PROGRAMMABLE ASICs

There are two types of programmable ASICs: programmable logic devices (PLDs) and field-programmable gate arrays (FPGAs). The distinction between the two is blurred. The only real difference is their heritage. PLDs started as small devices that could replace a handful of TTL parts, and they have grown to look very much like their younger relations, the FPGAs. We shall group both types of programmable ASICs together as FPGAs.

An FPGA is a chip that you, as a systems designer, can program yourself. An IC foundry produces FPGAs with some connections missing. You perform design entry and simulation. Next, special software creates a string of bits describing the extra connections required to make your design-the configuration file . You then connect a computer to the chip and program the chip to make the necessary connections according to the configuration file. There is no customization of any mask level for an FPGA, allowing the FPGA to be manufactured as a standard part in high volume.

FPGAs are popular with microsystems designers because they fill a gap between TTL and PLD design and modern, complex, and often expensive ASICs. FPGAs are ideal for prototyping systems or for low-volume production. FPGA vendors do not need an IC fabrication facility to produce the chips; instead they contract IC foundries to produce their parts. Being fabless relieves the FPGA vendors of the huge burden of building and running a fabrication plant (a new submicron fab costs hundreds of millions of dollars). Instead FPGA companies put their effort into the FPGA architecture and the software, where it is much easier to make a profit than building chips. They often sell the chips through distributors, but sell design software and any necessary programming hardware directly.

All FPGAs have certain key elements in common. All FPGAs have a regular array of basic logic cells that are configured using a programming technology . The chip inputs and outputs use special I/O logic cells that are different from the basic logic cells. A programmable interconnect scheme forms the wiring between the two types of logic cells. Finally, the designer uses custom software, tailored to each programming technology and FPGA architecture, to design and implement the programmable connections. The programming technology in an FPGA determines the type of basic logic cell and the interconnect scheme. The logic cells and interconnection scheme, in turn, determine the design of the input and output circuits as well as the programming scheme.

The programming technology may or may not be permanent. You cannot undo the permanent programming in one-time programmable (OTP) FPGAs. Reprogrammable or erasable devices may be reused many times. We shall discuss the different programming technologies in the following sections.

4.1 The Antifuse

4.2 Static RAM

4.3 EPROM and EEPROM Technology

4.4 Practical Issues

4.5 Specifications

4.6 PREP Benchmarks

4.7 FPGA Economics

4.8 Summary

4.9 Problems

4.10 Bibliography

4.11 References

4.1 The Antifuse

An antifuse is the opposite of a regular fuse-an antifuse is normally an open circuit until you force a programming current through it (about 5 mA). In a poly-diffusion antifuse the high current density causes a large power dissipation in a small area, which melts a thin insulating dielectric between polysilicon and diffusion electrodes and forms a thin (about 20 nm in diameter), permanent, and resistive silicon link . The programming process also drives dopant atoms from the poly and diffusion electrodes into the link, and the final level of doping determines the resistance value of the link. Actel calls its antifuse a programmable low-impedance circuit element (PLICE ‘).

Figure 4.1 shows a poly-diffusion antifuse with an oxide-nitride-oxide (ONO) dielectric sandwich of: silicon dioxide (SiO_2) grown over the n -type antifuse diffusion, a silicon nitride (Si_3N_4) layer, and another thin SiO_2 layer. The layered ONO dielectric results in a tighter spread of blown antifuse resistance values than using a single-oxide dielectric. The effective electrical thickness is equivalent to 10nm of SiO_2 (Si_3N_4 has a higher dielectric constant than SiO_2 , so the actual thickness is less than 10 nm). Sometimes this device is called a fuse even though it is an anti fuse, and both terms are often used interchangeably.

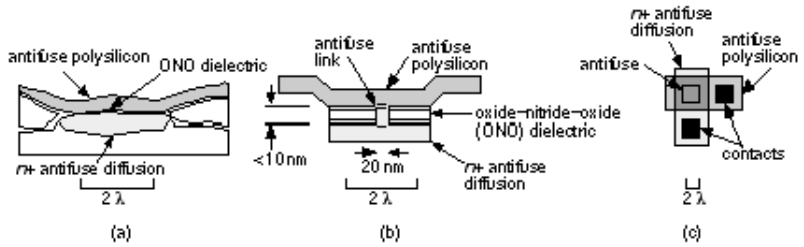


FIGURE 4.1 Actel antifuse. (a) A cross section. (b) A simplified drawing. The ONO (oxide-nitride-oxide) dielectric is less than 10 nm thick, so this diagram is not to scale. (c) From above, an antifuse is approximately the same size as a contact.

The fabrication process and the programming current control the average resistance of a blown antifuse, but values vary as shown in Figure 4.2 . In a particular technology a programming current of 5 mA may result in an average blown antifuse resistance of about 500 W . Increasing the programming current to 15 mA might reduce the average antifuse resistance to 100 W . Antifuses separate interconnect wires on the FPGA chip and the programmer blows an antifuse to make a permanent connection. Once an antifuse is programmed, the process cannot be reversed. This is an OTP technology (and radiation hard). An Actel 1010, for example, contains 112,000 antifuses (see Table 4.1), but we typically only need to program about 2 percent of the fuses on an Actel chip.

TABLE 4.1 Number of antifuses on Actel FPGAs.

