Quarterly newsletter for Mil/Aero engineers

Issue 1 | 2013

Our First Issue

Synopsys is pleased to announce the Mil-Aero Technical Bulletin. Its mission: to provide important technology information and insights to engineers and managers in the aerospace and defense sectors. In each edition, technologists address timely challenges and methods applicable to the design and verification of high-reliability electrical, electronic and multi-domain systems. The series will include a range of viewpoints, including thoughts on: FPGA implementation, prototyping and debug; modeling and simulation of multi-domain physical systems; high-level/model-based design for signal processing algorithms, fast simulation, creation of synthesizable RTL and more.

In this edition, Kurt Mueller of Synopsys describes how NASA uses the Saber platform to implement their simulation-based analysis methodology used for the mission-critical power system design on the Orion Multi-Purpose Crew Module. Additionally, Angela Sutton of Synopsys explains a number of synthesis techniques for the implementation of highly reliable FPGA designs. In the Q&A section, our panel of experts address questions on the topics of FPGA synthesis methods for safe finite state machine implementation, high-level simulation and tools for worst-case analysis to ensure proper system operation over a wide range of environmental and tolerance variations.

We hope you enjoy this issue!

With sincere regards,

The Synopsys Mil-Aero Team

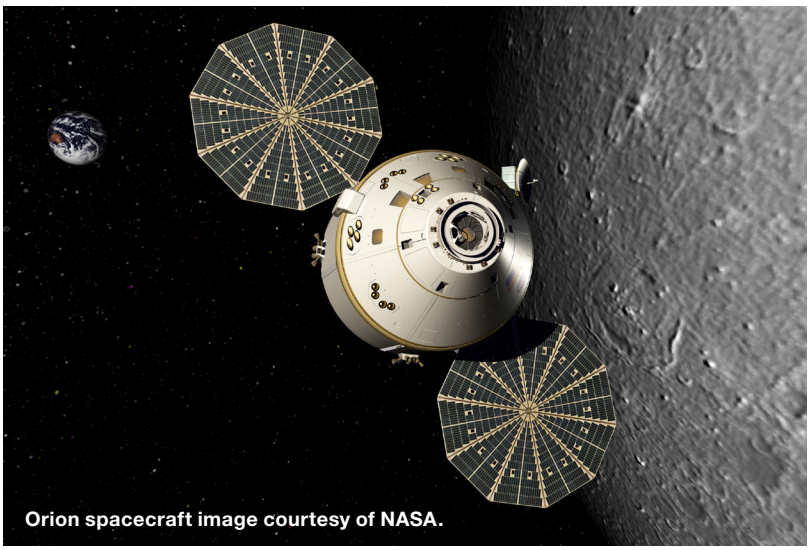
Contact the MTB Team
For inquiries and submissions to future issues:
mtb@synopsys.com

Military & Aerospace

Technical Bulletin

Space Vehicle Power Systems: The Ultimate Robust Design Challenge

The power systems design team at NASA uses Synopsys' Saber to perform the system and circuit analysis that is essential for checking power quality and verifying interface compatibility for the Orion Multi-Purpose Crew Vehicle. Kurt Mueller, Synopsys, helps explain what it takes to deliver mission-critical power engineering.



Orion spacecraft image courtesy of NASA.

In This Issue

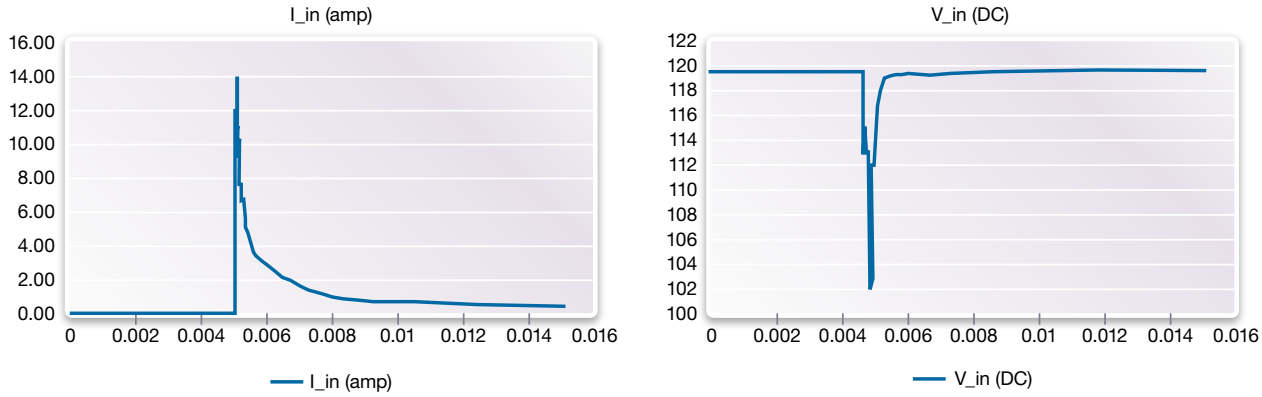


Figure courtesy of NASA.

