confused over the past few years as companies have attempted to compare and contrast their products in literature. Since such inconsistencies can be a source of confusion for readers, the most important terminology is defined in the list below. Note that some definitions shown may not be universally accepted, but usage will be consistent throughout this paper:

- *Programmable Logic Device (PLD)* — a general term that refers to any type of integrated circuit used for implementing hardware, where the chip can be configured by the end user to realize different designs. Programming of such a device usually involves placing the chip into a special programming unit, but some chips can also be configured 'in circuit'. Since PLDs are programmed "in the field" by the end user, they are also called User-Programmable or Field-Programmable devices.
- *PLA* — a Programmable Logic Array (PLA) is a specific type of PLD that contains two levels of logic, where both levels are programmable

0-7803-1886-2/94 $3.00

- *PAL*[1] — a Programmable Array Logic (PAL) is a programmable device that has one level of programmable logic followed by another level of fixed logic
- *SPLD* — refers to any simple type of PLD, usually either a PLA or PAL
- *CPLD* — a more Complex PLD that consists of a hierarchical arrangement of multiple PAL-like blocks on a single chip. Alternative names (that will not be used in this paper) for this style of chip are EPLD, Super PAL, Mega PAL, and others.
- *FPGA* — a Field-Programmable Gate Array is a PLD featuring a general structure that engenders very high logic capacity. Unlike SPLDs and CPLDs, FPGAs easily can implement several levels of logic. In some literature, CPLDs are also called FPGAs, but we will distinguish the two in this paper.
- *HCPLDs* — a single acronym that encompasses both CPLDs and FPGAs. This term has been coined in trade literature strictly for convenience to provide an easy way to refer to both types of devices.
- *Logic Block* — a relatively small circuit block that is replicated in an array. When a circuit is implemented in a PLD, it is first decomposed into smaller sub-circuits that can each be mapped into a logic block. The term logic block is mostly used in the context of FPGAs, but it could also refer to a block of circuitry in a CPLD.
- *Interconnect* — the wiring resources in a PLD
- Programmable Switch — a user-programmable switch that can connect a logic gate to an interconnect wire, or one interconnect wire to another
- *Logic Capacity* — the amount of circuitry that can be mapped into a single PLD. This is usually measured in units of "equivalent to an MPGA basic gate". In other words, the capacity of a PLD is measured by the size of MPGA that it is comparable to. In simpler terms, logic capacity can be thought of as "number of NAND gates". Another term often used for this is *logic density*.
- *Speed-Performance* — measures the maximum operable speed of a circuit when implemented in a PLD. For combinational circuits, it is set by the longest delay through any path, and for sequential circuits it is the maximum clock frequency for which the circuit will function properly.

Some of the above terminology will be defined in more detail later on in the paper. In the remainder of this section, the evolution of the various types of PLDs is briefly described and then the fundamental semiconductor technologies used in the manufacture of PLDs are presented.

2.1 Evolution of Programmable Logic Devices

The first type of user-programmable chip that could be employed for implementing logic circuits was the Read-Only Memory (ROM). In a ROM, address lines can be used as logic circuit inputs and data lines as outputs. However, logic functions rarely require more than a few product terms, and since a ROM contains a full decoder for its address inputs it is not an

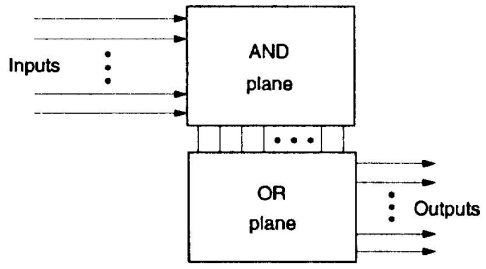

Figure 1 - *Structure of a PLA.*

efficient architecture for realizing logic circuits. A more appropriate type of device that was later developed specifically for implementing logic circuits is the Programmable Logic Array (PLA). A PLA consists of two levels of logic gates: a programmable AND-plane followed by a programmable OR-plane. As illustrated in Fig. 1, A PLA is structured so that any of the inputs to the chip (or their complements) can be AND'ed together in the AND-plane; each AND-plane output can thus correspond to *any* product term of the inputs. Similarly, each OR-plane output can be configured to produce the logical sum of any of the AND-plane outputs. Thus, PLAs are well-suited for implementing logic functions in sum-of-products form. The user-programmability of most PLAs is realized by *fuse* technology, which will be described shortly. The main drawbacks of the PLA is that it's expensive to manufacture and it features somewhat poor speed-performance. Both disadvantages are due to having two levels of configurable logic, because programmable logic planes are difficult to manufacture and introduce significant propagation delays.

