Quarterly newsletter for Mil/Aero engineers

Revised edition Issue 2 | 2013

Military & Aerospace

Technical Bulletin

About This Issue

The mission of the MTB is to provide important technology information and insights to engineers and managers in 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.

In this issue, Emmanuel Liégeon of Thales Alenia Space details his team's good track record of developing advanced ASICs for satellite payloads through their continued focus on tools, methodology and flows. Additionally, Chris Eddington of Synopsys explains how design teams targeting high-throughput and high-reliability satellite communications are using advanced system-level flows to support rad-hard FPGA and ASIC implementation and verification, and ultimately reduce their "time-to-flight". In the Q&A section, our team of experts addresses questions on the topics of validating the quality of RTL functional verification environments, reducing turnaround time on ASIC place-and-route, automated tracking and correction scrubbing techniques for SEU-induced errors for FPGAs and modeling lightning strike modes and scenarios for safe airframe design.

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

Focus on Tools to Take the Lead in the ASIC Space Race

Emmanuel Liégeon, design group manager with Thales Alenia Space France, describes how his team's early adoption of advanced design tools helps it to address the challenges unique to developing ASICs for space satellite applications.

Our core business is to design and build satellite technology for a range of applications including telecommunications, navigation and earth observation, for both commercial and military use.

Designing systems for deployment in space brings a number of challenges, which are quite different from those found in other industries. For example, design teams that tackle consumer and communications applications can call on a rich legacy of standards. The use of standards lends itself to a particular system-on-chip methodology, which is able to take advantage of standards-based IP and extensive design reuse.

By contrast, even if standardized interfaces and communication protocols enable the use of IP cores, we often find ourselves starting each new design from scratch. Indeed, due to longer product life cycles, the gaps between two generations of products are wider than for consumer electronics, with major algorithms evolutions and performance increase. Therefore, we begin with an engineering study of the algorithm at

In This Issue

a mathematical abstraction level using MATLAB. We devise an architecture that will support the algorithm and refine that using VHDL.

It's essential to get the architecture right and our system architects and ASIC designers work closely to iterate the design and evolve the architecture until we're sure that it will meet the system specifications. As well as using MATLAB, the team in charge of the signal processing study creates 'C' code to capture the algorithm. We then use high-level synthesis to automatically create RTL VHDL for use with Design Compiler. The high-level synthesis tool has knowledge of the target cell library, and can call on different variants of the operators within the library, depending on whether timing or area is the priority. This enables us to create an architecture based on a pipeline that will stand a good chance of meeting the timing performance.

Designing for Space

When developing products for the space sector, we don't have access to a broad range of commercial semiconductor technologies. Typically, when a foundry develops a new process we will generally use it for a period of ten years or more before moving to the next process node. Inevitably our designs become a lot more complex over such a long period of time. At the launch of a new process we can typically accommodate our designs with margin to spare, while at the end of a process's lifetime we find ourselves using it at the limits of its capabilities.

Because foundry processes for space are long-lived, we also find that we move between several generations of technology when we transition to a new process node. For example, we are currently moving from a 0.18-µm to a 65-nm process. Taking such a step means that we have to be sure that our tool flow will handle the requirements of the new technology.

Another issue that influences our approach to chip development is the nature of our project deadlines. Missing a deadline doesn't only mean that we suffer the financial consequences of being "late to market" by a week or two. Being late means that we miss the rocket launch. This constraint focuses us on minimizing the risk to schedules.

Partnering for Manufacture

Our backend design flow must support our route to manufacturing through our foundry partner. While we perform synthesis, the foundry manages the place and route of our designs.

Several years ago, we introduced physical synthesis to our flow as a result of experiencing issues trying to achieve timing closure on a large ASIC for a telecommunications payload. At that time we had transitioned from technologies in the class between 0.5µm and 0.35µm to technologies in the class between 0.25µm and 0.13µm. We had to go through several iterations of the design, passing it back and forth between the founder and our engineers, which was very time consuming. We worked with Synopsys to improve our design flow by adopting floor planning tools and custom wire-load models. This enabled our front-end designers, the team that knows the architecture very well, to place the main macros and memories. This particular device incorporated more than two hundred memory blocks, and the paths between the memories and the processing units were critical to achieving timing closure. Before adopting Synopsys physical synthesis, the best we could manage was to provide the engineers at the foundry with constraints in an Excel spreadsheet.