Figure 1: Load start-up behavior showing typical inrush current and voltage characteristics

Designing equipment for space travel requires the ultimate in robust system engineering. Hardware must be capable of withstanding extremes of vibration, shock, temperature and radiation if it is to perform successfully and enable the crew to achieve its mission objectives. There is no margin for error.

The Orion Multi-Purpose Crew Vehicle (MPCV) is NASA's next-generation exploration vehicle that will carry crews into space, sustain them during space travel and provide safe re-entry from deep-space return velocities. NASA's power systems design team is responsible for the inline design, analysis and test of the complete power system for Orion, including power distribution, interface compatibility, and power quality.

Current Challenges

Typically, sub-contractors will deliver subsystems that meet the power quality requirements specification and that conform to their own tests. Ideally, the requirements spec has captured everything, but in the real world, that cannot be guaranteed. The rigor of each sub-contractor's circuit analysis varies widely, as do their test capabilities. Simply relying on compliance with the specification isn't enough for one-of-a-kind space systems. To guarantee that the MPCV works as a system, the NASA team performs subsystem and full system tests and analyses using Saber. These ensure that the components are going to work correctly once they come together in the system. The aim is to close all of the interface design issues as early as possible using analysis before physical testing is performed in the lab.

Modeling and simulation is performed by NASA's engineering team using Synopsys' Saber platform, which enables them to look at the whole system or drill down to analyze a specific interface between a load and the upstream switchgear. The NASA systems team develops the majority of the models they use. Some of the subsystem models are derived from SPICE descriptions. Sometimes other sources, such as contractors or vendors, who are working on particular sub-systems, will supply the models. Regardless of the source, all of the models are brought together in Saber for the system-level analysis.

Beyond ensuring that the system meets the requirements specification, the design team also has to address specific design interface issues. For example, they have performed extensive analysis on the Orion power interfaces to refine the control of inrush current, to ensure that the hardware is compatible with the switchgear (Figure 1). Any mismatch between interfaces can result in system-level instabilities in the power network. In one case, an interface

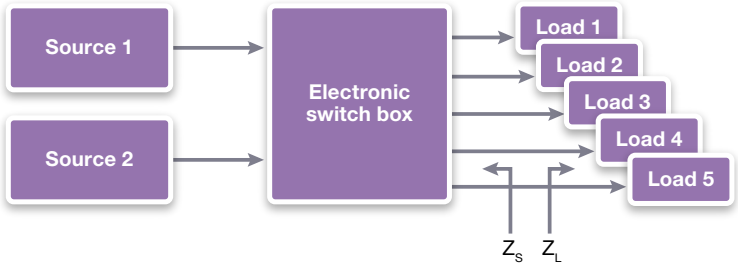


Figure courtesy of NASA.

Figure 2: Block diagram showing how source (Z_S) and load (Z_L) impedances relate to hardware

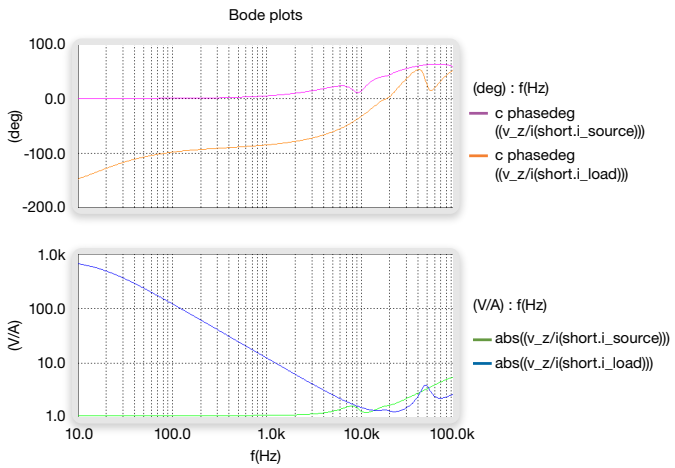


Figure courtesy of NASA.

