Heterogeneous System Composition, Simulation, and Validation with the assistance of IP-XACT

Gary Delp, PhD EE

Mayo Clinic

Special Purpose Processor Development Group (SPPDG)

Purpose and Outline

Convince you of the utility of standards and standard based tools when assembling, simulating, and validating (heterogeneous) components and interconnections

Outline

- Introductions
  - SPPDG
  - IEEE Stds
- IP Reuse
- Interconnections & Integration
  Making your job harder with automation
- **The Talk**
- Conclusions and Questions

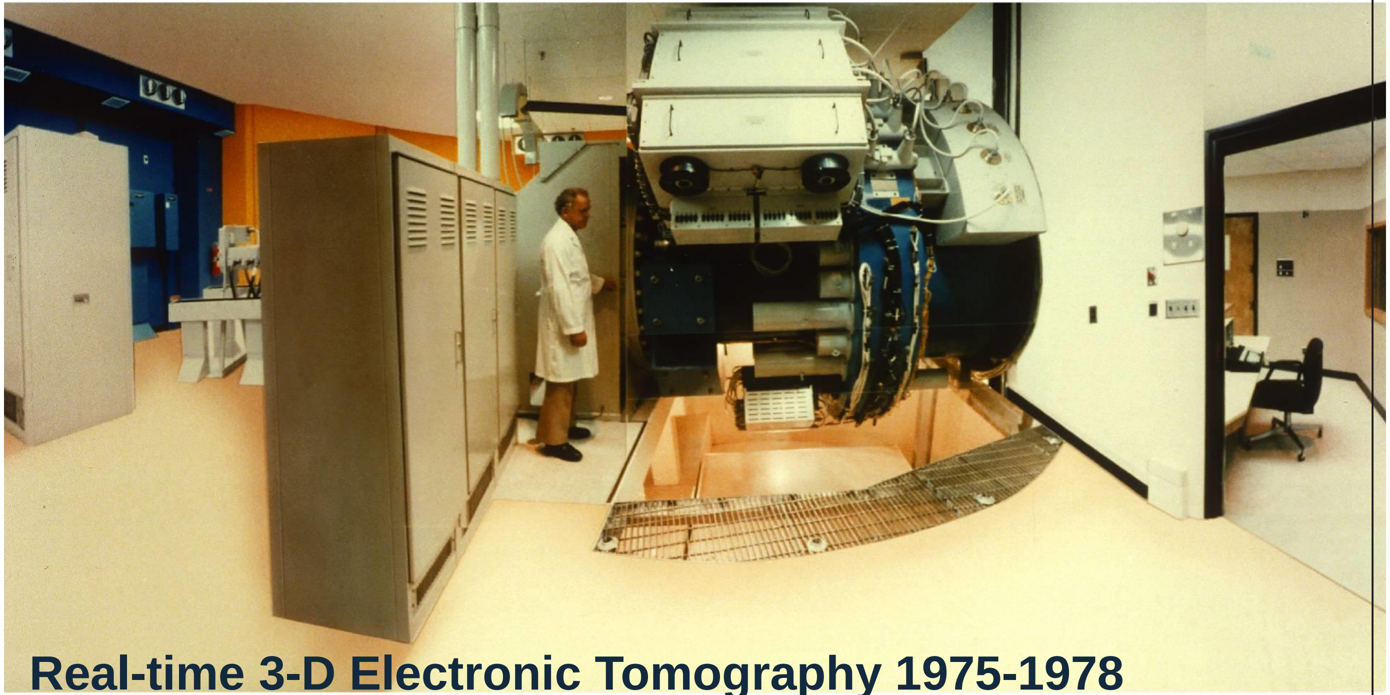
- Structure and Resources IP-XACT
- Verification (+ coverage) UVM
- Power UPF
- Quality Assessment QIP
- Tagged traceability Soft & Hard Tags
- Additional Semantics and constraints

no recommendations expressed or implied
#include <std/disclaimer.h>

Special Purpose Processor Development Group (SPPDG) Overview

- Research group in the department of Physiology and Biomedical Engineering at Mayo Clinic
  - Founded in 1971 with initial work in high-speed medical signal processing
  - Still operating under the same Director: Dr. Barry Gilbert
- Since 1978, primarily funded by U.S. Government agencies
  - R & D in high performance electronics and related systems
  - *Charter for Mayo:* transfer technology to biomedical research and clinical practice
  - *SPPDG* operates as a government-trusted, unbiased (non-competitive), third-party for TRL-1-3 research, proof of concept development, and bleeding edge evaluation and validation

PHOTOGRAPH OF EXPERIMENTAL DYNAMIC SPATIAL RECONSTRUCTOR
THREE DIMENSIONAL X-RAY COMPUTED TOMOGRAPHY IMAGING MACHINE
(Side View of Rotating Gantry, Showing 14 X-ray Tubes and
14 Video Image Chains Mounted on Gantry; Designed in 1975;
Fabricated at Raytheon, Sudbury, MA; Installed at Mayo in 1978)



Real-time 3-D Electronic Tomography 1975-1978

xxx / 1978 / BKG / 6128v2

Now:

- Power Packaging Cooling
- Technology evaluation
- Medical (and other) Sensing
- Signal and Power integrity
- Specialized Mathematics
- 6 transistor analysis
- Architecture, analysis & processing
- graph & neural dense SIMD
- Sparse array
- In Memory
- Composites/ optics
- Novel technologies and extreme temperatures

■ ■ ■ ■

Medical Applications and Beneficial Potentials for Hybrid Computing Platforms

Medical Areas		Dense linear algebra	Sparse linear algebra	Graph Algorithms	Frequency Analysis	Data Retrieval/ Filtering/ Sorting	Stochastic processes	Monte Carlo Particle methods
Basic Bio-medical Modeling	Disease Processes		X	X			X	
	Devices / Physics	X			X		X	X
	Biology						X	X
Clinical science	Population Statistics	Trad	Graph	Graph	Trad	Both	Both	
Clinical Practice	Image Formation	X			X		X	
	Image Analytics	X	X	X	X	X	X	X
	Genomic Analysis	X				X		
	Decision Support		X	X		X	X	
Health Management	Trend Analytics	Trad	Graph	Graph	Trad	Both	Both	
	Privacy Protection					X		

Key Messages & Goals for the talk

Goals of Structured Interconnections

Useful

Functional

Logical

Physical

Correct

Consistent

Easy

Verify-able

Build-able

Test-able

Abstract-able

Reuse-able

Transport-able

Implement-able

Goals of the Talk

In 25 minutes you will:

- Know about electronic design automation (EDA) standards as *providers of capability*:
  - IP-XACT IEEE Std 1685-2014
  - Unified verification methodology (UVM)
    - IEEE Std 1800.2
  - Power intent IEEE Std1801-2015
  - IP Quality IEEE Std 1734-2011
  - IP Tagging - VSIA => Accellera Standard
- Know where to find tools to successfully interconnect heterogeneous is systems
  - See Backup for 4 pages of links

OK to Click Next

accellera SYSTEMS INITIATIVE™

UVM

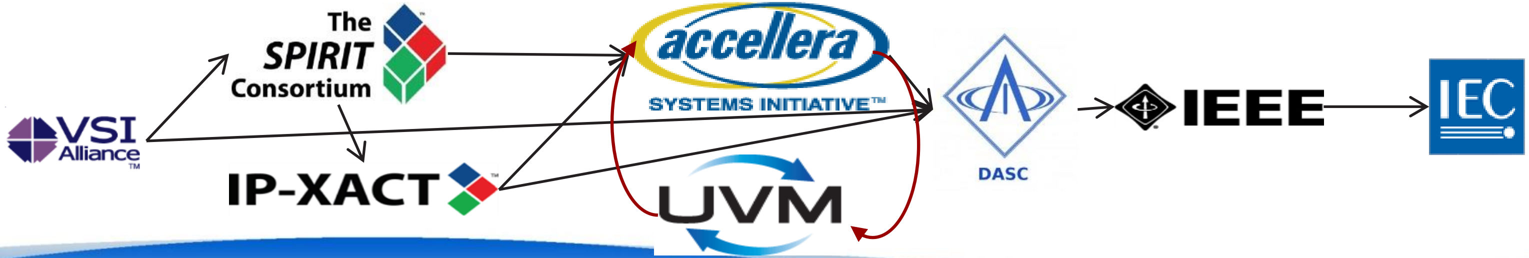
IEEE

IP-XACT

Grass Roots to International agreement potential paths to "simplified structure"

(occasionally over 50 standards in an area each with 1000+ pages)

- The UN International Standards association (ISO) has accredited:
  - International Electrotechnical Commission (IEC)
- The American National Standards Institute ANSI has accredited
  - IEEE Standards Association
    - IEEE Computer Society has accredited the Design Automation Standards Committee (DASC)
  - The DASC charters working groups for different proposed EDA standards



Outline of the Talk & Progress Toward the Goals

Purpose: Convince you of the new capabilities of standards and standard-based tools for assembling, simulating, and validating (heterogeneous) components and interconnections

- □ Structure and Resources: IP-XACT
- □ Verification (+ coverage): UVM
- □ Power Management: UPF
- □ Quality Assessment: QIP
- □ Traceability: Soft & Hard Tags
- □ Additional Semantics and constraints
- □ Conclusions and Questions

Goals of Structured Interconnections

Useful
Functional
Logical
Physical
Correct
Consistent
Easier

Verify-able
Build-able
Test-able
Abstract-able
Reuse-able
Transport-able
Implement-able

Outline of the Talk & Progress Toward the Goals

Purpose: Convince you of the new capabilities of standards and standard-based tools for assembling, simulating, and validating (heterogeneous) components and interconnections

- □ Structure and Resources: IP-XACT
- □ Verification (+ coverage): UVM
- □ Power Management: UPF
- □ Quality Assessment: QIP
- □ Traceability: Soft & Hard Tags
- □ Additional Semantics and constraints
- □ Conclusions and Questions

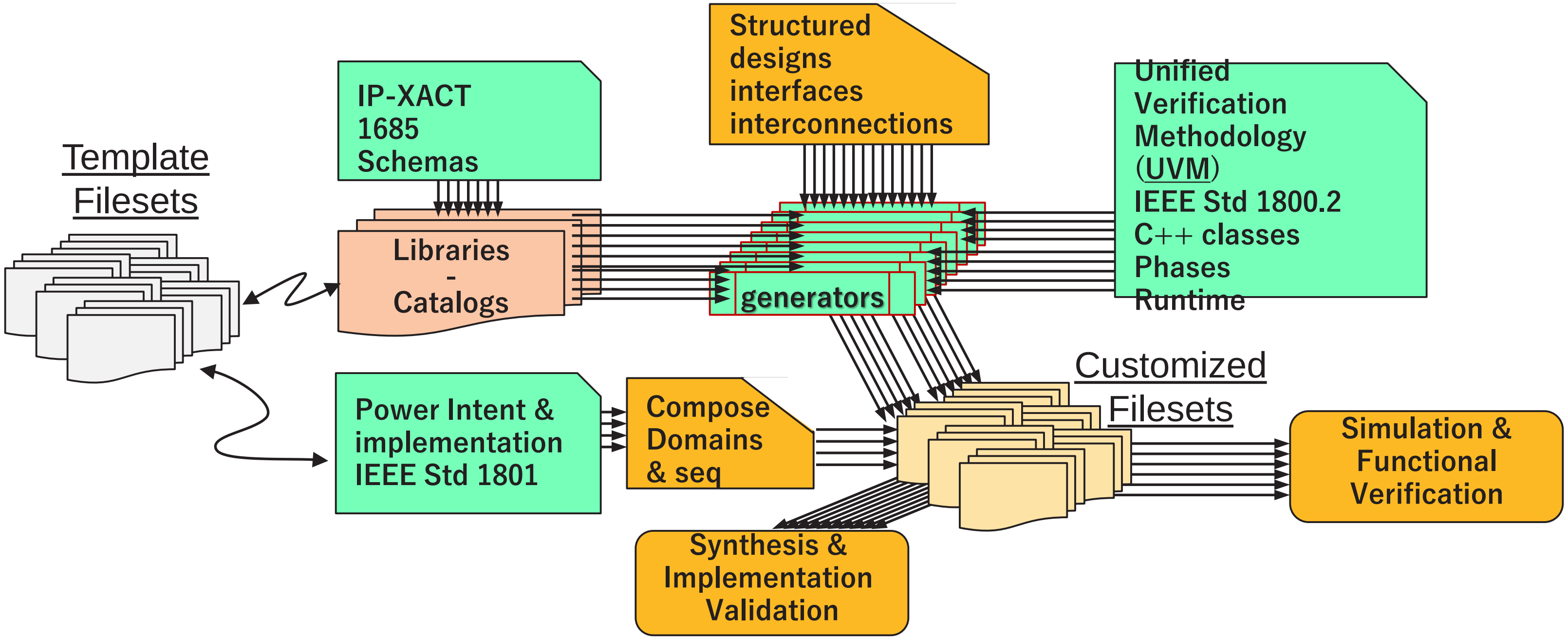
Goals of Structured Interconnections

Useful
Functional
Logical
Physical
Correct
Consistent
Easier

Verify-able
Build-able
Test-able
Abstract-able
Reuse-able
Transport-able
Implement-able