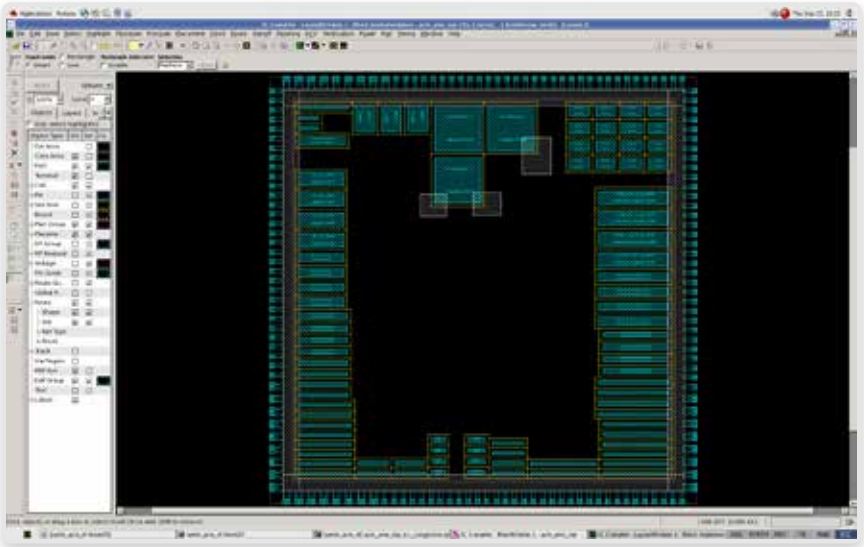


Figure 1: Floorplanning within IC Compiler-PC

As a result of adopting physical synthesis, we closed the timing on the design and we continue to work this way today with Synopsys IC Compiler-PC technology (a netlist to post- CTS placed gates solution for all technology nodes). We perform an initial placement to measure the performance of the design. We then typically transfer the netlist to our manufacturing partner (without the clock tree) with our global placement data (DEF) and scripts. Our partner will legalize the placement and perform the detailed routing. Following detailed routing we confirm the performance and the netlist goes back to the foundry for final signoff using Synopsys Star-RCXT.

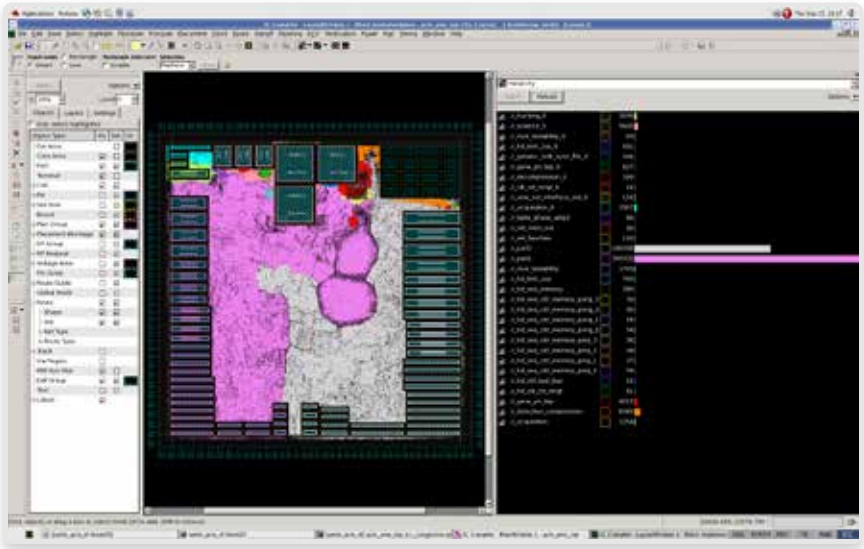


Figure 2: Hierarchical floorplanning using IC Compiler-PC

Predicting the Future

We have learned that when it comes to design methodology, it is far better to adopt the tools you will need before you hit a problem. By doing this, we are more confident about meeting our deadlines and we are in a much better position to develop new designs in new technology. The alternative, which is to evolve a flow as the design progresses, is too susceptible to delays.

For our next project, a large design for a telecom payload of several tens of Million gates and large memory areas, we will target a technology in the class below 100nm. For this project we will address clock-tree generation during the front-end design in order to manage all the critical paths at the very beginning of the flow. We will write the clock tree synthesis script to infer the clock tree for the design. Once we have validated the script and checked that the clock tree is feasible, we will remove it from the design netlist and give the database and script to the foundry to implement.

Another issue that we are anticipating for our future designs is how to deal with the increased power dissipation. Since we cannot cool chips by using a fan in space, as our designs get bigger we will need to accurately predict the power and adopt new design techniques for power reduction.

