

Jasper Design Automation

OCP and Verification of Configurable OCP Interfaces

Øystein Kolsrud, Holly Stump

OCP Configurable Implementation

- A blessing and a curse
 - Flexible and widely applicable to many different SoC hardware applications
 - Many implementation variants present a challenge for the verification team
- Formal verification of protocols is a superior approach because formal is comprehensive; exhaustively removing all doubt of incorrect interface behavior
 - Jasper configurable OCP formal verification IP generator automatically creates appropriate OCP properties for a specific design
 - Silicon-tested method of developing a verification plan for OCP designs

Straightforward OCP Configuration File

```
module top {
  addr_def "" {
    base_addr 0xB
    size 0x1FFF
  }
  interface "internal" bundle "ocp2" revision 1 {
    interface_type slave
    param addr 1
    param addr_width 13
    param enbl 1
    port Sclk net Clk
    port MAddr net MAddr
    port MCmd net MCmd
    port MData net MData
    port STagID net STagID
  } # close interface top definition
} # module close
```

- Flexibility of OCP configurations requires protocol interface to be defined within a text-based configuration file
- File identifies design-specific parameters (data command subsets and modes, data and address widths, burst sequence and alignment, data handshaking, and optional signals)
- Parameters can be explicitly declared in the file or loaded from default settings

Challenges of OCP Verification

- Formal verification provides highest confidence level, strengths for protocol verification
 - Configuration file provides great flexibility for customizing the OCP details for a specific architecture
 - Verification teams with access to formal-friendly OCP verification kits reduce verification cycle time, and increase confidence in design behavior
- Configuration freedom: engineers determine if properties in the verification sets are meaningful or can be safely ignored
 - Requires knowledge of what parts of the protocol checking are not required
 - Time saved by using a pre-packaged set of OCP checkers can easily be lost in debugging
 - A better solution is needed

Simplifying OCP Verification Effort

- Jasper Design Automation OCP proof kit IP Generator
 - Produces OCP property suites tailored for specific OCP configurations
 - Property generator creates a subset of the entire protocol automatically, based upon design information
 - Setup straightforward without requiring deep understanding of the protocol; requires no manual changes to the property set once completed
 - Resulting output not restricted to a specific design or property language, and optimized for formal analysis
 - Run times nearly instantaneous; resulting output files available in Verilog or VHDL
- Supports property specification in PSL or SVA

Jasper OCP Proof Kit Results

- Users report reduced verification debugging time right from the outset because of the finely-tuned scope of the property sets
- For design specification changes over time, updating the verification environment is a simple matter of rerunning the IP Generator on the updated OCP configuration file
 - Subsequent design family generations with additional functionality no longer need to endure long debugging efforts to repeatedly identify and remove unused property sets
- IP Generator ensures that verification environment includes only the properties necessary for verification of the specific design implementation

Essentials of Verification Planning

- Begin by verifying that the protocol is implemented correctly with a fully tailored set of formal-friendly OCP properties
 - Protocol verification often involves a large number of interrelated tasks which must be accounted for before the design can be certified as compliant
 - Planning helps coordinate the effort across multiple engineers to ensure verification is completed on schedule
 - Planning identifies prime formal analysis targets in the protocol early in the verification cycle
- Design a verification test plan to leverage formal verification effectively alongside simulation where it can provide the greatest value

Verification Planning

- Address four fundamental questions in the verification flow:
 - What is the design functionality that will be verified?
 - How will each functionality be verified?
 - What is the relative importance or priority of each feature to be verified?
 - How will the team measure when the verification process is complete?
- Answering these questions introduces predictability into the verification schedule, crucial in addressing time-to-market pressures

