Programmable logic devices

David Smith - May 15, 1986

Sidebars:

The basics of PLD architecture

PLD design software: straining to keep pace

Advances in programmable logic devices' architectures are making PLDs suitable for a plethora of applications. The devices now find use in state machines and other highly sequential circuits. In the past two years, bipolar PLD operating speeds have doubled, and emerging CMOS PLDs offer four times the circuitry per chip than the older bipolar versions offered. CMOS PLDs also offer lower operating currents, erasability and reprogrammability, and testability that promises close to a 100% yield of working parts.

To evaluate the design potential of PLDs, you must understand their architectural aberrations from the traditional PLA circuit (see box, "The basics of PLD architecture"). Each aberration enables the device to implement certain types of circuits that the original couldn't manage. An understanding of how the benefits of CMOS PLDs affect your particular system will help you determine whether the higher price of the CMOS parts is worth those benefits. Bipolar PLDs are seeing speed improvements that render them useful in higher-speed systems traditionally served by SSI and MSI chips. But when evaluating these chips, don't consider only functional specs. You should also consider factors like programming yield, software support, testability, and reprogrammability and how they influence the design and manufacturing process.

Most new PLD architectures expand on the basic AND-OR structure of the original PLDs. They offer programmable output cells that let you set the polarity of the output signal; the cells often include an output-enable feature as well. These cells can also include a register and several feedback lines, in which case the cells are called macrocells. Some macrocells, like those in Altera's devices, also include part of the logic arrays. The extent to which they expand upon or deviate from the basic PLA architecture is the most distinguishing characteristic of any PLD.

Deviations from established PLD architectures vary from the barely noticeable to the dramatic. The CMOS PLDs from Sprague Solid State are an example of the former. They're identical in function to Monolithic Memories' bipolar PAL devices, and you can use the Sprague PLDs as direct replacements for PAL devices. Unlike most older devices, however, the Sprague devices include a security bit that you can set to keep a competitor from reading out your design's programming pattern. Most new PLDs include this security bit.

A simple variation on the PAL theme comes from Panatech Semiconductor. The company is the American arm of Ricoh (Osaka, Japan), the only Japanese company presently involved in PLDs.
Panatech's CMOS devices contain twice as many product terms per output as their PAL counterparts. The extra product terms allow you to implement more complex algorithms, including XOR instead of OR summing logic. In other functional respects they are identical to PAL devices.

Other companies are offering PAL-compatible devices that include more circuitry than do the standard parts. VLSI Technology offers CMOS PAL replacements with what it calls an X-pander cell, which connects the product terms of two adjacent output cells, doubling the number of product terms available to each output. In addition, the cell can perform an OR or an XOR function on the product terms available to it. Atmel, a year-old company offering EPROMs and electrically erasable ROMs, will use a 1.25-µm CMOS technology to create a 20-pin PLD. Scheduled for release by the end of the second quarter, the company's V750 will include twice the number of sum terms, registers, and feedback lines that are normally associated with 20-pin devices.

Another new architecture implements a PAL superset in a chip that you can program to operate like a PAL device. Lattice Semiconductor offers such an architecture in its GAL devices, for example. The GAL (generic array logic) devices, for which VLSI Technology is an alternate source, can replace most 20- and 24-pin PAL devices. For example, the GAL 16V8 20-pin PLD can replace any of the following PAL Series chips: 10X8, 12X6, 14X4, 16X2, 16X8, 16X6 and 16X4 (where X stands for L, H, P, or RP). By including the options found in all of those devices, the 16V8 can offer some combinations of options not available in any one PAL device. Furthermore, the availability of all the options in one device allows you to stock only one chip type instead of as many as 21 different types.

The GAL devices also feature reprogrammability and lower power consumption than the PAL devices do. The GAL advantages come at a higher cost, however, one with a 25-nsec propagation delay (the time it takes a change in state at a PLD's inputs to reach its outputs) costs $8.80 in a plastic DIP; the device it can replace cost from $3 to $7.

Another PLD that can replace a variety of 24-pin PAL devices is Cypress Semiconductor's EPROM based 24-pin PLD C 20G10. It specs a 25-nsec propagation delay and consumes 55 mA, which is lower than the GAL 20-V8 24-pin PLD's 90 mA and the 180-mA standard for 24-pin bipolar PLDs. Prices for the PLD C 20G10 start at $6.75.