Device	Antifuses
A1010	112,000
A1020	186,000
A1225	250,000
A1240	400,000
A1280	750,000

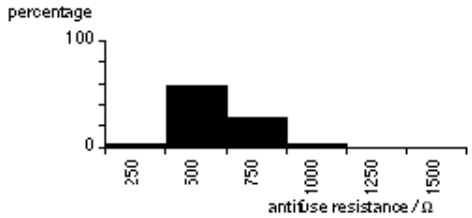


FIGURE 4.2 The resistance of blown Actel antifuses. The average antifuse resistance depends on the programming current. The resistance values shown here are typical for a programming current of 5 mA.

To design and program an Actel FPGA, designers iterate between design entry and simulation. When they are satisfied the design is correct they plug the chip into a socket on a special programming box, called an Activator , that generates the programming voltage. A PC downloads the configuration file to the Activator instructing it to blow the necessary antifuses on the chip. When the chip is programmed it may be removed from the Activator without harming the configuration data and the chip assembled into a system. One disadvantage of this procedure is that modern packages with hundreds of thin metal leads are susceptible to damage when they are inserted and removed from sockets. The advantage of other programming technologies is that chips may be programmed after they have been assembled on a printed-circuit board-a feature known as in-system programming (ISP).

The Actel antifuse technology uses a modified CMOS process. A double-metal, single-poly CMOS process typically uses about 12 masks-the Actel process requires an additional three masks. The n- type antifuse diffusion and antifuse polysilicon require an extra two masks and a 40 nm (thicker than normal) gate oxide (for the high-voltage transistors that handle 18 V to program the antifuses) uses one more masking step. Actel and Data General performed the initial experiments to develop the PLICE technology and Actel has licensed the technology to Texas Instruments (TI).

The programming time for an ACT 1 device is 5 to 10 minutes. Improvements in programming make the

programming time for the ACT 2 and ACT 3 devices about the same as the ACT 1. A 5-day work week, with 8-hour days, contains about 2400 minutes. This is enough time to program 240 to 480 Actel parts per week with 100 percent efficiency and no hardware down time. A production schedule of more than 1000 parts per month requires multiple or gang programmers.

4.1.1 Metal-Metal Antifuse

Figure 4.3 shows a QuickLogic metal-metal antifuse (ViaLink ‘). The link is an alloy of tungsten, titanium, and silicon with a bulk resistance of about 500 mW cm.

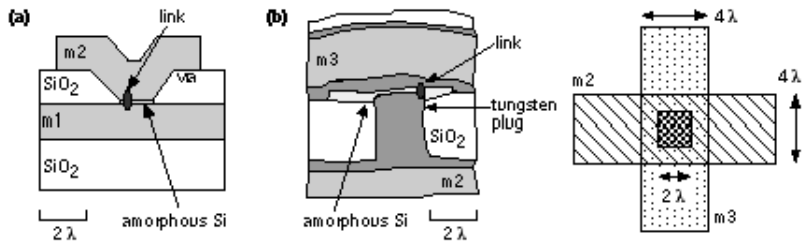
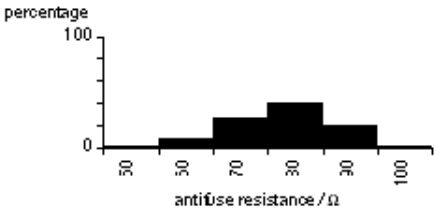


FIGURE 4.3 Metal-metal antifuse. (a) An idealized (but to scale) cross section of a QuickLogic metal-metal antifuse in a two-level metal process. (b) A metal-metal antifuse in a three-level metal process that uses contact plugs. The conductive link usually forms at the corner of the via where the electric field is highest during programming.

There are two advantages of a metal-metal antifuse over a poly-diffusion antifuse. The first is that connections to a metal-metal antifuse are direct to metal-the wiring layers. Connections from a poly-diffusion antifuse to the wiring layers require extra space and create additional parasitic capacitance. The second advantage is that the direct connection to the low-resistance metal layers makes it easier to use larger programming currents to reduce the antifuse resistance. For example, the antifuse resistance R ? 0.8/ I , with the programming current I in mA and R in W , for the QuickLogic antifuse. Figure 4.4 shows that the average QuickLogic metal-metal antifuse resistance is approximately 80 W (with a standard deviation of about 10 W) using a programming current of 15 mA as opposed to an average antifuse resistance of 500 W (with a programming current of 5 mA) for a poly-diffusion antifuse.

FIGURE 4.4 Resistance values for the QuickLogic metal-metal antifuse. A higher programming current (about 15 mA), made possible partly by the direct connections to metal, has reduced the antifuse resistance from the poly-diffusion antifuse resistance values shown in Figure 4.2 .



The size of an antifuse is limited by the resolution of the lithography equipment used to makes ICs. The Actel antifuse connects diffusion and polysilicon, and both these materials are too resistive for use as signal interconnects. To connect the antifuse to the metal layers requires contacts that take up more

space than the antifuse itself, reducing the advantage of the small antifuse size. However, the antifuse is so small that it is normally the contact and metal spacing design rules that limit how closely the antifuses may be packed rather than the size of the antifuse itself.

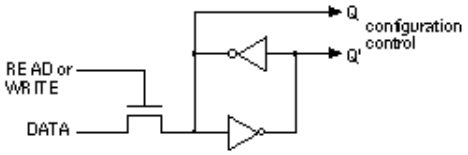
An antifuse is resistive and the addition of contacts adds parasitic capacitance. The intrinsic parasitic capacitance of an antifuse is small (approximately 1-2 fF in a 1 m m CMOS process), but to this we must add the extrinsic parasitic capacitance that includes the capacitance of the diffusion and poly electrodes (in a poly-diffusion antifuse) and connecting metal wires (approximately 10 fF). These unwanted parasitic elements can add considerable RC interconnect delay if the number of antifuses connected in series is not kept to an absolute minimum. Clever routing techniques are therefore crucial to antifuse-based FPGAs.

