Testing and Programming Actel Field Programmable Gate Arrays (FPGAs)

Testing has long been a struggle for users of masked gate arrays. To avoid board-level, system-level, or even possible field failures, the system designer must expend great effort in developing test vectors for gate array designs. Even after the vectors are developed, fault coverage for typical designs may be only about 70 percent, with about 95 percent coverage being the best possible. With a 70 percent fault coverage, typical masked gate array designs are likely to have 2 to 5 percent defective devices.

In general, field programmable logic devices have allowed users to avoid the need to develop test vectors. These devices allow tests to be performed by the semiconductor vendor prior to programming. However, most one-time programmable logic devices have not yet achieved the functional quality levels of other semiconductor devices, because they don’t allow the chip manufacturer to access and test all internal gates. Early one-time programmable devices had poor test coverage, and users were often disappointed to see functional failure rates of more than 10 percent on parts that had passed programming. Over time, on-chip test circuits and testing techniques have greatly improved, and now one-time programmable devices have functional defect rates in the range of 0.1 to 1 percent. Although this failure rate is low for individual chips, putting 10 such chips on a single board can still mean a board failure rate of 5 to 10 percent.

The Actel FPGA Product Family

There are five Actel FPGA product families. The ACT 1 family offers 1200 (A1010) and 2000 (A1020) gate products. The ACT 2 family consists of three products—A1225A, A1240A, and A1280A—with 2500, 4000, and 8000 gate array equivalent gates, respectively. This family offers improved performance and number of I/Os compared with ACT 1 products. The 1200XL products (A1225XL, A1240XL, and A1280XL) are functionally equivalent to the ACT 2 products but offer even better performance. The 3200DX family currently consists of two products with 6500 and 14,000 gates (3265DX and 32140DX) but will soon expand to include products up to 40,000 gates. The ACT 3 family contains five products (A1415A, A1425A, A1440A, A1460A, and A14100A) with 1500, 2500, 4000, 6000, and 10,000 equivalent gates, respectively. The ACT 3 products offer the highest performance of the five families.

Testability of Actel FPGAs

Although Actel’s FPGA families use a one-time programmable technology, the device’s unique architecture permits a degree of testability comparable to reprogrammable devices. Special test modes allow functional testing of unprogrammed devices at essentially 100 percent fault coverage. This testability is independent of the large number of equivalent gates in the A1010 (1200 gates) through the 32140DX (14,000 gates). To show how this is accomplished, we will first review the architecture of the Actel FPGAs and describe how they are programmed.

Architecture

The basic building block of all Actel FPGAs is the logic module. Each logic module is programmable and capable of implementing all two-input logic functions, most three-input functions, and many other functions up to eight inputs. With an architecture similar to a channeled gate array, logic modules are organized in rows and columns across the chip (Figure 1). Adjacent to each row of logic modules are routing channels. Horizontal routing channels are shown in the figure, but vertical channels also run through the logic modules. These are used to configure a logic module and connect inputs and outputs of logic modules to implement a design. Surrounding the array of logic modules and routing channels are I/O buffers and test circuits.

Within the routing channels are programmable antifuse (PLICE) elements. The antifuse is normally open and is programmed to form an electrical connection between routing elements. An antifuse that connects a horizontal routing track to a vertical track is called a cross-antifuse. An example of a logic module interconnection (or a net) is shown in Figure 2. Here the output from Module 3 is connected to a horizontal routing track by programming a cross-antifuse. Another cross-antifuse is programmed to connect an input to Module 4. In a similar manner, the output of Module 3 is connected to the input of Module 2. Notice that not all horizontal tracks are continuous across the chip. Often, tracks are broken into a series of smaller tracks called segments. Segments are useful because it is often desirable to connect logic modules that are close to each other, and a full horizontal track would waste routing resources and slow down circuit performance. Sometimes, however, it is necessary to connect two segments to form a longer segment.
Figure 1 • Logic Module Architecture

Figure 2 • Logic Module Interconnection
This can be done by programming a special type of antifuse referred to as a horizontal antifuse. As an example, the output of Module 3 is also connected to the input of Module 1 by programming two cross-antifuses and one horizontal antifuse. Vertical antifuses are used to connect two vertical segments (not shown).