Figure 3a: Bode plot shows impedance as a function of frequency, used in analyzing margin of stability and control system compensation circuitry

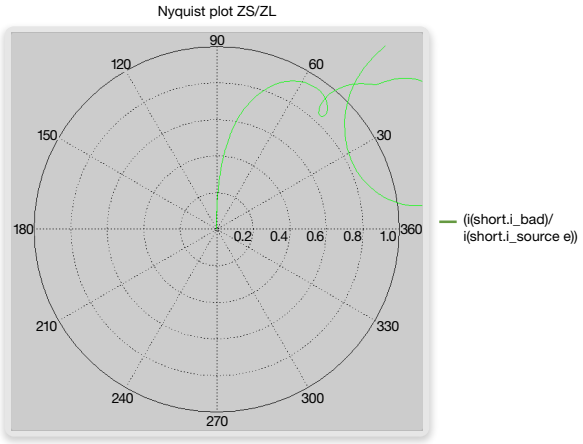


Figure courtesy of NASA.

Figure 3b: Nyquist plot enables analysis of system stability

problem meant that the switchgear was unable to activate the equipment attached to the power bus. This resulted in the equipment erroneously shutting down when it should have started up. Obviously, failures of this kind could be catastrophic were they to occur during an actual mission.

It is vital that any design changes that the design team makes do not result in further instabilities, so a combination of tests are used at the component level, alongside system and sub-system analysis. NASA employs a range of different analysis techniques to evaluate various aspects of system stability (Figure 3a and 3b) including:

- Small signal frequency domain analysis (to ensure small-signal steady-state stability)
- Time domain analysis (for ensuring large-signal AC system stability and to analyze inrush current profiles)
- Fault injection analysis (to validate protection and safety functions in the system)
- Monte Carlo analysis (to evaluate system level power stability margins)

Analysis is focused at the system level, enabling the assimilation of all the loads and all the different conditions simultaneously (Figure 2).

Avoiding Mission-Critical Problems

Fully integrated, the MPCV system comprises around 30 different power distribution loads, which gives rise to hundreds of possible test permutations. The engineering team simply doesn't have time to conduct all of those tests on the physical system, and hardware tests are expensive to carry out. To mitigate these challenges, the critical tests are done in hardware while Saber is used to validate a much wider range of operational scenarios through simulation.

Hardware integration and test time is at a premium for space vehicle design and development, due to the one-of-a-kind nature of the projects. It can be hard to quantify the benefits from a change in the methodology, because there is no production line to compare improvements against. Risk reduction is NASA's primary objective in using Saber. Employing analysis techniques for power system and load interfaces reduces the risk of encountering problems during the downstream hardware integration phases. It is critical for the team to identify and solve potential system design compatibility issues earlier in the project lifecycle. Late-stage problems can cost millions of dollars to solve and have the potential to throw an entire program off the tracks.

The Sum of the Parts

Environment, especially temperature, plays an important role in modeling. However, it is usually too time-consuming to develop a complete physical model while in the early development stages. For sub-circuit and power system interface compatibility analysis, the design team isolates the external dependencies by making assumptions to bound the problem so that they can perform a purely electrical analysis to make the design robust. The aim is to think through the

simplifying assumptions in order to be conservative, but not overly constraining. The performance boundary is explored by establishing pass-fail criteria using sensitivity analysis, which enables the team to validate their assumptions about the effects of environmental conditions on the electrical system.

Correct modeling of the components in the system is the key to accurate and useful simulation results. Physical modeling of components is generally a complex and time-intensive task. In order to balance these two divergent design challenges, the team uses the Saber model libraries to construct component models from basic building blocks that are readily available in the libraries. At times, it is necessary to customize or create a new type of model that isn't available in the libraries; in these cases, the team uses Saber's open Hardware Description Language (HDL), MAST, to write their own models. This combination of readily-available base models and the capability for customization or new model development provides the flexibility needed to meet both accuracy and schedule requirements.

From experience, the design team has a good idea about which corner conditions will give rise to the worst-case behaviors. They will typically run multiple simulations in Saber to identify the worst case scenarios. They can then explore and stress the system at the worst-case corners, which gives them more confidence that when they test the actual physical equipment, it will meet their requirements.

Saber allows the design space to be explored by calling on multiple methodologies to capture detailed models, and, where appropriate, use simplified models. This flexibility allows a focus on solving specific "real world" problems important at the time.