Increasingly, we will see a need for more flexibility and programmability in satellite applications. This will again influence our design flow as we consider using new FPGAs and microprocessor cores to enable our customers to more easily update their payloads over the typical 20-years lifespan of a satellite chip.

Managing Schedule Risk

As a result of adopting physical synthesis we are far more likely to meet our performance goals, which are becoming more demanding. We provide our manufacturing partner with a database, which includes the initial placement that we have evaluated, to run with its layout tools. We receive back an accurate indication of the timing performance. This becomes increasingly important when we are pushing the limits of our design by trying to squeeze the last drop of performance out of a mature semiconductor process. We can now proceed with layout knowing that the design is feasible.

Improved accuracy and performance is a major plus, however the key benefit for us is that physical synthesis reduces the risk of missing our schedules. By adopting this approach we have reduced the number of iterations between our team and backend designers, which has halved our layout time.

Key Ingredients for Success

There are several key reasons why we have a good track record of developing advanced ASICs for satellite payloads.

Our early and continued focus on tools, methodology and flows is essential to our success. We have a dedicated team to support and develop our flow and we involve our systems and ASICs teams in our design strategy. We aim to anticipate and adopt new EDA technologies before we have to deploy them on a critical design.

Image courtesy of Thales Alenia Space.

Our design team is organized in such a way that we design for a variety of applications within our sector. When we introduce a new technology for a very demanding application, the whole team benefits from the knowledge that we acquire on these leading-edge designs, which is a key differentiator.

Finally, our tool suppliers play an important role. Because they have a wealth of experience working with design teams in many other sectors – not just space – they help us to anticipate and plan for the problems we have not yet encountered.

Conclusion

As the technology nodes continue to decrease, thereby allowing the great improvement of design capabilities and bringing along new inherent constraints, tools availability and control are of utmost importance. This is true whatever the application area and space domain does not escape these rules.

Project profile	
Algorithm design	Mathworks MATLAB
C-to-RTL synthesis	Calypto Catapult C
RTL synthesis	Synopsys Design Compiler
Physical synthesis	Synopsys IC Compiler – Physical Compiler (ICC-PC)
Foundry signoff	Synopsys StarRC

About the Author

Emmanuel Liégeon is manager of the ASIC and FPGA design group at Thales Alenia Space, France. His team develops integrated circuits for satellite payloads.

He graduated from ENSEEIHT (Electrical Engineering), Toulouse, in 1997, and has worked at Thales Alenia Space France since 1998 as a CAD Engineer for the ASIC design flow and ASIC/FPGA designer. He has developed several ASICs and FPGAs for telecommunication, navigation and observation applications. He has been responsible for the development of multi-million gate ASICs for telecom equipment.

Emmanuel is a professor at ENSEEIHT and ISAE engineering schools in ASIC/FPGA design.

www.thalesaleniaspace.com

Reliable Verification Flows for Signal Processing ICs in Spacecraft Systems

Chris Eddington, Synopsys, explains how design teams targeting high-throughput satellite communications and high-reliability are using advanced system-level flows to support radiation-hardened FPGA and ASIC implementation and verification, and ultimately reduce their "time-to-flight".

Modern satellite communications systems are increasingly tasked with routing multiple gigabits of data per second as new applications demand increasing capacity, coverage and connectivity. Designing electronics for satellite payloads covers a diverse range of uses, including control and flight processors as well as high-capacity wireless communications. Whatever the application, satellite communications designs have exacting requirements that design teams must address in order to achieve project success.

High-Speed Signal Processing: Implementation Challenges

Complex signal processing algorithms intended for satellite payloads require devices that can support broadband communication rates and giga-sample per second (GS/s) throughputs. Because of size, weight, and power restrictions, these throughput requirements may require parallel architectures using slower system clocks and/or lower power. Exploring various architecture tradeoffs becomes the major part of the project effort and timeline, i.e. exploring tradeoffs in area, power, and target technology choices like FPGAs (antifuse-based, flash-based, SRAM-based) or ASIC.

Implementing complex signal processing solutions requires a sophisticated design and test methodology to ensure robust operation of integrated products. The process involves significant and rigorous methodologies for:

- up-front system modeling of performance to validate the design
- fixed-point modeling at both the module- and system-level to facilitate bit-true verification of RTL implementations
- reliable RTL synthesis flows with fast turnaround for architecture exploration
- gate-level testing and final hardware verification to ensure coverage of all potential input stimuli

These rigorous requirements make architecture exploration difficult. They also reduce the ability to re-target designs into new technologies without significant effort or impact on reliability.