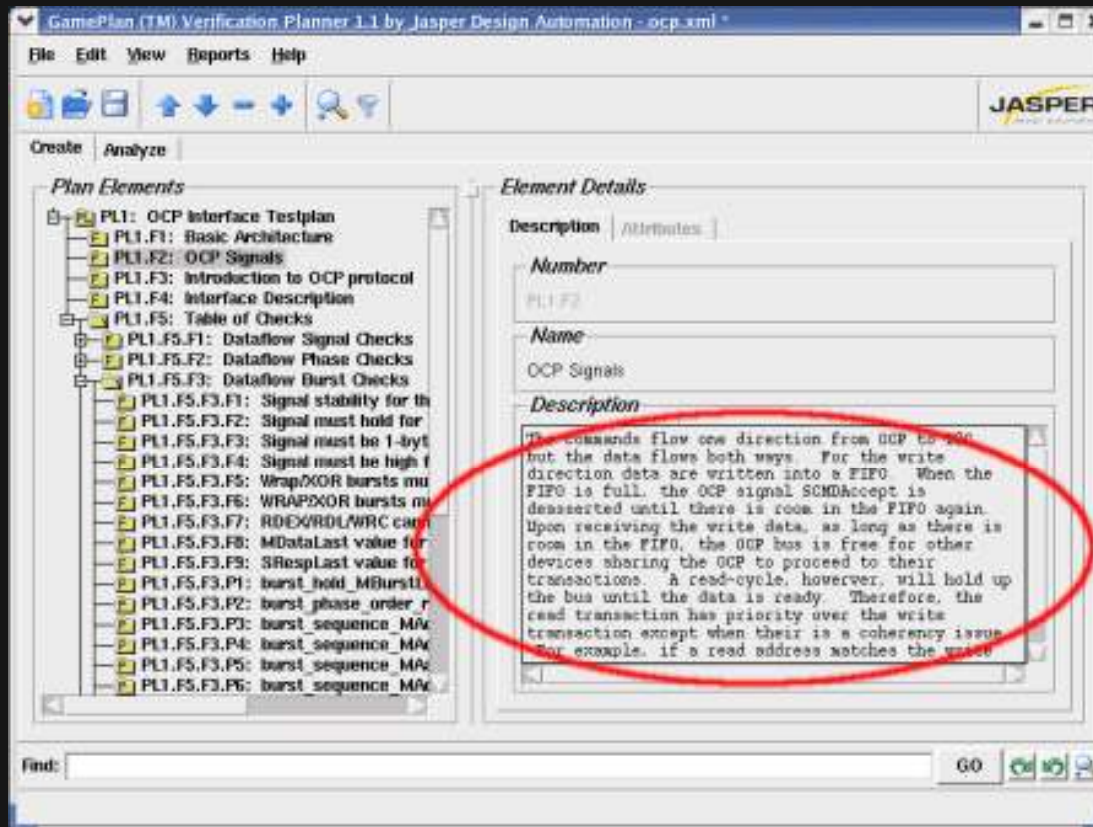
Jasper GamePlan™ Verification Planner

- Jasper GamePlan: easily organize design features to test
 - Powerful analysis and reporting
- Hierarchical verification planning
 - Ensures all high-level issues are covered with extensive coverage at block level
- Planning begins with the natural-language statement of the intended behavior of the design under test.
 - Detailed description essential to understand and communicate exactly what functionality you will test
- Emphasis on what should be tested, not how it will be tested

Verifying OCP Protocols

- Interface is one of the primary features to verify
 - Identify and document properties of each interfacing block to be checked
 - Note any open issues identified upon examination of the connecting blocks
 - Prioritize properties in terms of their critical nature; identify key areas of focus throughout the verification flow and assign appropriate resources
- For interface components of the connecting logic, formal is a good technology choice for protocol verification, regardless of the data destination or source

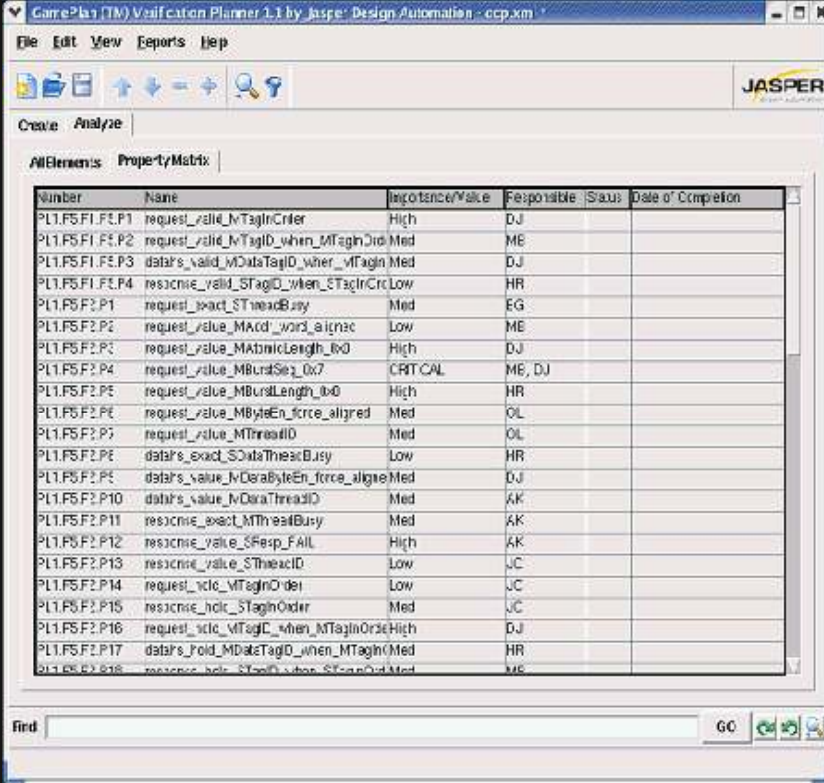
Verification Planning and Design Intent



- Verification planning starts with describing design intent
- Example of the early planning stages in a typical OCP interface design