The long-term reliability of antifuses is an important issue since there is a tendency for the antifuse properties to change over time. There have been some problems in this area, but as a result we now know an enormous amount about this failure mechanism. There are many failure mechanisms in ICs-electromigration is a classic example-and engineers have learned to deal with these problems. Engineers design the circuits to keep the failure rate below acceptable limits and systems designers accept the statistics. All the FPGA vendors that use antifuse technology have extensive information on long-term reliability in their data books.

4.2 Static RAM

An example of static RAM (SRAM) programming technology is shown in Figure 4.5 . This Xilinx SRAM configuration cell is constructed from two cross-coupled inverters and uses a standard CMOS process. The configuration cell drives the gates of other transistors on the chip-either turning pass transistors or transmission gates on to make a connection or off to break a connection.

FIGURE 4.5 The Xilinx SRAM (static RAM) configuration cell. The outputs of the cross-coupled inverter (configuration control) are connected to the gates of pass transistors or transmission gates. The cell is programmed using the WRITE and DATA lines.



The advantages of SRAM programming technology are that designers can reuse chips during prototyping and a system can be manufactured using ISP. This programming technology is also useful for upgrades-a customer can be sent a new configuration file to reprogram a chip, not a new chip. Designers can also update or change a system on the fly in reconfigurable hardware .

The disadvantage of using SRAM programming technology is that you need to keep power supplied to the programmable ASIC (at a low level) for the volatile SRAM to retain the connection information. Alternatively you can load the configuration data from a permanently programmed memory (typically a programmable read-only memory or PROM

4.3 EPROM and EEPROM Technology

Altera MAX 5000 EPLDs and Xilinx EPLDs both use UV-erasable electrically programmable read-only memory (EPROM) cells as their programming technology. Altera's EPROM cell is shown in Figure 4.6 . The EPROM cell is almost as small as an antifuse. An EPROM transistor looks like a normal MOS transistor except it has a second, floating, gate (gate1 in Figure 4.6). Applying a programming voltage V_{PP} (usually greater than 12 V) to the drain of the n- channel EPROM transistor programs the EPROM cell. A high electric field causes electrons flowing toward the drain to move so fast they "jump" across the insulating gate oxide where they are trapped on the bottom, floating, gate. We say these energetic electrons are hot and the effect is known as hot-electron injection or avalanche injection . EPROM technology is sometimes called floating-gate avalanche MOS (FAMOS).

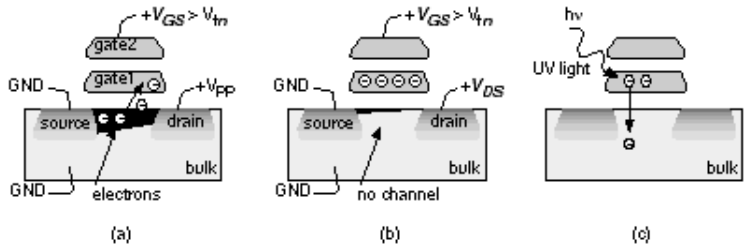


FIGURE 4.6 An EPROM transistor. (a) With a high (> 12 V) programming voltage, V_{PP} , applied to the drain, electrons gain enough energy to "jump" onto the floating gate (gate1). (b) Electrons stuck on gate1 raise the threshold voltage so that the transistor is always off for normal operating voltages. (c) Ultraviolet light provides enough energy for the electrons stuck on gate1 to "jump" back to the bulk, allowing the transistor to operate normally.

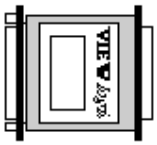
Electrons trapped on the floating gate raise the threshold voltage of the n- channel EPROM transistor (Figure 4.6 b). Once programmed, an n- channel EPROM device remains off even with VDD applied to the top gate. An unprogrammed n- channel device will turn on as normal with a top-gate voltage of VDD . The programming voltage is applied either from a special programming box or by using on-chip charge pumps. Exposure to an ultraviolet (UV) lamp will erase the EPROM cell (Figure 4.6 c). An absorbed light quantum gives an electron enough energy to jump from the floating gate. To erase a part we place it under a UV lamp (Xilinx specifies one hour within 1 inch of a 12,000 m Wcm^{-2} source for its EPLDs). The manufacturer provides a software program that checks to see if a part is erased. You can buy an EPLD part in a windowed package for development, erase it, and use it again, or buy it in a nonwindowed package and program (or burn) the part once only for production. The packages get hot while they are being erased, so that windowed option is available with only ceramic packages, which are more expensive than plastic packages.

4.4 Practical Issues

System companies often select an ASIC technology first, which narrows the choice of software design tools. The software then influences the choice of computer. Most computer-aided engineering (CAE) software for FPGA design uses some type of security. For workstations this usually means floating licenses (any of n users on a network can use the tools) or node-locked licenses (only n particular

computers can use the tools) using the hostid (or host I.D., a serial number unique to each computer) in the boot EPROM (a chip containing start-up instructions). For PCs this is a hardware key, similar to the Viewlogic key illustrated in Figure 4.7 . Some keys use the serial port (requiring extra cables and adapters); most now use the parallel port. There are often conflicts between keys and other hardware/software. For example, for a while some security keys did not work with the serial-port driver on Intel motherboards-users had to buy another serial-port I/O card.