Signal Processing Verification and Reliability

A combination of MATLAB and Simulink is a very common environment of choice for most algorithm design. Typically, a systems team will develop and refine its algorithms in MATLAB and then pass a floating-point specification over to the implementation or design team. The design team will then create an architecture that implements the algorithm using fixed-point arithmetic.

Maintaining a verification flow from the high-level specification is a critical part of an effective design methodology. Furthermore, the ability to reuse design and verification data across multiple targets (FPGA and ASIC) can significantly reduce effort, schedule, and risk for satellite IC designs.

> *"Using Synphony Model Compiler, and with extensive collaboration with Synopsys, we have streamlined our design flow at Boeing to eliminate the manual translation of our system-level description to RTL for targeting FPGAs and ASICs. The resulting efficiencies have reduced the time-to-flight on our new designs."*
>
> **Michael E. Holmes**
> Senior Manager, Supplier Management, Boeing Space and Intelligence Systems

Teams using FPGAs for prototyping face the added complexity of designing the FPGA before the ASIC. Whether they are using an in-house prototyping board or one supplied by a third party, they must still create a suitable architecture—one that will support the algorithm and map to the resources available on the target FPGA. Having proven the algorithm on the FPGA, the next challenge is to preserve and reuse as much of the engineering investment as possible in moving to the ASIC design. Management expectation is that FPGA prototyping will provide more reliable results for the algorithm implementation and also significantly reduce the schedule.

Solutions for Signal Processing: High-Level IP, Implementation and Verification

Synopsys has many customers tasked with developing military standard signal processing FPGA and ASIC designs. Using Synplify Premier and Synphony Model Compiler (MC), they have explored and created a range of parallel architectures for algorithms specified in MATLAB/Simulink. The Synopsys environment generates module-level RTL for both FPGA prototyping and ASIC implementation, and integrates with Synopsys low-power tools for ASIC optimization. Using such a flow, as shown in Figure 1, design teams are able to target and optimize for several "space-grade" radiation-hardened FPGAs, which means that they can use FPGA technology for both prototyping and flight hardware, if they wish. Furthermore, because of Synphony MC's automation for verification, they are able to re-target designs more quickly and reliably without major re-verification effort.

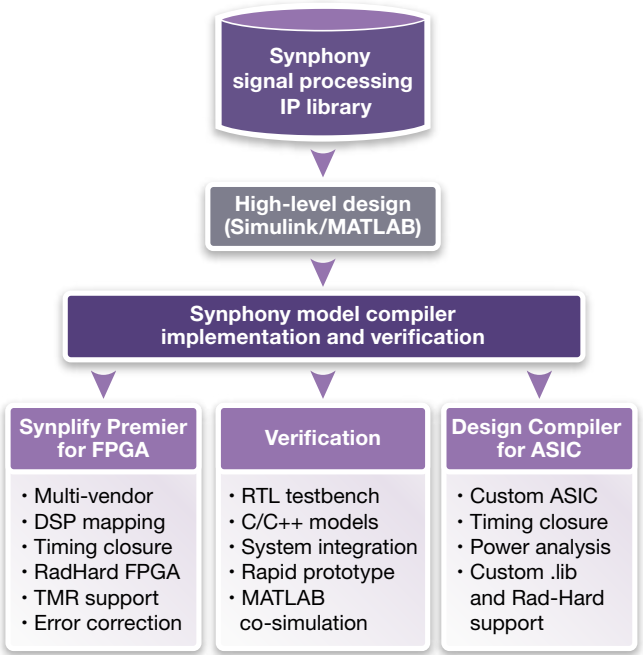


Figure 1: Synphony Model Compiler increases verification reliability, enables easier system prototyping and facilitates design re-use for multiple radiation-hardened FPGA and ASIC technologies. Combined with the Synplify Premier high-reliability features, IC designers can reduce schedules by as much as 80% for commercial and military space applications.