Intended to overcome the drawbacks of PLAs, the next type of logic device developed was the Programmable Array Logic (PAL). PALs feature only a single level of programmability, consisting of a programmable AND-plane that feeds a fixed OR-plane. The structure of a typical PAL is illustrated in Fig. 2. To compensate for the lack of generality in PALs, incurred because the OR plane is fixed, manufacturers produce several variants with different numbers of inputs and outputs, various sizes of OR-gates, and the like. Also, PALs usually contain flip-flops connected to the OR-gate outputs so that sequential circuits can be realized. PAL devices are very

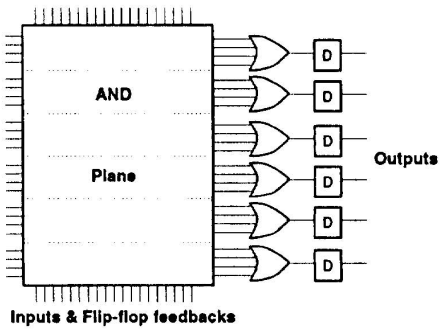

Figure 2 - *Structure of a PAL.*

1. PAL is a trademark of Advanced Micro Devices.

important because when introduced they had a profound effect on digital hardware design, and also they are the basis for some of the newer, more sophisticated architectures that will be described shortly.

PALs provide very high speed-performance of circuits and can be configured for a wide variety of applications. However, they are limited in terms of logic capacity and as technology has advanced a need for logic chips with higher density has developed. The highest density general purpose logic chips available today are called *Mask-Programmable Gate Arrays (MPGAs)*. MPGAs are mostly pre-designed chips in that they consist of an array of transistors whose placement on the chip is pre-defined, but MPGAs can be customized to implement a specific user's circuit. Customization is performed during chip fabrication by specifying mask layers for metal interconnect, and this means that in order for a user to employ an MPGA a large setup cost is involved and manufacturing time is long. Although MPGAs are clearly not PLDs, they are mentioned here because they motivated the design of the user-programmable equivalent: Field-Programmable Gate Arrays (FPGAs). Like MPGAs, FPGAs comprise an array of uncommitted circuit elements, called *logic blocks*, and interconnect resources, but in an FPGA configuration is performed in the field by the end user. The structure of an FPGA is depicted in Fig. 3. As the first type of PLD that supported very high logic densities, FPGAs were responsible for a major shift in the way digital circuits were designed and are considered to have had a major impact. In fact, high-capacity PLDs are currently the largest growing segment of the semiconductor industry.

Because of a rapidly growing market for large PLDs, other manufacturers developed new device architectures to compete with FPGAs; the most important of these are known as Complex Programmable Logic Devices (CPLDs). Referring to the terminology that was listed at the beginning of this section, recall that small devices like PALs are referred to as Simple PLDs. Complex PLDs are so named because they consist of multiple SPLD-like blocks on a single chip. The structure of a typical CPLD is depicted in Fig. 4.

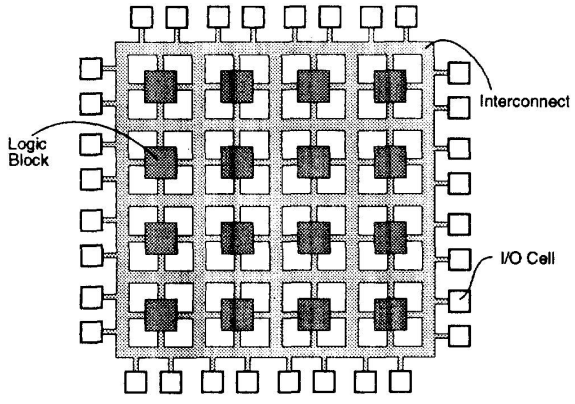

Figure 3 - *Structure of an FPGA.*

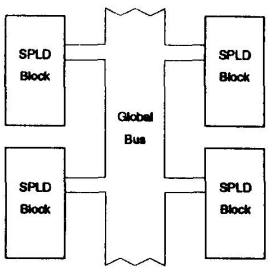

Figure 4 - *Structure of a CPLD.*

Besides the three categories of PLDs mentioned so far, other special purpose devices optimized for specific applications (e.g. state machines, analog gate arrays, large interconnection problems), have also been developed. However, since use of such devices is limited they will not be described here. The remainder of this section will present the methods used to implement the user-programmable switches that are the key to the usefulness of PLDs.