FIGURE 4.7 CAE companies use hardware security keys that fit at the back of a PC (this one is shown at about one-half the real size). Each piece of software requires a separate key, so that a typical design system may have a half dozen or more keys daisy-chained on one socket. This presents both mechanical and software conflict problems. Software will not run without a key, so it is easily possible to have $60,000 worth of keys attached to a single PC.



Most FPGA vendors offer software on multiple platforms. The performance difference between workstations and PCs is becoming blurred, but the time taken for the place-and-route step for Actel and Xilinx designs seems to remain constant-typically taking tens of minutes to over an hour for a large design-bounded by designers' tolerances.

A great deal of time during FPGA design is spent in schematic entry, editing files, and documentation. This often requires moving between programs and this is difficult on IBM-compatible PC platforms. Currently most large CAD and CAE programs completely take over the PC; for example you cannot always run third-party design entry and the FPGA vendor design systems simultaneously.

There are many other factors to be considered in choosing hardware:

- Software packages are normally less expensive on a PC.
- Peripherals are less expensive and easier to configure on a PC.
- Maintenance contracts are usually necessary and expensive for workstations.
- There is a much larger network of users to provide support for PC users.
- It is easier to upgrade a PC than a workstation.

4.4.1 FPGAs in Use

I once placed an order for a small number of FPGAs for prototyping and received a sales receipt with a scheduled shipping date three months away. Apparently, two customers had recently disrupted the vendor's product planning by placing large orders. Companies buying parts from suppliers often keep an inventory to cover emergencies such as a defective lot or manufacturing problems. For example, assume that a company keeps two months of inventory to ensure that it has parts in case of unforeseen problems. This risk inventory or safety supply, at a sales volume of 2000 parts per month, is 4000 parts, which, at an ASIC price of $5 per part, costs the company $20,000. FPGAs are normally sold through distributors, and, instead of keeping a risk inventory, a company can order parts as it needs them using a just-in-time (JIT) inventory system. This means that the distributors rather than the customer carry inventory (though the distributors wish to minimize inventory as well). The downside is that other customers may change their demands, causing unpredictable supply difficulties.

There are no standards for FPGAs equivalent to those in the TTL and PLD worlds; there are no standard pin assignments for VDD or GND, and each FPGA vendor uses different power and signal I/O pin arrangements. Most FPGA packages are intended for surface-mount printed-circuit boards (PCBs). However, surface mounting requires more expensive PCB test equipment and vapor soldering rather than bed-of-nails testers and surface-wave soldering. An alternative is to use socketed parts. Several FPGA vendors publish socket-reliability tests in their data books.

Using sockets raises its own set of problems. First, it is difficult to find wire-wrap sockets for surface-mount parts. Second, sockets may change the pin configuration. For example, when you use an FPGA in a PLCC package and plug it into a socket that has a PGA footprint, the resulting arrangement of pins is different from the same FPGA in a PGA package. This means you cannot use the same board layout for a prototype PCB (which uses the socketed PLCC part) as for the production PCB (which uses the PGA part). The same problem occurs when you use through-hole mounted parts for prototyping and surface-mount parts for production. To deal with this you can add a small piece to your prototype board that you use as a converter. This can be sawn off on the production boards-saving a board iteration.

Pin assignment can also cause a problem if you plan to convert an FPGA design to an MGA or CBIC. In most cases it is desirable to keep the same pin assignment as the FPGA (this is known as pin locking or I/O locking), so that the same PCB can be used in production for both types of devices. There are often restrictions for custom gate arrays on the number and location of power pads and package pins. Systems designers must consider these problems before designing the FPGA and PCB.

4.5 Specifications

All FPGA manufactures are continually improving their products to increase performance and reduce price. Often this means changing the design of an FPGA or moving a part from one process generation to the next without changing the part number (and often without changing the specifications).

FPGA companies usually explain their part history in their data books. 1 The following history of Actel FPGA ACT 1 part numbers illustrates changes typical throughout the IC industry as products develop and mature:

- The Actel ACT 1 A1010/A1020 used a 2 m m process.
- The Actel A1010A/A1020A used a 1.2 m m process.
- The Actel A1020B was a die revision (including a shrink to a 1.0 m m process). At this time the A1020, A1020A, and A1020B all had different speeds.
- Actel graded parts into three speed bins as they phased in new processes, dropping the distinction between the different die suffixes.
- At the same time as the transition to die rev. ’B’, Actel began specifying timing at worst-case commercial conditions rather than at typical conditions.

From this history we can see that it is often possible to have parts from the same family that use different circuit designs, processes, and die sizes, are manufactured in different locations, and operate at very different speeds. FPGA companies ensure that their products always meet the current published worst-case specifications, but there is no guarantee that the average performance follows the typical specifications, and there are usually no best-case specifications.

There are also situations in which two parts with identical part numbers can have different performance-when different ASIC foundries produce the same parts. Since FPGA companies are fabless, second sourcing is very common. For example, TI began making the TPC1010A/1020A to be equivalent to the original Actel ACT 1 parts produced elsewhere. The TI timing information for the TPC1010A/1020A was the same as the 2 m m Actel specifications, but TI used a faster 1.2 m m process. This meant that "equivalent" parts with the same part numbers were much faster than a designer expected. Often this type of information can only be obtained by large customers in the form of a qualification kit from FPGA vendors.