Another type of architecture, introduced this spring, combines both AND and OR logic arrays into one array, a technique called folding. Feedback lines from the output of the array attach to the inputs, permitting implementation of multilevel logic. For example, inputs to the arrays can form product terms that can attach to the feedback lines. The product terms themselves become inputs to the array, allowing you to sum those terms for a sum-of-products logic circuit. You can assign any term to a feedback line, and you can create a design that effects product-term sharing.

The Erasic (electrically reprogrammable application-specific IC) from Excel Microelectronics is the first commercially available PLD to employ the folded-logic-array architecture. The company refers to the product's architecture as a configurable logic array, which consists of one programmable array with feedback lines. Because the number of times you route signals back into the array is arbitrary (as many as 32 product terms can feed back into the array), you implement nested logic more efficiently than you can using an AND-OR architecture. You can also implement a single level of logic that, by bypassing a second logic plane, yields a shorter propagation delay than would a 2-level logic device.

The CMOS Erasic consumes 35 mA max and specs a 35-nsec propagation delay for one level of logic. It's pin compatible with many 24-pin PAL devices. The device contains feedback lines from its output registers to its logic array, so you can use it to create state machines with buried registers. Like many CMOS ICs, it doesn't drive as large a load as bipolar devices can; it sinks and sources only 4
mA, so it can't drive more than two TTL loads.

Signetics is developing a folded-logic-plane architecture for its bipolar parts. Devices with that architecture will contain buried registers and a flexible interconnect structure. The parts could implement state machines running at 80 to 100 MHz, according to the company. The first ones won't have registers, however, but will offer only a 4-nsec propagation delay through the logic plane.

Not all PLD vendors see the folded-logic array as the best architecture for PLDs. Altera's PLD group logic circuitry, registers, feedback lines, and I/O-control circuitry into macrocells. The macrocells connect to local and global buses that distribute input and feedback signal lines. In addition, these devices have features not commonly found in PAL devices: synchronous and asynchronous clocking, buried registers, variable product-term distribution, and a security bit. Altera offers a family of these devices ranging in complexity from about 300 to 1800 equivalent 2-input gates.

Intel offers devices identical to the 300-, 600-, 900-, and 1200-gate Altera devices. Altera exchanged the device designs for the Intel EPROM technology from which the devices are made. Intel plans to introduce a zero-standby-power version of the 300-gate device that will also deliver a 25-nsec propagation delay and 25-mA current consumption at 10 MHz (a typical PAL device specs a 25-nsec delay and 180-mA power consumption).

Although most complex PLDs are CMOS devices, bipolar PLDs are clearly getting more complex as well. An early example, the 22V10 from Advanced Micro Devices, brought programmable-output macrocells to 24-pin PLDs. The 22V10 can replace a variety of 24-pin PAL devices. The usefulness of this design is not lost on makers of CMOS parts; Cypress offers CMOS versions of the 22V10, and Texas Instruments and Lattice Semiconductor will have CMOS versions by the end of the year. The CMOS ICs consume as much as 50% less power than their bipolar counterparts, and they are reprogrammable.

AMD continues its expansion of bipolar-PLD architecture with its AmPAL23S8. The most noteworthy features of the device are six buried registers. These registers have an average of eight product-term inputs, which in effect add another 48 product terms to a 20-pin PAL-like architecture. These buried registers allow you to build sequential circuits like state machines without using any output pins. By not routing feedback lines from the output registers through output buffers, the buried registers can also clock at higher frequencies than registers in output macrocells.

The AmPAL23S8 offers other features not found on most bipolar PLDs. You can set the polarity of each output separately, and you can program each output buffer so that it's enabled by its own product term. The eight outputs consist of four output registers and four output macrocells that offer a choice of registered or unregistered operation and a choice of several feedback modes. One product term can operate as a synchronous preset, and another can operate as an asynchronous reset. Both product terms operate on all registers in the device.

Large bipolar PLDs do not have an enviable history. MMI attempted to realize large architectures in bipolar technology via its MegaPAL devices. They maintain the PAL AND-OR structure but offer as many as 32 inputs, 32 outputs, and 255 product terms. This fixed-PAL structure makes it difficult to apply these devices, and their 640-mA max operating current renders them too power hungry for most designs. The AND-OR structure that works well for collecting several MSI and SSI functions doesn't seem to implement larger circuits nearly as well. Nevertheless, the manufacturer claims that the devices' popularity is slowly increasing.

