Reliability of Antifuse-Based Field Programmable Gate Arrays for Military and Aerospace Applications

John McCollum, Roy Lambertson, Jeewicka Ranweera, Jennifer Moriarta, Jih-Jong Wang, and Frank Hawley, Arun Kundu
Actel Corporation, Sunnyvale, CA

Abstract

The differentiating structure in an antifuse-based field programmable gate array (FPGA) is, obviously, the antifuse. This two-terminal device, which is used as the configuration element in this programmable microcircuit, must be highly reliable for critical applications. This paper will briefly review the structures of different types of antifuses, the reliability history, and the efforts and results from hardening this programmable element. The antifuses must be shown reliable in both the programmed and unprogrammed states. In the programmed state, currents will flow through the structure. In the unprogrammed state, there are several reliability issues. First, the element must be stable over time under the full military temperature and voltage operating conditions; that is, time dependent dielectric breakdown (TDDB) is not a concern for military and aerospace systems that have long service lives. Secondly, the element must be reliable in the circuit; that is, normal operations such as handling with exposure to some level of ESD and soldering operations, where high temperatures are seen. Lastly, the biased, unprogrammed antifuse must be reliable to heavy ion irradiation. The reliability analysis will focus on the 0.25 µm SX-S devices, designed for space use, with hardened antifuses. Issues with antifuses for smaller geometry processes will also be discussed.

Antifuse-based FPGAs are high-density devices, with integration levels from 10^4 to 10^5 gates per microcircuit. The reliability of this class of programmable microcircuit will be examined. The discussion will start out with the basics of how reliability is determined for this device. That is, how are the devices tested, the fault coverage, and how that is achieved. Following this introduction, a summary of the reliability of different products, spanning over a decade of experience, will be presented. In particular, we shall present data showing the reliability of the devices as a function of product life and product introduction. This will quantitatively show the level of risk, for critical systems, of using newer technologies vs. older, more established devices. To put the reliability of this class of programmable devices in perspective, the failure rate for devices will be compared with historical measures, starting with the earliest microcircuits built in the 1960s for Apollo and continuing through the decades, as the industry's gained experience and increased integration levels.

Unlike traditional microcircuits, the design of the device is completed by the end user and programmed in the laboratory. Additionally, while systems comprised of traditional circuits are designed with only the simplest computer-aided engineering (CAE) tools such as programs that generate netlist, the designs of FPGAs require an extensive suite of sophisticated software. The *field reliability* of the device is now dependent not solely on the skills of the device designers and engineer, but the intellectual property embedded in CAE tools as well as the models that they operate on, and the tool users experience. The tool suite for military and aerospace FPGAs typically consists of:

- Logic synthesis
- Simulation
- Optimization
- Placement and routing
- Parameter extraction
- Timing analysis
- Programming and verification

The impact of these tools on field reliability will be discussed.

Introduction

With the advent of FPGAs the system designer is actually responsible in part for the design of the integrated circuit and its SEU tolerance. In this design process he must use software tools and logic modules with which he probably only has a limited understanding. In this paper we will present the various levels of reliability that must be dealt with to produce a reliable product.

Antifuse

Antifuse based FPGAs are founded on a device (antifuse) that is not based on mainstream technology and therefore the manufacturer of such a part bears a bigger burden in reliability testing than do companies using purely mainstream technologies. The first antifuse technology used in FPGAs was the ONO (oxide, nitride, oxide) antifuse shown in Figure 1 and Figure 2. The first concern is that the dielectric must sustain Vcc across it for the life of the part without shorting. This needs accelerated testing known as TDDB (time dependant dielectric breakdown). The time to breakdown is measured for various voltages far exceeding the normal operating voltage. The time to failure can then be determined by extrapolating the TDDB curve (Figure 3). HTOL (High Temperature Operating Life) is also performed, which is an industry standard. This check typically tests the quality of the dielectric and verifies the TDDB testing. Secondly, the programmed antifuse must sustain a large number of AC logic pulse currents through it without significantly increasing its resistance. This is similar to electro-migration in metal lines. The conductive filament is shown in Figure 4. In the case of ONO this conductive filament actually reduces its resistance with increasing current stress (Figure 5). To verify that this holds true for long times at high temperature, antifuses were stressed at their programming current at 250 C until failure as shown in Figure 6. Failure analysis (Figure 7) showed that the limitation was the standard contact to poly-silicon and not the antifuse filament itself.