Critical Functionality

- Weight relative importance of property definition and resource assignments
- Critical functional areas of the design should use formal verification for the highest confidence in the logic.
- Identify dependencies between properties at this stage.
 - Documenting dependencies helps ensure the robustness of the plan



The screenshot shows the 'Property Matrix' window in the Jasper Design Automation Verification Planner. The table lists various properties with their names, importance values, responsible parties, and status.

Number	Name	Importance/Value	Responsible	Status	Date of Completion
P11.F5.F1.FE.P1	request_valid_MTagInCntrl	High	DJ		
P11.F5.F1.FE.P2	request_valid_MTagID_when_MTagInCntrl	Med	ME		
P11.F5.F1.FE.P3	datafs_valid_MDataTagID_when_MTagInCntrl	Med	DJ		
P11.F5.F1.FE.P4	resrcnrc_valid_SFtagID_when_SFtagInCntrl	Low	HR		
P11.F5.F2.P1	request_exact_SThreadBusy	Med	EG		
P11.F5.F2.P2	request_value_MAcid_word_alignac	Low	ME		
P11.F5.F2.P3	request_value_MAtomicLength_b0	High	DJ		
P11.F5.F2.P4	request_value_MBurstSeq_bx7	CRITICAL	ME, DJ		
P11.F5.F2.P5	request_value_MBurstLength_b0	High	HR		
P11.F5.F2.P6	request_value_MByteEn_force_alignrd	Med	OL		
P11.F5.F2.P7	request_value_MThreadID	Med	OL		
P11.F5.F2.P8	datafs_exact_SDataThreadBusy	Low	HR		
P11.F5.F2.P9	datafs_value_MDataByteEn_force_alignrd	Med	DJ		
P11.F5.F2.P10	datafs_value_MDataThreadID	Med	AK		
P11.F5.F2.P11	resrcnrc_exact_MThreadBusy	Med	AK		
P11.F5.F2.P12	resrcnrc_value_SFresp_FAIL	High	AK		
P11.F5.F2.P13	resrcnrc_value_SThreadID	Low	JC		
P11.F5.F2.P14	request_vclc_MTagInCntrl	Low	JC		
P11.F5.F2.P15	resrcnrc_hold_SFtagInCntrl	Med	JC		
P11.F5.F2.P16	request_vclc_MTagID_when_MTagInCntrl	High	DJ		
P11.F5.F2.P17	datafs_fold_MDataTagID_when_MTagInCntrl	Med	HR		
P11.F5.F2.P18	resrcnrc_hold_SFtagInCntrl_when_SFtagOutCntrl	Med	ME		

Multiple Verification Technology Coverage

- For coverage holes, schedule additional simulation or formal verification testing
 - Ensure the plan is complete with identified and documented all feature dependencies
- With this plan in place, verification can proceed and a team can readily track it over the lifetime of the project

Coverage Matrix

Coverage			Properties				
			ID	P1.F1.P1	P1.F1.P2	P1.F2.P1	P1.F2.P2
			Method	Formal Verification	Formal Verification	Directed Test, Constrained Random	Constrained Random
			Threshold	FP3	PP3	N/A	N/A
			Status	Completed	Completed	Completed	Completed
ID	Type	Threshold Hierarchy					
P1.F1.P1.C1	Code	99% Block	X				
P1.F1.P2.C1	Functional	N/A Block		X			
P1.F1.P2.C2	Functional	N/A Block		X			
P1.F2.P1.C1	Code	100% Block			X		
P1.F2.P1.C2	Functional	100% Block System			X		
P1.F2.P2.C1	Code	100% Block				X	
P1.F2.P2.C2	Functional	80% Block System				X	

- Coverage matrix shows both formal and simulation

Summary

- Designing a reusable OCP verification environment presents challenges due to highly customizable protocol
- For formal verification, leverage Jasper's OCP IP Generator to automatically deduce the required subset of properties specific to the design
 - The flexibility of the output is readily adaptable to mixed-RTL designs and requires little additional OCP knowledge
 - OCP IP Generator produces output that is efficiently modeled for formal verification, providing the highest confidence levels possible
- GamePlan hierarchical verification eases work necessary to set up and maintain a verification test plan