Back to the Future

The Saber-based analysis environment meets the design team's requirements today, but looking forward, there are some design flow changes and improvements that can further enhance their methodology.

NASA is always on the lookout for ways to accelerate schedules from development to final delivery. For example, the team is currently investigating how the vehicle system requirements can be verified in parallel with the vehicle assembly, integration and test. They are examining ways to streamline the hardware integration test-case matrix through pre-screening analysis to reduce the time between vehicle assembly and verification to launch.

Test and analysis are viewed as complementary verification activities, and, to refine the methodology, opportunities to integrate these processes more closely will be explored. The team captures a lot of transient data in the test lab, and, in the past, engineers have spent significant time manually comparing the test results with the analysis results to see where they differ. The use of Saber's TCL/TK-based scripting language, AIM, is being explored to automate comparison of the test and analysis environments by integrating test data into Saber.

NASA's current verification methodology could also benefit from modeling more of the non-electrical, physical domain components. By modeling electrical motors within the system, the accuracy of time-domain analysis could be improved.

No Margin for Error

Integrating and testing the power system with all of the other vehicle avionics and subsystems is a huge challenge, which involves bringing all the hardware, wiring and software into a single system during ground assembly test, complete functional checkout, and ultimately to the launch pad. Everything must work together flawlessly, and there is no margin for error. The verification effort leading up to the launch is extremely rigorous, and it is a monumental challenge to achieve flight readiness. Schedules are aggressive, and test time is at a premium. Using a variety of analysis methods with Saber builds confidence and reduces time-consuming physical test procedures, helping manage these challenges and assure mission safety.

About the Author

Kurt Mueller is currently a Business Development Manager for the Saber Product line at Synopsys. He is focused on developing new markets for Saber products with emphasis in power electronics related applications and is currently active in both Japan and Brazil. Prior to his current role, Kurt held R&D management positions in the Saber Simulation Technology and Modeling groups at Synopsys. Prior to joining Synopsys, Kurt held positions at Intel and Avant! Kurt received his MSEE in device physics and microwave engineering from Portland State University in 2003.

Creating Highly Reliable FPGA Designs

Angela Sutton of Synopsys explains how design teams can use automated features within Synopsys' Synplify design solution to protect their FPGA designs from soft errors.

Radiation-induced soft errors—"glitches"—became widely known in the 1970s with the introduction of dynamic RAM chips. The problem emerged as a result of radioactive contaminants in chip packaging, which emit alpha particles as they decay and subsequently disturb electrons in the semiconductor. This disturbance can result in an unwelcome change in voltage levels in digital logic.

In combinational logic, the voltage disturbance will most likely be transient; an unwanted transient signal is known as a single event transient (SET). However, synchronous logic—such as state machines, registers and memory—can store and propagate the transient error, which is likely to result in hardware failure. Such a stored error is known as a single event upset (SEU) (Figure 1).

As far back as 1996, researchers[1] at IBM estimated that each 256MB of RAM suffers one error per month as a result of soft errors. The error rate grows as logic densities increase, switching voltage levels decrease and switching speeds rise. Today's bigger, faster FPGAs will suffer from higher soft-error rates.

Beyond Aerospace and Defense Applications

Indeed, soft errors still occur today as a result of radiation from space—even within electronic equipment operating at sea level. For many years, design teams working in aerospace and defense have been aware of the need to protect their designs against SEUs. Today, engineers working in other market sectors are adopting techniques to guard against SEUs. We are increasingly dependent on the safe operation in automotive systems and medical equipment, but high reliability is no longer purely a safety-critical issue; it is a growing concern even for networking and industrial automation systems that demand high quality-of-service and uptime.

Detecting and Protecting Against SEUs

For some applications, design teams choose to use radiation hardened ("rad-hard") devices that are physically resistant to soft errors. However, rad-hard devices such as MicroSemi's RT ProASIC 3 and Xilinx's Virtex-5-QVR FPGAs come at a price premium and, as a consequence, find use mainly in mission-critical space projects.

Fortunately, there are design-based techniques that engineers can use to detect and protect against soft errors in normal sequential logic FPGA structures. Synopsys' Synplify Premier enables design teams to automatically apply techniques that build safety into the design. These techniques include triple modular redundancy (TMR) and fault-tolerant Finite State Machine (FSM) implementation.

