Power Matters

Microsemi

Design Techniques for Implementing Highly Reliable Designs using FPGAs

Microsemi Space Forum Russia – November 2013

Feodor Merkelov
Applications Consultant, Synopsys

Design Reliability Involves Many Things

Complete and Accurate Design Specification

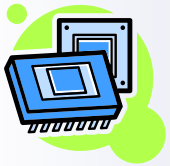
- Constraints and syntax checking
- Design specification checking

Built-in Safety

- Triple Modular Redundancy (TMR)
- Safe Finite State Machines (Safe FSMs)

Evaluate / Debug Correct Chip Operation

- Implement and preserve debug logic during synthesis
- Debug chip at the RTL level in hardware
- Debug and develop proof of concept using prototyping hardware



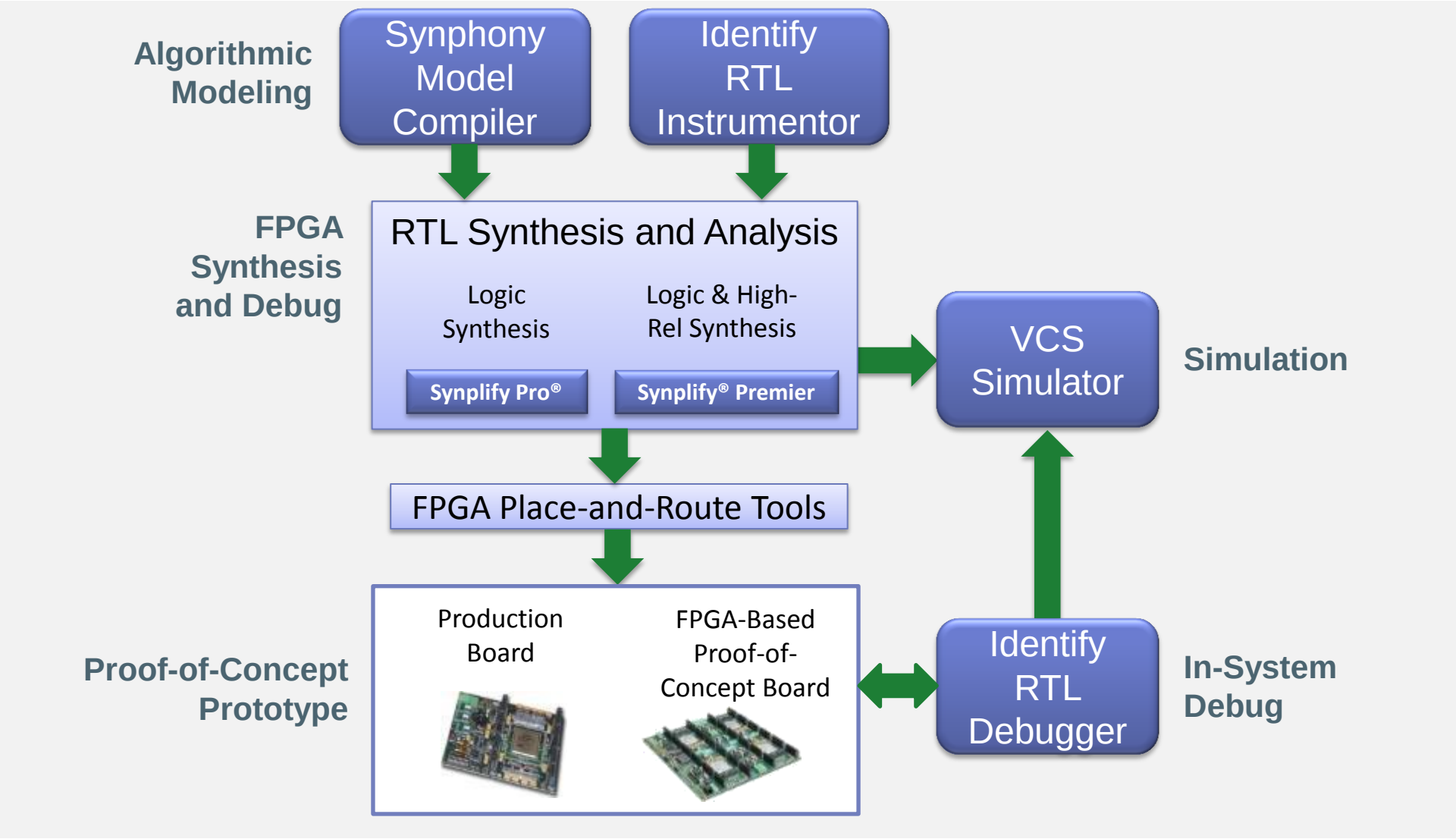
Reproducible, Documented Design Process

- Documentation, archiving and restoration
- DO-254 and process compliance

Verification and Equivalence Checks

- Disabling optimizations that obstruct requirements tracing
- RTL debug in operating hardware

Synopsys FPGA Design Suite – Design Steps



Why There is a Reliability Problem

High Reliability is Not Just for Space Applications

- CMOS transistors at today's process geometries have become susceptible to soft errors at ground level
  - Reduction in core voltage
  - Decrease in transistor geometry
  - Increase in switching speeds
- Needed for space applications
- A reality in terrestrial applications including, communications/ network equipment, automotive, medical, and industrial

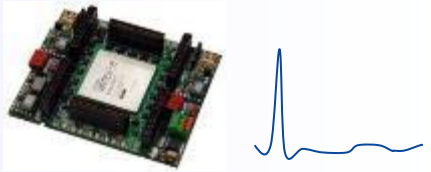




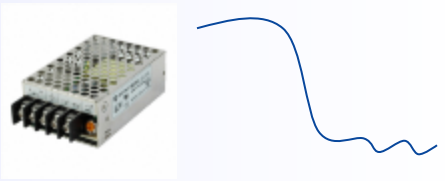
Why There is a High Reliability Problem

Radiation-induced circuit glitches (SEUs, SETs)

Power rail glitch



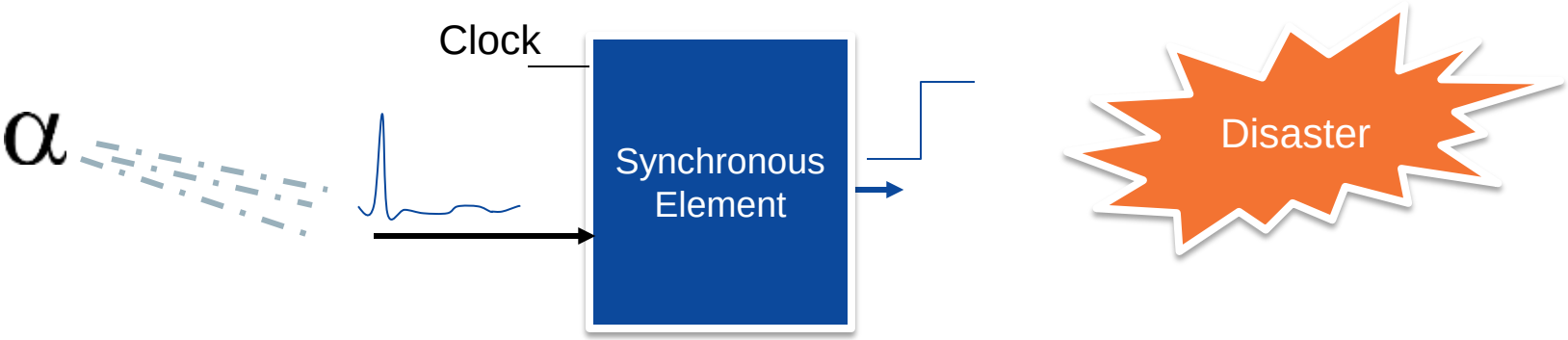
Power supply glitch or brown-out



Meta-stability due to design flaws (failure to synchronize the circuit)