A more detailed example of the Actel FPGA architecture is shown in Figure 3. Six logic modules (two rows, three columns) are shown. Between the two rows are six horizontal tracks. Down each column are five vertical tracks. The circles at the intersections of vertical and horizontal tracks represent cross-antifuses. There are also circles at certain points on the horizontal tracks; these are horizontal antifuses. No vertical antifuses are shown. Notice the transistors that connect both horizontal and vertical tracks. By turning on selected transistors, various horizontal or vertical tracks can be connected even though an antifuse has not been programmed. This ability to connect tracks in unprogrammed devices is used extensively during antifuse programming and is one of the key elements responsible for the excellent testability of the Actel FPGAs.

Logic configuration of modules is interesting because there are no dedicated antifuses in the module to accomplish this. Instead, the inputs (and outputs) of logic modules extend into the cross-antifuse array. Each logic module has eight to ten inputs and one output. By programming appropriate antifuses, an input can be connected to a dedicated horizontal ground line, a Vcc line, or a horizontal routing track. The logic module implements a particular logic function by tying appropriate unused inputs to ground or Vcc.

Figure 3 • Programmable Interconnect
Programming

The following discussions about programming and testing modes are specific to the ACT 1 family of FPGAs. However, basic concepts also apply to all other antifuse FPGA families.

An antifuse is programmed by applying a sufficiently high voltage across it. This voltage is referred to as Vpp. To access an antifuse deep inside the chip, it is necessary to create electrical paths from Vpp and ground to the antifuse. This is done by turning on the appropriate horizontal and vertical pass transistors. (In normal chip operation, these transistors are always off.) The transistors are turned on by applying Vpp to their gates. In Figure 4, we see an example of programming a typical cross-antifuse. Vpp is applied to a vertical track at the top of the chip, and ground is applied to a horizontal track on the right side. The design of the A1010/A1020 actually allows Vpp or ground to be applied from the top, bottom, left, or right, as is most appropriate to access a particular antifuse. Notice that Vpp is also applied to the gates of the horizontal and vertical pass transistors on the tracks accessing the cross-antifuse. The circled cross-antifuse now has Vpp applied to it on one side and ground on the other. This voltage breaks down the antifuse’s dielectric and creates an electrical connection between the horizontal and vertical routing tracks.

There is one other important consideration when programming an antifuse. Notice that the cross-antifuses in the same vertical track as the antifuse to be programmed also have Vpp applied to them on one side. This is true until the track is broken by a vertical pass transistor, below it, that is turned off. However, the potential on the other side of the antifuses is not being driven. Should this potential be at ground, the other cross-antifuses on the vertical segment could be accidentally programmed.

The same logic applies to other antifuses on the same horizontal track. Here, one side of the antifuse is being driven to ground, and if the other side were at Vpp, extra antifuses could be programmed. This problem is solved by first applying what is referred to as a precharge cycle. During the precharge cycle, all horizontal and vertical tracks are charged to Vpp/2. As a result, there is no voltage across the

Figure 4 • Programmable Interconnect
antifuses. The appropriate vertical track is then driven to Vpp, and a horizontal track to ground (Figure 5). At this point, other antifuses on the vertical track have a potential of Vpp/2 across them (Vpp on one side and Vpp/2 on the other). This Vpp/2 voltage is not sufficient to program the antifuses. Other antifuses on the same horizontal track also have Vpp/2 across them (Vpp/2 on one side and ground on the other). Most other antifuses in the chip still have Vpp/2 on both sides and will not be programmed.