2.2 User-Programmable Switch Technologies

Used in SPLDs, the first type of user-programmable switch developed was the fuse. There are two types in use today: lateral fuses and vertical fuses. A lateral fuse is simply a small section of metal alloy positioned between two interconnect wires. In series with the fuse is a Bipolar transistor controlled in a way such that if the fuse needs to be blown, the transistor generates the required current. Fig. 5 shows how fuses might appear in an AND-plane of an SPLD. In the figure, the vertical wires are the AND-plane inputs and the horizontal wires are the outputs. Notice that multiple fuses are connected to each wire marked *product wire* and these wires are pulled high through resistors. Since each *input wire* can connect to a product wire through a fuse, this allows the implementation of *wired-AND* functions.

In contrast to lateral fuses, which normally connect two wires and can be blown to separate them, vertical fuses are normally open circuits. Vertical fuses do not contain a separate section of metal alloy, but rather each fuse is *part of* a Bipolar transistor. The operation of this type of fuse is that the Bipolar transistor in its normal state is equivalent to two back-to-back diodes that isolate the two wires. However, if suffi-

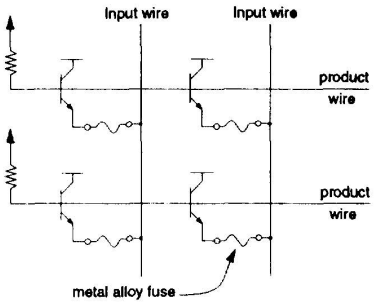

Figure 5 - *Lateral Fuse Technology.*

cient current is forced through the transistor a phenomenon known as the avalanche effect occurs and one of the diodes becomes permanently shorted, connecting the two wires.

Although fuse technology works well for SPLDs, fuses are not easy to manufacture, and also they are based on Bipolar technology. For higher density devices, where CMOS technology is dominant, different approaches to implementing programmable switches have been developed. For CPLDs the dominant switch technologies are EPROM and EEPROM, and for FPGAs they are SRAM and antifuse. Each of these are briefly discussed below.

EEPROM and EPROM transistors are used as programmable switches for CPLDS (and also for some SPLDs) in much the same way as fuses are in SPLDs. That is, an EPROM or EEPROM transistor is placed between two wires in a way that facilitates the implementation of wired-AND functions. This is illustrated in Fig. 6, which shows EPROM transistors as they might be connected in an AND-plane of a CPLD; the diagram would look very similar for EEPROM transistors.

While fuses, EPROM and EEPROM are appropriate for SPLDs and CPLDs, where the structure is based on AND/OR logic planes, they are not as useful for FPGAs. In FPGAs, the type of switch that is needed is one that can simply connect or isolate two wires, rather than combining wires in a wired-AND or wired-OR fashion. The two dominant switch technologies found in FPGAs are SRAM and antifuses.

An example of the application of SRAM-controlled switches is illustrated in Fig. 7. The figure shows two usages of SRAM cells: for controlling the gate nodes of pass-transistor switches and to control the select lines of multiplexers that drive logic block inputs. Note that whether pass-transistors or multiplexers are used depends on the particular product.

Featured in FPGAs from Actel and Quicklogic are programmable switches built using antifuse technology. In terms of the other programming technologies discussed so far, antifuses are most like lateral fuses because they are originally open-circuits and take on low resistance only when programmed. Antifuses are suitable for FPGAs because they can be built using modified CMOS technology. As an example, the Actel antifuse structure is illustrated in Fig. 8.

Table 1 summarizes the most important characteristics of the various programming technologies discussed in this section. Referring to the table, the left-most column indicates

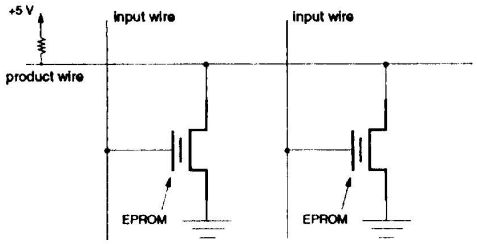

Figure 6 - *EPROM Programmable Switches.*

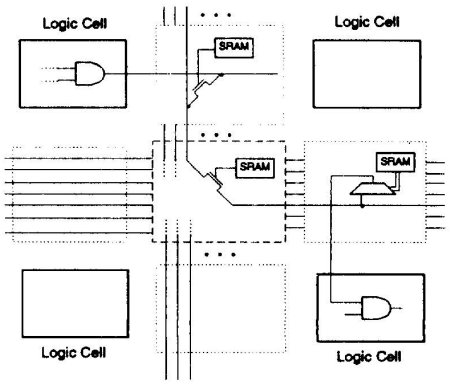

Figure 7 - *SRAM-controlled Programmable Switches.*

whether the programmable switches are one-time programmable (OTP), or can be re-programmed (RP). The next column lists whether the switches are volatile, and the last column names the underlying transistor technology.