Improvements in the FPLA architecture are appearing as well, although they're less dramatic than the improvements in PAL architectures. For example, Signetics's PLS167 and PLS168 have a preset
option that allows you to initialize output registers. The devices also include a complement array that generates the negation of any product term you choose and feeds it back into the AND array. You can then use the term in subsequent product terms.

Only one vendor, Xilinx, has abandoned the PLA-based PLD design altogether in favor of an architecture more like that of a gate array. Xilinx's PLD, the XC 2064, comprises 64 individual blocks of logic in a sea of interconnects. Both the blocks and the interconnects are programmable. Because the circuits you build don't conform to a PLA-like structure, you have more freedom in using registers than you do in the PLA-based devices.

You use the logic in a logic block in a way similar to the manner in which you would define the function of cells in a gate array. Each logic block contains a 4-input logic cell that can implement logic functions. The blocks also contain a register and feedback lines. The XC 2064 can implement as many as 1500 equivalent gates of logic, so you can use it instead of a small gate array, in place of 15 to 75 SSI and MSI ICs, or as substitutes for as many as four PAL devices, according to the vendor.

Most of the companies new to the PLD market are concentrating on CMOS devices, because the technology can yield PLDs that are amenable to more circuit designs than are bipolar PAL devices. The principal advantage of CMOS is that, by virtue of its simpler layout and lower power consumption per unit of silicon area, it can create much more dense and complex circuitry. Also, any CMOS part should have better supply-voltage tolerance and noise margins than corresponding bipolar parts. CMOS vendors list other benefits—lower power consumption in its own right, testability, and erasability—but you should inspect them carefully to determine whether they actually benefit your design.

For example, though CMOS devices generally consume less power than equivalent bipolar devices, at high frequencies they may actually consume the same amount of power, because CMOS power consumption is proportional to operating frequency. Check the power consumption at the frequencies at which you expect your circuit will operate. Quarter- and half-power bipolar devices may consume the same amount of power as some CMOS devices.

AMD offers half- and quarter-power versions of its 18P8. The half-power version specs a 25-nsec propagation delay and 80-mA max operating current. The quarter-power version specs a 35-nsec propagation delay but a cool 45-mA max current. Comparable CMOS devices, Lattice Semiconductor's quarter-power GAL16V8 and Cypress Semiconductor's PAL C 16XXL-25, spec a 25-nsec delay and 45-mA current consumption.

Look for the frequency at which the vendor determines maximum power consumption. Maximum power consumption for CMOS devices is often measured at 10 MHz, but if you plan to use the full speed of a 35-nsec CMOS PLD, the device will use more power than it would when operating at its 10-MHz rating. Ask the manufacturer to estimate the power consumption at your frequency requirements.

Even in those cases where the CMOS part consumes roughly the same power as a bipolar part during operation, the CMOS part will still save you power when it's idle. Unlike bipolar devices, which have a significant quiescent current, CMOS devices fall into a quiescent mode that consumes much less power than does the active state. Zero-power CMOS PLDs can consume microwatts when idle, but it takes some time for most PLDs to enter a quiescent mode. If that power-down time is longer than the time your circuit is typically idle, then the PLD won't achieve quiescent operation very often.

CMOS devices like Altera's can drop into a quiescent state immediately once the outputs stop
switching. Others, like the Xilinx part, contain an inactive power-down pin that you must assert to put the device in that mode. The Xilinx part consumes only 50 nA in quiescent mode.

Harris offers CMOS replacements for bipolar PAL devices and chip-select decoders. The relatively slow CMOS PAL replacements (typical propagation delay is 125 nsec) spec 6-mA/MHz power consumption and a 150-μA standby current, so they are appropriate for circuits with slow and infrequently changing signals. The Harris chip-select decoders can replace two to seven ICs in memory and I/O chip decoding circuits. They spec a 60-nsec propagation delay.

CMOS is also used to create hard-wired versions of PAL devices. Vendors of the hard-wired versions customize their parts through use of a metal mask during processing. The devices are therefore fixed, not programmable. Monolithic Memories' HAL (hard-wired array logic) devices and VLSI Technology's mask-programmable logic arrays use less power than PAL devices, aren't quite as fast, and might not sink as much current, but in large quantities they will cost much less than programmable parts.