Utilizing the antifuse in space necessitated the testing of the part under heavy ion bombardment. It was shown that this thin ONO could be ruptured (SEDR) when a LET 53 ion passed through it with 5.6 megavolts/centimeter E-Field was applied (Figure 8). To achieve good reliability in this environment, it was necessary to thicken the ONO above that used in commercial parts. This resulted in longer programming times. This would not be tolerable for large commercial runs, but for the small quantities used in space this is quite acceptable.

With the advent of amorphous-silicon metal to metal antifuses, this allows higher performance and density. However the reliability must be verified again. Figure 9 shows a cross-section of a metal to metal antifuse. The TDDB curve is shown in Figure 10. At low fields this curve actually turns up as the leakage of amorphous silicon prevents the aging effects of trapped charge as seen ONO. Hence the TDDB for amorphous silicon is extremely good. Care must be taken in this measurement as high frequency voltage spikes can cause an erroneous rupture. While switch off of programmed antifuses is not a concern for ONO, it is for amorphous silicon. A programmed filament as shown in Figure 11, can open up if current equal to its programming current is passed through it (Figure 12). This process has been carefully studied and the SX and SXA FPGAs have been designed very conservatively to limit the current through the antifuse during operation. For this reason amorphous silicon antifuse parts go through a reliability test at -55C and high voltage to increase the current stress in the filament (LTOL, low temperature operating life). To further test the guard band, parts are programmed at various low programming currents and placed on LTOL to determine the failure point. As can be seen in Figure 12 the SX and SXA parts are greatly over designed in this respect. The SXA parts were further tested for SEDR and found to be excellent as shown in Figure 13 with no failures within the operating spec voltage.

.

ESD/PID

ESD is always a concern for integrated circuits, and especially in antifuse based ICs. For this reason no antifuses are connected to any pin. In this way Actel parts can exceed 2000 volts on each pin, or Class 2 ESD. A closely related phenomena known as PID (process induced damage) was pioneered on the 1010 product. It was discovered that antifuses could be damaged in process by Ion Implanters and Plasma Etchers. These equipment could apply approximately 20 volts on metal lines during processing, thus damaging or destroying antifuses. Antifuses proved to be especially susceptible to this as the ONO BVG had to be lower than the gate oxide BVG. This showed up as sort yield fails and antifuse stress failures (Figure14). As a result Actel lead the effort at the wafer foundries in detecting and curing this problem. All Actel products are also tested for this effect with stress tests at wafer sort and final test to assure that no such defect leaves the factory.

Soldering

Packaging and soldering temperature cycling places stress on the die. This is a concern as this can place stress on bond wires, metalization and via structures, as well as antifuses. Solder reflow effects are therefore emulated by a standard ‘preconditioning’ flow performed prior to submitting units for reliability testing. Preconditioning is an inherent requirement for qualification. Failure analysis is performed on any units that fail the Preconditioning step.

Summary of reliability as a function of product life and product introduction

Actel's FIT rate comprehends all failure modes including 168hr HTOL/LTOL which some may consider as infant mortality. HTOL testing is based upon programmed units as it is necessary to do life testing as the devices will be used in the field. It is worth noting that units used to generate the ORT (Ongoing Reliability Testing) data are not processed through an 883B equivalent flow, i.e. no burn-in was performed on blank units prior to the gathering of this data. For units processed through a military flow requiring some burn-in, we would expect a significant improvement in the FIT rate.