Hierarchical Composition of Systems

Simplicity / ~~Divide Effort~~ ~~Multiply~~ / Conquer the problem => Simplicity / Conquers the problem



Template Filesets

IP-XACT
1685
Schemas

Libraries
-
Catalogs

Structured
designs
interfaces
interconnections

generators

Unified
Verification
Methodology
(UVM)
IEEE Std 1800.2
C++ classes
Phases
Runtime

Power Intent &
implementation
IEEE Std 1801

Compose
Domains
& seq

Customized Filesets

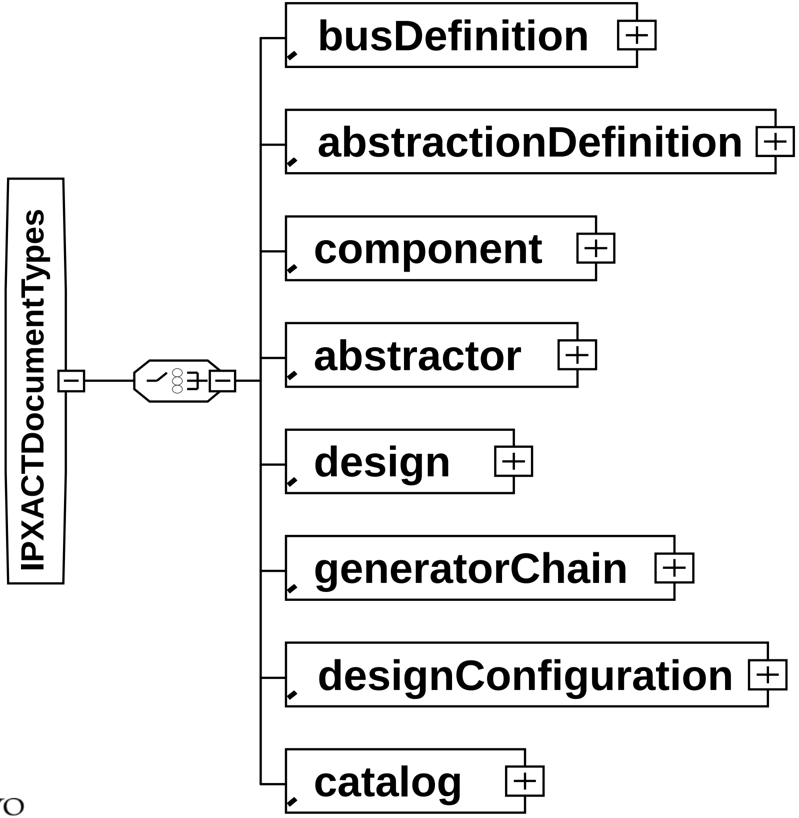
Simulation &
Functional
Verification

Synthesis &
Implementation
Validation

Standard Synergies – Data & MetaData

- IP-XACT is Meta-Data
  - information about
    - the information about
      - the design – Electronic Databook
- Provides for codified data transfer semantics across languages & tools
- Provides both Concrete and Abstract *Abstraction Descriptions*
  - Systems composed of components
    - Composed of systems
      - Composed of Components
- Integration Constraints
  - IP-XACT can express limitations of or requirements for adjoining IP’s not available in VHDL or Verilog
  - Generators can check the use of an IP
- Standard Xtensions for Power Management, VIP, Bus Interconnections, Version tracking, . . .

The IP-XACT XML Schema Defines the Syntax for the Documents that Enable System Composition …



- The <u>*SCHEMA*</u> of each of the top level documents are used to codify the required/allowable structure for the types of elements. (*Syntax*)
- Those documents the describe individual “classes of buses, components, etc.
- Instances of documents based on the “classes” of elements hold the Meta-Data for each element of a design.
- *Semantic* Consistency Rules ensure:

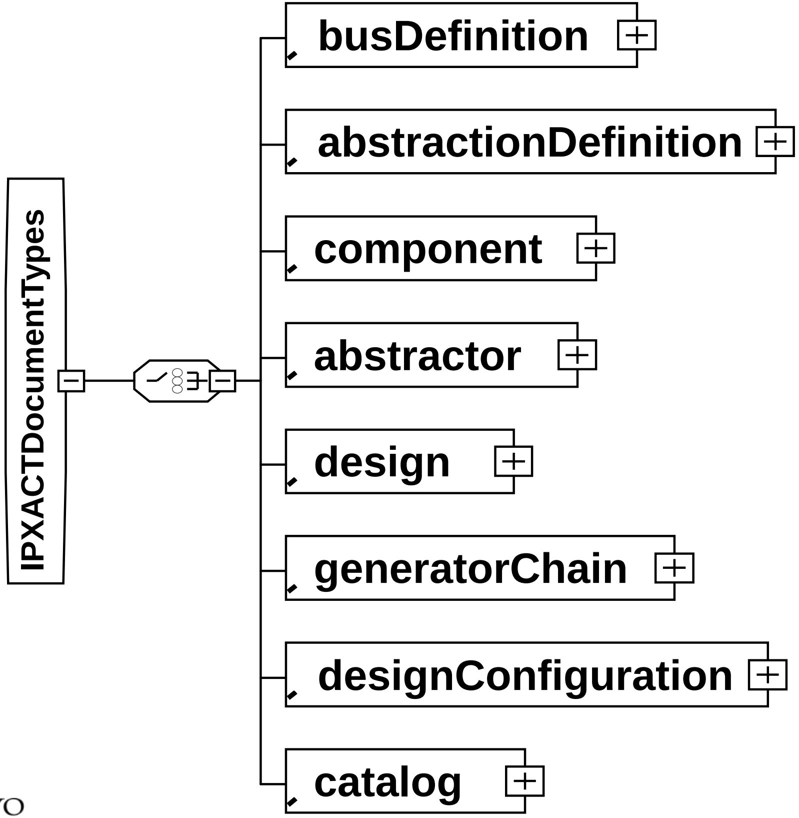
  That the *Humpty Dumpty Principal* is followed, “When I use a word, it means exactly what I want it to mean – no more – no less. Who will be the master, you, or the word?”

```
<xs:schema xmlns:ipxact="http://www.accellera.org/XMLSchema/IPXACT/1685-2014"
           xmlns:xs="http://www.w3.org/2001/XMLSchema"
           targetNamespace=http://www.accellera.org/XMLSchema/IPXACT/1685-2014
           elementFormDefault="qualified">
```

```
<xs:schema
    <xmlns:ipxact="http://www.accellera.org/XMLSchema
     /IPXACT/1685-2014"
    xmlns:xs="http://www.w3.org/2001/XMLSchema"
    targetNamespace="http://www.accellera.org
     /XMLSchema/IPXACT/1685-2014" elementFormDefault="qualified">
<xs:include schemaLocation="busDefinition.xsd" />
<xs:include schemaLocation="component.xsd" />
<xs:include schemaLocation="design.xsd" />
<xs:include schemaLocation="designConfig.xsd" />
<xs:include schemaLocation="abstractionDefinition.xsd"/>
<xs:include schemaLocation="catalog.xsd" />
<xs:include schemaLocation="abstractor.xsd" />
<xs:group name="IPXACTDocumentTypes">
   <xs:annotation>
    <xs:documentation>This IP-XACT schema documentation is part of the
       IP-XACT standard deliverables. The diagrams in this
       documentation represent the relationships between elements of
       the schema together with their attributes and expected values.
       Valid IP-XACT XML files must have a top-level type that is one of
       the elements listed here.</xs:documentation>
    </xs:annotation>
       <xs:choice>
          <xs:element ref="ipxact:busDefinition">
             <xs:annotation>
                <xs:documentation>To define all elements and attributes
                   supported when defining a bus.</xs:documentation>
             </xs:annotation>
             </xs:element>
```

```
<xs:element ref="ipxact:abstractionDefinition" />
<xs:element ref="ipxact:component">
   <xs:annotation>
      <xs:documentation>To define all elements and attributes
         supported when defining a
         component.</xs:documentation>
   </xs:annotation>
</xs:element>
<xs:element ref="ipxact:abstractor" />
<xs:element ref="ipxact:design">
   <xs:annotation>
      <xs:documentation>To define all elements and attributes
         supported when defining a design and its configured
         components</xs:documentation>
   </xs:annotation>
</xs:element>
<xs:element ref="ipxact:generatorChain">
<xs:annotation>
   <xs:documentation>To define all elements and attributes
      supported for defining generator
      chains.</xs:documentation>
</xs:annotation>
</xs:element>
<xs:element ref="ipxact:designConfiguration" />
<xs:element ref="ipxact:catalog" />
      </xs:choice>
   </xs:group>
</xs:schema>
```

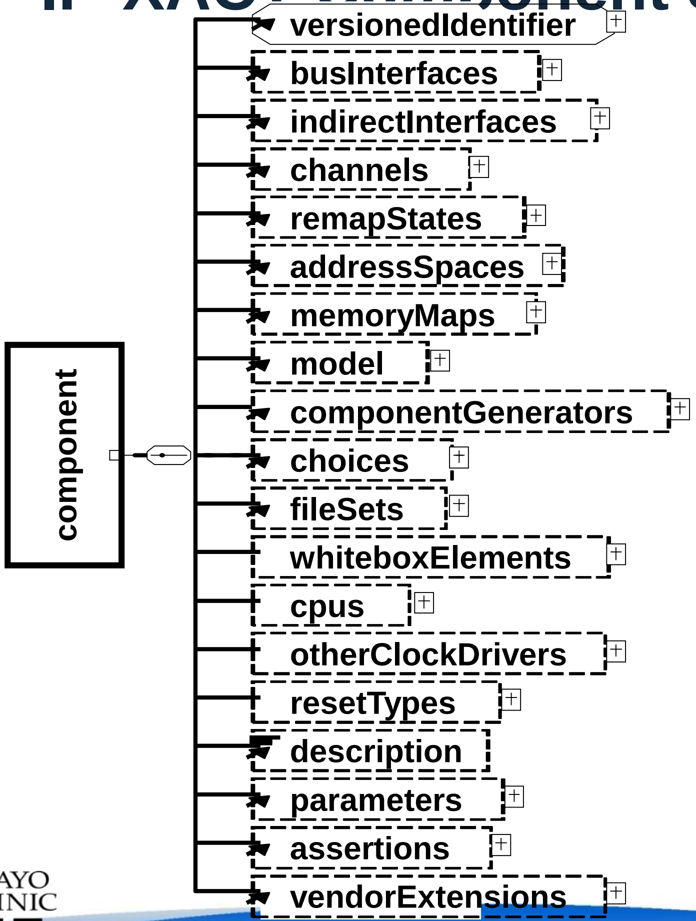
The IP-XACT XML Schema Defines the Syntax for the Documents that Enable System Composition …



- The *SCHEMA* of each of the top level documents are used to codify the required/allowable structure for the types of elements. (*Syntax*)
- Those documents the describe individual “classes of buses, components, etc.
- Instances of documents based on the “classes” of elements hold the Meta-Data for each element of a design.
- *Semantic* Consistency Rules ensure:

  That the *Humpty Dumpty Principal* is followed, “When I use a word, it means exactly what I want it to mean – no more – no less. Who will be the master, you, or the word?”

IP-XACT Component Schema (elements of a component)