SETs and SEUs

- Glitches clocked into a synchronous element can cause operation errors and ultimately system failure
  - Impacts memories, registers, state machines



SET (Single Event Transient)

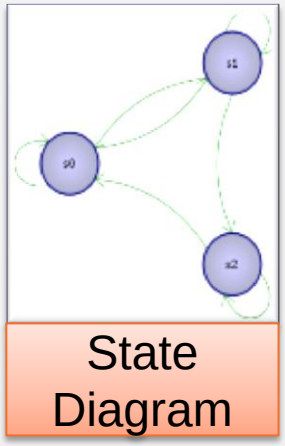
A current spike or "glitch" in a signal that occurs due to ionization or electromagnetic radiation

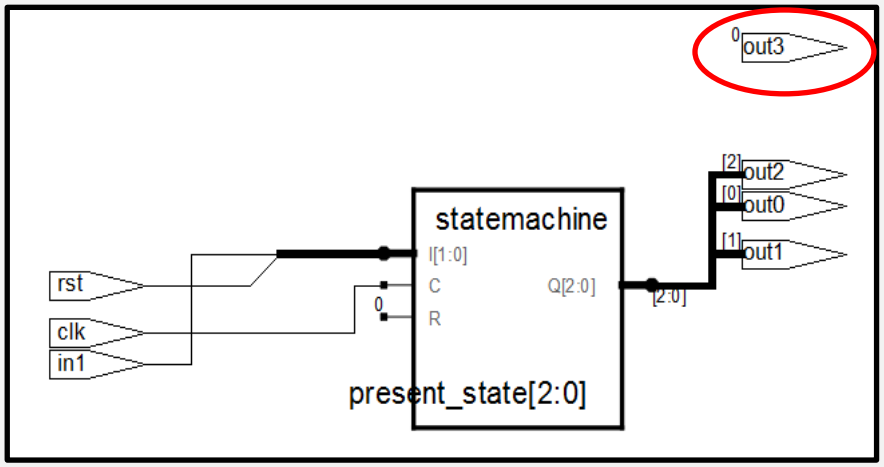
SEU (Single Event Upset)

Incorrect signal (SET) captured by a synchronous element that impacts its internal state and / or output

SEUs Cause Incorrect Behavior or Failure of FSMs

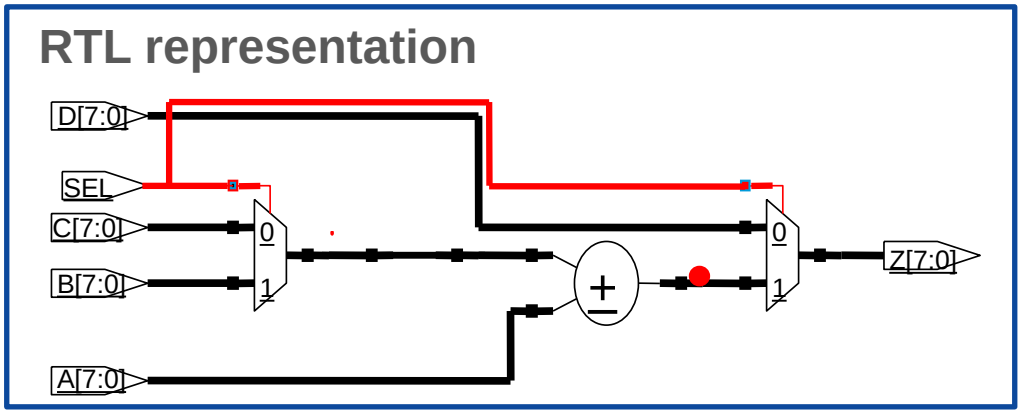
- A flip in a register bit within an FSM can cause what was assumed to be an "unreachable" or "invalid" state to be reached
  - FSM becomes STUCK in the invalid state





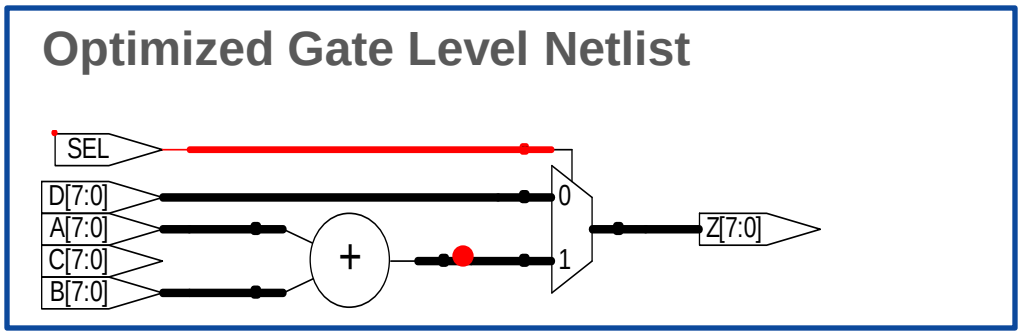
RTL *others* clause that would handle state transition behavior out of an unreachable state is optimized away by default by the synthesis tool

Synthesis Optimizations Make Requirements Traceability, Debug a Challenge



Synthesis

- *RTL Enumerated TYPES appear as 1's and 0's in netlist*
- *RAM, DSP inferencing and sequential optimization optimize node in RED away*

