SPACE CODESIGN

A System Design and Rapid Prototyping Methodology Applied to a Fault-Tolerant Vision-Based Navigation Algorithm Targeting MPSoC-FPGA

Guy Bois, Fabien Portas, Anthony Dentinger, Hubert Guérard

Space Codesign Systems Inc.
4388 Rue Saint-Denis Suite 200 #705
Montreal H2J 2L1 Canada

Agenda

- Space Codesign
- SpaceStudio
- Siemens Precision Hi-Rel
- Proposed Workflow
- Use case
- Conclusion

Agenda

- Space Codesign
- SpaceStudio
- Siemens Precision Hi-Rel
- Proposed Workflow
- Use case
- Conclusion

Space Codesign, 2010-2025

- Software publisher specialized in development environment for easing and speeding up time-to-market cycle of embedded systems (SpaceStudio)
- Long experience in FPGA
- Business model relies on time-based licenses of SpaceStudio and highly specialized services on top of SpaceStudio

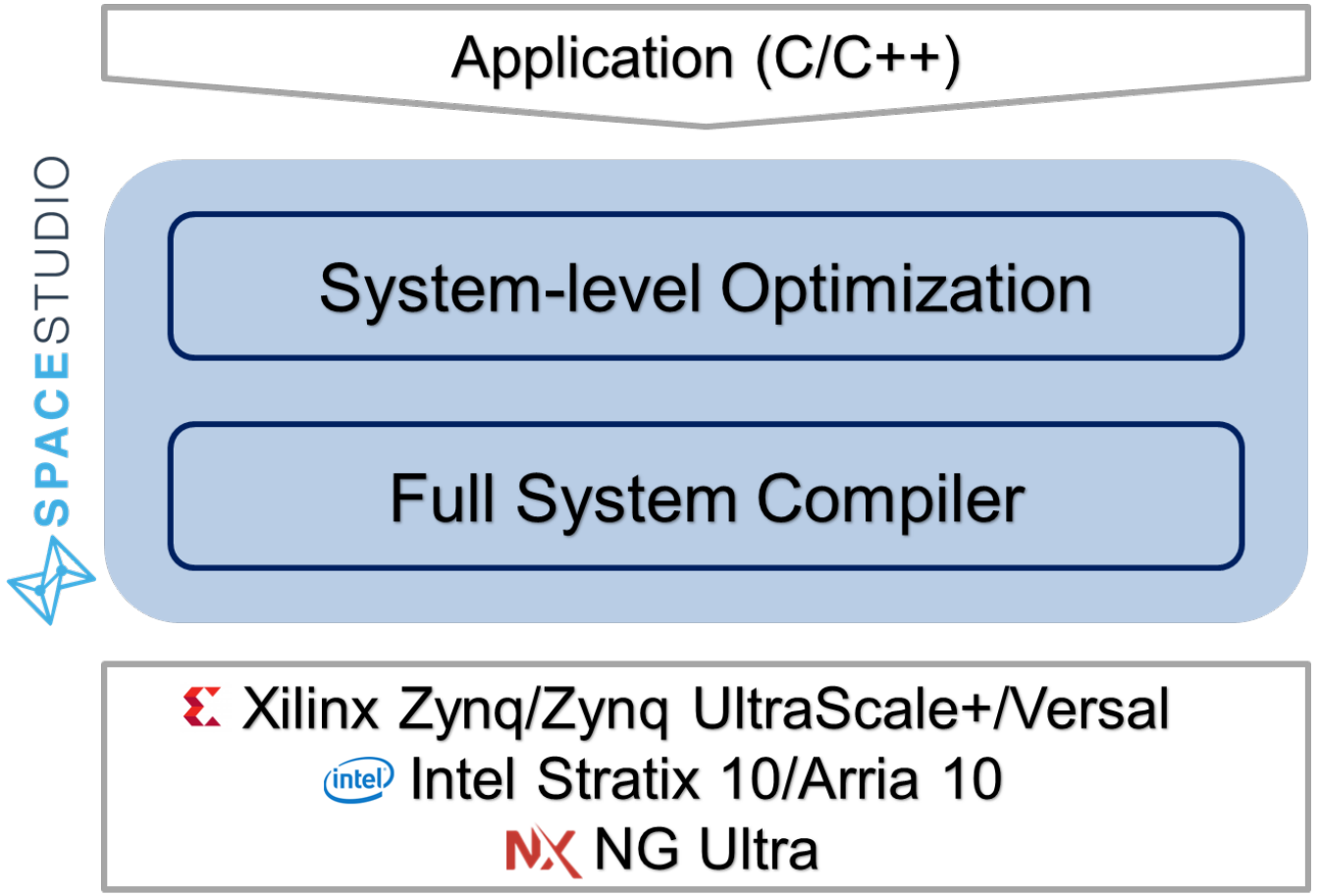
Agenda

- Space Codesign
- SpaceStudio
- Siemens Precision Hi-Rel
- Proposed Workflow
- Use case
- Conclusion

What is SpaceStudio?

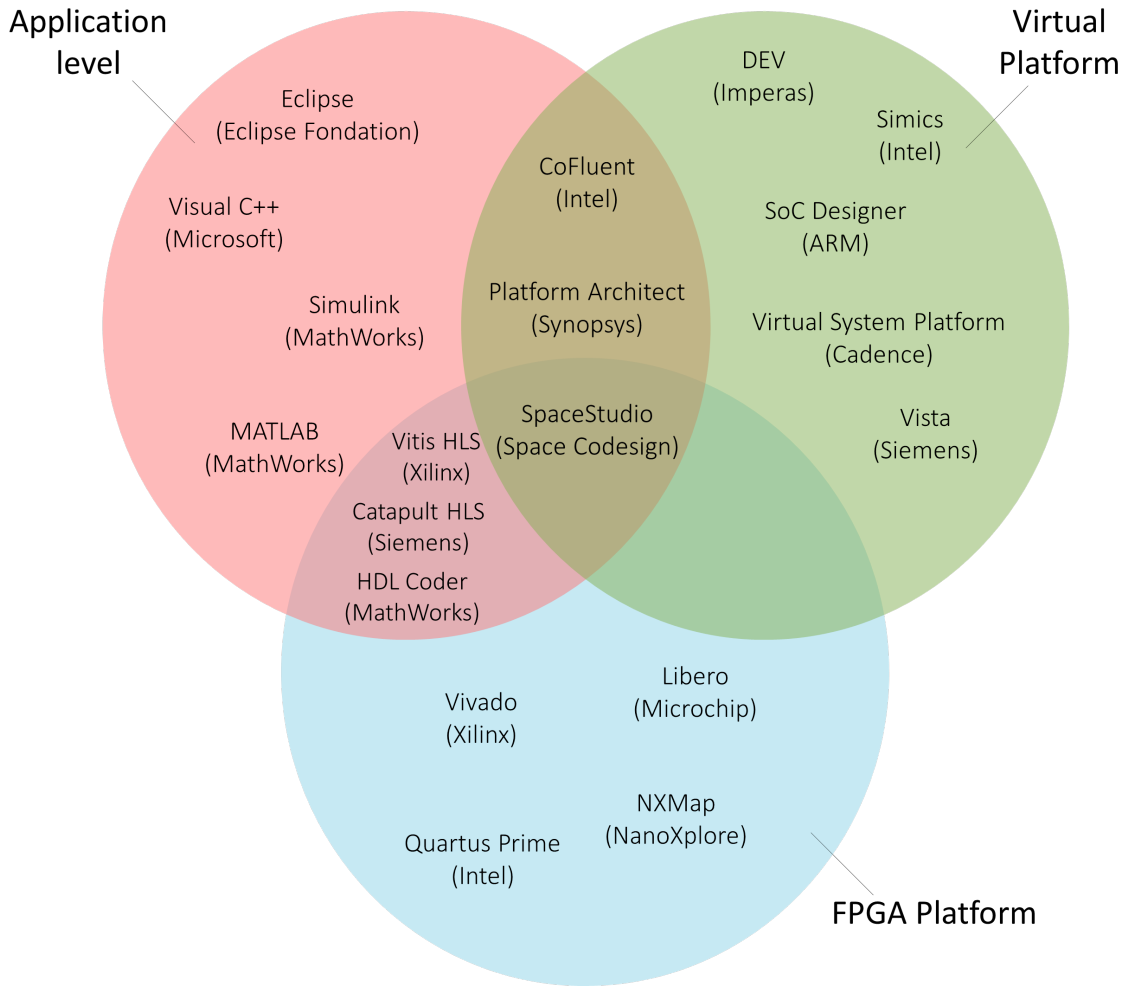
- Software development environment
  - Consolidate HW/SW team in a joint workflow
  - System centric
- Design Space Exploration (DSE)
  - Provides system performance prediction
- Rapid prototyping on physical board
  - Generate complete RTL for downstream tool (i.e., Xilinx, Intel)
  - Leverage on HLS tools (i.e., Vivado HLS, Catapult HLS)