Table 1 - *Summary of Programming Technologies.*

Name	RP ?	Vol. ?	Tech.
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+

Table 2 summarizes the characteristics of PLDs, by listing the switch technologies available for each category and the maximum logic capacity available in commercial products. Note that the capacity numbers are approximate, since it is difficult to measure this and manufacturer's capacity claims vary widely. The rest of this paper will now discuss the three categories of PLDs in more detail.

Table 2 - *Summary of Categories of PLDs.*

Category	Switch Technologies	Maximum Capacity
SPLD	Fuse, EPROM, EEPROM	750 gates
CPLD	EPROM, EEPROM	5000 gates
FPGA	SRAM, ANTIFUSE	12000 gates

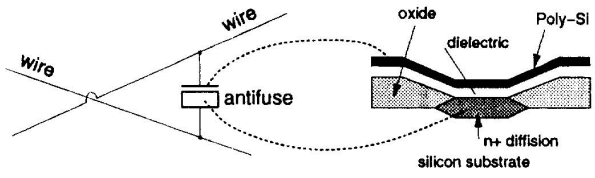

Figure 8 - *Actel Antifuse Structure.*

3 Simple Programmable Logic Devices (SPLDs)

As the staple for digital hardware designers for the past two decades, SPLDs are no doubt very important devices. However, they are also fairly straight-forward and well understood, so this paper will discuss them only briefly.

Two of the most popular SPLDs are the PALs produced by Advanced Micro Devices (AMD) known as the 16R8 and 22V10. Both of these devices are industry standards and are widely second-sourced by various companies. The name "16R8" means that the PAL has a maximum of 16 inputs (there are 8 dedicated inputs and 8 input/outputs), and a maximum of 8 outputs. The "R" refers to the type of outputs provided by the PAL and means that each output is "registered" by a D flip-flop. Similarly, the "22V10" has a maximum of 22 inputs and 10 outputs. Here, the "V" means each output is "versatile" and can be configured in various ways, some configurations registered and some not.

Another widely used and second sourced SPLD is the Altera Classic EP610. This device is similar in complexity to the above PALs, but it offers more flexibility in the way that outputs are produced and has larger AND- and OR- planes. In the EP610, outputs can be registered and the flip-flops are configurable as any of D, T, JK, or SR.

In terms of software CAD tool design cycles, typical steps performed for SPLDs include: initial design entry, logic optimization, device fitting, simulation, and configuration. This design flow is illustrated in Fig. 9, which also indicates how some stages feed back to others. Referring to the figure, design entry may be done either by creating a schematic diagram with a graphical CAD tool or by using a text-based system to describe a design in a simple hardware description language. It is also possible for sections of a design to be created in different ways and then translate and merge the pieces into a complete circuit, but this is normally done only for larger designs. Since initial logic entry is not usually in an optimal form, CAD tools are employed to optimize the circuits, after which additional programs analyse the resulting logic equations and "fit" them into the SPLD. Simulation is used to verify correct operation, and the user would go back to design entry to fix errors. When a design simulates correctly it can be loaded into a programming unit and an SPLD configured. One final detail to note about Fig. 9 is that only the original design entry step is performed manually by the designer; all other steps are carried out automatically, possibly with hints from the user.

In addition to the SPLDs mentioned above many other products are available from a wide array of companies. All SPLDs share common characteristics, like some sort of logic planes (AND, OR, NOR, or NAND), but each specific product offers unique features that may be particularly attractive for some applications. A partial list of companies that offer SPLDs includes: AMD, Altera, ICT, Intel, Lattice, National, Cypress, and Philips-Signetics. Since some of these SPLDs have complexity approaching that found in CPLDs, the paper will now move on to more sophisticated devices.