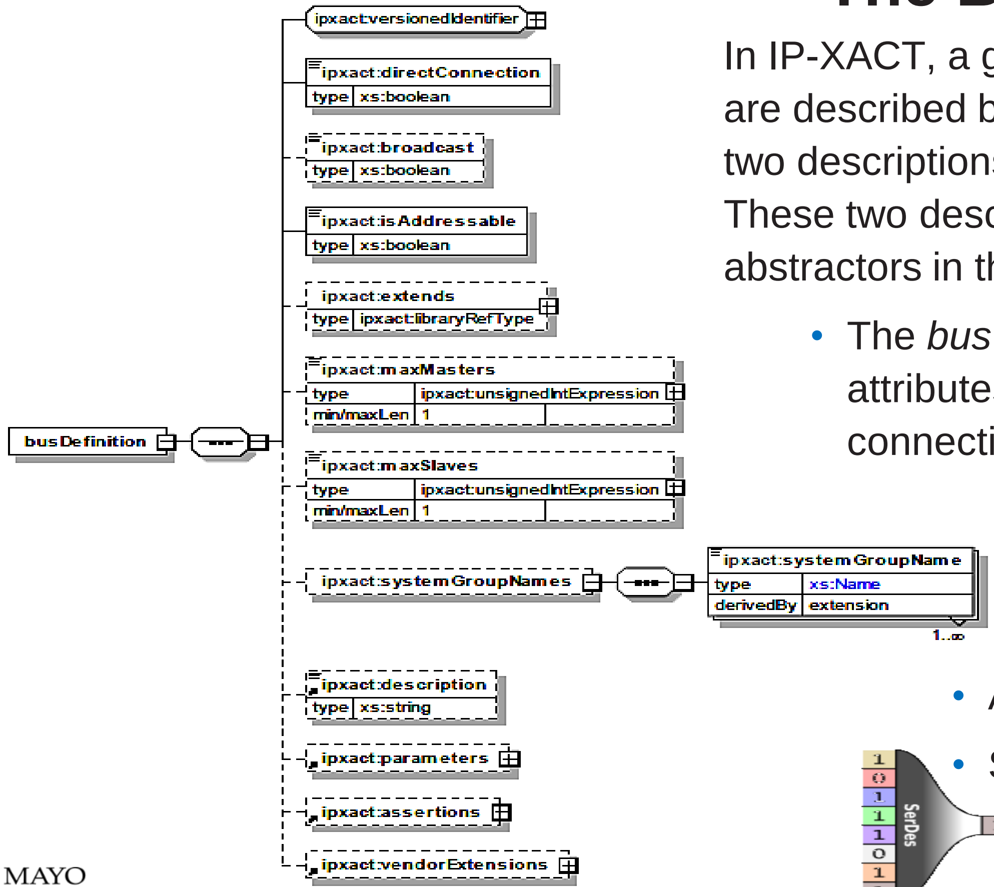
The component is the composable unit which includes:

- VLNV – Vendor Library Name Version
- Interfaces
- Memory & address spaces including
  - Full-Featured Registers
  - CPUs
  - White Box Elements
- Component generators based on interconnections, choices, file sets, parameters, vendorExtentions, etc.
- Vendor Extensions

The Busdefinition Document

In IP-XACT, a group of ports that together perform a function are described by a set of elements and attributes split across two descriptions: a *bus definition* and an *abstraction definition*. These two descriptions are referenced by components or abstractors in their bus or abstractor interfaces.

- The *bus definition* description contains the high-level attributes of the interface, including items such as the connection method and the indication of addressing.



Examples include *bus definitions* for interfaces like

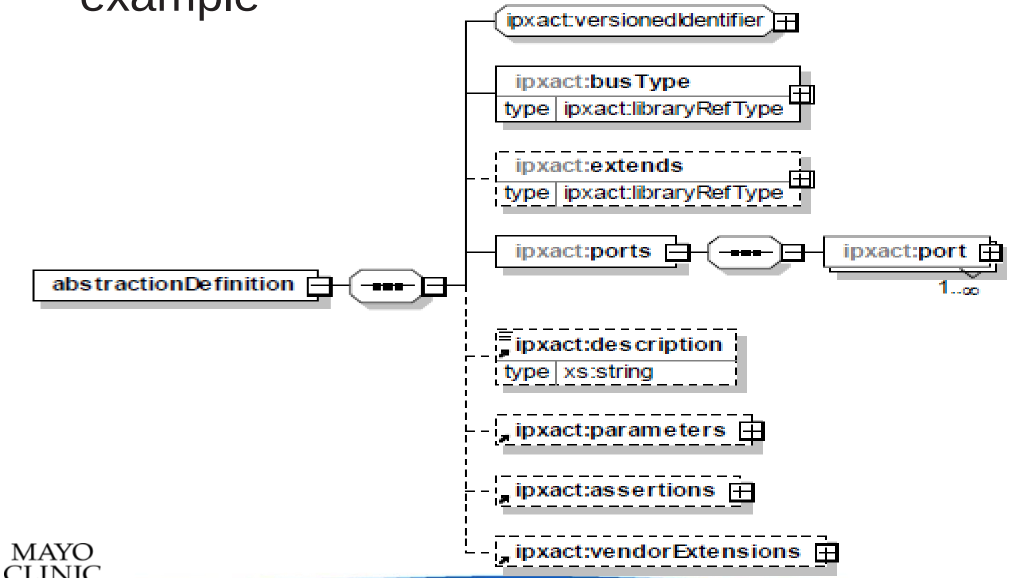
- AMBA Peripheral Bus (APB)
- SerDes

At many levels of abstraction

The Abstraction Definition

The *abstraction definition* contains the low-level attributes of the interface, including items such as the name, direction, and width of the ports. This is a list of logical ports that may appear on a bus interface for that bus type.

- Multiple abstractions definitions can be associated with a single bus definition, for example



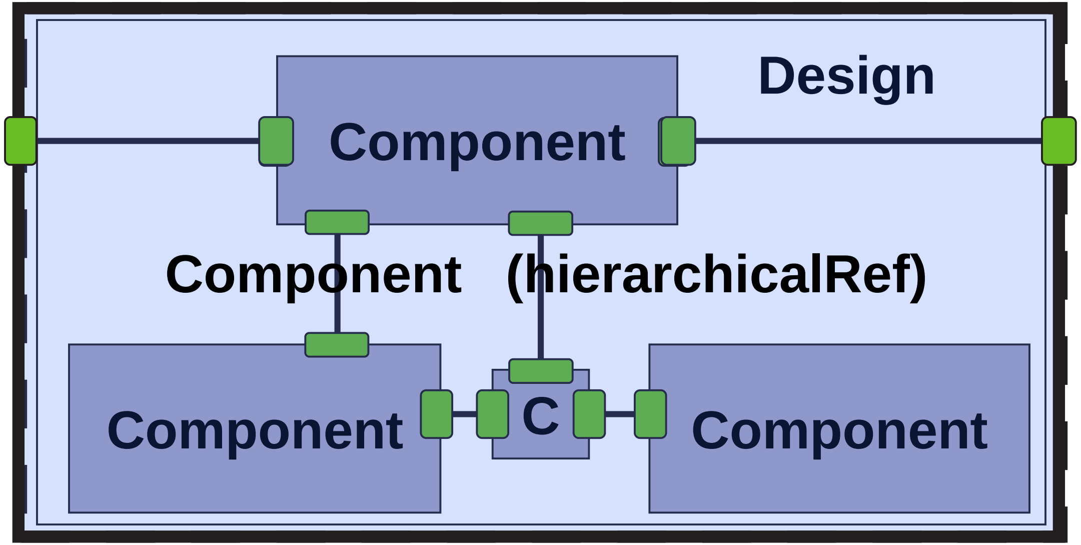
The Ethernet physical layer encompasses coaxial, twisted pair and fiber-optic physical media interfaces (PMIs) Common forms are 10BASE-T and 1000BASE-T. These three use twisted pair cables and 8P8C modular connectors. Fiber optic variants are also very common in larger networks.