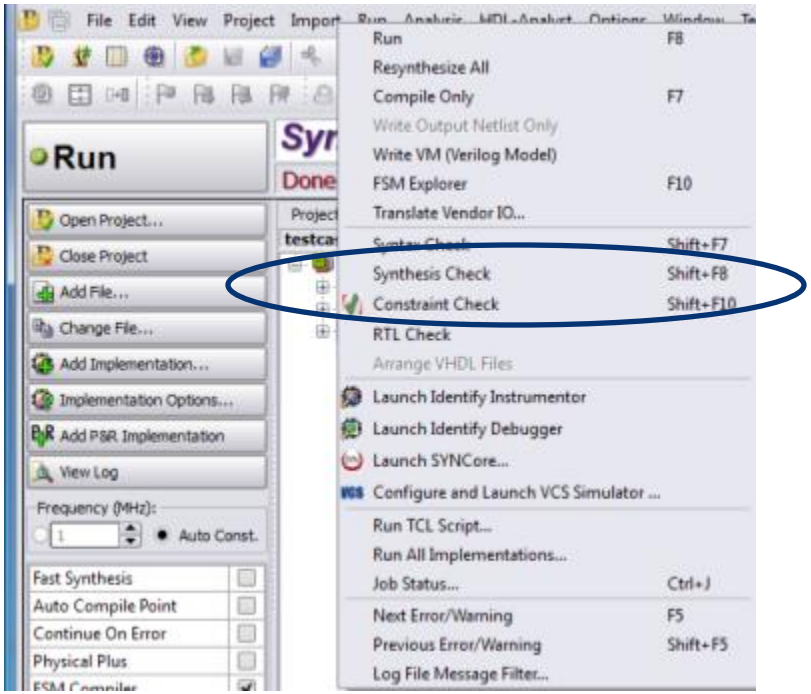


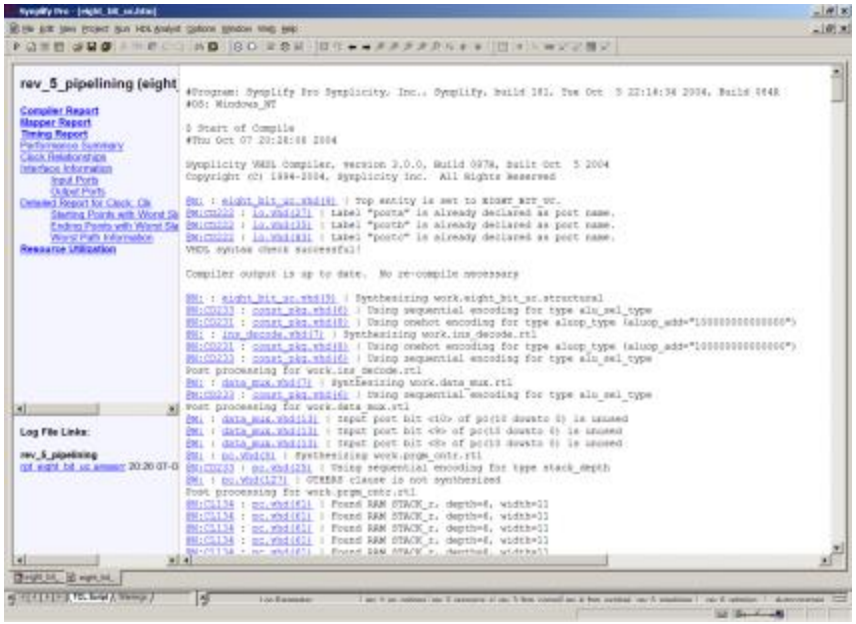
Some RTL Nodes in the Circuit get Optimized Away

Synthesis performs optimizations that can result in a debug node of interest being completely absorbed or transformed, so you can no longer trace or probe these nodes or relate their behavior on the board back to your RTL

High Reliability Design Solutions

Complete Design Specification Constraint Checking



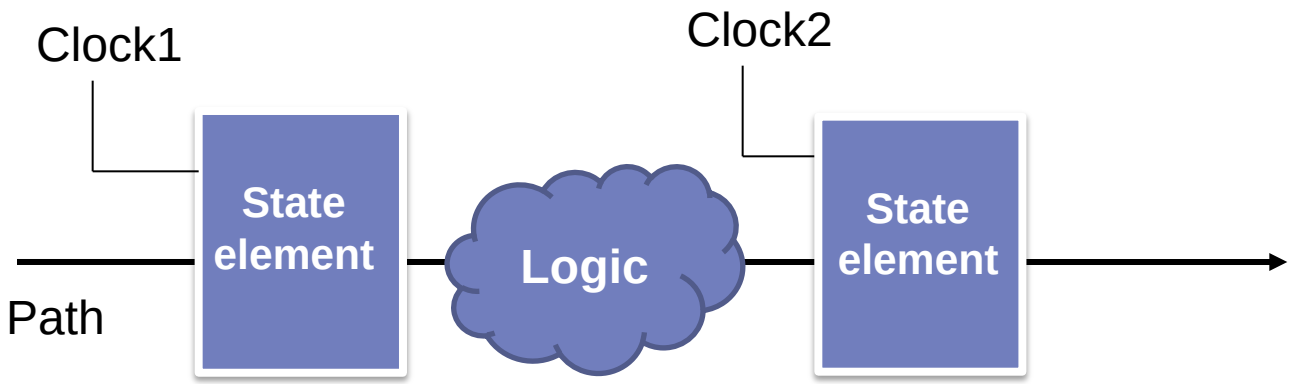



```
synplify_premier -batch <design>.prj -run constraint_check
```

- Checks constraints prior to RTL synthesis
- Checks constraints syntax and for timing constraints applied to non-existent or invalid types of arguments/objects

Complete Design Specification
Clock Domain Synchronization Assurance

Find combinatorial paths that are crossing clock domains without synchronization



Clock1 and Clock2 are in <u>different</u> clock domains and are assumed synchronous by default unless you use the –asynchronous option

```
set_clock_groups -asynchronous -group {A0 A1} -group {B0 B1}
```

- Synplify generates report for all paths that:
  - Start at state element in one clock domain (clock group)
  - End at state element in different clock domain (different clock group)
  - Reports longest path between these 2 points
- User can then synchronize the path if synchronization was intended

Design Practices that Build-in Safety
SEU Mitigation Practices for Antifuse/Flash Devices

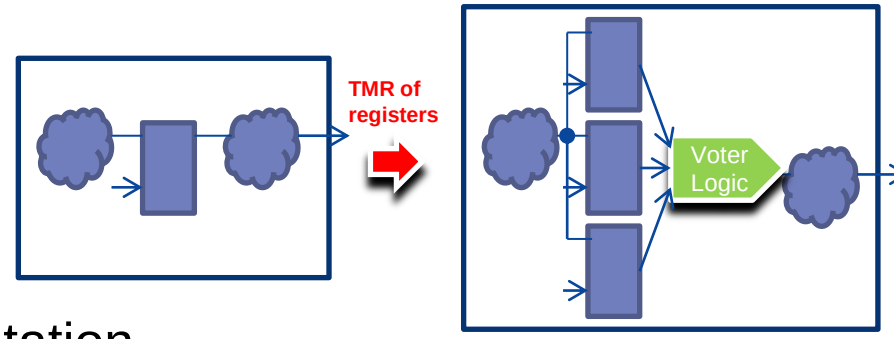
- Antifuse/flash device registers can be moderately susceptible to SEUs
- FSMs contain registers
- The following solutions will help

Resource	Recommended solution
Registers	Local TMR
FSMs	Hamming-3 Automatic Error Correction Specify how to return FSM from “unreachable state” to safe state by specifying and preserving the RTL “others” clause

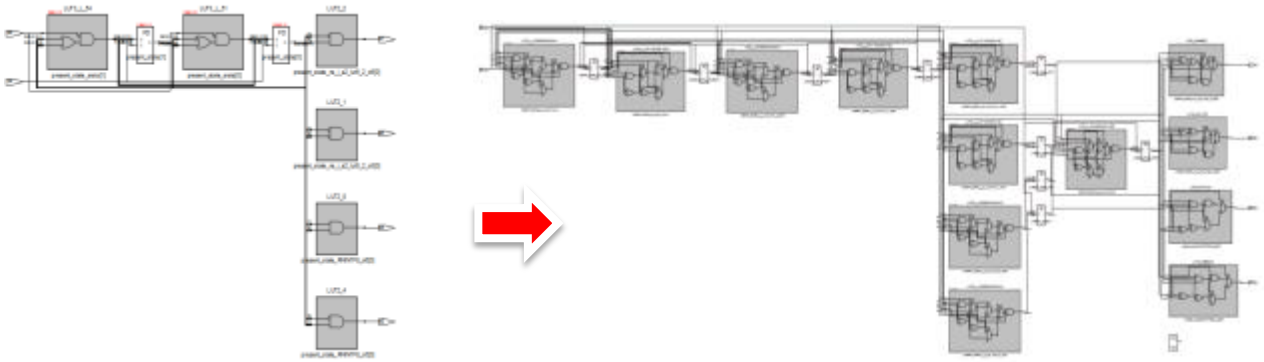
Be sure to instruct the synthesis software to preserve error-mitigation circuitry

Automated Techniques to Design Highly Reliable Circuitry for Antifuse/Flash

- Local TMR with voting logic
  - SEU-immunity for critical registers (e.g. control logic) by creating redundant circuitry