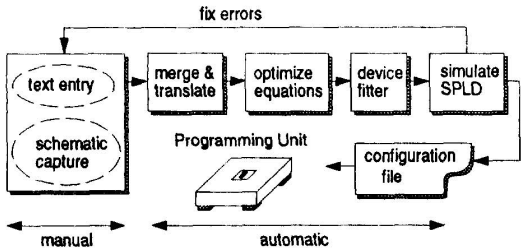

Figure 9 - *CAD Design Flow for SPLDs.*

4 Complex Programmable Logic Devices (CPLDs)

The simplest way of describing a Complex Programmable Logic Device (CPLD) is to say that a CPLD consists of multiple SPLD-like blocks on a single chip. This statement is fairly accurate, but CPLDs are much more sophisticated than SPLDs, even at the level of their basic SPLD-like structures. As an example of CPLDs, this section will first discuss the devices offered by Altera, since they are among the most widely used in industry.

Altera offers two families of chips that fit within the CPLD category: MAX 5000 and MAX 7000. Here, the discussion will focus on the MAX 7000 series, because while the two families have similar features the 7000 is more advanced and offers higher logic capacity and speed-performance.

The general architecture of the Altera MAX 7000 series is depicted in Fig. 10. It comprises an array of blocks called Logic Array Blocks (LABs), and interconnect wires called a Programmable Interconnect Array (PIA). The PIA is capable of connecting any LAB input or output to any other LAB. Also, the inputs and outputs of the chip connect directly to the PIA and to LABs. A LAB can be thought of as a more complex SPLD-like structure, and so the entire chip can be considered to be an array of SPLDs.

Each LAB consists of two sets of eight *macrocells*, where a macrocell comprises a programmable AND-plane that feeds an OR-gate and a flip-flop. Each set of macrocells, therefore, is similar to a PAL device, and a LAB corresponds roughly to two PALs. However, as illustrated in Fig 11, the number of

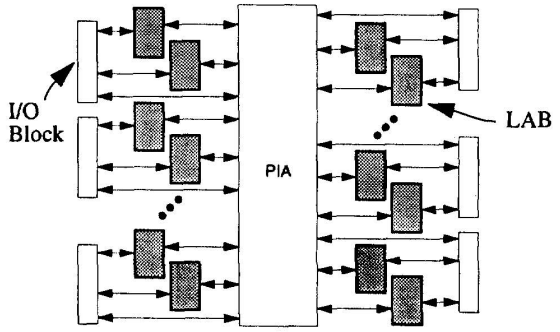

Figure 10 - *Altera MAX 7000 Series.*

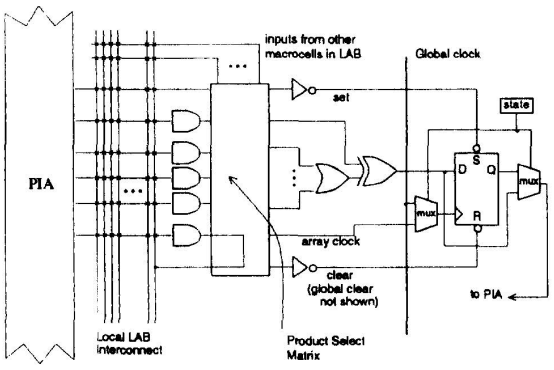

Figure 11 - *MAX 7000 Macrocell.*

inputs to the OR-gate in a macrocell is variable; the OR-gate can be fed from as many or few of the five product terms in the macrocell as is desired, and in addition can have up to 15 extra inputs from other macrocells in the same LAB. Altera claims that this makes the 7000 series LAB more efficient in terms of chip area because typical logic functions do not need more than five sum terms, and the architecture supports wider functions when they are needed. The flip-flops in the 7000 series can be configured as D type, JK, T, SR, or can be transparent. Referring to Fig11, it is important to realize that the individual wires drawn within the macrocells are local to the LAB and are not part of the PIA. This aspect of the chip represents one of the key features shared by all CPLDs, namely that there is both *local* interconnect for use within a single SPLD-like block, as well as separate *global* interconnect that spans the entire device. Because of these two levels of interconnect, CPLDs are also sometimes called *hierarchical* PLDs.

Besides Altera, several other companies produce devices that can be categorized as CPLDs. For example, AMD manufacturers the MACH family, Lattice has the (i)pLSI series, Xilinx produces a CPLD series that they call 7000 (unrelated to the Altera 7000 series), Intel offers their FLEX devices, National produces the MAPL family, and ICT has the PEEL array. While it would be of interest to discuss the unique features of each of these families of CPLDs in turn, it's not practical in this paper due to space limitations. In lieu of a detailed presentation, suffice it to say that all CPLDs share common features and offer similar capabilities; they differ mostly in the way in which their features are implemented, such as with larger or smaller AND- and OR- planes, different numbers of flip-flops configurable in various ways, a range of maximum logic capacities, different programming technologies (fuses, EPROM, EEPROM), varying numbers of interconnect wires, in-circuit versus out-of-circuit programmability, and others. The reader is encouraged to consult the individual data sheets from the manufacturers to learn more of the details.