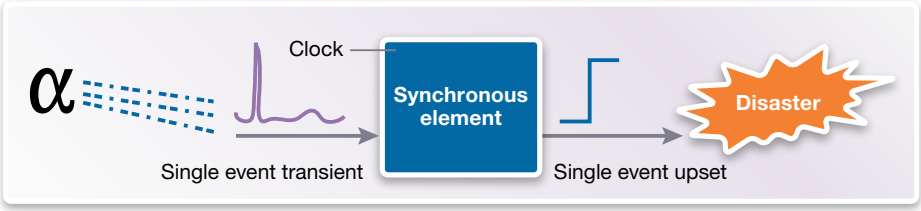


Figure 1: Single event upset (SEU) results from storing an unwanted transient event

Safe Finite State Machines*

A flipped bit in a state machine's state register can put the FSM into what the design team assumed would be an "unreachable" state under normal

[1] http://www.ece.rochester.edu/~garg/documents/garg.cs456survey05.pdf

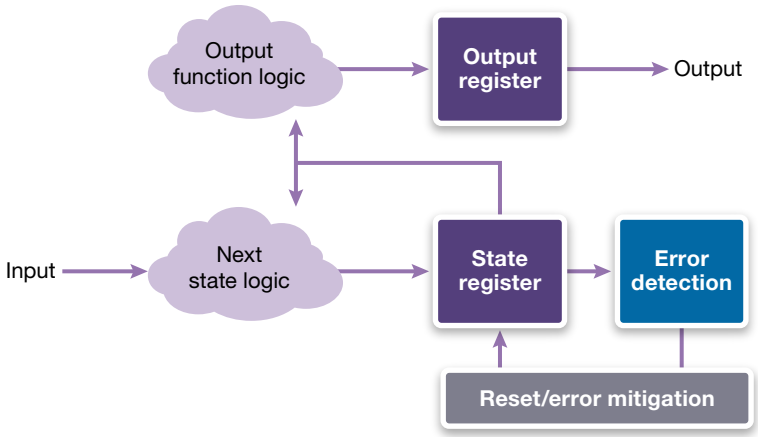


Figure 2: Additional error detection and mitigation circuitry is created to ensure correct FSM operation in the event of radiation-induced soft errors (SEUs)

circumstances. The FSM can become stuck in the invalid state, which is potentially disastrous in, for example, a control logic module.

Safe FSM implementation involves using error-detection circuitry to force a state machine into a reset state or into a user-defined error state so the error can be handled in a specific way. The Synplify synthesis software can be instructed to automatically add error detection circuitry to identify errors and create additional error mitigation circuitry to return the FSM into a safe state, so that the chip resumes correct operation (Figure 2).

For state machines that use "1-hot" state encoding, the error detection circuitry could be a parity checker, which ensures only one state register bit is high at any time. Once an error is detected, the state machine is then returned to a "safe" or "reset" state.

Fault-tolerant FSMs with Hamming-3 encoding can be used to detect and correct single-bit errors with a Hamming distance of 3, ensuring that the content of a state register erroneously reaching an adjacent state would be detected and that correct operation of the FSM would continue.

Deadlock occurs when a state machine enters a state from which it is not able to exit. Design teams can avoid deadlock by automatically inserting timeout counters on critical state machines.

Protecting Redundant Logic

Synthesis tools are designed to optimize away redundant logic, since the tool seeks to meet timing goals in the smallest possible chip area. Many of the structures that help to mitigate soft errors contain logic that synthesis tools would like to remove. Synopsys provides synthesis tool attributes such as syn_keep and syn_preserve in order to preserve the error detection and mitigation logic that has been created to improve reliability.

Designers can use the RTL "others" clause to specify a fault-tolerant or safe FSM. The "others" clause describes the behavior of the FSM or sequential logic, should an SEU cause it to enter a state that is nominally unused (that is, unreachable), but that in fact can be entered when the SEU causes a bit flip to occur. For example, the code fragment below specifies that the FSM returns to the IDLE state if it enters an unused state:

```
when others =>
     next_state <= IDLE ;
```

By default, synthesis would optimize away the "others" clause. Designers can now instruct the Synplify synthesis tool to preserve the "others" clause when optimizing Safe FSMs or sequential logic.

Error Correcting Code (ECC) Memories

Design teams can use error-correcting codes (ECCs) to detect and correct single-bit errors. Designers simply have to indicate in the RTL or constraints file which memory functions are safety critical for design. The Synplify Premier software infers the ECC memories offered by many FPGA vendors and automatically makes the proper connections.