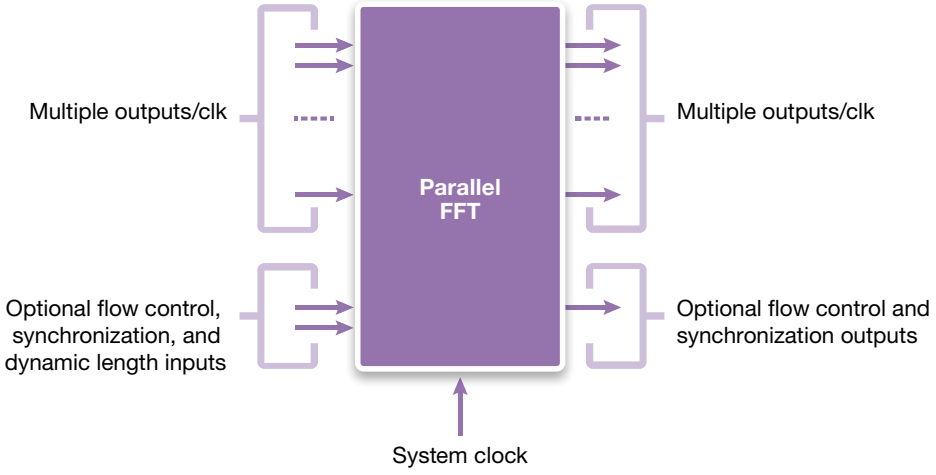


Figure 2: Parallel FFT architectures process multiple samples at a time to achieve greater throughput than the target device clock frequency

Design Example

A parallel FFT is a good example of a common signal processing function that may have multiple architecture choices depending on system requirements. A parallel architecture accepts multiple samples per clock and processes them concurrently to deliver multiple output samples per clock (Figure 2). While throughput increases, the payoff is greater area.

Synphony MC's library includes a parallel FFT IP block, which design engineers can use to target advanced FPGA devices like Xilinx's Virtex-7. The FFT is a custom IP block that uses arithmetic primitives to create an architecture based on user-specified options such as length, precision, flow control, and dynamic programmability.

Table 1 shows tradeoffs for different implementations of the FFT algorithm mapped to a Virtex-7 FPGA. The data shows that as parallel throughput increases, the area utilization of the multipliers increases with a slightly lower multiple (better than linear). Slower system clocks and timing closure yield sub-linear throughput growth as the architectures become more parallel; on modern FPGAs this degradation is getting smaller. Overall, it's possible to achieve better-than-linear throughput/area growth. Finally, latency decreases as parallelism increases.

Targeting ASICs and Analyzing Power Tradeoffs

To optimize their implementation for flight hardware, some design teams choose to target ASIC technology after they have proven the design in an FPGA. Synphony MC provides a more automated way to target custom ASIC technologies, including radiation-hardened libraries, while also closing timing and using Synopsys' low-power optimization tool flows. Furthermore, using ASIC technologies one can choose more parallel architectures, lower system clocks, and make significant reductions in power[1].

Synphony MC uses scripts to direct Synopsys' low-power optimization tools, which enables ASIC design teams to produce effective results at the gate level without having expert knowledge of the low-power flow. The ASIC low-power RTL implementation flow estimates a design's power dissipation and optimizes it for power during the synthesis process. Design Compiler and Power Compiler[3] call on a range of gate-level optimizations, such as clock-gating and operand isolation. These tools use

FFT Architecture Length 1024 16-bit	Number of Complex Input Samples	Max System Clock Achieved	FFT Throughput (Samples/s)	Hardware Multiplier Utilization	Latency (System Cycles)
Streaming	1	500 MHz	500 MS/s	32	1260
Parallel x2	2	500 MHz	1 GS/s	64	630
Parallel x4	4	490 MHz	1.968 GS/s	128	360
Parallel x8	8	490 MHz	3.92 GS/s	260	220
Parallel x16	16	440 MHz	7.088 GS/s	408	145

Note that the specific numbers in Table 1 are only valid for a given target and configuration of the FFT, which, in this case, is length = 1024, 16-bit input, dynamic length programmability (4-1024), and flow control. A designer was able to explore this range of architectures using Synphony MC in less than a day.

Table 1: Typical performance and area tradeoffs for parallel FFTs on Virtex-7 class devices.

Parallel FFT[1]	FFT Throughput (Samples/s)	System Clock	Relative Dynamic Power[2,3]	Relative Area[2]	Relative Leakage Power	Latency (system cycles)
Parallel x2	2 GS/s	1 GHz	1X	1X	.0003	597
Parallel x4	2 GS/s	500 MHz	.72	1.23	.0004	332
Parallel x8	2 GS/s	250 MHz	.59	1.64	.0008	200
Parallel x16	2 GS/s	125 MHz	.49	2.51	.0008	102

1. Using Synphony Model Compiler, parallel Radix2-MDC architecture and configuration as described in previous section
2. Implementation flow with Synphony Model Compiler 2013.03, Design Compiler 2011.09-SP4
3. Dynamic power estimated using Synphony generated design and testbench of 4 frames, VCS E-2011.03 for activity data, and Power Compiler 2011.09-SP1

Table 2: Parallel FFT power results in TSMC 40-nm LP process using Synphony Model Compiler's ASIC power estimation flow. Lower power per frame is achieved using more parallelism for a given throughput at the expense of additional area. Power is estimated using activity data generated by the Synphony MC testbench for a fixed number of frames.

the power characterization specified in the target library and switching activity to estimate power dissipation, which is available from the automated RTL testbench simulation from the user's high-level Synphony MC design.