All Protocols Should Talk Nicely Like Padlipski
Verification through the Abstractions

- Application
- Presentation
- Session
  - Setup, Initialize, maintain state
- Transport
  - Turns Frames into data stream/ address accesses, MESI transactions
- Network
  - Get from one place to another
    - Naming addressing routing
- Link coding across wires electrical or optical connection
- Physical / Mechanical

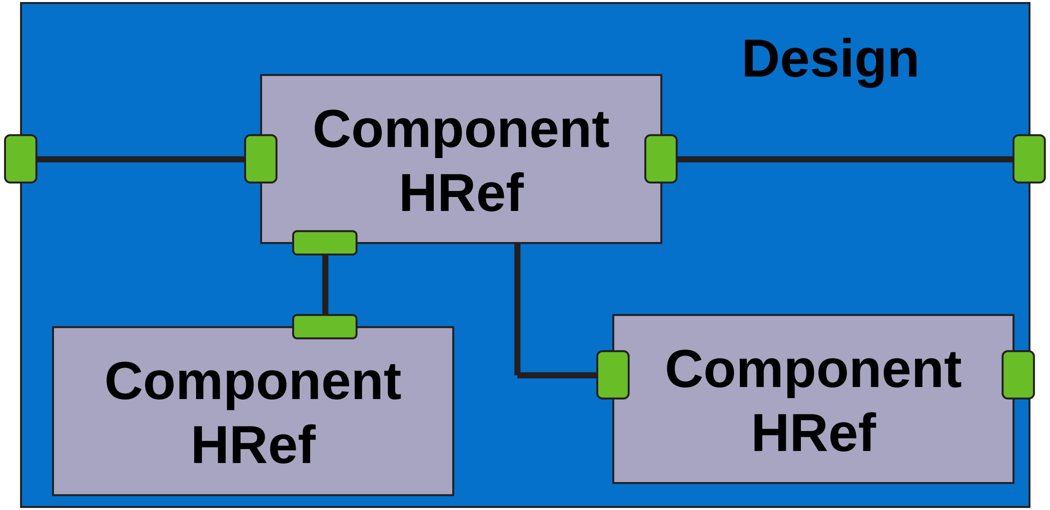
Design/Component Hierarchy

- Managing Complexity:
  - Design and Conquer
  - Divide and Conquer
  - Reuse
  - Provide Robust Interfaces



OK to Click Next

Design/Component Hierarchy



- The Interface specification is common to:
  - the design of the component and
  - the Reuse of the component.
- It is the contract made at the boundary between design teams.
  - Top–down and Bottom–up are really just two views of the Component/Interface–based design flow.

OK to Click Next

IP-XACT eco-system is applicable to the 3-D composition space

- Components are composed into systems
- IP-XACT (consistent) interfaces are defined, used, and tested
- Constraints may be simple or complex
- Bundling the descriptions requires agreed semantics
- The *IP-XACT Architecture and Xtensions workgroup* in Accellera
  - Is a good place for converging on semantic agreements
- Progressive refinement of Metadata Descriptions
  - Provide successive approximations to the working systems

this is often just a part of the process

OK to Click Next

Outline of the Talk & Progress Toward the Goals

Purpose: Convince you of the new capabilities of standards and standard-based tools for assembling, simulating, and validating (heterogeneous) components and interconnections

- □ Structure and Resources: IP-XACT
- □ Verification (+ coverage): UVM
- □ Power Management: UPF
- □ Quality Assessment: QIP
- □ Traceability: Soft & Hard Tags
- □ Additional Semantics and constraints
- □ Conclusions and Questions

Goals of Structured Interconnections

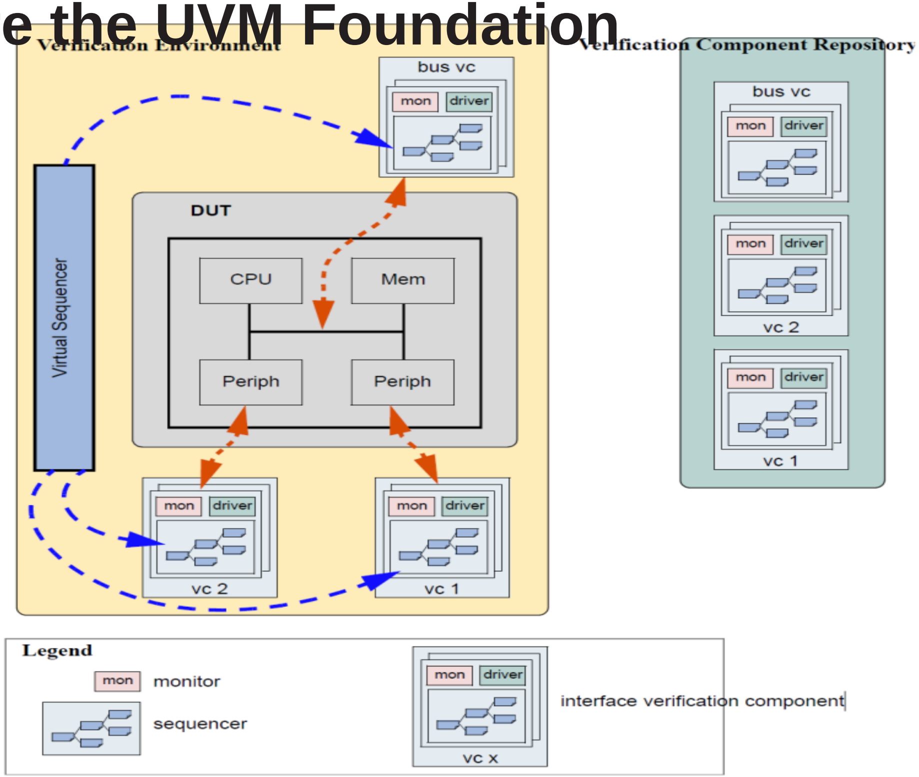
Useful
Functional
Logical
Physical
Correct
Consistent
Easier

Verify-able
Build-able
Test-able
Abstract-able
Reuse-able
Transport-able
Implement-able