Distributed TMR with Voting Logic

Design teams have used Triple Modular Redundancy (TMR) for years to help mitigate SEUs in sequential circuits. TMR triplicates part or all of the logic in a circuit and then uses "voting" logic to determine the best two from three results in case a signal is changed due to a soft error.

Figure 3 shows how a cone of logic is replicated three times to create identical cones along with voting logic. If one cone fails, the output from the voting logic will pass through to the output the signal with the two-thirds majority vote.

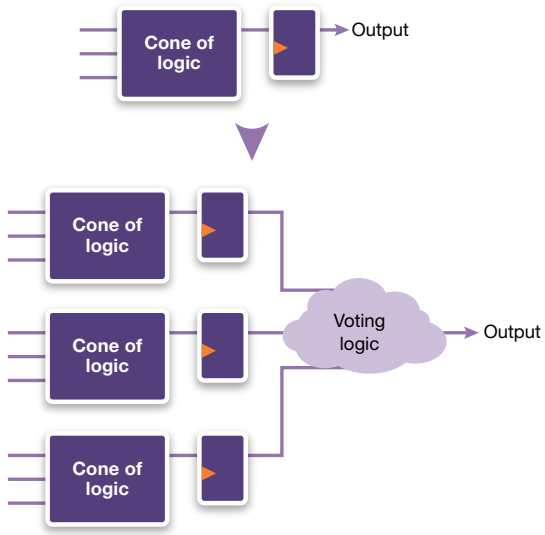


Figure 3: TMR helps mitigate SEUs induced by radiation effects by inserting redundancy during synthesis with triplicated circuitry and voting logic

For certain applications, especially those that cannot tolerate going into a reset or error handling state, TMR can be a good way to mitigate soft errors. The disadvantage of TMR is that it takes a lot of extra logic, that is, chip resources, to implement and can impose additional latency in the output of the cone of logic.

In general, the design team will want to selectively implement TMR at a local, block or system level. The Synplify Premier software lets designers decide which parts of the design would benefit from redundancy and automatically implements TMR for those areas.

Summary

Radiation-induced soft errors impose an increasing threat to the reliable operation of mil-aero, communications, automotive and industrial designs alike. Design teams can protect their FPGA designs against soft errors by incorporating redundancy and by developing safe sequential logic and fault-tolerant state machines with custom error mitigation logic. Such techniques ensure safe design operation by returning the design to a known safe state of operation, should a soft error occur. This logic can ensure high system availability in the field and provide reliable system operation. Synopsys Synplify Premier provides designers with the ability to automatically create this circuitry in FPGAs that are not radiation hardened, and the flexibility to control where and how these techniques are applied to the design.

*See the Q&A section of this newsletter for more technical details about designing safe finite state machines.

About the Author

Angela Sutton brings over 20 years of experience in the field of semiconductor and design tools to her role as staff product marketing manager for FPGA Implementation products at Synopsys. Before joining Synopsys, Ms. Sutton worked as senior product marketing manager in charge of FPGA implementation tools at Synplicity, Inc., which was acquired by Synopsys in May 2008. She has a B.Sc. in Applied Physics from Durham University UK, and a Ph.D. in Engineering from Aberdeen University UK.

Q&A

Ask our panel of experts

Robert Efram
Applications Consultant

Balaji Siva Prasad Emandi
Applications Engineer, Saber products

Chris Eddington
Director, High Level Synthesis and System Level products

Q How do I ensure my state machine is truly safe and that any unspecified state is covered? I noticed modern synthesis tools perform detailed logic and reachability analysis of state machines to achieve the best timing and area. Since reachability analysis shows that the “default” or “others” clause can never be reached, the logic for these branches are optimized out by synthesis tools.

Robert Efram
Applications Consultant

A Synthesis tools are very efficient in removing redundant and unused logic to achieve the best timing and area. Since the “default” or “others” clause is never reached during normal FSM (finite state machine) operation, this logic by default is optimized out. This means that under an SEU (single event upset) condition, the FSM could be set into an undefined state. Fortunately, there are several options in synthesis tools to preserve both the safeness of the FSM with SEU conditions and the timing and area optimizations critical for achieving performance.