The flow provides algorithm designers with a quick estimate of the power dissipation of their design at a very early stage in the project[2]. Designers can then optimize the micro-architecture of their design to meet the power targets.

Using the parallel FFT example, while throughput is maintained at 2GS/s across all the design variants, adding parallelism to the architecture makes it possible to significantly reduce the clock rate with a corresponding saving in dynamic power. Although leakage power increases because of the increased gate count, the increase is negligible compared to the considerable savings in dynamic power. Exploring and analyzing the four architectures in Table 2 manually would take many days or weeks, but can take only hours using the Synphony MC-based flow.

Summary

Design teams faced with building modern satellite communications systems have to overcome multiple engineering challenges, including meeting the system performance requirements and schedule constraints, without compromising reliability. High-throughput GS/s algorithms—common to satellite communications systems—benefit from architecture choices that tune performance, power, and area for various target technologies that include radiation-hardened FPGAs and ASICs.

By using a Synopsys solution comprising Synphony Model Compiler, Synplify Premier, and Design Compiler, design teams can quickly explore multiple architectures for mapping complex signal processing algorithms to FPGA devices and flight-ready, lower-power ASIC implementations. Using high-level IP and design flows from MATLAB/Simulink, space design teams can improve reliability and reduce design and verification task schedules by up to 80%.

References

[1] DSP Architecture Design Essentials, Markovic, Brodersen

[2] 'Early and accurate' power analysis: myth or reality?, Gupta

[3] Power Compiler datasheet

See the Additional Resources section on the back page for more information

About the Author

Chris Eddington is Sr. Technical Marketing Manager for High-Level Synthesis at Synopsys and has over 20 years of experience in ASIC and FPGA design. He has held various roles in technical marketing, algorithm development and IC design at semiconductor companies that develop video and audio conferencing ICs and wireless communications systems. He holds an MS engineering degree from the University of Southern California and an undergraduate degree in Physics and Math from Principia College.

Q&A

Ask our panel of experts

Steve Swanson
Applications Engineer, Saber products

Jafar Safdar
Product Marketing Manager, IC Compiler

Angela Sutton,
Product Marketing Manager, FPGA Implementation Products

George Bakewell
Director of Applications Engineering, Certitude

Q There are multiple lightning strike scenarios and related specifications to consider for airframe design. Is it possible to model a lightning flash in Saber?

Steve Swanson
Applications Engineer, Saber products

A The term lightning flash is commonly used to describe the entire event. But a flash can be one or several shorter discharges which repeat rapidly. These individual discharges are called strokes. Three commonly referenced lightning events are the single stroke, multiple stroke and multiple burst.

A single stroke is composed of one stroke with three components—a rise and fall time (typically exponential), and a continuing current.

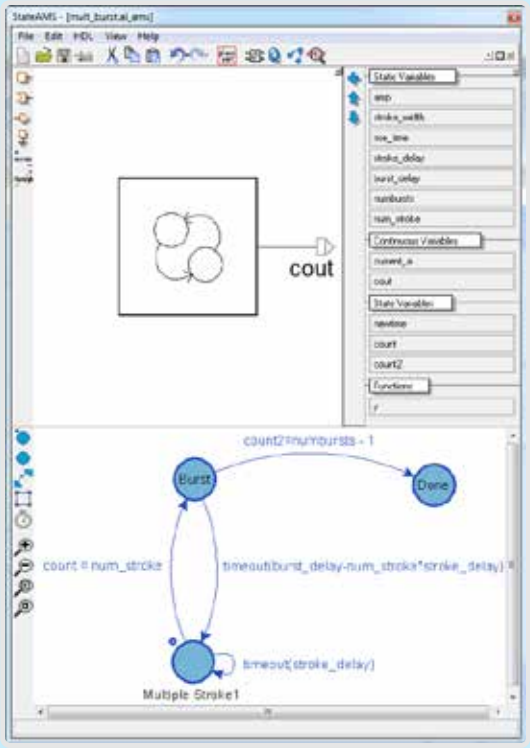
A multiple stroke strike is composed of several single stokes (without the continuing current component) each delayed by a statistical variation.