After noting the new architectures and the emergence of CMOS parts, you'll observe that today's PLDs have become faster as well, and it's little wonder that they have. Performance is the most important parameter for PLDs; many PLD vendors describe their customers as "speed freaks" who are always crying for faster devices than are available at any given time. Bipolar ECL PLDs now offer propagation delays as short as 6 nsec. TI and Fairchild plan to introduce 3-nsec devices by the end of the year. Because PLDs replace the equivalent of two or three levels of logic, the present ECL versions are roughly equivalent in performance to discrete ECL parts (which typically spec a propagation delay of 2 nsec).

TTL PLDs are also getting faster and are superior to CMOS parts in speed. For example, Monolithic Memories' PAL 24B Series and 20B Series devices, TI's Impact PAL devices, and Fairchild's 16P8B device furnish a propagation delay of only 15 nsec. These companies all plan to introduce 10-nsec TTL devices by the end of the year. The fastest CMOS devices feature a propagation delay of 25 nsec. The high-speed bipolar PLDs can require as much as 210 mA max power-supply current but are otherwise identical to regular PAL devices.

Typical TTL bipolar parts, with propagation delays of approximately 25 nsec, compete directly with CMOS parts. As noted, the CMOS parts consume as little as 50% of the power of bipolar equivalents, but you pay a price premium to use them. Remember also that half-power TTL devices can consume as little as 90 mA, and quarter-power TTL parts as little as 45 mA. Though these lower-power PLDs are slower than their full-power counterparts, the lower-power versions may consume roughly the same amount of power as equally fast CMOS parts.

By integrating more logic, CMOS parts could compensate for their relative slowness and achieve competitive system-throughput levels. With more feedback lines allowing more complex state machines, your system design could be as fast with one CMOS part as it would be with two or more PAL devices or a PAL device and other discrete logic. However, CMOS parts may not be able to drive the high currents that bipolar parts can, so you must consider the load that your system places on your PLD.

The complexity you can realize from the simple layout of CMOS devices does make timing analysis of the parts more difficult. The multilevel devices can theoretically include any level of nesting (up to the number of feedback terms into the logic array). Ideally, your design software should verify the dynamic performance of your part, but not all PLD design tools can simulate the newer, more complex PLDs.
Higher complexity—in any type of device—exacerbates problems related to testing. Functional testing is not always necessary with PAL designs, but feedback terms will introduce a variety of dynamic signal paths that you may have to verify. Also, on-chip registers require a preset or preload function to put them into a known state. Fortunately, most PLDs with registers include such functions.

From the manufacturer's point of view, CMOS's biggest asset is its testability. CMOS PLDs usually use reprogrammable technologies based on EPROM or EEPROM cells. The vendor can set the cells to test the device and then erase them before shipping. The eras ability of EPROM and EEPROM cells makes the parts good candidates for prototype development; you can design incrementally and fix minor bugs without rewiring your boards. Bipolar fuses, on the other hand, cannot be restored once they are blown, so vendors can't test the fuse array and circuitry immediately adjacent to it.

Lattice Semiconductor encourages its customers to forego incoming and dynamic testing because the company completely tests its electrically erasable CMOS PLDs (although it will supply Sentry test vectors if asked). The savings gained by eliminating some device testing become significant at higher production volumes, in the form of reduced handling, testing, and inventory costs. In addition, the costs of sorting and returning devices are eliminated.

Still, most manufacturers of bipolar parts say that they can achieve a greater than 99% postprogramming functional yield (PPFY), so if the 1% fallout is acceptable, testability may not even be an important factor in your evaluation of PLDs. (The PPFY denotes the percentage of functionally good parts after the entire lot of PLDs has been programmed.) Because the design and testing of fuse-programmable PLDs is a mature art, vendors of bipolar devices declare that they can test virtually all functional circuitry without programming the entire logic array. According to Mitch Richman, product marketing manager at AMD, internal testing circuitry allows the company to deliver parts that achieve a PPFY of greater than 99.9%

The 99% figure that vendors of bipolar PLDs cite is usually mentioned in reference to the simpler PAL devices. More complex architectures, with buried feedback terms and registers, require fuses within the macrocells to effect these features. As noted, these fuses can't be blown for testing, so the testing problem for complex fuse-programmable devices is more intractable than it is for simpler architectures. Therefore, don't expect the PPFY figure to be as high for the big bipolar PLDs.

One manufacturer of bipolar PLDs, AMD, mitigates the testing problem by making its buried registers in the AmPAL23S8 accessible to the output pins. One product term enables a preset mode for the registers, allowing you to load the registers through output pins that are otherwise connected to output macrocells and registers. This arrangement allows AMD (and the user) to set the buried registers into a known state to begin testing.