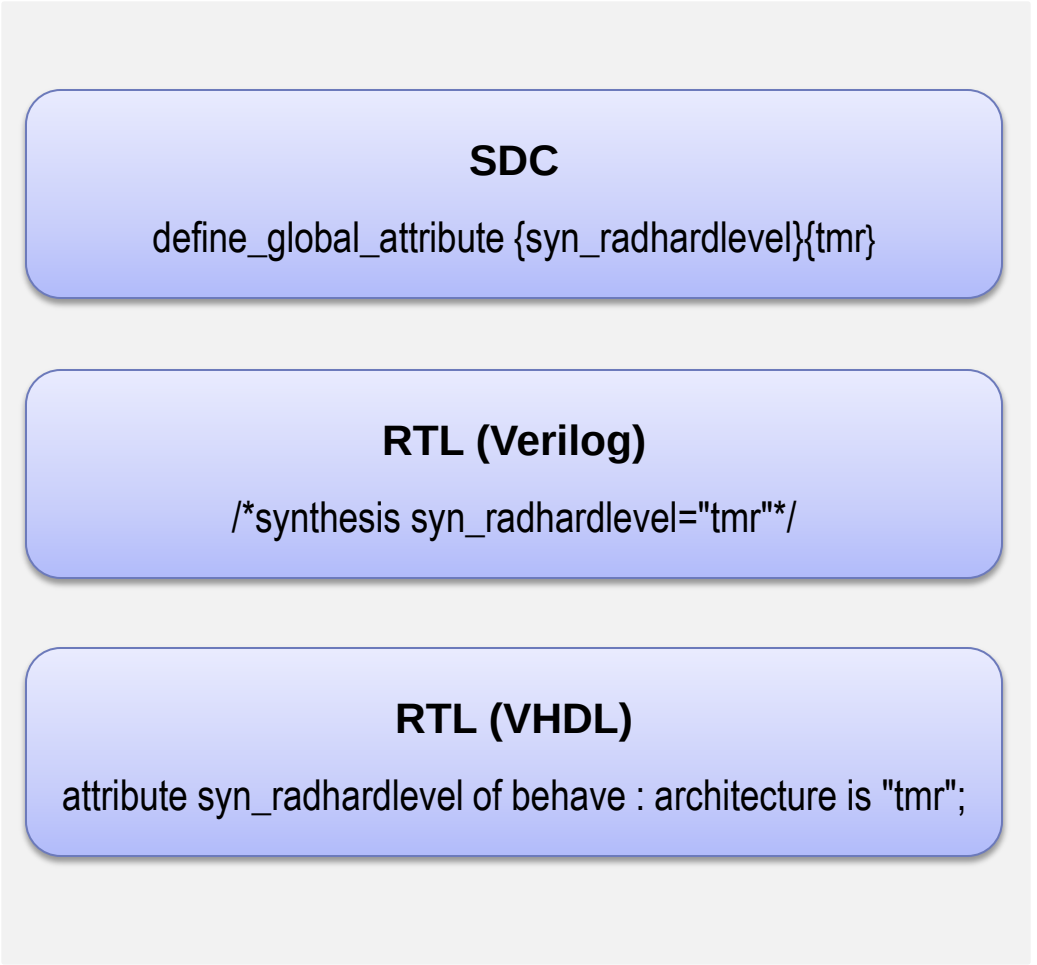


- Safe and fault-tolerant FSM implementation
  - RTL “others” clause for custom error mitigation (return FSM to safe state)
  - Hamming-3 encoding for automatic single bit error correction



Local TMR Logic Creation by Synplify Pro

- Use syn_radhardlevel attribute for Microsemi FPGA devices
  - Antifuse: RT, RH and RD radhard devices
  - Flash: ProASIC3/ProASIC3E and ProASIC3L
- Microsemi Axcelerator RT devices have built-in TMR in silicon
  - syn_radhardlevel attribute ignored by Synplify

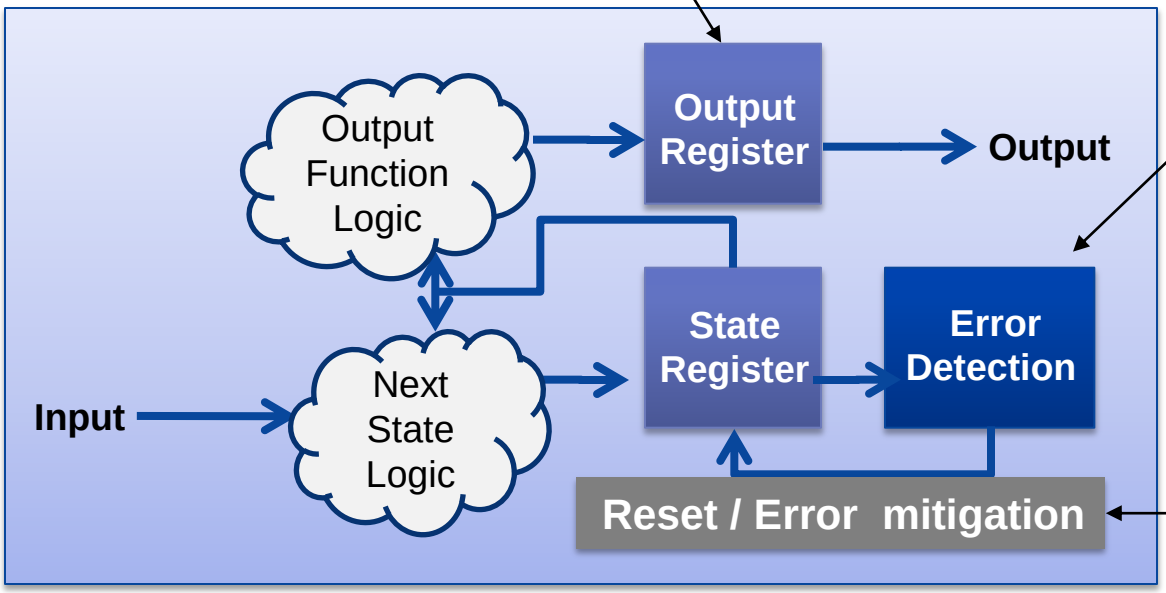


Creating Safe Finite State Machines

State Transition Behavior
(**Next State** and **Output Function** Logic)
syn_state_machine.
syn_encoding or hand-coded RTL

Using “One Hot” State Encoding for automatic detection of invalid FSM States

Error Detection & Mitigation Logic
Hamming-3
syn_radhardlevel / TMR redundancy.
RTL code (reset behavior)



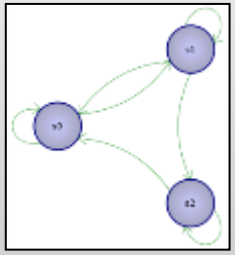
Preserve Debug Logic or Nodes
syn_keep, syn_preserve,
syn_hier, syn_noprune
syn_probe

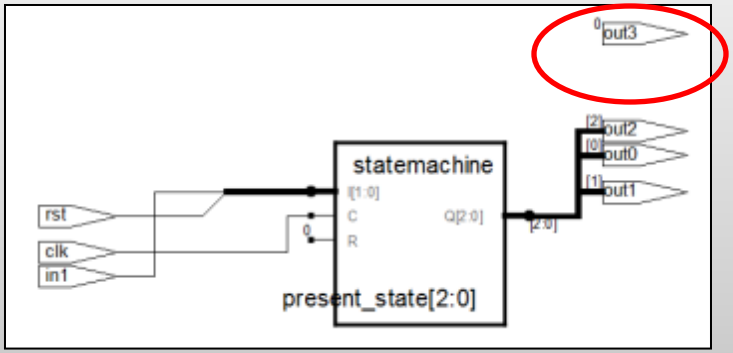
Automatic FSM Error Detection and Correction in Synplify Premier