A similar situation arises when the FPGA manufacturer adjusts its product mix by selling fast parts under a slower part number in a procedure known as down-binning . This is not a problem for synchronous designs that always work when parts are faster than expected, but is another reason to avoid asynchronous designs that may not always work when parts are much faster than expected.

---

1. See, for example, p.1-8 of the Xilinx 1994 data book.

4.6 PREP Benchmarks

Which type of FPGA is best? This is an impossible question to answer. The Programmable Electronics Performance Company (PREP) is a nonprofit organization that organized a series of benchmarks for programmable ASICs. The nine PREP benchmark circuits in the version 1.3 suite are:

1. An 8-bit datapath consisting of 4:1 MUX, register, and shift-register
2. An 8-bit timer-counter consisting of two registers, a 4:1 MUX, a counter and a comparator
3. A small state machine (8 states, 8 inputs, and 8 outputs)
4. A larger state machine (16 states, 8 inputs, and 8 outputs)
5. An ALU consisting of a 4 ¥ 4 multiplier, an 8-bit adder, and an 8-bit register
6. A 16-bit accumulator
7. A 16-bit counter with synchronous load and enable
8. A 16-bit prescaled counter with load and enable
9. A 16-bit address decoder

The data for these benchmarks is archived at http://www.prep.org . PREP’s online information includes Verilog and VHDL source code and test benches (provided by Synplicity) as well as additional synthesis benchmarks including a bit-slice processor, multiplier, and R4000 MIPS RISC microprocessor.

4.7 FPGA Economics

FPGA vendors offer a wide variety of packaging, speed, and qualification (military, industrial, or commercial) options in each family. For example, there are several hundred possible part combinations for the Xilinx LCA series. Figure 4.8 shows the Xilinx part-naming convention, which is similar to that used by other FPGA vendors.

FIGURE 4.8 Xilinx part-naming convention.

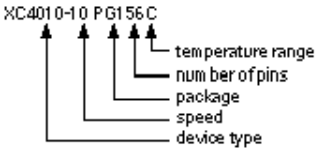


Table 4.2 shows the various codes used by manufacturers in their FPGA part numbers. Not all possible part combinations are available, not all packaging combinations are available, and not all I/O options are available in all packages. For example, it is quite common for an FPGA vendor to offer a chip that has more I/O cells than pins on the package. This allows the use of cheaper plastic packages without having to produce separate chip designs for each different package. Thus a customer can buy an Actel A1020 that has 69 I/O cells in an inexpensive 44-pin PLCC package but uses only 34 pins for I/O-the other 10 (= 44 - 34) pins are required for programming and power: three for GND, four for VDD, one for MODE (a pin that controls four other multifunction pins), and one for VPP (the programming voltage). A designer who needs all 69 I/Os can buy the A1020 in a bigger package. Tables in the FPGA manufacturers' data books show the availability, and these matrices change constantly.

TABLE 4.2 Programmable ASIC part codes.

Item	Code	Description	Code	Description
Manufacturer's	A	Actel	ATT	AT&T (Lucent)
code	XC	Xilinx	isp	Lattice Logic
	EPM	Altera MAX	M5	AMD MACH 5 is on the device
	EPF	Altera FLEX	QL	QuickLogic
	CY7C	Cypress		
Package	PL or PC	plastic J-leaded chip carrier, PLCC	VQ	very thin quad flatpack, VQFP
type	PQ	plastic quad flatpack, PQFP	TQ	thin plastic flatpack, TQFP
	CQ or CB	ceramic quad flatpack, CQFP	PP	plastic pin-grid array, PPGA
	PG	ceramic pin-grid array, PGA	WB, PB	ball-grid array, BGA
Application	C	commercial	B	MIL-STD-883
	I	industrial	E	extended
	M	military		

TABLE 4.3 1992 base Actel FPGA prices.

Actel part	1H92 base price
A1010A-PL44C	$23.25
A1020A-PL44C	$43.30
A1225-PQ100C	$105.00
A1240-PQ144C	$175.00
A1280-PQ160C	$305.00

TABLE 4.4 1992 base Xilinx XC3000 FPGA prices.

Xilinx part	1H92 base price
XC3020-50PC68C	$26.00
XC3030-50PC44C	$34.20
XC3042-50PC84C	$52.00
XC3064-50PC84C	$87.00
XC3090-50PC84C	$133.30

4.7.1 FPGA Pricing

Asking "How much do FPGAs cost?" is rather like asking "How much does a car cost?" Prices of cars are published, but pricing schemes used by semiconductor manufactures are closely guarded secrets. Many FPGA companies use a pricing strategy based on a cost model that uses a series of multipliers or adders for each part option to calculate the suggested price for their distributors. Although the FPGA companies will not divulge their methods, it is possible to reverse engineer these factors to create a pricing matrix.

Many FPGA vendors sell parts through distributors. This can introduce some problems for the designer. For example, in 1992 the Xilinx XC3000 series offered the following part options:

TABLE 4.5 Actel price adjustment factors.