IP-XACT and UVM Support
Automated Generation and Testing [of Interconnections]
System C++ Classes are the UVM Foundation

Environment
Verification
Component
Repository
vc 1
mon driver
mon driver
bus vc mon driver
bus vc

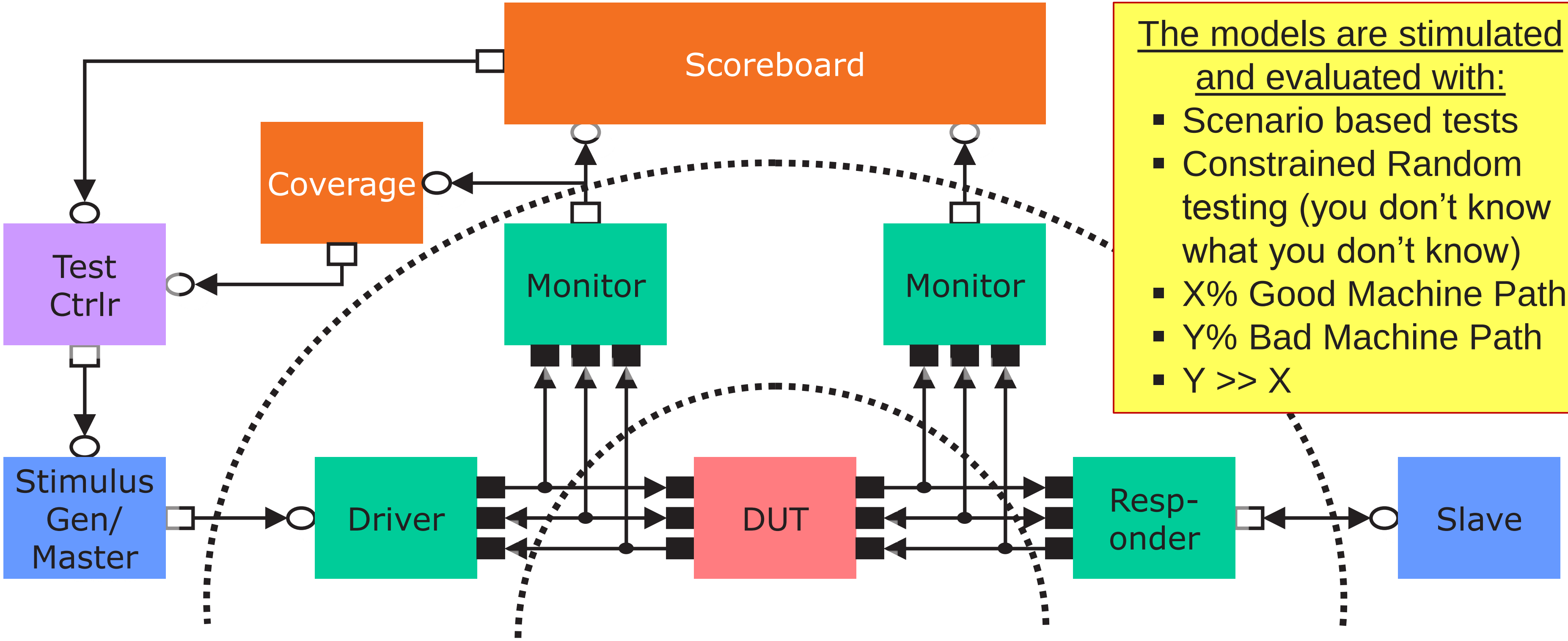
vc 2
mon driver
vc 1
mon driver
mon
vc 2
mon driver
Legend
monitor
sequencer
vc x
mon driver
interface verification
component



- https://workspace.accellera.org/downloads/standards/uvm

OK to Click Next

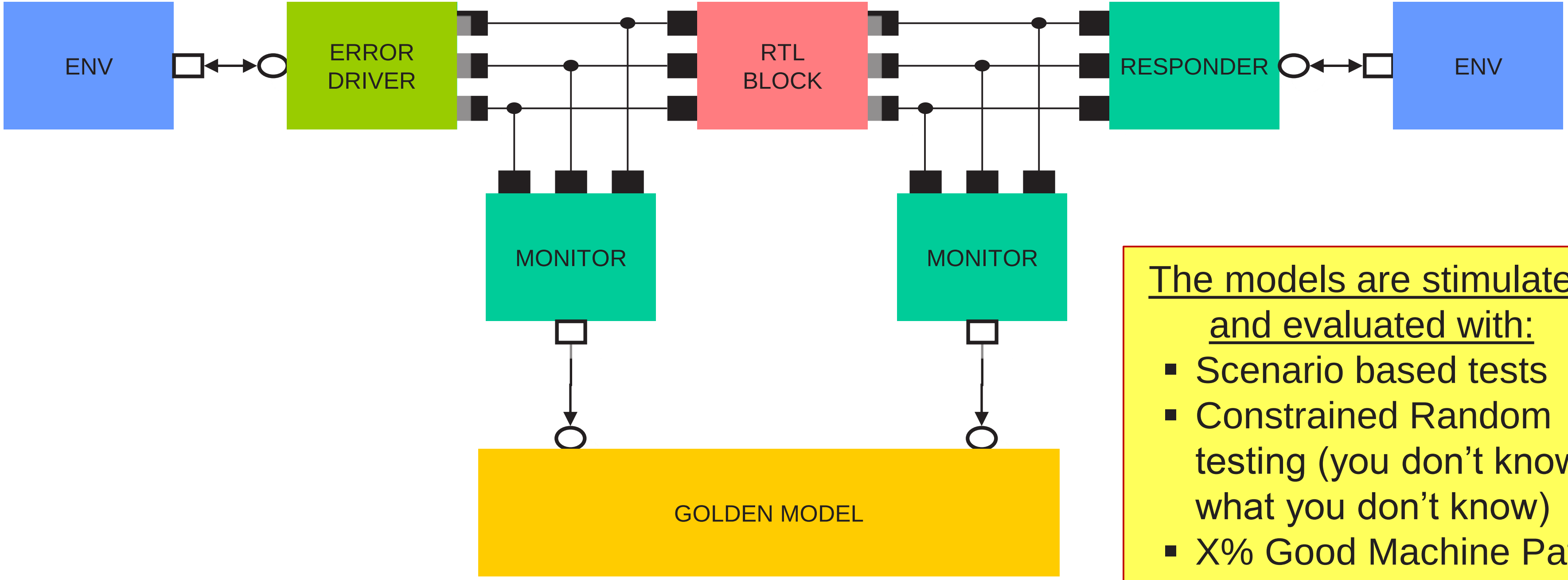
UVM Generic Testbench Organization
Device Under Test (DUT) Simulation and Verification Environment