The need to develop tests for PLDs has come to the attention of vendors of PLD design systems and of test systems. They are introducing software to assist you in creating tests for your PLDs. These tests work from a description of the device and your programming pattern to develop test vectors. In addition to developing test vectors, software from Teradyne (Boston, MA) runs a fault simulation, a necessary step for PLDs that incorporate thousand-gate complexities.

If you have narrowed your PLD candidates to CMOS parts, differences between EPROM-based PLDs and electrically erasable (EE) PLDs may affect your choice of device. Vendors of EPROM-based parts must place the devices under a lamp to erase programming patterns when reprogramming the parts for more tests. EE PLDs can be erased in the test equipment. Still, EPROM-based vendors claim to provide as high a level of testing as EE PLD vendors. For example, Cypress Semiconductor borrows a technique from bipolar vendors to test its PLDs. One portion of a part's logic array is programmed
at the factory for functional and ac testing; this phantom array remains programmed so you can test the part upon delivery.

The basic prices of EE PLDs are higher than those of EPROM-based PLDs because the process technology is more difficult to control. In addition, the programming-cell size of EE PLDs is larger by two to three times than EPROM cells (estimates depend on the technology used, and few companies release exact figures). As a result, EE PLDs tend to cost more. Lattice Semiconductor's 20-pin EE PLD, the GAL 16V8, costs $8.80 (1000), while an EPROM-based device of similar complexity, Altera's EP310, costs $6 in a plastic DIP. To take advantage of the EP310's reprogrammability, however, you'd order the part in a package with a window at a cost of $10.75.

As PLDs become more complex, which usually means more I/O circuitry and feedback, the percentage of area covered by programming cells will decrease. Consequently, the size advantage of EPROM-based devices will shrink. Because the memory-cell-size advantage should decrease and packages with windows will always be considerably more expensive than plastic package, EE PLDs should eventually become the most cost-effective of the complex CMOS PLDs.

EE cells also offer the designer the capability to reprogram the chip within the circuit design. However, for production runs, the value of reprogrammability is uncertain. Few current applications require hardware that configures itself, although in the words of Alex Goldberger, technical marketing manager at Exel Microelectronics, "Give it [in-situ reprogramming] to designers and they'll figure out a way to use it."

The Xilinx CMOS PLD, with its gate-array-like architecture in which RAM cells control macros, doesn't employ the expensive process of EE PLDs, nor does it feature the window and reprogramming time of EPROMs. It is volatile, however, and it's expensive (prices start at $55). Xilinx designed the part so that it can load its programming pattern from an EPROM by itself, but the added design effort and complexity of that programming-pattern memory must be taken into account when choosing parts.

With the availability of a variety of PLD types, speeds, and power-consumption specs, the devices will find use in a wider variety of applications than you may have thought possible. You can use the smaller PSAL devices in faster and faster systems. Usually these parts have been used to create multiplexing and encoding circuits, and they remain ideal for those tasks. ECL PAL devices can replace parts in mainframes and minicomputers. The CMOS devices, with more than a thousand gates, serve in applications for which you might otherwise have considered small (100-gate) gate arrays.

To review the spectrum of circuits that PLDs can implement, examine the PLD-design handbooks that most PLD vendors offer. These handbooks typically include several application descriptions. For example, AMD's handbook shows PLD designs for a 4-bit registered barrel shifter, an interface between the Z Bus and an 8086 µP, and a Multibus arbiter. (See also the April 17, 1986, issue of EDN, pg 200, for an application of a PLD in a motor-control circuit.) These examples demonstrate that PLDs are much more than mere generators of Boolean logic.
The basics of PLD architecture

The first programmable-logic devices implemented Boolean sum-of-products logic to replace the growing number of SSI and MSI chips proliferating on pc boards. The early PLDs usually included the coding, decoding, multiplexing, and demultiplexing logic needed to allow LSI devices to communicate with each other. Boolean algebra was chosen as the basis for PLD logic because it was taught in most engineering schools.

A PLD uses a programmable logic array (PLA) whose exact function you can program, either by melting fuses or by loading memory elements. A PLA (Fig A) can generate any possible sum of products for a particular set of inputs. When you program the device, you in effect select the sums of products that you need.

The first PLDs, Signetics's field-programmable logic arrays (FPLAs), contained a PLA in which all product terms and sums were fuse selectable. Although these parts were flexible, they were too complicated for many designers, who were just getting used to designing with SSI and MSI chips.