Programming Algorithm
In concept, the Actel FPGAs are programmed in a manner very similar to many other programmable logic devices, and similar to memories such as EPROMs. The programming algorithm consists of the following steps:

1. An addressing sequence to select the antifuse to be programmed
2. A programming sequence whereby Vpp is applied in pulses until the antifuse is programmed
3. A soak or “overprogram” step to ensure uniform, low antifuse resistance
4. A verify step to make sure the antifuse was properly programmed

Unlike a memory in which an antifuse is addressed by applying a parallel address, the FPGAs are addressed in a serial manner by using the special DCLK (Data Clock) and SDI (Serial Data In) pins. There is a large shift register that travels around the periphery of the chip. Bits in this shift register can be used to drive tracks to ground, Vcc, Vpp, or float. It is also possible to sense the level on the track (high or low) and to load this information into the shift register. By shifting in the correct address, any antifuse can be selected for programming. The shift register also plays a key role in testing the chip. This will be discussed later.

The programming sequence starts with the precharge pulse whereby Vpp/2 is applied to the Vpp pin. This is followed by a programming pulse that applies Vpp to the pin. Following the

Figure 5 • Programmable Interconnect
program pulse, the voltage on the Vpp pin is returned to a nominal value (about 6 V). See Table 1 for a typical Vpp waveform. The precharge/program pulse sequence is repeated until either the selected antifuse programs or a maximum number of pulses is exceeded (in which case the antifuse is considered unprogrammable and the device is rejected).

Confirmation that an antifuse has been programmed is determined by monitoring the current on the Vpp pin. This current is very low (typically < 10 μA) until an antifuse is programmed. Once an antifuse is programmed, an electrical connection is made between Vpp and ground, in which case currents in the range of 3 to 15 mA may be observed on Vpp. Once this current is observed, the antifuse is considered programmed and enters the soak or “overprogram” cycle. Here, extra pulses are applied to the antifuse to achieve minimum antifuse resistance. Figure 6 shows the Vpp waveform for ACT 1 devices.

Test Modes of Actel FPGAs
The unique architecture of Actel FPGAs allows outstanding testability of unprogrammed devices at the factory. Details of the various test modes are as follows:

- The shift register circling the periphery of the chip can be both downloaded and uploaded. This allows the use of various test patterns to ensure that the shift register is fully functional.
- All vertical and horizontal tracks can be tested for continuity and shorts. There are several ways to implement these tests. One way of doing continuity testing is to precharge the array, turn on all vertical or horizontal pass transistors on a track, drive the track low from one side of the chip, and read a low on the other side. Shorts can be detected by driving every other track low after precharge and reading back on the other side. Note that these tests also confirm that the vertical and horizontal pass transistors will turn on.
- It is important for programming to make sure that all tracks can hold the precharge level. By charging a track, floating it, and waiting a predetermined amount of time, the track can be read back and confirmed to be still high.
- Leakage of vertical and horizontal pass transistors can be tested by driving one side of a track to a voltage via the Vpp pin and grounding the other side. All pass transistors except the one being tested are turned on. If excess current is detected on the Vpp pin, the pass transistor is considered defective.
- There are one or two dedicated clock buffers that travel across all horizontal channels. These buffers can be tested by driving with the clock pin and reading for the proper levels at the sides of the array.
- There are two special pins referred to as Probe A and Probe B (Actionprobes). By entering a test mode, the shift register can be made to address the internal output of any logic module. This output is then directed to one of two dedicated vertical tracks, which in turn can be observed externally on the Probe A or Probe B pin. This ability to observe internal signals (even on unprogrammed parts) allows Actel to perform a large number of functional tests. The first such test is the input buffer test. Input buffers on all I/O pins can be tested for functionality by driving at the input pad and reading the internal I/O output node through the probe pins.
- Test modes exist to drive all output buffers low, high, or tristate. This allows testing of Vol, Voh, Iol, Ioh, and leakage on all I/Os.