Method 1: Enabling a “safe” mode for the FSM by using an attribute such that the optimizer inserts optimal logic to drive the FSM to a reset state when an undefined state is reached. This “safe” mode is the optimal method for both timing and area since a minimal set of logic is introduced to the design that utilizes the dedicated reset pins on the FPGA registers. The FSM performance is not degraded and since dedicated reset lines are used, minimal area increase is introduced. The FSM is rendered safe by detecting any undefined state and driving the FSM back to a reset state. This method does not honor the “default” or

"others" clause of the FSM. This method has been used for safe FSMs in the Synplify tools for over a decade.

Method 2: Enabling a "safe" mode in the FSM that honors the "default" or "others" clause in the FSM using an attribute. This method is often requested by designers who require the FSM to be driven to a state other than reset when an undefined state is reached. It is often requested to honor an error status bit that is defined under this condition. With this attribute, the synthesis tool maps all undefined and unreachable states in the FSM to the "default" or "others" defined logic. If the default logic is defined to be the reset state, the results are identical to Method 1. However, if a complex set of logic is used to a non-reset state, the area will typically increase, and the timing performance will typically decrease. The advantage is complex logic in the default clause will be implemented which typically includes the setting of error status flags. This method was introduced in the Synplify tools in the 2011.09 release.

Method 3: Enabling a "safe" mode that performs error detecting and single bit error correction using an attribute. This method automatically inserts Hamming-3 error-detection and single bit error correction logic to the FSM. This error correction logic will dynamically correct any single-bit error in the FSM state. The obvious advantage is the FSM will continue to operate as designed through an SEU condition and will not transition to reset and an error state. The disadvantage is due to the extra logic introduced, which typically increases area and reduces performance. This method was introduced in Synplify Premier in the 2012.03 release.

Method 4: Custom FSM error correction logic. The ability to disable all FSM optimizations, preserve FSM encodings, and preserve all FSM state bit registers is possible using attributes in synthesis tools. Using tool specific attributes, synthesis optimizations are disabled and the "default" and "others" clauses are honored. The results are exactly as written in the HDL. Typically, there is a large area and timing performance penalty. Synplify has attributes such as syn_statemachine to disable FSM reachability optimizations, syn_preserve to prevent FSM state register logic optimizations, and syn_keep to prevent signal/wire optimizations.

Q In aerospace applications, it is critical and challenging to make sure that systems operate as per design specifications and are optimized for efficiency, while ensuring safety is not at risk, even under worst case scenarios like extreme environmental or operating conditions. Does Saber have any analysis methodology to ensure this in the design stage?

Balaji Siva Prasad Emandi
Applications Engineer, Saber products

A Saber is not only a design and simulation tool, but also a design optimization tool. It has the capability to analyze the nominal and worst case behavior of a system and enables robust design. In the design stage, the engineer follows the specification and/or datasheet information to create an initial design. Then, in the verification stage, the design is simulated and verified for its nominal operation. However, additional verification measures are required to create a robust design. Specifically, all possible corner cases must be analyzed in order to avoid any potential hazards.

Saber Worst-Case Analysis (WCA), introduced in the D-2009.12 release of Saber, provides the capability to find the set of design parameters which might cause the design to operate outside the acceptable and safe range. Saber WCA also helps to identify the parameter combinations which could give optimal/expected design performance, enabling design optimization—for example, in synthesizing a filter design. This is an automated tool which takes user inputs on the parameters that are to be optimized. With a given set of goals, the tool has the capability of altering the design parameters automatically within their tolerance range to find the right combination of parameters to achieve the goal—for example, maximum power output. WCA in Saber is performed by looking not only at the corner values (min and max) of tolerances—which would eventually fail to identify the real worst/best case—but rather, by using a guided EVA (extreme values analysis) search approach to quickly identify parameter combinations from the entire design space that cause extreme worst/best performance, thus making the results trustworthy.

In addition to optimizing designs to match an objective function, the WCA tool can also be used to tweak the design/model parameters to match a given measured waveform or a graph. After importing the waveform into Saber, the WCA tool analyzes

the differences between the actual output waveform and the desired, imported waveform and optimizes the design parameters to achieve the desired model characterization.

In summary, Saber WCA provides a powerful, automated capability to find the optimum parameters of a design that satisfies a given objective function (worst/best) by executing an exhaustive set of inbuilt optimization algorithms. Please contact Synopsys for more details.