Overview of SpaceStudio



Commercial ecosystem for system design

SpaceStudio position in the ecosystem



SpaceStudio

- **Application first!**
- Application drives the complete flow
- GUI is based on a design block interface
- Command line interface based on Python
- Ships with ready-made platform IPs (memory, interconnect, processor, etc.)
- Designer creates and focuses on application IPs
- Runs on Ubuntu and Windows

Agenda

- Space Codesign
- SpaceStudio
- Siemens Precision Hi-Rel
- Proposed Workflow
- Use case
- Conclusion

Siemens Precision Hi-Rel

- Based on Siemens Precision RTL Synthesis tool
  - High quality and verifiable RTL synthesis
  - Wide array of supported FPGA vendors (AMD, NanoXplore, Intel, …)
- Automatic insertion of fault-tolerance in synthesized designs
  - Triple Modular Redundancy with voters
  - Multiple TMR strategies to achieve resource/safety tradeoff
  - SEU-tolerant or SEU-Detection FSMs
- Fault tolerance without hardware support
  - Enables hardening of existing FPGA hardware at no extra cost

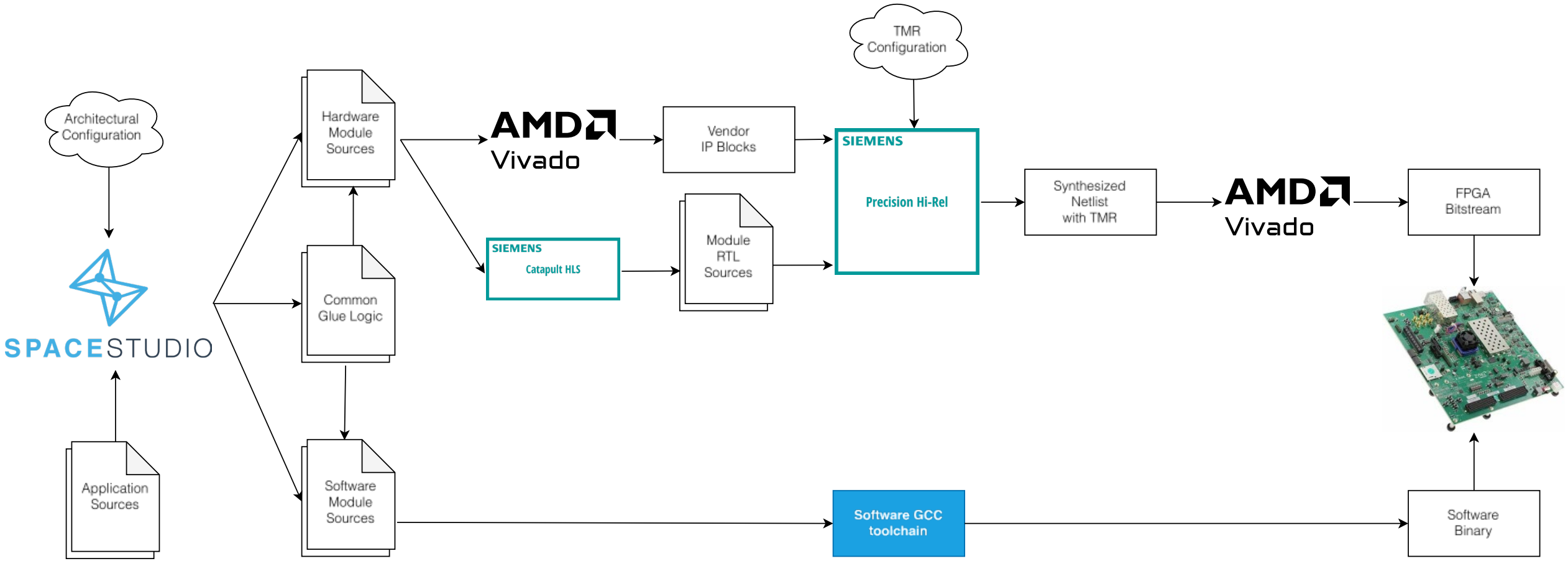
SIEMENS

Precision Hi-Rel

Agenda

- Space Codesign
- SpaceStudio
- Siemens Precision Hi-Rel
- Proposed Workflow
- Use case
- Conclusion

Proposed Workflow (main objective)



Proposed Workflow (cont.)

Automated generation, refinable results

- Vivado and Precision projects automatically generated by SpaceStudio
  - Project can be refined by the user afterwards
- SpaceStudio quickly generates a working prototype automatically, which can be explored further manually:
  - Hardware communication glue logic (e.g., interconnect, DMA, timers, etc.)
  - Software Linux kernel drivers
  - Complete downstream tools automation

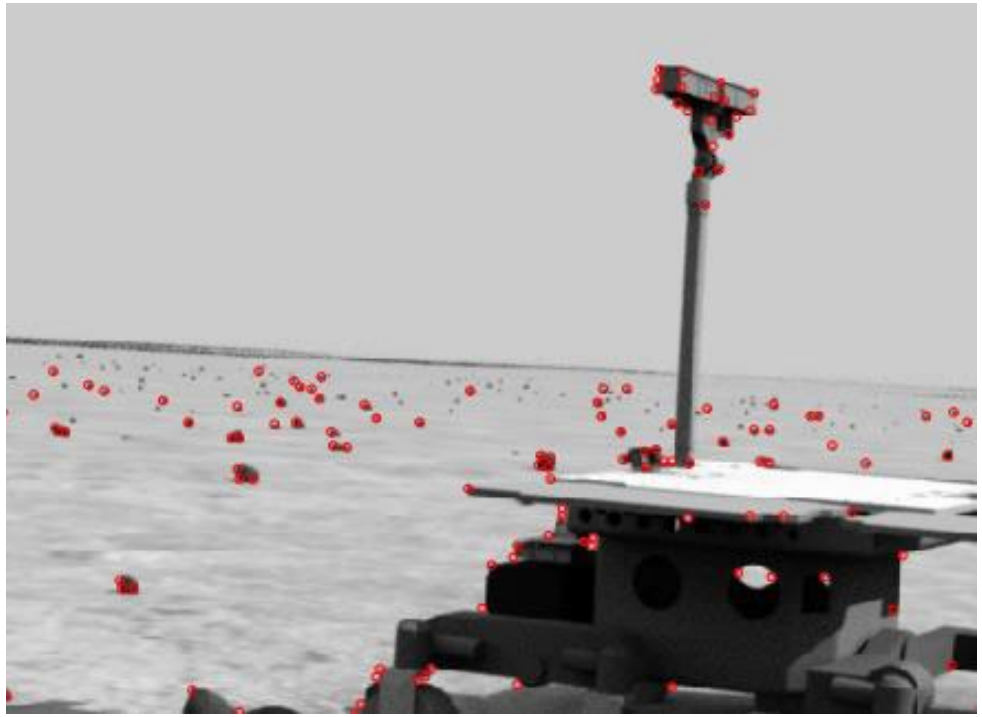
Agenda

- Space Codesign
- SpaceStudio
- Siemens Precision Hi-Rel
- Proposed Workflow
- Use case
- Conclusion