Product Reliability

The most common defect in any MOS integrated circuit that results in a field failure is gate oxide breakdown. This is usually tested for by stress testing (applying a voltage larger than spec). Stress testing always results in yield loss and no manufacturer wants to apply more stress than absolutely necessary. The temptation is always to stress at the lowest possible voltage to maximize yield. Actel parts though are by their very design required to withstand very large voltages during programming, even the low voltage gates. Thus very few gate oxide failures have ever been reported on Actel parts.

Manufacturers often post their ongoing reliability tests on their web sites. Figure 15 is a typical example. This one shows 15 years of Actel data combined with previous technology. The FIT rate of the Apollo onboard computer is a testament to the care taken by the Apollo team. Note that the older the Actel product the lower the FIT rate. As shown in Figure 16, this is really due to the fact that the it takes a long time to collect this data and so the plotted FIT is reflecting the limited data and not real defects. This data also notes an interesting trend in that only the oldest parts showed any defects. These defects were actually via failures dating from the early days of multilevel metalization. As the industry gained experience with multilevel metalization these problems were cured. The subsequent generations have not had these problems so the reliability is now actually better than it has ever been. This, combined with the higher level of integration today, has made systems much more reliable (Figure 17). The current production technology is only a scaled version of that which was developed 10 years ago and as such is very mature. However, the industry is beginning to introduce new technologies, namely Copper and Low K dielectric. These will bring with them new defects as well and their subsequent learning curve.

Fault Coverage

These very high integration levels exacerbate the testing problem. In a gate array or full custom circuit, logic can be buried very deep in a system making it difficult to test every possible logic state. In Figure 18 one can see that a large gate array that yields a very respectable 50% and has an outstanding 95% fault coverage has a defect rate of 3%, a normally unacceptable number for a merchant product. Hence the customer must put a great deal of effort into any custom design to insure adequate testing. FPGAs, on the other hand, are fully tested before personalization. All of the logic modules are tested in each state and all of the interconnect is verified to be continuous with no shorts. The key tests which allow Actel FPGA's to be fully testable are described below:

1) A shift register circles the periphery of the chip that can be loaded and read back during testing. The various shift register patterns are used to ensure that different portions of the internal circuits are fully functional.
2) All vertical and horizontal tracks are tested for continuity and shorts.
3) All horizontal and vertical pass transistors are tested for leakage and functionality.
4) The clock buffers are fully tested by driving with the clock pins and reading the proper levels at the side of the array.
5) There exist two special pins noted as Probe A and Probe B, which by use of the shift register, can be made to address the output of every logic module inside the array to ensure functionality. Every logic module in the blank FPGA is fully tested.
6) Through Probe A and Probe B, numerous other functional tests are allowed on a blank FPGA such as verification of the Input and Output buffer on every I/O.
7) Test modes exist to drive all output buffers to low, high or tri-state for testing. Thus Vol, Voh, Iol, Ioh standby current, and leakage tests are performed
8) There exists one or two dedicated columns on the FPGA which are transparent to the customer but allow Actel to program antifuses connecting inputs and outputs of modules into what is known as the "binning

circuit". The programming is performed at conditions identical to those utilized by the programmer while programming a customer design. This allows Actel to ensure functionality of programmed antifuses and guarantee speed performances of the FPGA.

9) There are numerous tests to ensure that the programming circuitry is fully functional to ensure >90% programming yield. The programming circuitry is exercised at levels slightly greater than the programming conditions used by the programmer.
10) There are special high voltage devices used to isolate low voltage logic from the elevated voltages during programming. No low voltage logic will be exposed to any high voltage signals during testing.
11) There are tests to ensure that all antifuses are un-programmed and to ensure reliability and quality of the antifuse in the FPGA. These proven electrical screens to ensure antifuse reliability are performed prior to and after the tests that exercise the programming conditions of the FPGA. It is also important to note that the architectural design of the FPGA ensures the minimal stress on antifuses.