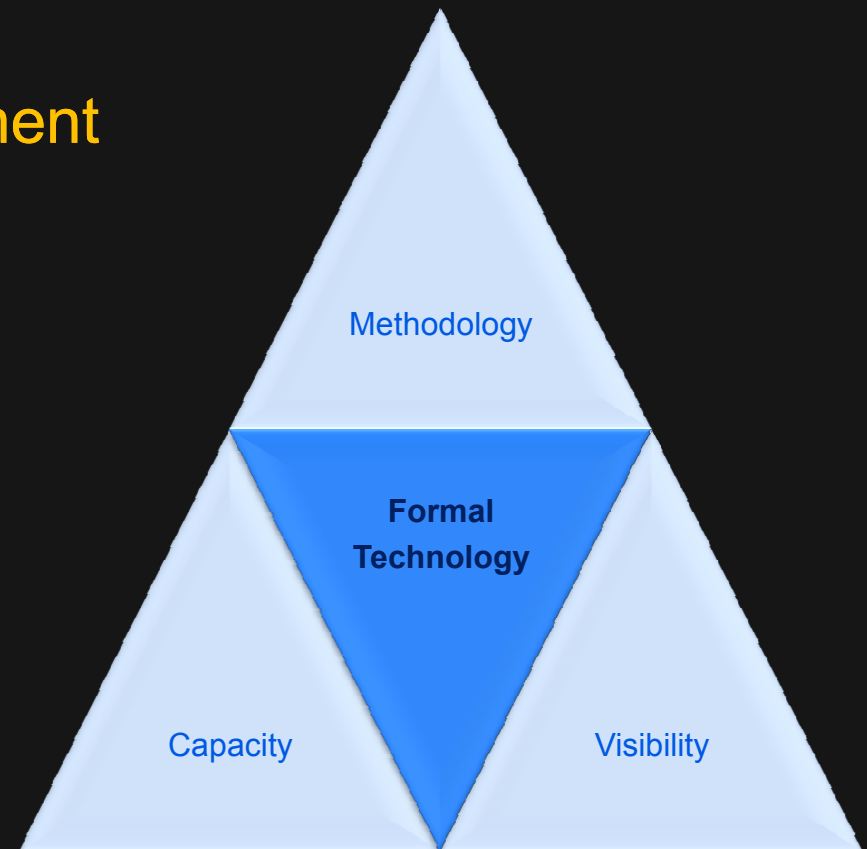
JasperGold Leads Industry Deployment

JasperGold Technology Deployment

- Methodology
- Capacity
- Visibility

- Supported by Jasper Services

- Patented formal technology
 - 11 patents, 5 pending



Design

Verification

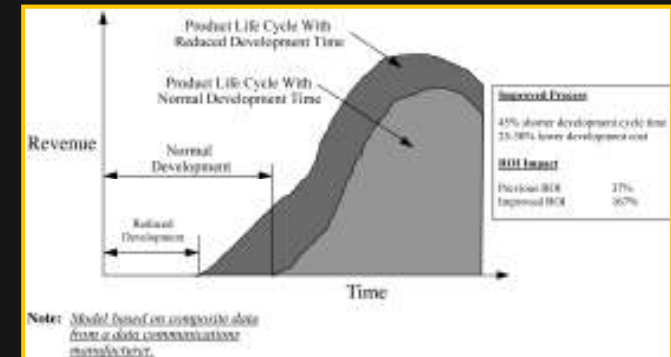
Reuse

Jasper Design Delivers Targeted ROI

Jasper Design Automation deploys advanced formal verification technology for the competitive advantage of its customers, across the SoC design cycle

Competitive advantage through targeted ROI:

- Risk
 - Avoid fatal bugs and respins
- Resources
 - Reduce verification and SoC integration engineering effort
 - Saves engineering effort with design reuse
- Time to market
 - Accelerate schedule and silicon delivery



Jasper for Competitive Advantage

Customers in a wide variety of electronics markets rely on Jasper to solve their most critical business and project challenges

- Wireless, consumer, computing, networking, graphics and more
- Over 100 successful chip designs with JasperGold



“Eliminating top project risks with JasperGold ultimately shortens the schedule and reduces overall project costs”

Jasper Formal Technology for Targeted ROI



For more information on OCP-IP verification, please
contact:

Jasper Design Automation
www.jasper-da.com

Reference:

<http://www.eetimes.com/showArticle.jhtml?articleID=199501643>