	Hamming-3 Error Detection / Correction in FSM	Safe FSM
	Automatic error detection and correction of 1-bit state error	Automatic error detection and reset
What Happens Upon Error	Error corrected automatically and FSM functions as normal	FSM goes to a default or reset state as specified in the user's RTL in the "others" clause

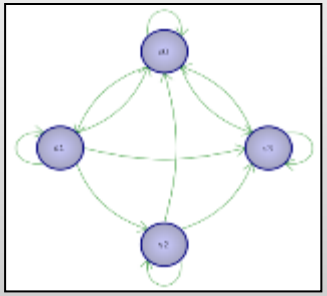
Implementing SEU Detection/Mitigation Circuitry using the “Others Clause”

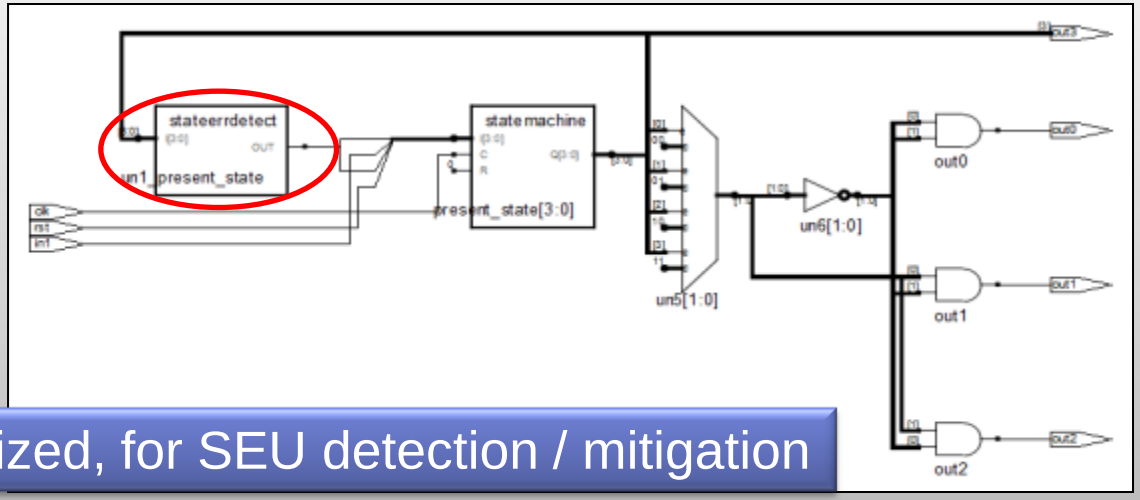
- Use in a Safe FSM (or in any sequential logic)





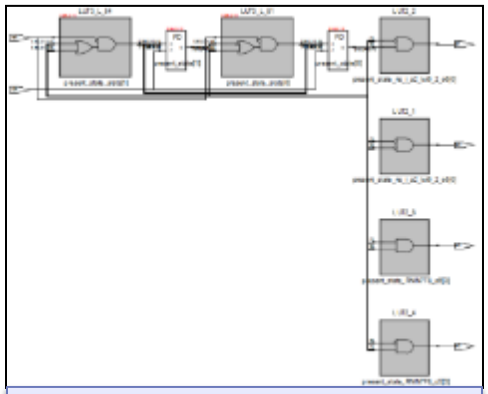
others clause optimized away by default



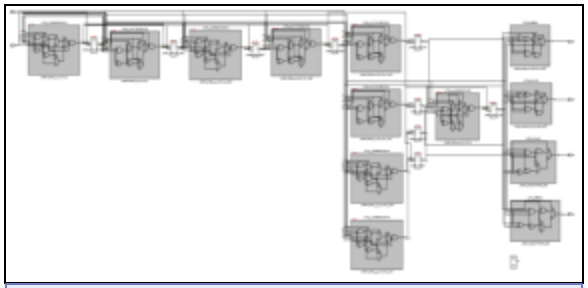


others clause synthesized, for SEU detection / mitigation

Hamming-3 Error Detection/Correction for FSMs

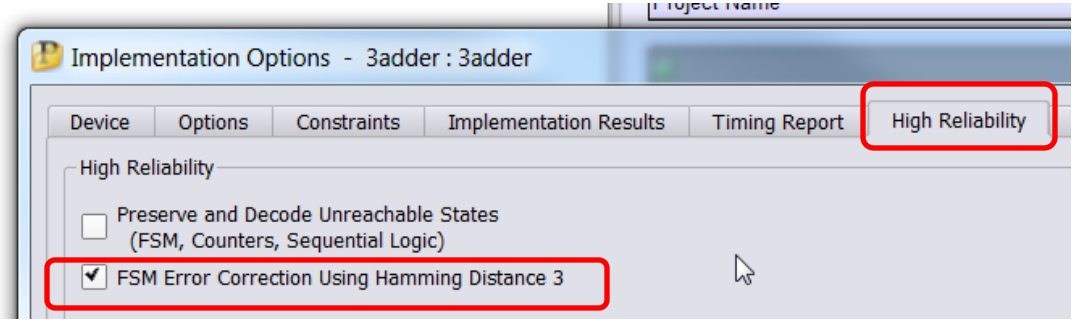


Without Error Correction



With Hamming-3 error correction
- Comparator circuits and parity flip-flops added

Perform Hamming-3 error correction automatically so that FSM continues to operate



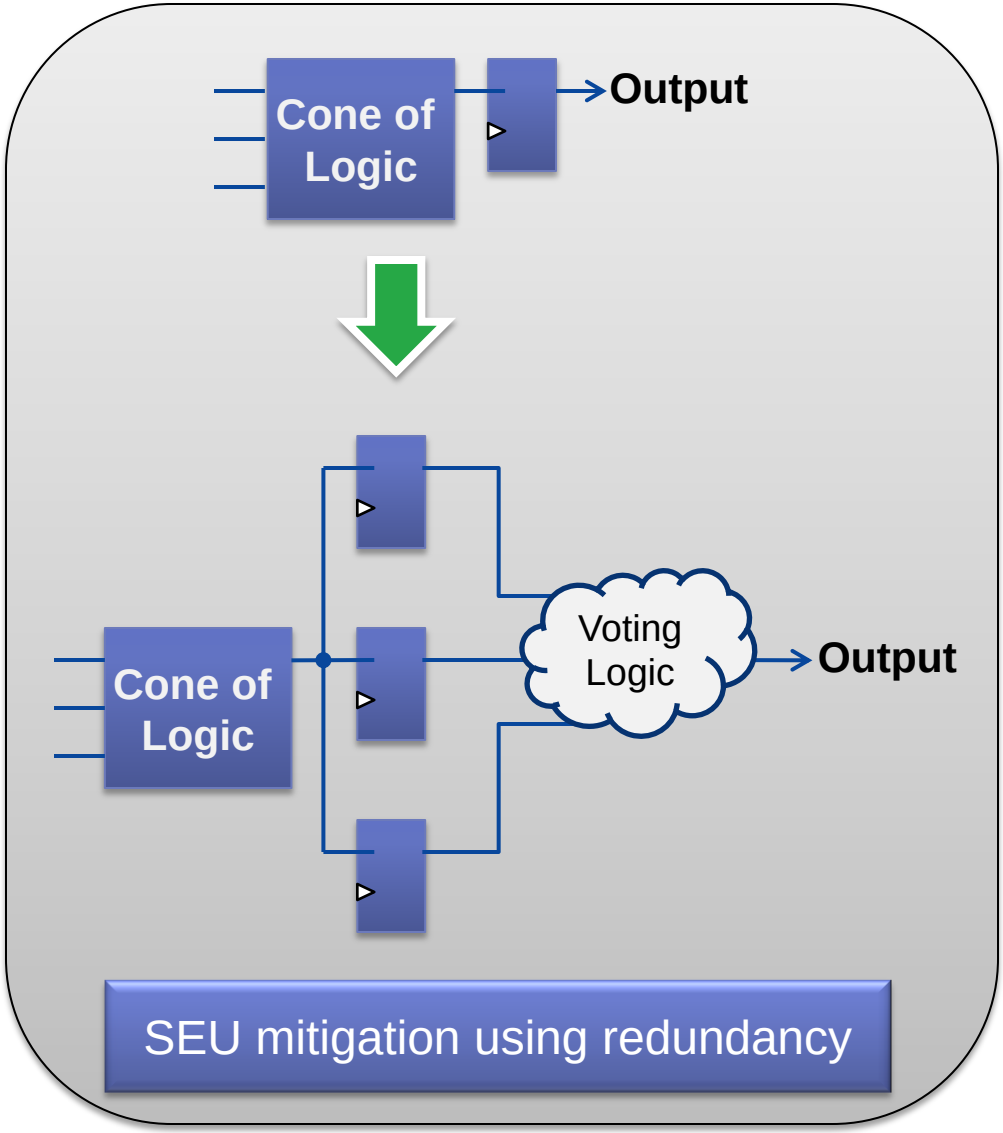
With Hamming-3 error correction, comparator circuits and parity flip-flops are automatically added during synthesis