A multiple burst is composed of a series of multiple strokes with each burst again delayed by a statistical variation.

Using either of the two modeling languages, MAST or VHDL-AMS associated with the SaberHDL simulator, or using library primitives, it is possible to build high fidelity models of any of the various discharges described above. This includes statistical variations associated with multiple burst and multiple stroke as well as variations in amplitude and pulse width.

A third alternative is Saber's StateAMS (see figure). StateAMS creates both MAST and VHDL-AMS models in an easy-to-use state machine format. Where other state machine modeling tools allow only behavior that changes instantly in time (discontinuous in time), StateAMS accommodates continuous analog, state, and digital logic behaviors. This allows the engineer to concentrate on their design and model without knowing a modeling language.

By running such a simulation using lumped sum models with both thermal and electrical characteristics, designers are able to identify weak links in the grounding structure to improve the airframe's ability to handle such an occurrence.



Q One of the key challenges of the aerospace industry is to meet chip design schedules. What are the recommended methodologies I can use in IC Compiler to reduce the turn-around-time (TAT) and minimize the risk of missing my chip design schedule?

A In every release of IC Compiler we provide improvements that help reduce runtime and enable faster design closure. Here are the key recommended methodologies:

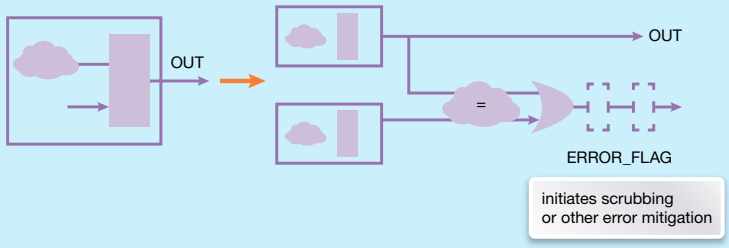
1. Use IC Compiler Recommended Methodology (RM flow) available on SolvNet. The Synopsys AC team can also help review your scripts, to ensure the settings and the flow are in-line with the IC Compiler release being used in the production flow. This helps reduce TAT and allows users to take advantage of the latest tool capabilities.
2. In the early part of the design flow, use feasibility analysis capability to improve SDC constraints. Clean constraints will reduce runtime and enable you to more quickly meet your design objectives, such as timing, power and area.
3. Take advantage of the multi-threading support. With one license of IC Compiler (and IC Compiler-PC) you can use four cores. This will help reduce the runtime very significantly in all phases of the design flow.
4. Follow Synopsys guidelines to setup appropriate settings in IC Compiler and the industry standard signoff tools, StarRC and PrimeTime. These recommended settings help minimize correlation issues and engineering change orders (ECOs) and enable faster design closure.

Jafar Safdar
Product Marketing Manager, IC Compiler

Q Is there a way to track whether SEU-induced errors are occurring in my FPGA design? I'd like to gather statistics on whether there is a problem and have errors trigger error correction scrubbing procedures.

Angela Sutton,
Product Marketing Manager,
FPGA Implementation Products

A Automated techniques are available that do this, as of the 2013.03 release of Synplify Premier. First of all, Synplify Premier provides an automated way to create Error flag signals that indicate the presence of an error on any piece of synchronous circuitry. The error flag can be used to drive Custom Error Correction or Scrubbing. This is done by enabling the "Duplicate with Compare" capability. Duplicate with Compare (DWC) is a method that is used to detect, but not correct errors such as those caused by radiation effects. A synchronous piece of circuitry is created in duplicate and output states of the duplicates compared. If an error occurs in just one of the synchronous logic blocks an output error flag from the comparison logic will indicate an error. (See figure)



To create Duplicate with Compare circuitry, apply the following synthesis attribute to the synchronous module of interest:

```
syn_radhardlevel {duplicate_with_compare}
```

Synplify Premier additionally provides automated Access to Error Status for Error Monitoring on Automatically error-correcting Circuitry for ECC RAMS, Distributed TMR circuitry, Distributed TMR with RAM circuitry, and Safe State Machines. To achieve Error monitoring and access, apply the following synthesis attributes:

```
syn_create_err_net { –name {<Non existing net name>}
    –inst {i:<hier path of HighRel instantiation>}
    –err_pipe_num {<no. of pipeline stages> Default: 0}
    –err_clk {n:<hier path to clock net>}
    –err_reset {n:<hier path to reset net>}
    –err_set {n:<hier path to set net>}
    –err_enable {n:<hier path to enable net>}
    –err_synch {<Boolean value to specify synch or asynch
        set/reset> Default: TRUE} }
syn_connect { –from {n:<net name specified in syn_create_
        err_net command>}
    –to {n:<hier path to an existing net> |
         t:<hier path to the input pin of EMIP> |
p:<hier path to the top error port>} }
```