4.1 CAD Design Cycle for CPLDs

The steps involved for implementing circuits in CPLDs are ostensibly the same as for SPLDs (see Fig. 9), but the tools themselves are more sophisticated. Because the devices are complex and can accommodate large designs, commonly a mixture of design entry methods is used for different modules of a complete circuit. For instance, some modules might be designed with a small hardware description language like ABEL, others drawn using a symbolic schematic capture tool, and still others described via a full-featured hardware description language such as VHDL. This implies that for CPLDs an especially important step in the CAD system is the merging of pieces of a design into a single large circuit and then "fitting" that circuit into the CPLD. Because of the complexity involved, this task of mapping a circuit into the device must be performed automatically by sophisticated CAD programs, which are supplied either by the CPLD manufacturer or a third party. Referring back to Table 2 shown earlier, the maximum size of circuit that will fit in the largest available CPLDs is about 5000 gates; for larger circuits it's necessary to turn to the more general structure found in FPGAs.

5 Field-Programmable Gate Arrays (FPGAs)

As one of the largest growing segments of the semiconductor industry, the FPGA market-place is extremely volatile. As such, the pool of companies involved changes rapidly and it is somewhat difficult to say which products will be the most significant when the industry reaches a stable state. For this reason, and to provide a more focused discussion, this paper will not mention all of the FPGA manufacturers that currently exist, but will instead focus on those companies whose products appear to be in somewhat widespread use at this time.

There are two basic categories of FPGAs on the market today: 1. SRAM-based FPGAs and 2. antifuse-based FPGAs. In the first category, Xilinx is the leading manufacturer in terms of number of users, with the major competitors being Altera and AT&T. For antifuse-based products, Actel is the major player and Quicklogic also offers a competing product.

Since Xilinx is the largest FPGA manufacturer (and the inventor of FPGAs) this paper will discuss Xilinx products as an introduction to FPGA architectures. The basic structure of Xilinx FPGAs is *array-based*, meaning that each chip comprises a two-dimensional array of logic blocks that can be interconnected via horizontal and vertical routing channels. An illustration of this type of structure was shown in Fig. 3. Xilinx introduced the first FPGA family, called the XC2000 series, in about 1985 and now offers two more generations: XC3000 and XC4000. Although the XC3000 devices are very widely used, this paper will focus on the more recent XC4000 family, because they represent state-of-the-art FPGAs.

Like all SRAM-based FPGAs, the XC4000 features a logic block, called a Configurable Logic Block (CLB), that is based on look-up tables (LUTs). A LUT is simply a small one bit wide memory array, where the address lines for the memory are inputs of the logic block and the output from the memory (one bit) is the LUT output. A LUT with K inputs would then correspond to a 2^K x 1 bit memory, and would be able to implement any logic function of its K inputs. The XC4000

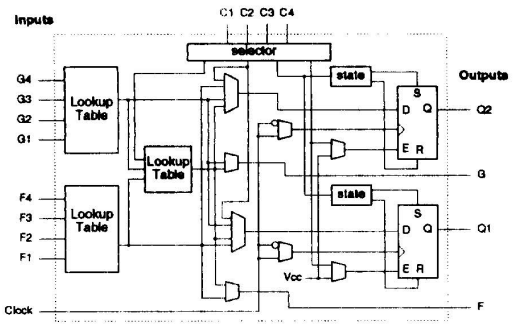

Figure 12 - *Xilinx XC4000 Configurable Logic Block.*

CLB contains three separate LUTs, in the configuration shown in Fig 12. Referring to the figure, there are two 4-input LUTS that are fed by CLB inputs, and the third LUT can be used in combination with the other two. This arrangement allows the CLB to implement a wide range of logic functions of up to nine inputs, two separate functions of four inputs or other possibilities. Each CLB also contains two flip-flops.

As part of the effort to provide high density devices that support the integration of entire systems, the XC4000 chips have "system oriented" features. For instance, each CLB contains circuitry that allows it to efficiently perform arithmetic (i.e., a carry chain) and also the LUTs in a CLB can be configured as read/write RAM cells. Finally, each XC4000 device includes very wide AND planes around the periphery of the logic block array to facilitate implementing circuit blocks such as wide decoders.