Corner Detection Algorithm

Introduction of the use case

- Algorithm used in Vision-Based Navigation (VBN) for:
  - Feature tracking
  - SLAM (Simultaneous Localization And Mapping)
- VBN used for exploration/landers, automatic docking/rendez-vous
  - Requires acceleration and fault tolerance during critical phases (landing, maneuver close to the target)
- Operations are on matrixes/vectors
  - Convolution, addition, maximum detection
  - Basic operation on matrixes elements



Corner Detection Algorithm (cont.)

Introduction of the use case

- Written in C++: perfect for SpaceStudio
- Detects key features in 1Kx1K images captured from spacecraft
  - Computationally intensive
  - Required to be responsive for position estimation
  - Space-borne : susceptible to interference
- Initially a fully software implementation
  - Split computationally intensive code to hardware module using HLS
  - Implemented on Zynq Ultrascale+ MPSoC
- Can benefit greatly from fault-tolerance, but complex to integrate manually
  - Our workflow allows us to go from code to SEU-tolerant implementation quickly

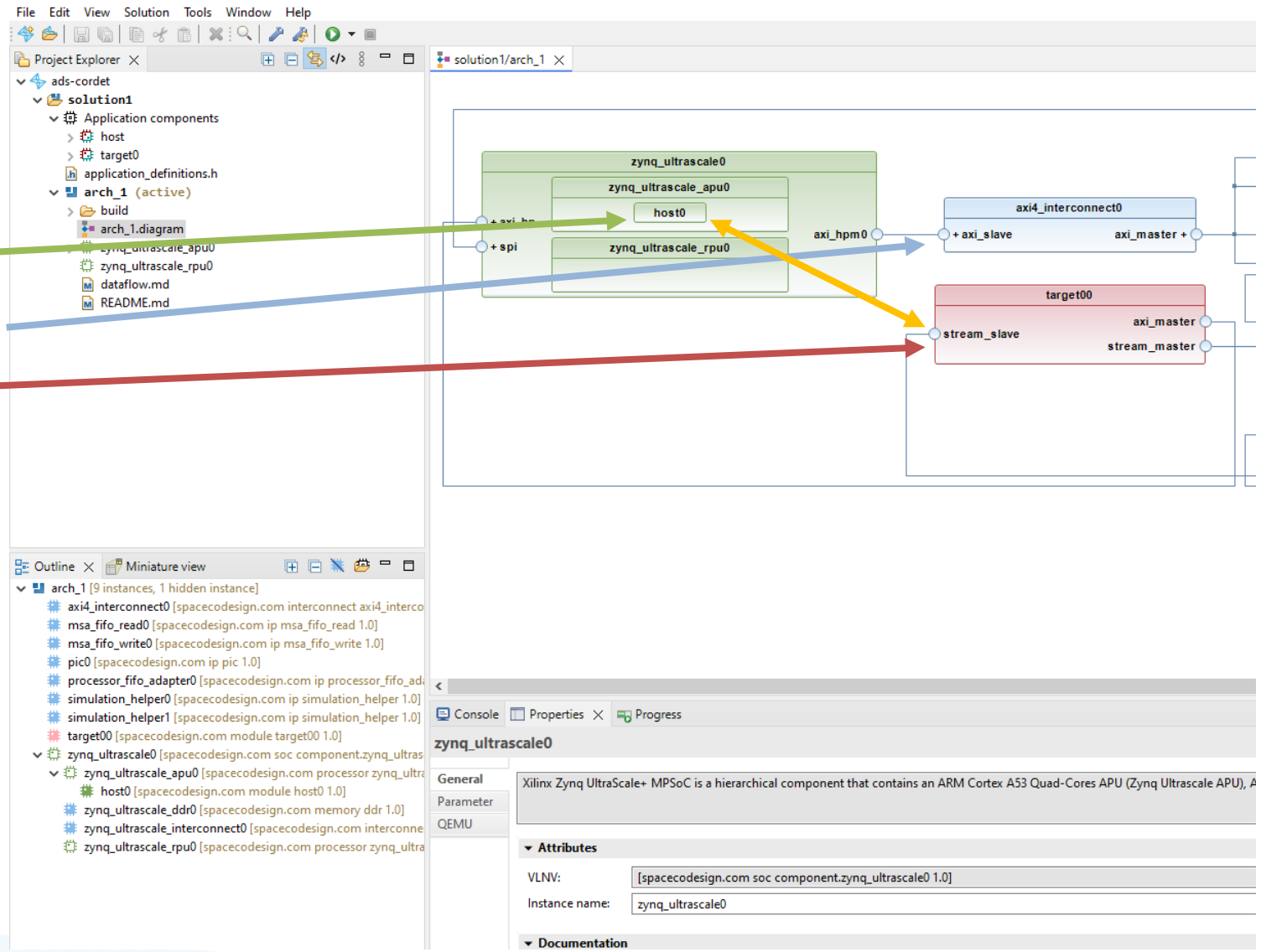
Test Environment

- AMD Zynq UltraScale+™ MPSoC ZCU102
- Vivado 2018.3 for placement and routing netlists
- FPGA clocked at 100 MHz
- Four setups compared
  - Baseline (plain Vivado)
  - Precision, without TMR
  - Precision, Global TMR (GTMR)
  - Precision, Intelligent Selective TMR (ISTMR)

Results

SpaceStudio

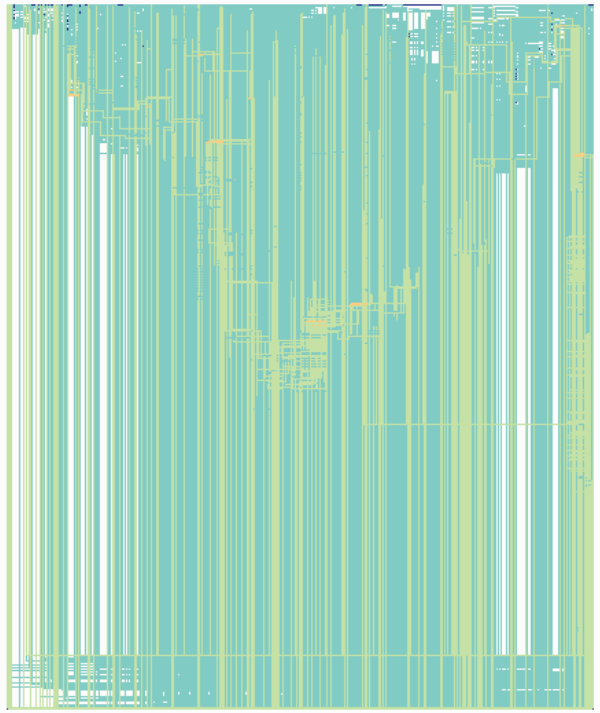
- Block design
  - Green: software task
  - Blue: automatically generated IP
  - Red: accelerated IP
- Architecture
  - Combination of HW/SW
  - Functionally validated in a simulated environment