Project File

```
set_option -hamming3  1
```

Triple Modular Redundancy (TMR) with Voting Logic

- TMR helps mitigate SEUs induced by radiation
- Insert redundancy during synthesis with triplicated circuitry + majority voting logic
- Configure type of TMR to be used (register, block…)
- And/or create custom TMR architectures and invoke settings that ensure redundant circuitry is not optimized away



Preserve Debug Logic During Synthesis Attributes and Settings

- During synthesis, preserve nodes in the design for probing/debug or equivalence validation

syn_keep	1/0	Preserve a net
syn_probe	1/0	Preserve a net for probing
syn_preserve	1/0	Preserve a cell / sequential component
syn_hier	1/0	Preserve a block
syn_noprune	1/0	Preserve an instantiated component

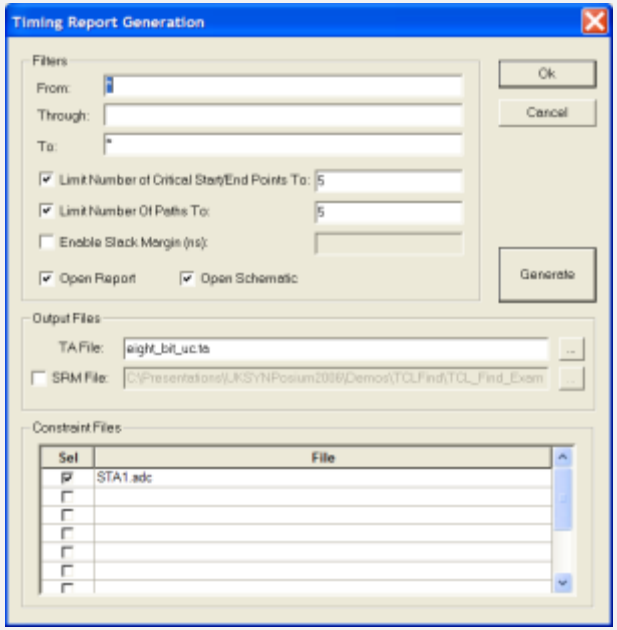
Note: P&R will not optimize away redundant logic - P&R does not alter the input netlist – it only removes floating or unconnected gates

Ensure better matching of RTL: netlist name correspondence (with some loss in QoR)

```
set_option -no_sequential_opt 1
```

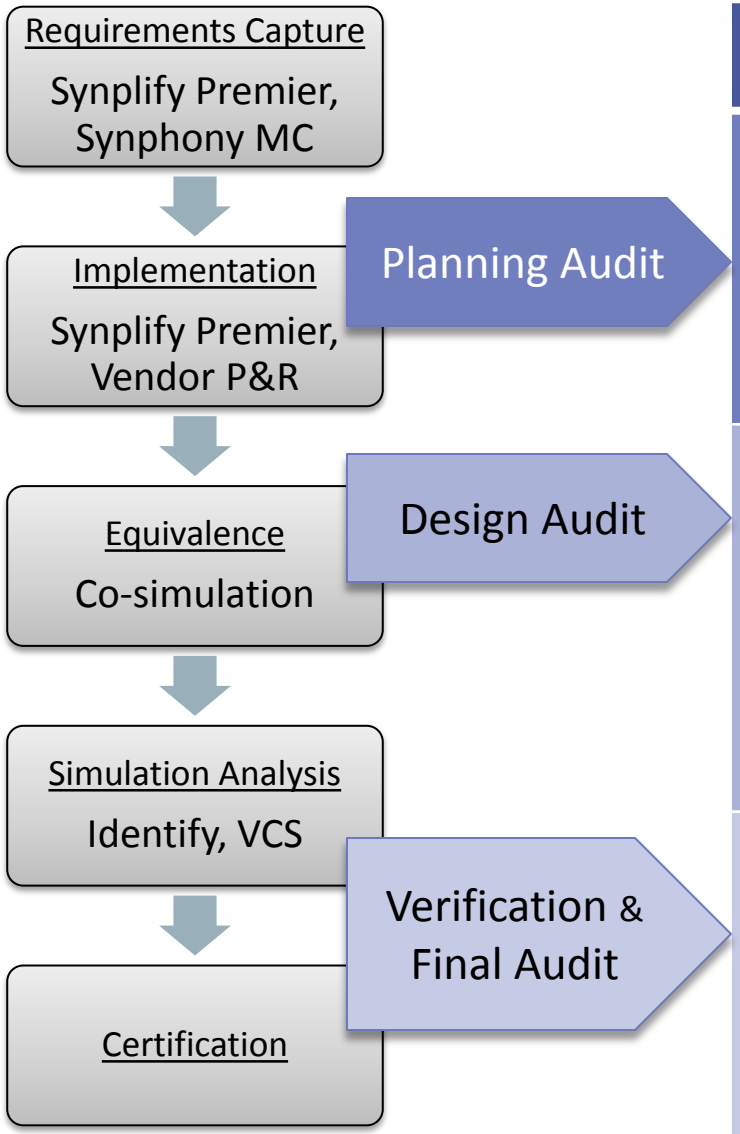
Reproducible and Documented Design Process

- Complete design reporting
  - Custom and Tcl / Find based reports
- Complete project archiving
- Synplify synthesis results are reproducible for given FPGA/speed grade
  - Same software version, machine and OS
- Synplify includes “path group” technology
  - Reduces difference in results with minor changes to RTL and constraints
- Incremental and design preservation flows are available
  - Reduce changes when only a small part of the design specification changes