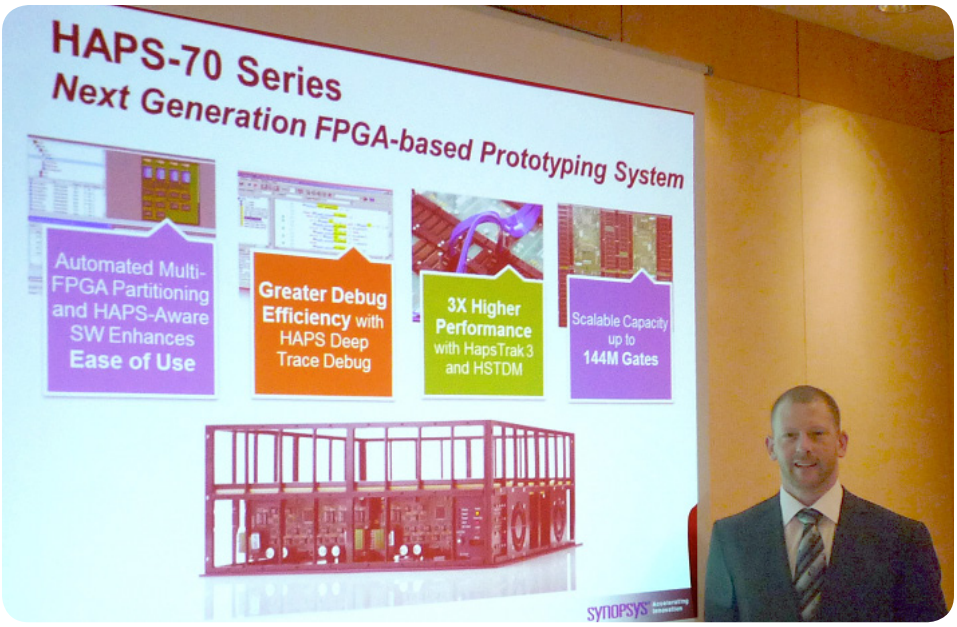
Q How can high-level design and simulation model generation help me with high-reliability and verification?

A When a design can be captured in a more concise and succinct way, this typically leads to fewer errors and much easier debugging. Furthermore, if a design can be captured in an environment that is more intuitive to the designer's domain (for example, algorithm domain, logic domain, analog domain, etc.) then the time to complete verification (design capture, debug, and verification) can significantly be reduced. Simulation environments that support multiple domains can also increase the coverage of such integrated systems. Synopsys provides signal processing algorithm implementation flows for FPGA and ASIC that allow mixed high-level and RTL design methodologies that ease the adoption while improving verification productivity and reliability.

Chris Eddington
Director, High Level Synthesis and System Level products

Have Questions?

Have a burning question you want answered? Submit questions to our panel of experts. Please send your email to **mtb@synopsys.com**.



Mick Posner, Synopsys Director, FPGA-Based Prototyping Solutions at the Design Automation Conference (DAC 2012)

Additional Resources

Saber website:
www.synopsys.com/saber

SaberRD Student/Demo Edition FREE software download:
www.synopsys.com/saber-sw-demo

SaberRD Datasheet:
www.synopsys.com/saber-ds

SaberRD Quick Start Introduction Video (English, 20min):
www.synopsys.com/saber-quick-start-video

FPGA Design website:
www.synopsys.com/fpga

Synplify Premier FPGA Brochure:
www.synopsys.com/synplifypremier-fpga

No Room for Error: Creating Highly Reliable, High-Availability FPGA Designs White Paper:
www.synopsys.com/fpga-high-reliability-wp

High-Level Block Design website:
www.synopsys.com/blockdesign

FPGA-Based Prototyping website:
www.synopsys.com/fpga-based-prototyping

Certitude Functional Qualification website:
www.synopsys.com/certitude

Certitude, Functional Qualification System Datasheet:
www.synopsys.com/certitude-ds

Mil/Aero Solution website:
www.synopsys.com/mil-aero

DO-254 Datasheet:
www.synopsys.com/do254

Upcoming Events

Synopsys plans to present on FPGA Hi-Rel at these events:

CMSE—Components for Military & Space Electronics
Los Angeles, CA
February 19-22

SEE & MAPLD—Single Event Effect Symposium and Military Aerospace Programmable Logic Devices Conference
La Jolla, CA
April 9-12

MILCOM
San Diego, CA
November 17-20

Share this by email

Feedback and Submissions

We welcome your comments and suggestions. Also, tell us if you are interested in contributing to a future article. Please send your email to **mtb@synopsys.com**.

SYNOPSYS® Accelerating Innovation

Synopsys, Inc. • 700 East Middlefield Road • Mountain View, CA 94043 • www.synopsys.com

01/13.TT.CS2547.