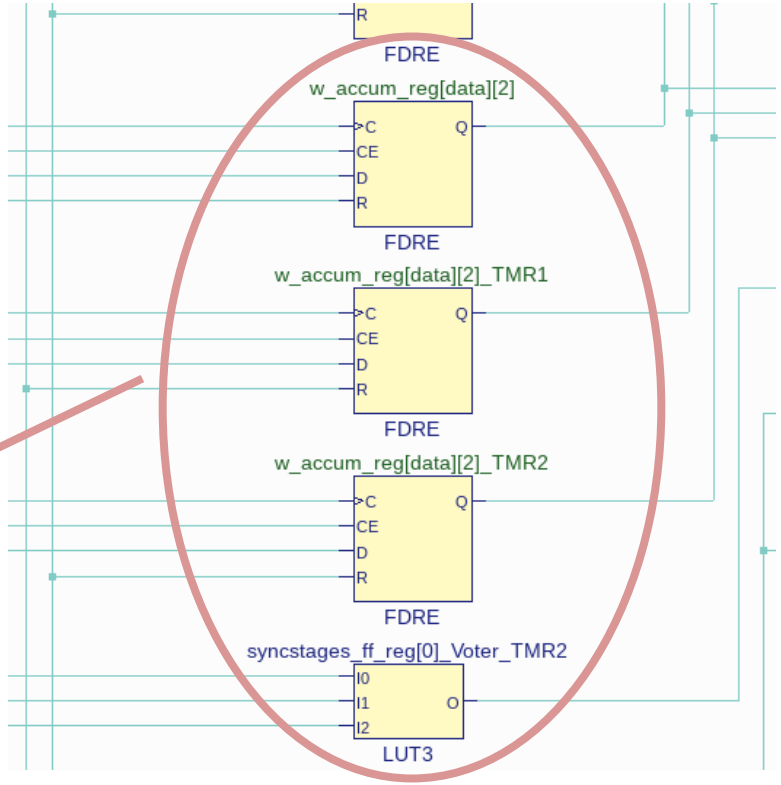
Results (cont.)

Precision Hi-Rel

- Design remains functional through Precision synthesis
- TMR has indeed been introduced by the tool



Some Triplicated Register Instance and a Voter Instance inside Precision Hi-Rel



Results (cont.)

Resource Usage

Method	LUTs		Flip-Flop		BRAM	
	# (%)	Variation	# (%)	Variations	# (%)	Variation
Baseline (plain Vivado)	2371 (0.87 %)		2459 (0.45 %)		1 (0.11 %)	
Precision No TMR	2228 (0.81 %)	-6.0 %	2470 (0.45 %)	+0.5 %	1 (0.11 %)	+0 %
Precision Global TMR	14650 (5.35 %)	+517.96 %	7322 (1.34 %)	+197.8 %	3 (0.33 %)	+200 %
Precision Intelligent TMR	13731 (5.01 %)	+479.1 %	7322 (1.34 %)	+197.8 %	3 (0.33 %)	+200 %

- As expected, resource usage is greatly increased
  - Intelligent TMR somewhat mitigates this, but resource usage still skyrockets
- Takeaway: TMR is most applicable to large FPGAs with extra available resources

Results (cont.)

Runtime Performance

- Very little to no impact on performance
  - Expected, as TMR is calculated in parallel

Method	Execution Time (ms)	
Baseline (plain Vivado)	479.077	
Precision No TMR	478.2958	-0.16 %
Precision Global TMR	478.9728	-0.02 %
Precision Intelligent TMR	478.8582	-0.05 %

Results (cont.)

Timing

- Coarse global TMR causes timing failures
- Intelligent TMR provides a good compromise
  - Meets timing while increasing fault tolerance
- Takeaway: timing is an important consideration when using TMR

Method	Worst Negative Slack (ns)	
Baseline (plain Vivado)	0.265	
Precision No TMR	0.253	-4.53 %
Precision Global TMR	-0.736	-377.74 %
Precision Intelligent TMR	0.063	-76.23 %

Agenda

- Space Codesign
- SpaceStudio
- Siemens Precision Hi-Rel
- Proposed Workflow
- Use case
- Conclusion

Conclusion

- Effective, automated workflow for generating fault-tolerant systems for standard MPSoC-FPGA systems from high-level application code
- Seamless integration of Siemens' and FPGA vendor's toolsets
- Quickly generates a working prototype ready for further user refinement

What is next?

- Further optimizations (e.g. performance, area and power)
- Demonstration with the NanoXplore NG-ULTRA
- Upcoming in-depth seminar in 2025Q2 with Siemens EDA:
  - Siemens Catapult HLS
  - Siemens Precision Hi-Rel

- Connect with me: guy.bois@spacecodesign.com
- Visit us: www.spacecodesign.com