Guaranteeing Programming

In antifuse FPGAs it is not possible to test that every antifuse will program, therefore a small percentage will fail programming. They are however guaranteed to be 100% functional once programmed. This is achieved by: testing that the required antifuse is in fact programmed, that no nets are left floating, and that there are no shorts between nets. The Actel programming algorithm serially identifies each antifuse requiring programming and applies a voltage in pulses to program the antifuse. A soak "overprogram" step is performed to ensure that the antifuse is fully programmed and the resistance of the antifuse is uniform across the chip. During programming, the programmer has the ability to ensure that each identified antifuse to be programmed is fully programmed and only that antifuse is programmed else it identifies the part as a programming failure. There are numerous tests performed before and after every antifuse is blown to ensure correct functionality after programming. A standby-ICC measurement is recorded pre-programming and post programming to ensure that the chip ICC characteristics have not changed due to programming. Through the extensive electrical tests and screens performed during regular production testing at Actel in combination with the extensive tests performed during programming, Actel is able to ensure functionality and reliability of its programmed FPGA.

CAE Tools

CAE tools, while very reliable in translating RTL code to a logic design, still require that care must be taken to produce a functionally reliable design. Current VHDL Code is much like Assembler Code, far more efficient than coding in Binary, but easy to create many bugs. Behavior Level Code would be far less prone to bugs, but not particularly efficient in silicon use or fast in speed. These tools are not yet available and not likely to be used until Silicon area and speed is not a premium in design. As a result a designer in the high reliability space market must be aware of the idiosyncrasy of the VHDL version he is using as well as the target FPGA. As an example, a designer may wish to insert a logic delay to account for clock skew. The VHDL code will happily delete the delay as redundant. The designer must make sure this stays in by using the *preserve* attribute.

Another problem area is using combinatorial circuits as clocks to registers. The HDL coding style can avoid this. The coding style can steer the logic mapping tools to infer FFs with built in Enable functions. For example, the following Verilog code will synthesize to a two-input AND gate the output of which will clock the register.

```
module gatedFF(Q, Data, Clock,
Enable);
    input Clock, Data, Enable;
    output Q;
    reg Q;
    wire GC;

    assign = (Clock && Enable);
    always @(posedge GC)
    begin
        Q = Data;
    end
endmodule
```

Once you rewrite the Verilog in the following way, the tools are able to infer the Enable-FF.

```
module enableFF(Q, Data, Clock,
Enable);
    input Clock, Data, Enable;
    output Q;
```

```
    reg Q;

    always @(posedge Clock)
    begin
        if (Enable)
            Q = Data;
   end
  endmodule
```

The synthesis methodology can help you instantiate the special CLKINT or CLKBUF drivers for high fanout and clock signals. This will avoid the synthesis tool building a buffer tree that will adversely affect the timing and area of the design.

Register duplication is not a problem with Synopsys tools. While using Synplify, there is an option to turn this feature off.

Another area of concern is generating a design tolerant to SEU. For instance a user may wish to make a flip flop utilizing two or even 4 combinatorial modules and a logic optimizer may collapse it to a more efficient design which is not SEU immune. It may generate multiple pipelines to optimize speed, but may generate SEU sensitive designs that may have possible illegal states. Synthesis tools have improved over the years, but care still needs to be taken when using them.

The default logic design technique, using standard commercial sequential elements (S-FFs), produces designs that are most susceptible to SEU (single event upset) effects. There are two unique techniques (the CC-FF and the TMR technique) that can eliminate the SEU sensitivity and improve the SEU rate on Actel's radiation-hardened FPGAs. In addition, a fully synchronous design methodology creates a more stable circuit with shorter debug time for timing verification and it is also easier to migrate between process geometries and architectures.

CC-FF Technique

The CC-FF technique avoids the S-FF portion of the sequential element and uses combinatorial cells with feedback to implement storage. The Macro Library Guide lists all the macros that are implemented by combinatorial cells. The ACT 1, 40MX and RH1020 family of devices do not have S-FFs and only use the CC-FF design technique.