Synplify Premier also provides Error Status monitoring and Correction at i/o's using a choice of automated "Duplicate with Compare" or "Distributed TMR" techniques. To do this, set the following synthesis attribute on your i/o connectors: `syn_highrel_ioconnector`

These capabilities are available for Xilinx—Virtex 4/5/6/7 devices and Altera—Arria V, Cyclone III/IV/V, Stratix III/IV/V devices.

Be sure to switch on Synplify Premier "Enhanced Optimization" when using these features.

For more information, please refer to the "demos and examples" section of the Synplify FPGA Graphical User Interface, and to the Synplify FPGA 2013.03 User Guide section: `Inferring High-Level Objects->Implementing High-Reliability Designs`

Q My company designs IC components used in mission-critical mil-aero systems and I'm concerned that our testing of the RTL design is not sufficient to prevent functional bugs from slipping through the process undetected. How can I get objective information about the quality of my verification environment?

George Bakewell
Director of Applications Engineering, Certitude

A You may have tools and techniques at your disposal, such as code coverage and functional coverage, that provide useful information about the quality of your verification efforts. Unfortunately, these methods have significant limitations that can provide a false sense of security.

Code coverage only provides information about how well your tests "activate" the logic in your design, but it's quite easy to create a verification environment with 100% code coverage that can't detect significant functional bugs.

Functional coverage provides a way to determine if you're executing all important functional aspects of your design, but the problem is that it's subjective—it relies on engineers to define a finite set of coverage points—and if too few are defined, corner-case bugs can be missed.

The Certitude Functional Qualification Solution provides a significant advance in measuring verification effectiveness for RTL designs. Certitude provides two types of feedback: An objective assessment of your verification environment's ability to activate, propagate and detect potential bugs and pointers to specific weaknesses, such as missing assertions or incomplete test scenarios, that must be fixed to make your environment more robust.

Certitude is based on a technique familiar to verification engineers who at some point have suspected that they're not really verifying some aspect of a design: Why not modify the RTL code to break that part of the design and make sure that at least one of the tests fails? Of course, this manual technique is fine in isolation, but it's not practical to use it to get a comprehensive assessment of the verification environment.

Certitude implements this approach in a systematic, intelligent way—by automatically inserting "faults"—that can be applied to even the largest and most complex designs. These faults are classified and prioritized through a number of automated processes, which is key to the practical application of this approach.

When Certitude finds that a fault is not detected, it means that there is a weakness in your verification environment, caused by a number of potential problems. Certitude guides you to the source of these problems and provides the results in an easy-to-use HTML-based report with links to the RTL code to show exactly what was changed for a given fault.

Certitude can also run in a statistical mode to quickly provide an overall measure of verification quality.

For more details on Certitude, see the Additional Resources section on the back page

Have Questions?

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

Image courtesy of Thales Alenia Space.

Additional Resources

IC Compiler Comprehensive Place and Route System website:
www.synopsys.com/iccompiler

Synphony High-Level Synthesis Solution website:
www.synopsys.com/hls

Multi-Gigahertz FPGA Signal Processing article:
www.synopsys.com/insight-fpga-dsp

FPGA Design website:
www.synopsys.com/fpga

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

Power Compiler, Power Optimization in Design Compiler datasheet:
www.synopsys.com/powercompiler

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

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

Saber website:
www.synopsys.com/saber

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

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

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

Understanding DO-254 Compliance for the Verification of Airborne Digital Hardware white paper:
www.synopsys.com/do254-wp1

A Methodology for a DO-254 Compliant Verification Flow using HVP white paper:
www.synopsys.com/do254-wp2

Upcoming Events

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

Microsemi Space Forum
Ahmedabad, India
July 30

Microsemi Space Forum
Bangalore, India
August 1

MILCOM
San Diego, CA, USA
November 17-20

SUBSCRIBE
Get the latest bulletin automatically via email

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
©2013 Synopsys, Inc. All rights reserved. Synopsys is a trademark of Synopsys, Inc. in the United States and other countries. A list of Synopsys trademarks is available at http://www.synopsys.com/copyright.html. All other names mentioned herein are trademarks or registered trademarks of their respective owners.
05/13.TT.CS2945.CPR/325.