Besides its logic blocks, the other key feature that characterizes an FPGA is its interconnect structure. The XC4000 family's interconnect is arranged in horizontal and vertical channels and comprises wire segments of various lengths. Each channel contains some number of short wire segments that span a single CLB (the number of segments in each channel depends on the specific part number), longer segments that span two CLBs, and very long segments that span the entire length or width of the chip. Programmable switches are available (see Fig. 7) to connect the inputs and outputs of the CLBs to the wire segments, or to connect one wire segment to another. A small section of a routing channel representative of an XC4000 device appears in Fig. 13. The figure shows only the wire segments in a horizontal channel, and does not show the vertical routing channels, the CLB inputs and outputs, or the routing switches. An important point worth noting about

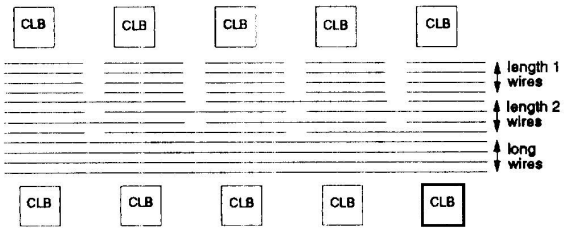

Figure 13 - *Xilinx XC4000 Wire Segments.*

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 wire segments used. Thus, the speed-performance of a circuit implemented in a Xilinx device depends in part on how the wire segments are allocated to individual signals by the CAD tools.

As mentioned above, SRAM-based FPGAs are also offered by Altera and AT&T. Rather than delving into the details of these chips, for brevity only some of their features will be highlighted. Altera's SRAM-based PLDs are called FLEX 8000. They consist of a two-dimensional array of logic blocks, called LABs. Each LAB consists of a set of eight 4-input LUTs, where each LUT's output can be registered by a flip-flop. Any LUTs within a LAB can be connected together via a local bus that consists of wire segments that each span the entire LAB. Connections are realized through SRAM-controlled switches. At a more global level, the inputs or outputs of any LABs can be interconnected by a global bus consisting of long wires that span the entire chip. It's worth noting that the Altera interconnect structure is very different from that in Xilinx because it consists of only one length of wire segment at the LAB level and one length at the global level. This means that the number of switches a signal passes through to reach one LUT from another is always the same, and thus the speed-performance of circuits is more predictable. In this sense, FLEX 8000 is similar to CPLDs and so Altera considers them to be a mixture of FPGA and CPLD technologies. An illustration of the FLEX 8000 would look similar to the MAX 7000 shown in Fig. 10, but with many more LABs.

The AT&T FPGA, called ORCA, is also based on SRAM-controlled switches and LUTs, and the overall architecture has a general array-based structure similar to Xilinx. One novel feature of the ORCA chips is that the LUTs in their logic blocks can be configured in several ways: as four 4-input LUTs, or two 5-input LUTs, or one 6-input LUT. In each configuration, the outputs of the individual LUTs are available as logic block outputs; this is more efficient in terms of area usage, since it's unnecessary to waste a large LUT if only a small logic function is needed. Like the XC4000, ORCA logic blocks contain carry-chain circuitry for arithmetic and the LUTs can be configured to act as RAM. Organized as a 4-bit bus structure, the ORCA interconnect is meant to be particularly suited to data path designs.

In contrast to FPGAs described above, the devices manufactured by Actel are based on antifuse technology. Actel is the second largest manufacturer of FPGAs and offers three families: Act 1, Act 2, and Act 3. Although all three generations have similar features, this paper will focus on the most recent devices, since they are apt to be more widely used in the longer term. Unlike the FPGAs described above, Actel devices are based on a structure similar to that in conventional mask-programmed gate arrays; the logic blocks are arranged in rows and there are horizontal routing channels between adjacent rows. This architecture is illustrated in Fig. 14. The logic blocks in the Actel devices are relatively small in com-

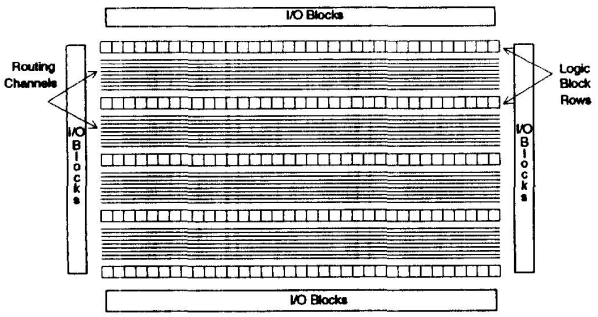

Figure 14 - *Structure of Actel FPGAs.*

parison to the LUT-based ones described above, and are based on multiplexers. Fig. 15 illustrates the logic block in the Act 3 and shows that it comprises an AND and OR gate that are connected to a multiplexer-based circuit block. The multiplexer circuit is arranged such that, in combination with the two logic gates, a very wide range of functions can be implemented in a single logic block. About half of the logic blocks in an Act 3 device also contain a flip-flop.