Purchase quantity, all types

(1-9)	(10-99)	(100-999)
100 %		
	96 %	84 %

Purchase time, in (100-999) quantity

1H92	2H92	93
100 %		
	80-95 %	60-80 %

Qualification type, same package

Commercial	Industrial	Military	883-B
100 %	120 %	150 %	230-300 %

Speed bin 1

ACT 1-Std	ACT 1-1	ACT 1-2	ACT 2-Std	ACT 2-1
100 %				
	115 %	140 %	100 %	120 %

Package type

A1010:	PL44, 64, 84	PQ100	PG84		
	100 %	125 %	400 %		
A1020:	PL44, 64, 84	PQ100	JQ44, 68, 84	PG84	CQ84
	100 %	125 %	270 %	275 %	400 %
A1225:	PQ100	PG100			
	100 %	175 %			
A1240:	PQ144	PG132			
	100 %	140 %			

A1280:	PQ160	PG176	CQ172
	100 %	145 %	160 %

- Five different size parts: XC30{20, 30, 42, 64, 90}
- Three different speed grades or bins: {50, 70, 100}
- Ten different packages: {PC68, PC84, PG84, PQ100, CQ100, PP132, PG132, CQ184, PP175, PG175}
- Four application ranges or qualification types: {C, I, M, B}

where {} means "Choose one."

This range of options gave a total of 600 possible XC3000 products, of which 127 were actually available from Xilinx, each with a different part code. If a designer is uncertain as to exact size, speed, or package required, then they might easily need price information on several dozen different part numbers. Distributors know the price information-it is given to each distributor by the FPGA vendors. Sometimes the distributors are reluctant to give pricing information out-for the same reason car salespeople do not always like to advertise the pricing scheme for cars. However, pricing of the components of a microelectronics system is a vital factor in making decisions such as whether to use FPGAs or some alternative technology. Designers would like to know how FPGAs are priced and how prices may change.

4.7.2 Pricing Examples

Table 4.3 shows the prices of the least-expensive version of the Actel ACT 1 and ACT 2 FPGA families, the base prices , in the first half of 1992 (1H92). Table 4.4 shows the 1H92 base prices for the Xilinx XC3000 FPGA family. Current FPGA prices are much lower. As an example, the least-expensive XC3000 part, the XC3020A-7PC68C, was $13.75 in 1996-nearly half the 1992 price.

Using historical prices helps prevent accusations of bias or distortion, but still realistically illustrates the pricing schemes that are used. We shall use these base prices to illustrate how to estimate the sticker price of an FPGA by adding options-as we might for a car. To estimate the price of any part, multiply the base prices by the adjustment factors (shown in Table 4.5 for the Actel parts).

The adjustment factors in Table 4.5 were calculated by taking averages across a matrix of prices. Not all combinations of product types are available (for example, there was no military version of an A1280-1 in 1H92). The dependence of price over time is especially variable. An example price calculation for an Actel part is shown in Table 4.6 . Many FPGA vendors use similar pricing models.

TABLE 4.6 Example Actel part-price calculation using the base prices of Table 4.3 and the adjustment factors of Table 4.5 .

Example: A1020A-2-PQ100I in (100-999) quantity, purchased 1H92.

Factor	Example	Value
Base price	A1020A	$43.30
Quantity	100-999	84 %

Time	1H92	100 %
Qualification type	Industrial (I)	120 %
Speed bin 2	2	140 %
Package	PQ100	125 %
Estimated price (1H92)		$76.38
Actual Actel price (1H92)		$75.60

Some distributors now include FPGA prices and availability online (for example, Marshall at http://marshall.com for Xilinx parts) so that is possible to complete an up-to-date analysis at any time. Most distributors carry only one FPGA vendor; not all of the distributors publish prices; and not all FPGA vendors sell through distributors. Currently Hamilton-Avnet, at http://www.hh.avnet.com , carries Xilinx; and Wyle, at http://www.wyle.com , carries Actel and Altera.

---

1. Actel speed bins are: Std = standard speed grade; 1 = medium speed grade; 2 = fastest speed grade.

2. The speed bin is a manufacturer's code (usually a number) that follows the family part number and indicates the maximum operating speed of the device.

4.8 Summary

In this chapter we have covered FPGA programming technologies including antifuse, SRAM, and EPROM technologies; the programming technology is linked to all the other aspects of a programmable ASIC. Table 4.7 summarizes the programming technologies and the fabrication processes used by programmable ASIC vendors.

TABLE 4.7 Programmable ASIC technologies.

	Actel	Xilinx LCA 1	Altera EPLD	Xilinx EPLD
Programming technology	Poly-diffusion antifuse, PLICE	Erasable SRAM ISP	UV-erasable EPROM (MAX 5k) EEPROM (MAX 7/9k)	UV-erasable EPROM
Size of programming element	Small but requires contacts to metal Special: CMOS	Two inverters plus pass and switch devices. Largest.	One n - channel EPROM device. Medium.	One n - channel EPROM device. Medium.

Process	Special: CMOS plus three extra masks.	Standard CMOS	Standard EPROM and EEPROM	Standard EPROM
Programming method	Special hardware	PC card, PROM, or serial port	ISP (MAX 9k) or EPROM programmer	EPROM programmer
	QuickLogic	Crosspoint	Atmel	Altera FLEX
Programming technology	Metal-metal antifuse, ViaLink	Metal-polysilicon antifuse	Erasable SRAM. ISP.	Erasable SRAM. ISP.
Size of programming element	Smallest	Small	Two inverters plus pass and switch devices. Largest.	Two inverters plus pass and switch devices. Largest.
Process	Special, CMOS plus ViaLink	Special, CMOS plus antifuse	Standard CMOS	Standard CMOS
Programming method	Special hardware	Special hardware	PC card, PROM, or serial port	PC card, PROM, or serial port

All FPGAs have the following key elements:

- The programming technology
- The basic logic cells
- The I/O logic cells
- Programmable interconnect
- Software to design and program the FPGA

---

1. Lucent (formerly AT&T) FPGAs have almost identical properties to the Xilinx LCA family.

4.9 Problems

* = Difficult, ** = Very difficult, *** = Extremely difficult

4.1 (Antifuse properties, 20 min.) In this problem we examine some of the physical and electrical features of the antifuse programming process.