Reduced size and delay

With its PAL devices, Monolithic Memories simplified the PLA by assigning product terms to specific outputs (Fig B). Fixing the OR array reduces the PAL IC's size by reducing the number of programming elements, and it lowers the propagation delay through the PLD. However, the practice limits the number of product terms in each sum and doesn't allow any product term to go to more than one sum (i.e., it doesn't allow product sharing).

Most important, the simplification of architecture allowed Monolithic Memories to introduce software—Palasm—that facilitates the design process. Palasm accepts your Boolean equations and generates the programming pattern that implements the equations. With PAL devices and Palasm, you don't need to understand the circuit within the PLD to use the device.

As they've evolved, PLDs have incorporated other features that allow the devices to effect circuitry more complex than that which can be represented by straight Boolean algebra. Fig C shows modern PLD architecture with some of these features:

- PLDs contain registers that store the sums of products. You use the registers to design synchronous circuits and sequential logic, such as a state machine.
- You can program PLDs that have programmable output polarity for either active-high or active-low output signals.
- Feedback from an output to the AND array allows you to use that output as a bidirectional I/O line. Alternatively, you can drive the contents of the register back into the AND array, creating a state machine.
- Some PLDs contain product terms that enable the output buffers. Consequently, your design can enable the buffers individually and asynchronously.
- Other product terms may connect to the clock input of internal registers (a programmable clock), allowing your logic to clock individual registers.
- Product terms can drive the set and reset lines of internal registers.
Design software isolates you from PLD architecture, allowing you to design more naturally (as you would with discrete ICs). Just as Palasm translates Boolean algebra into PAL fuse maps, more advanced design software translates your state-machine designs and truth tables into programs for more complex PLDs.

The proliferation of new PLD architectures, however, is making it difficult for vendors of PLD design software to provide design support for all types of PLDs. When you consider using a complex PLD, first determine which design tools can create the programming patterns for it.

Relieve the tedium

You can design most PLDs by hand. To do so, you need to understand how a PLD's circuitry implements logic functions, and then you fill out a program table for that PLD. Program tables contain rows and columns of boxes that represent locations of programming elements on the PLD. Creating a PLD design this way is a long and tedious process.

Basic design tools such as MMI's Palasm and Signetics' Amaze automate the construction of a fuse map. You enter Boolean and state-machine equations into the software, and the software generates a file with the programming code. These tools also include functional simulators that let you verify the function of your PLD design. Like most PLD design tools, Palasm and Amaze can run on personal computers.

Not all designers use Boolean logic equations, especially for designs containing hundreds or thousands of gates, and these alternative design approaches require more advanced design tools. Third-party companies (not the PLD vendor or its customer) are now offering such design tools. Abele from Data I/O (Redmond, WA) and Cupl from Assisted Technology (San Jose, CA) are examples of these tools; you can use truth tables, state diagrams, and schematic capture in addition to Boolean equations. Another benefit of using third-party tools is that they support a wider range of PLDs than tools from a PLD vendor.

Keeping up with architectures

The expansion of PLD architectures has put a strain on the PLD software that was developed for PAL and FPLA architectures. The existence of this software guarantees the utilization of those devices for some years until more flexible software becomes widely distributed. In the meantime, vendors of new architectures generally offer their own design environments. These more recent design tools are usually more powerful, but they are also exclusive to a particular family or vendor.

Because its devices have an architecture that's significantly different from those of PALs and FPLAs, Altera offers a PLD-design package. The A+Plus design package includes design software and programming hardware that consists of a card for your computer and a programming box. When running A+Plus on an IBM PC, you can enter your design as a schematic or a net list, and you can enter Boolean equations and state-machine diagrams. Intel, which uses Altera's EPLD designs, offers a similar package.

Xilinx's PLDs differ radically from other PLDs, so the company offers the Xact development system, which runs on the IBM PC. Xact presents you with a graphic depiction of the chip's architecture. You place the logic function you need on one logic cell in the architecture and identify connection points; the software then assigns a programming pattern to the logic cell and finds an interconnection path. When you've completed placing your logic functions, the program can functionally simulate your design. Xact also performs timing simulation.

The most unusual feature of Xact is its ability to perform in-circuit emulation. You place an emulation pod, which is connected to the PC, over the XC-2064 PLD on your prototype pc board. You can download designs into the chip through the pod and transfer the state of the chip's registers back into the PC for debugging. As you test your prototype, you can observe the operation of the PLD and download design changes without removing the chip from the board.