As stated above, Actel's interconnect is organized in horizontal routing channels. The channels consist of wire segments of various lengths with antifuses to connect logic blocks to wire segments or one wire to another. Also, although not shown in Fig. 14, Actel chips have vertical wires that overlay the logic blocks, for signal paths that span multiple rows. In terms of speed-performance, it would seem probable that Actel chips are not as predictable as CPLDs, because the number of antifuses traversed by a signal depends on how the wire segments are allocated during circuit implementation by CAD tools. However, Actel provides a very rich selection of wire segments in each channel and has developed algorithms that guarantee strict limits on the number of antifuses traversed by any signal in a circuit. A corollary is that Actel offers among the highest speed-performance of any FPGA on the market.

The main competitor for Actel in terms of antifuse-based FPGAs is Quicklogic, whose family of devices is called pASIC. The pASIC's overall structure is array-based like a Xilinx FPGA and its logic blocks use multiplexers, similar to Actel. Like Actel, Quicklogic focuses on speed-performance and provides among the fastest FPGAs available.

5.1 CAD Design Cycle for FPGAs

The design process for implementing circuits in FPGAs is similar to that for CPLDs, but additional tools are needed to support the increased complexity of the chips. An example of a typical CAD system is illustrated in Fig. 16. Notice that the figure lists the names of some of the popular commercial products available for design entry. Comparing Fig. 16 to the SPLD and CPLD design flow that was depicted in Fig. 9, notice that the major difference is in the "device fitter" step that comes after logic optimization and before simulation, where FPGAs require at least three separate CAD tools: a technology mapper to map from basic logic gates into the FPGA's logic blocks, placement to choose which specific logic blocks to use in the FPGA, and a router to allocate the wire segments in the FPGA to interconnect the logic blocks. With this added complexity, the CAD tools might require a fairly long period of time (more than an hour) to complete their tasks, but the burden on the designer is usually minimal, since the CAD tools are intended to be fully automatic.

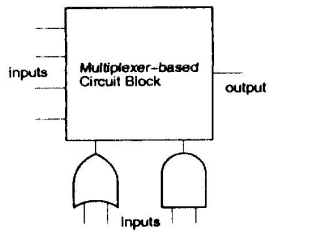

Figure 15 - *Actel Act 3 Logic Block.*

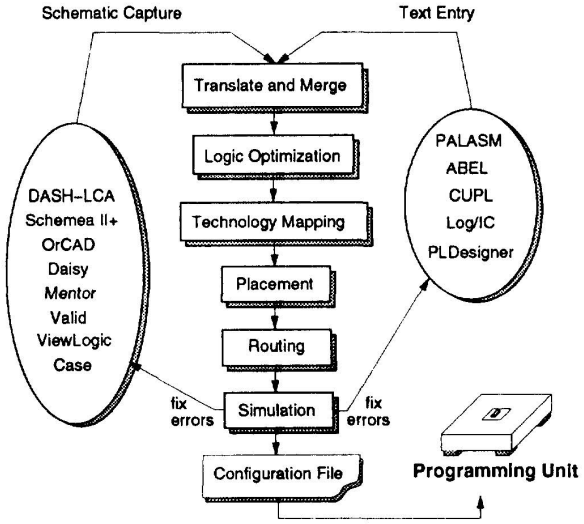

Figure 16 - *A Typical FPGA CAD System.*

6 Final Comments

This paper has presented an overview of the wide assortment of PLDs that are commercially available, categorizing them as either SPLDs, CPLDs, or FPGAs. The applicable technical jargon has been solidified, the basic technologies have been described, and commercial products in each category have been presented. For some topics, only a cursory discussion has been provided and the reader should consult companies' individual data sheets for more details, but the paper provides a comprehensive summary of the various types of PLD products that are available to designers today.

7 References

Rather than listing references to the various companies' data sheets, the interested reader is encouraged to directly contact the companies or their distributors and request the most recent documentation available.