- a. If the programming current of an antifuse is 5 mA and the link diameter that is formed is 20 nm, what is the current density during programming?
- b. If the average antifuse resistance is 500 W after programming is complete and the programming current is 5 mA, what is the voltage across the antifuse at completion of programming?
- c. What power is dissipated in the antifuse link at the end of programming?
- d. Suppose we wish to reduce the antifuse resistance from 500 W to 50 W . If the antifuse link is a

tall, thin cylinder, what is the diameter of a 50 W antifuse?

- e. Assume we need to keep the power dissipated per unit volume of the antifuse link the same at the end of the programming process for both 500 W and 50 W antifuses. What current density is required to program a 50 W antifuse?
- f. With these assumptions what is the required programming current for a 50 W antifuse? Comment on your answer and the assumptions that you have made.

4.2 (Actel antifuse programming, 20 min.) In this problem we examine the time taken to program an antifuse-based FPGA.

- a. We have stated that it takes about 5 to 10 minutes to program an Actel part. Given the number of antifuses on the smallest Actel part, and the number of antifuses that need to be blown on average, work out the equivalent time it takes to blow one antifuse. Does this seem reasonable?
- b. Because of a failure process known as electromigration, the current density in a metal wire on a chip is limited to about 50 k Acm $^{-2}$. You can exceed this current for a short time as long as the time average does not exceed the limit. Suppose we want to use a minimum metal width to connect the programming transistors: Would these facts help explain your answer to part a?
- c. What other factors might be involved in the process of blowing antifuses that may help explain your answer to part a?

4.3 (*Xilinx cell) Estimate the area components of a Xilinx cell as follows:

- a. (30 min.) Assume the two inverters in the cross-coupled SRAM cell are minimum size (they are not, the p- channels-or n- channels-in one inverter need to be weak-long and narrow-but ignore this). Assume the read-write device is minimum size. Estimate the size of the SRAM cell including an allowance for wiring (state your assumptions clearly).
- b. (15 min.) Assume a single n- channel pass transistor is connected to the SRAM cell and has an on-resistance of 500 W (equal to the average Actel ACT 1 antifuse resistance for comparison; the actual Xilinx pass transistors have closer to 1 k W on-resistance). Estimate the transistor size. Assume the gate voltage of the pass transistor is at 5 V, and the source and drain voltages are both at 0 V (the best case). Hint: Use the parameters from Section 3.1 , " Transistors as Resistors ."
- c. (15min.) Compare your total area estimates of the cell with other FPGA technologies. Explain why the assumptions you made may be too simple, and suggest ways to make more accurate estimates.

4.4 (FPGA vendors, 60 min.) Update the information shown in Table 4.7 using the online information provided by FPGA vendors.

4.5 (Prices) Adjustment factors, calculated from averages across the Xilinx price matrix, are shown in Table 4.8 (the adjustment factors for the Xilinx military and MIL-STD parts vary so wildly that it is not possible to use a simple model to predict these prices).

- a. (5 min.) Estimate the price of a XC3042-70PG132I in 100+ quantity, purchased in 1H92.
- b. (30min.) Use the 1992 prices in Figure 4.9 to derive as much of the information shown in Table 4.8 as you can, explaining your methods.

FIGURE 4.9 Xilinx XC3042 prices (1992). Problem 4.5 reconstructs part of Table 4.8 from this data.

3042	(1–24)	(25–99)	(100+)
50PC84C	$52.00	$47.30	$40.05
50PC84I	$67.30	$61.25	$51.80
70PC84C	$56.50	$51.40	$43.50
70PC84I	$73.30	$66.70	$56.45
100PC84C	$67.70	$61.60	$52.15
125PC84C	$114.00	$103.75	$87.80
50PP132C	$124.50	$113.30	$95.85
50PQ100C	$60.40		
50PG84C	$161.50		
50CQ100C	$194.50		
50PG132C	$191.20		

TABLE 4.8 Xilinx price adjustment factors (1992) for Problem 4.5

Purchase quantity, all types

(1-24)	(25-99)	(100+)	(5000+)
100 %	91 %	77 %	70 %

Purchase time, in (100-999) quantity

1H92	+18 months
100%	60%

Qualification type, same package

Commercial	Industrial	Military	883-B
100%			
	130%	varies	varies

Speed bin

50	70	100	125
100 %			
	110 %	130 %	220 %

Package type

3020:	PC68	PC84	PQ100	PG84	CQ100	
	100 %	106 %	127 %	340 %	490 %	
3030:	PC44	PC68	PC84	PQ100	PG84	
	100 %	107 %	113 %	135 %	330 %	
3042:	PC84	PQ100	PP132	PG84	PG132	CQ100
	100 %	175 %	240 %	310 %	370 %	375 %
3064:	PC84	PQ160	PP132	PG132		
	100 %	150 %	190 %	260 %		
3090:	PC84	PQ160	PP175	PG175	CQ164	
	100 %	130 %	150 %	230 %	240 %	

- c. (Hours) Construct a table (using the format of Table 4.8) for a current FPGA family. You may have to be creative in capturing the HTML and filtering it into a spreadsheet. Hint: In Microsoft Word 5.0 you can select columns of text by holding down the Option key.