High Reliability Process Considerations

DO-254 Process



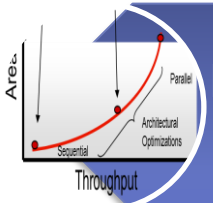
Traceability	Document (and archive)	Equivalence (repeatable)
• Synphony system level specs and schematics recorded • Synthesis RTL schematics generated and recorded	• Premier and Synphony project specifications and settings archived	• Cosimulation demonstrates Simulink® to RTL equivalence • Repeatable synthesis results and design re-use
• Premier design checks run (constraints, clock domain crossing) and logged in Premier project log files • Post-synthesis and P&R reports generated & logged	• Archive entire project including synthesis and P&R inputs and outputs, scripts, schematics, project settings, software config , timing reports, error messages	• VCS demonstrates RTL to netlist functional equivalence
• Verification Reports checked	• VCS based testbenches and verification reports archived	• Identify debug run to demonstrate board operational equivalence and correct behavior relative to the RTL • VCS coverage analysis

Reliable Verification Flows for Signal Processing

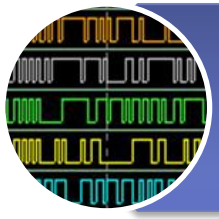
Synphony Model Compiler Overview
Productivity for Signal Processing Design & Verification

Save months in design & verification of signal processing hardware & systems

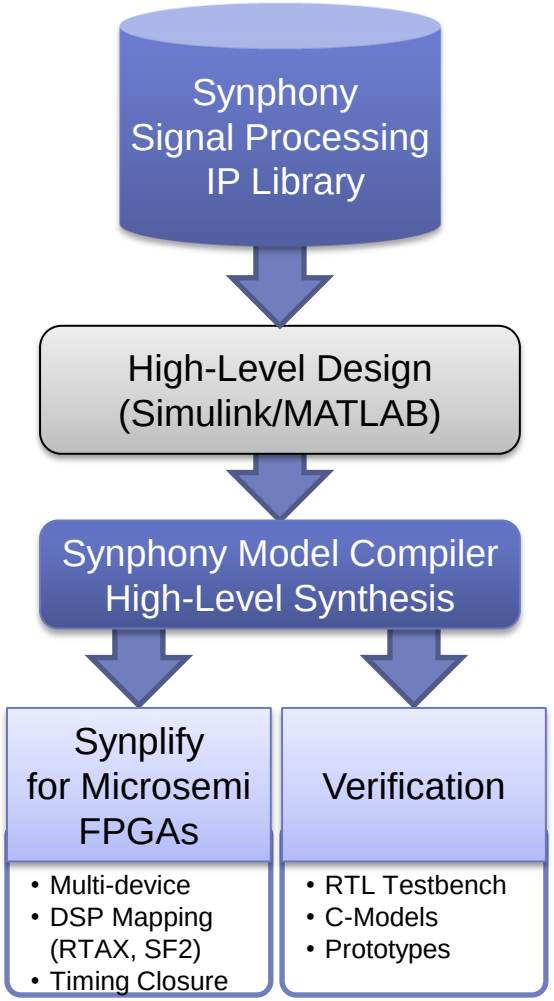
Signal Processing IP library



High-Level Design and Verification from Simulink/MATLAB



High-performance verification for RTL and C-model system simulation



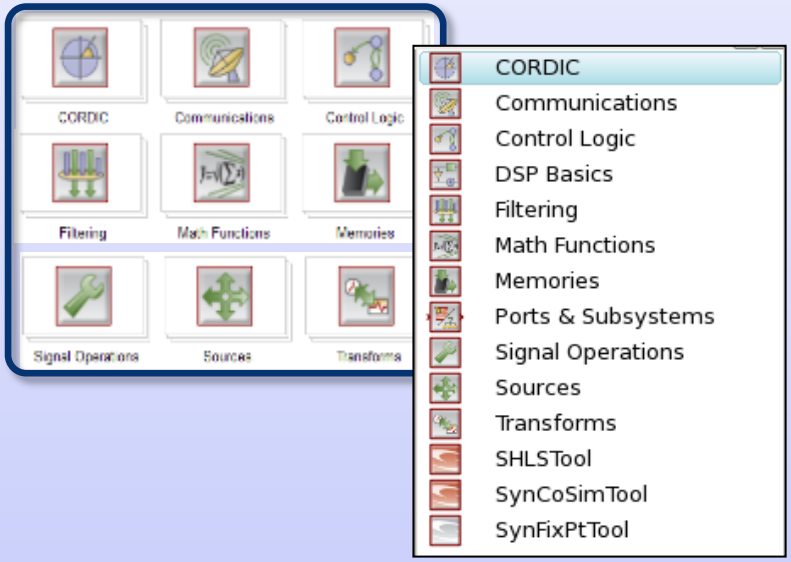
SMC Design & Verification Productivity Using MATLAB/Simulink

- Create & verify more reliably using Simulink/MATLAB environment
- High-level IP functions for wireless & communications applications
- Fixed-point tools for easy control of algorithm precision
- Vector math for fast and concise parallelism
- Full multirate support simplifies multiple clock designs
- Language blocks: use RTL & M-Control for easier design of control & interfaces