The synthesis methodology also supports the CC-FF technique.

- Synplify provides an attribute called *syn_radhardlevel* that can specify the register implementation to be "cc" at any level of a module, an architecture or an individual register of the design.
- Similarly, a Synopsys script called *sequential_combinatorial* is provided to implement this technique.

TMR Technique

Triple module redundancy (TMR) is a well-known technique for SEU mitigation. Instead of a single FF, TMR uses three FFs leading to a majority gate voting circuit. If one FF is flipped to the wrong state, the other two override it and the correct value is propagated to the rest of the circuit. The cost of this method is three to four times the area and twice the delay required for S-FF implementations.

The synthesis methodology also supports the TMR technique.

- Synplify attribute *syn_radhardlevel* can specify the register implementation to be "tmr" or "tmr_cc" at any level of a module, architecture, or an individual register of the design.
- Another Synopsys script called *sequential_triple_voting* is provided to implement this technique.

These techniques are, however, not needed with the RT54SX-S family which have self-refreshing TMR built into every register. This family of devices is practically ion proof, making the job of the designer much simpler.

New devices such as the SXS devices are made with the idea of taking advantage of past lessons. Antifuse devices have isolation transistors to keep programming voltages out of the logic circuits. During normal operation a charge pump drives these gates high to pass a logic signal. During power-up the logic is going through indeterminate states causing a power drain and an output spike. The new devices put all modules in a predetermined state until the charge pump has reached operating voltage and then all modules will immediately switch to the correct state. The outputs are tri-stated until the correct state is reached and then enabled. Additionally the outputs can be programmed to have a 50ua sink to ground or Vcc until the outputs are enabled. Slew control has also been added to the outputs to minimize ground bounce problems. These parts also contain triple module redundancy such that SEU does not have to

be considered by the designer. The JTAG TRST pin, which should always be programmed and grounded in space, is programmed before leaving the factory so that it is not inadvertently left unprogrammed by the user.

Synchronous Design Methodology

Synchronous design should always be used when designing with FPGAs. Many customers have utilized the design by test methodology often adding logic delays until a design functions. These often fail if the Vt of the transistors change causing slight changes in timing, something likely to happen in space. Often designers believe they are using synchronous design, but allow asynchronous design in innocuous areas. Additionally the timing analysis available in Actel tools is conservative and should always be used, particularly static timing analysis.

Fully synchronous designs follow certain basic rules. All registers that share the same data path , should be connected to a common clock network of the FPGA. Every element on the same clock should be triggered on the same clock edge to remove any dependencies on the duty cycle of the clock. If data must cross between multiple clock domains, the data must be re-synchronized between these domains using one or two registers to ensure that the data is always in a known state.

Common practices of clock gating, derived-clock and divided-clock situations violate these synchronous design principles. The enable function of the register can be employed in each of these situations to become fully synchronous. Other asynchronous situations of feedback loops, one-shots, data sampling, preset and clear should also be replaced with circuits that change only with the system clock.

User Testing

Testing of systems is also a critical area. This paper is not covering this, but there is one particular point users need to be aware of. Flip flops will remember their last state upto 24 hours. If power up is critical, always set the flip flops to the opposite state of the desired state for approximately one hour before power down followed by the power up sequence. In this manner the flip flop will power up in the wrong state to facilitate valid initialization testing. Also signals sent to the FPGA to initiate a power on reset should not be applied until the power supplies reach spec voltage, otherwise the logic modules may not be active.

Summary

Devices and CAE tools have improved a lot in the last 30 years and continue to improve. Additionally the very high levels of integration have made systems far more reliable. Ultimately the reliability of a specific design rests with the system designer and software engineers. Integrated circuits and CAE tools benefit from the heavy use by multiple users, which improve the quality of these systems. The user on the other hand has the problem of designing systems that may be used only once in a hostile radiation environment and failure may not be an option. This often may be the weakest link, and therefore the user must take care to know the products he uses and utilize sound design practice.