<table>
<thead>
<tr>
<th>ACT 1 Programming Algorithm</th>
<th>Current Parameters</th>
</tr>
</thead>
<tbody>
<tr>
<td>V Program</td>
<td>21 V</td>
</tr>
<tr>
<td>V Precharge</td>
<td>12.35 V</td>
</tr>
<tr>
<td>V Verify</td>
<td>6.0 V</td>
</tr>
<tr>
<td>t Program</td>
<td>150–300 μs</td>
</tr>
<tr>
<td>t Precharge</td>
<td>25 μs</td>
</tr>
<tr>
<td>I Threshold</td>
<td>–2.5 mA</td>
</tr>
<tr>
<td></td>
<td>(to detect programmed antifuse)</td>
</tr>
<tr>
<td>I Max</td>
<td>15 mA (clamp current)</td>
</tr>
<tr>
<td># Soak</td>
<td>30–800 pulses</td>
</tr>
<tr>
<td>Maxpulses</td>
<td>60,000</td>
</tr>
</tbody>
</table>
• The functionality of the programming circuitry can be tested functionally all internal logic modules. By turning on various vertical pass transistors and driving from the top or bottom of the chip, any of the eight to ten module inputs can be forced to a high or low. The output of the module can then be read through the Actionprobe pins. The logic module test allows 100 percent fault coverage of each module. In addition, the architecture allows modules to be tested in parallel for reduced test time.

• Actel FPGAs have one or two dedicated columns on the chip that are transparent to the user and used by the factory for speed selection. These columns are referred to as the Binning Circuit. Modules in the columns are connected to each other by programming antifuses. The speed of the completed test circuit can then be tested. The Binning Circuit allows the separation of units into different speed categories. It also allows the speed distribution within each category to be minimized.

• There are several tests to confirm that the programming circuitry is working. The first such test is a basic junction stress/leakage test. The program mode is enabled and Vpp voltage plus a guard band is applied to the Vpp pin. All vertical and horizontal tracks are driven to Vpp; thus, no voltage is applied across the antifuses. The Ipp current is then measured. If it exceeds its normal value, the device is rejected.

• There is a test to ensure that all antifuses are not programmed. This is referred to as the antifuse shorts test (or blank test). The array is precharged, and then the vertical tracks are driven to ground. The horizontal tracks are then read to confirm that they are still high. (A programmed or leaky antifuse would drive a horizontal track low.) The test is repeated by driving horizontal tracks low and reading vertical tracks.

• The functionality of the programming circuitry can be verified by programming various extra antifuses, on the chip, that are transparent to the user. Some of these antifuses were already described earlier when the Binning Circuit was discussed. Actel FPGAs also have a Silicon Signature. In the ACT 1 family, the Silicon Signature consists of four words of data, each word 23 bits in length. The first word is hardwired (no antifuses) and contains a manufacturer ID number as well as a device ID number. These numbers can be read by a programmer, and the proper programming algorithm can be automatically selected. The other words contain antifuses and are programmable. Actel is currently using bits in these words to store information such as the chip’s run number and wafer number. Thus, each Actel FPGA has traceability down to the wafer level. By programming this information, the functionality of the programming circuitry is also tested. Actel software also allows the user to program a design ID and check sum into the Silicon Signature. By later reading this back, the user can verify that the chip is correctly programmed to a given design.

• The most important antifuse test is the stress test. When this test is enabled, a voltage applied to the Vpp pin can be applied across all antifuses on the chip. (The other side is grounded.) The voltage applied is the precharge voltage plus a significant guard band. After the voltage is applied, the antifuse shorts test is again used to make sure no antifuses have been programmed. The antifuse stress test is effective at catching antifuse defects. Because the reliability of the antifuse is much more voltage dependent than it is temperature dependent, this test is also an effective antifuse infant mortality screen. See the “Actel Device Reliability Report” for details.