Answer: (a) $211.85 (the actual Xilinx price was $210.20).

4.6 (PREP benchmarks, 60min.) Download the PREP 1.3 benchmark results as spreadsheets from http://www.prep.org . Split the participating companies among groups and challenge each group to produce an averaging or analysis scheme that shows the group's assigned company as a "winner." For hints on this problem, consult advertisements in past issues of EE Times .

4.7 (FPGA patents) Patents are a good place to find information on FPGAs.

- a. Find U.S. Patent 5,440,245, Galbraith et al. "Logic module with configurable combinational and sequential blocks." Find and explain a method to paste the figures into a report.
- b. Conduct a patent search on FPGAs. Good places to start are the U.S. Patent and Trademark Office (PTO) at http://www.uspto.gov and the IBM patent resource at http://patent.womplex.ibm.com . Until 1996 the full text of recent U.S. patents was available at http://www.town.hall.org/patent ; this is still a good site to visit for references to other locations. Table 4.9 lists the patents awarded to the major FPGA companies up until 1996 (in the case of Actel and Altera the list includes only patents issued after 1990, corresponding roughly to patent numbers greater than number 5,000,000, which was issued in March 1990).

4.8 (**Maskworks, days) If you really want to find out about FPGA technology you tear chips apart. There is another way. Most U.S. companies register their chips as a type of copyright called a Maskwork . You will often see a little circle containing an "M" on a chip in the same way that a copyright sign is a circle surrounding the letter "C". Companies that require a Maskwork are required to deposit plots and samples of the chips with a branch of the Library of Congress. These plots are open for public inspection in Washington, D.C. It is perfectly legal to use this information. You have to sign a visitors' book, and most of the names in the book are Japanese. Research Maskworks and write a summary of its implications, the protection it provides, and (if you can find them) the rules for the materials that must be deposited with the authorities.

TABLE 4.9 FPGA Patents (U.S.).

QuickLogic	Xilinx	5,329,181	4,713,557	5,308,795	5,008,855	
5,416,367	5,436,575	5,329,174	4,706,216	5,304,871		
5,397,939	5,432,719	5,329,181	4,695,740	5,299,150	Altera	
5,396,127	5,430,687	5,321,704	4,642,487	5,286,992	5,477,474	5,280,203
5,362,676	5,430,390	5,319,254		5,272,388	5,473,266	5,274,581
5,319,238	5,426,379	5,319,252	Actel	5,272,101	5,463,328	5,272,368
5,302,546	5,426,378	5,302,866	5,479,113	5,266,829	5,444,394	5,268,598
5,220,213	5,422,833	5,295,090	5,477,165	5,254,886	5,438,295	5,260,611
5,196,724	5,414,377	5,291,079	5,469,396	5,223,792	5,436,575	5,260,610

	5,410,194	5,245,277	5,464,790	5,208,530	5,436,574	5,258,668
Intel	5,410,189	5,224,056	5,457,644	5,198,705	5,434,514	5,247,478
4,543,594 1	5,399,925	5,166,858	5,451,887	5,194,759	5,432,467	5,247,477
	5,399,924	5,155,432	5,449,947	5,191,241	5,414,312	5,243,233
Crosspoint	5,394,104	5,148,390	5,448,185	5,187,393	5,399,922	5,241,224
5,440,453	5,386,154	5,068,603	5,440,245	5,181,096	5,384,499	5,237,219
5,394,103	5,367,207	5,047,710	5,432,441	5,172,014	5,376,844	5,220,533
5,384,481	5,365,125	5,028,821	5,414,364	5,171,715	5,371,422	5,220,214
5,322,812	5,362,999	5,023,606	5,412,244	5,163,180	5,369,314	5,200,920
5,313,119	5,361,229	5,012,135	5,411,917	5,134,457	5,359,243	5,166,604
5,233,217	5,360,747	4,967,107	5,404,029	5,132,571	5,359,242	5,162,680
5,221,865	5,359,536	4,940,909	5,391,942	5,130,777	5,353,248	5,144,167
	5,349,691	4,902,910	5,387,812	5,126,282	5,352,940	5,138,576
Concurrent	5,349,250	4,870,302	5,373,169	5,111,262	5,350,954	5,128,565
5,218,240	5,349,249	4,855,669	5,371,414	5,107,146	5,349,255	5,121,006
5,144,166	5,349,248	4,855,619	5,369,054	5,095,228	5,341,308	5,111,423
5,089,973	5,343,406	4,847,612	5,367,208	5,087,958	5,341,048	5,097,208
	5,337,255	4,835,418	5,365,165	5,083,083	5,341,044	5,091,661
Plus Logic	5,349,248	4,821,233	5,341,092	5,073,729	5,329,487	5,066,873
5,028,821	5,343,406	4,820,937	5,341,043	5,070,384	5,317,210	5,045,772
5,023,606	5,337,255	4,783,607	5,341,030	5,057,451	5,315,172	
5,012,135	5,332,929	4,758,985	5,317,698	5,055,718	5,301,416	
4,967,107	5,331,226	4,750,155	5,316,971	5,017,813	5,294,975	
4,940,909	5,331,220	4,746,822	5,309,091	5,015,885	5,285,153	

1. Mohsen’s patent on the antifuse structure.