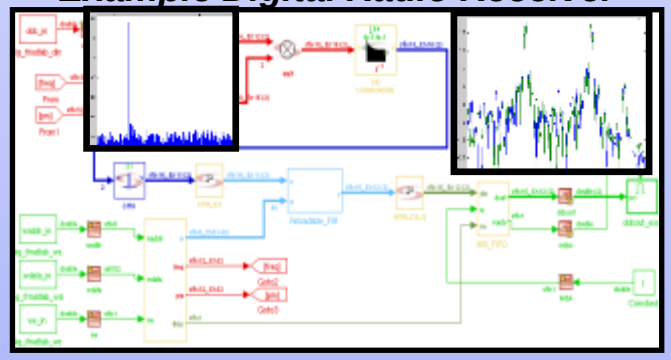
2-3X Productivity

Specification-to-Verified Fixed-Point Model

SMC Library (Blockset) for MATLAB/Simulink



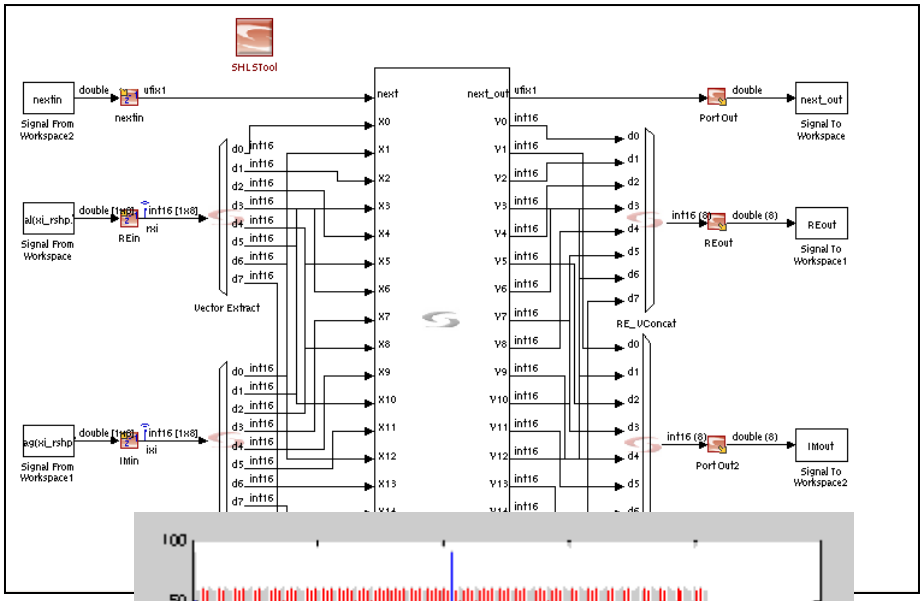
Example Digital Radio Receiver

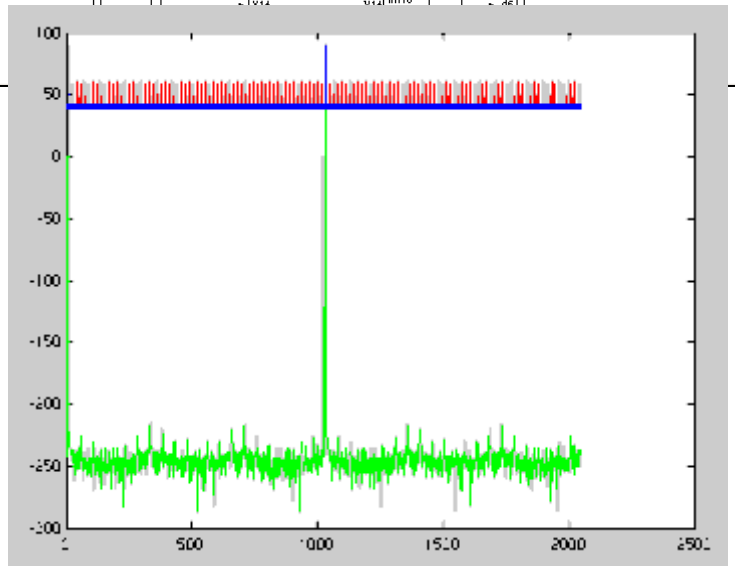


Example Verification Productivity
MATLAB-Driven Verification of RTL Blocks

Fixed-point verification of RTL using RTL Encapsulation Block (RTLE)

- Simple 3 step process
  - Instantiate RTLE block
  - Use To/From Workspace blocks to read & write block from MATLAB variables
  - Write MATLAB scripts to create stimulus & compare results to reference

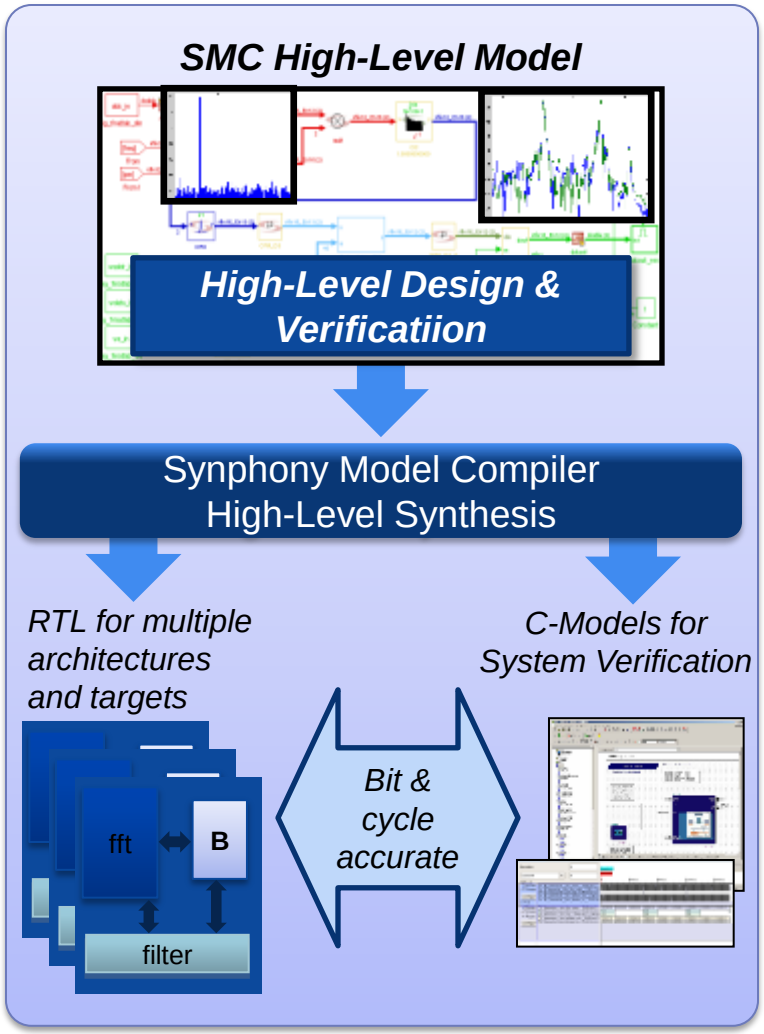




Example results of parallel FFT core (8-stride, 1024-point)
Green: spectrum of floating point data (double)
Blue: spectrum of quantized input data (sint16) (white gaussian noise)
Red: spectrum of parallel fft (red spurs above the noise floor)

C-Model Generation for Simulation Speed

- Easily create C-models of your hardware to
  - Increase simulation speeds
  - Validate system integration in external simulators
  - Verify SW/HW quickly
- Automated, flexible simulator support
  - Re-use in Simulink
  - RTL simulators (ModelSim, VCS)
  - Native ANSI-C direct executable
  - System C simulators
- Benefits
  - Earlier system validation by months
  - Higher reliability and coverage
  - 10X verification productivity



Optimized Signal Processing IP
Portable Signal Processing Functions for FPGA and ASIC

- Easy to use with high capacity and advanced features
- Achieve excellent DSP hardware mapping on advanced FPGAs
- Target multiple FPGA vendors and ASIC

Example High-Value Synphony IP Cores			
Function (Blocks)	**Key Features/ Modes**	**Architecture Optimizations**	**Target Optimizations**
FFT, FFT2	Multichannel, Parallel / Serial Flow Control Dynamic Length	Speed (HLS retiming) Area (HLS Folding) Micro-architecture (IP-specific)	Xilinx Altera ASIC
FIR, FIR2	Multi-Channel, Parallel / Serial Programmable Coef. Polyphase multirate Flow Control Symmetry-optimized		
DDS/NCO	Multichannel, Parallel / Serial SFDR mode Dithering Flow Control		
Viterbi Decoder / Convolutional Encoder	1/K Rate, Mother Codes, Traceback Length, …		

SMC FIR Optimized for SmartFusion2
Mapping to DSP Blocks for Very High Performance

- Optimized micro-architecture for SMC-FIR
- Excellent results that can exceed hand-coding and matching CoreFIR
- But includes benefits of MATLAB/Simulink verification flow!

Filter Parameters			Results				
Taps	Bit width		MACC	Resource Usage			Max Clock, MHz
	Coef	Data		Comb	Seq	Total	
24 (coreFIR)	18	18	24	51	1020	1071	423
24 (SMC)	18	18	24	63	926	989	423.19
72 (coreFIR)	18	18	72	155	3275	3430	371
72 (SMC)	18	18	72	63	3150	3213	383.73
144(symm) (SMC)	18	18	72	1296	10670	11966	304.14

1. All results using high effort layout in Libero SoC, target freq. 500 MHz.
2. For Symmetric FIR, critical path is the pre-adder

Summary

Synopsys FPGA solutions for highly reliable designs

- Safety-critical design techniques
- Complete and accurate design specification
- Debug, verification and analysis techniques
- Features for design traceability, documentation and to ensure equivalence for DO-254 compliance
- Flows for design proof-of-concept and prototyping to validate designs and win contracts
- High-level verification using MATLAB/Simulink and C-Model generation

For more info

- Solutions Overview
  - Webinars, Datasheets and Solutions Articles
  - http://www.synopsys.com/Tools/Implementation/FPGAImplementation/FPGASynthesis/Pages/HighReliability.aspx

- Synopsys Mil-Aero Technical Bulletin
  - A free quarterly Synopsys publication for mil-aero engineers, covering the latest methods for addressing the design and verification challenges. http://www.synopsys.com/MTB

Microsemi

Thank You