**Burn-In of Actel FPGAs**

As mentioned earlier, Actel has found that antifuse infant mortality failures can be effectively screened out during electrical testing, and it is thus unnecessary to do any kind of burn-in for standard commercial production units to screen out antifuse infant mortality failures. However, burn-in is still an effective screen for standard CMOS infant mortality failure mechanisms, and it is required for all military 883D products. MIL-883D Method 1005 allows several types of burn-in screens. These can be divided into two categories: steady state (static) and dynamic. Static burn-in applies DC voltage levels to the pins of the device under test. The device may or may not be powered up. Dynamic burn-in applies AC signals to device inputs with the unit powered up. These signals are selected so that the device receives internal and external stresses similar to those it may see in a typical application.

Static burn-in is by far the simplest to implement. By choosing appropriate biasing conditions and load resistors, it is possible to design a single burn-in circuit that can be used for both unprogrammed and programmed devices. It would not matter what pattern is programmed into the device. Static burn-in can be an effective screen for some types of failure modes, particularly those that may happen at device inputs or outputs (such as screening for mobile ionic contamination). It is not, however, very effective at stressing internal device circuits. Many internal nodes may be biased at ground without receiving any voltage or current stress. Signal lines will not toggle, and it may not be possible to screen failure modes such as metal electromigration.

A properly designed dynamic burn-in can effectively stress inputs, outputs, and internal circuits. However, dynamic burn-in of ASIC products can be very expensive because customer-specific burn-in circuits and burn-in boards must be designed and built to properly stress each design implemented in the ASIC. This results in large NRE costs and
long lead times to design and build these boards. From the standpoint of burn-in, a programmed FPGA is essentially the same as a mask-programmed ASIC, and it would require similar custom burn-in circuits to do a dynamic burn-in. However, Actel has been able to use the testability features of its FPGA products to allow effective dynamic burn-in of unprogrammed devices. This dynamic burn-in allows users to stress circuits in a way that static burn-in would be unable to duplicate.

During burn-in of unprogrammed units, test commands are serially shifted into each device by using the SDI pin and clocked by using the DCLK pin. There are three test modes shifted into each device. The first test stresses each cross-antifuse with a voltage of Vpp - 2V. (Vpp is normally set at 7.5–11V so that each antifuse gets 5.5–9V across it.) This voltage is applied to all vertical tracks while the horizontal tracks are grounded. Once enabled, the stress mode is held for 10 ms.

The second test mode is identical to the first except that the horizontal tracks are driven to Vpp - 2V while the vertical tracks are grounded. Note that both of these modes are similar to the antifuse stress test described earlier (although the stress voltage is lower during burn-in). Not only do these tests stress the antifuses, but they also toggle all routing tracks in the chip to Vpp - 2V and ground. All input and output tracks to the logic modules are also toggled.

The third test drives several I/O pins on the chip to a low state. Prior to this, they are at high impedance state and held at Vcc through pull-up resistors. This test confirms that the burn-in is being properly implemented by looking at these I/O pins to see if they display the proper waveform. It also passes current through each I/O as it toggles low.

Although the chip is unprogrammed, these tests allow users to apply stresses to the inputs, outputs, and internal nodes that are similar to what a programmed device may see in normal operation. Once burn-in is completed, post-burn-in testing, as specified by MIL-883D, is performed (including PDA) to ensure that fully compliant devices are shipped to the customers.

Conclusion

The description of the Actel FPGA architecture and the numerous test modes attest to the outstanding testability of these devices. All internal logic gates can be tested without programming antifuses other than the few for the Binning Circuit and Silicon Signature. Because Actel FPGAs are one-time programmable, the only item that is not fully tested at the factory is the programmability of all the individual antifuses. However, this is done on the programmer while the units are being programmed. Being able to test all internal gates allows Actel to achieve functional yields superior to other one-time programmable devices and equivalent to reprogrammable parts.

References:
2. AMD PAL Device Data Book, 1988, p. 3-106.