<u>The models are stimulated and evaluated with:</u>

- Scenario based tests
- Constrained Random testing (you don’t know what you don’t know)
- X% Good Machine Path
- Y% Bad Machine Path
- Y >> X

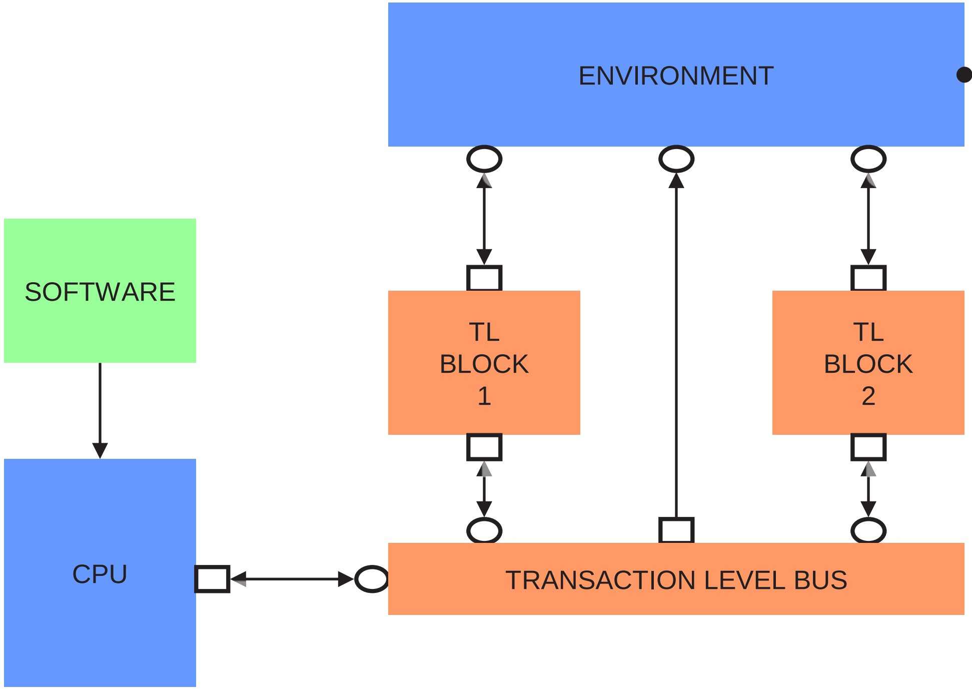
Golden Models



<u>The models are stimulated and evaluated with:</u>

- Scenario based tests
- Constrained Random testing (you don't know what you don't know)
- X% Good Machine Path
- Y% Bad Machine Path
- Y >> X

Transaction Level Model

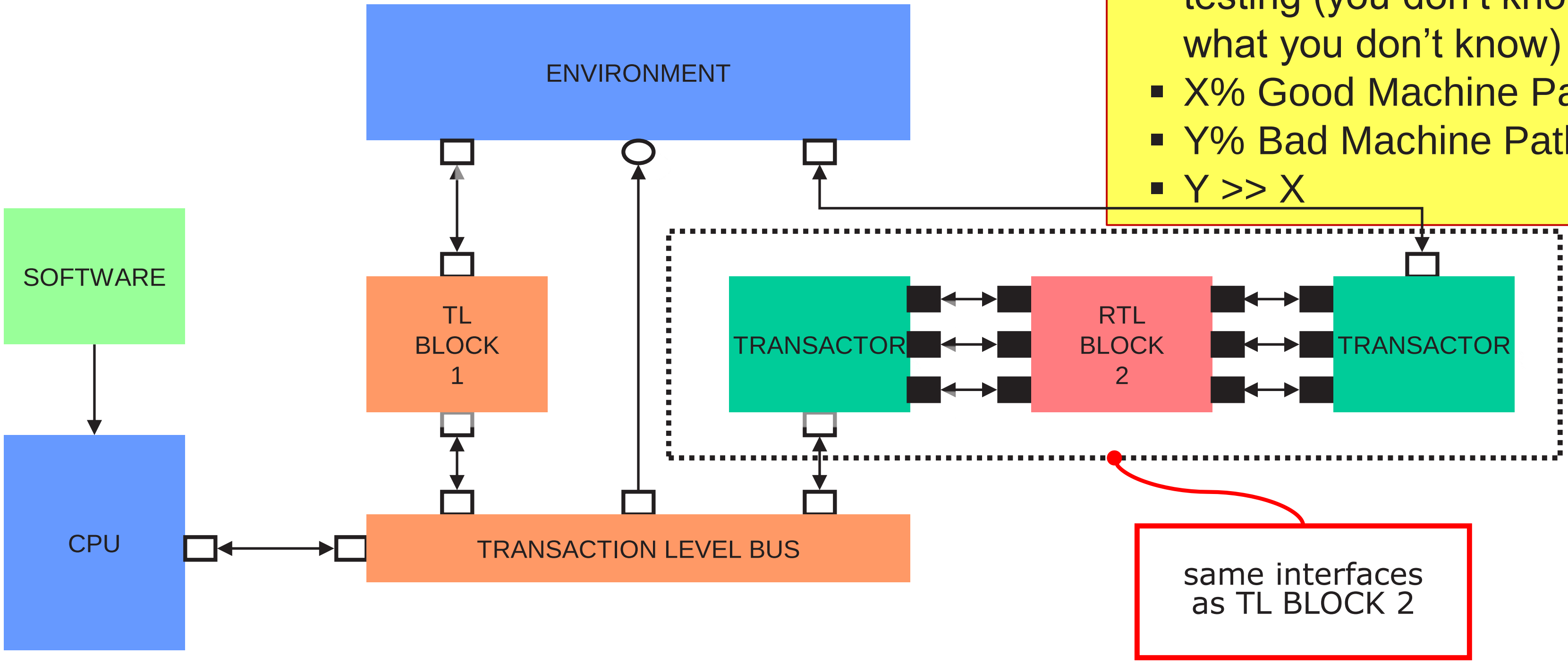


- Stimulus
- Response
- Coverage
- Analysis

<ins>The models are stimulated and evaluated with:</ins>

- Scenario based tests
- Constrained Random testing (you don't know what you don't know)
- X% Good Machine Path
- Y% Bad Machine Path
- Y >> X

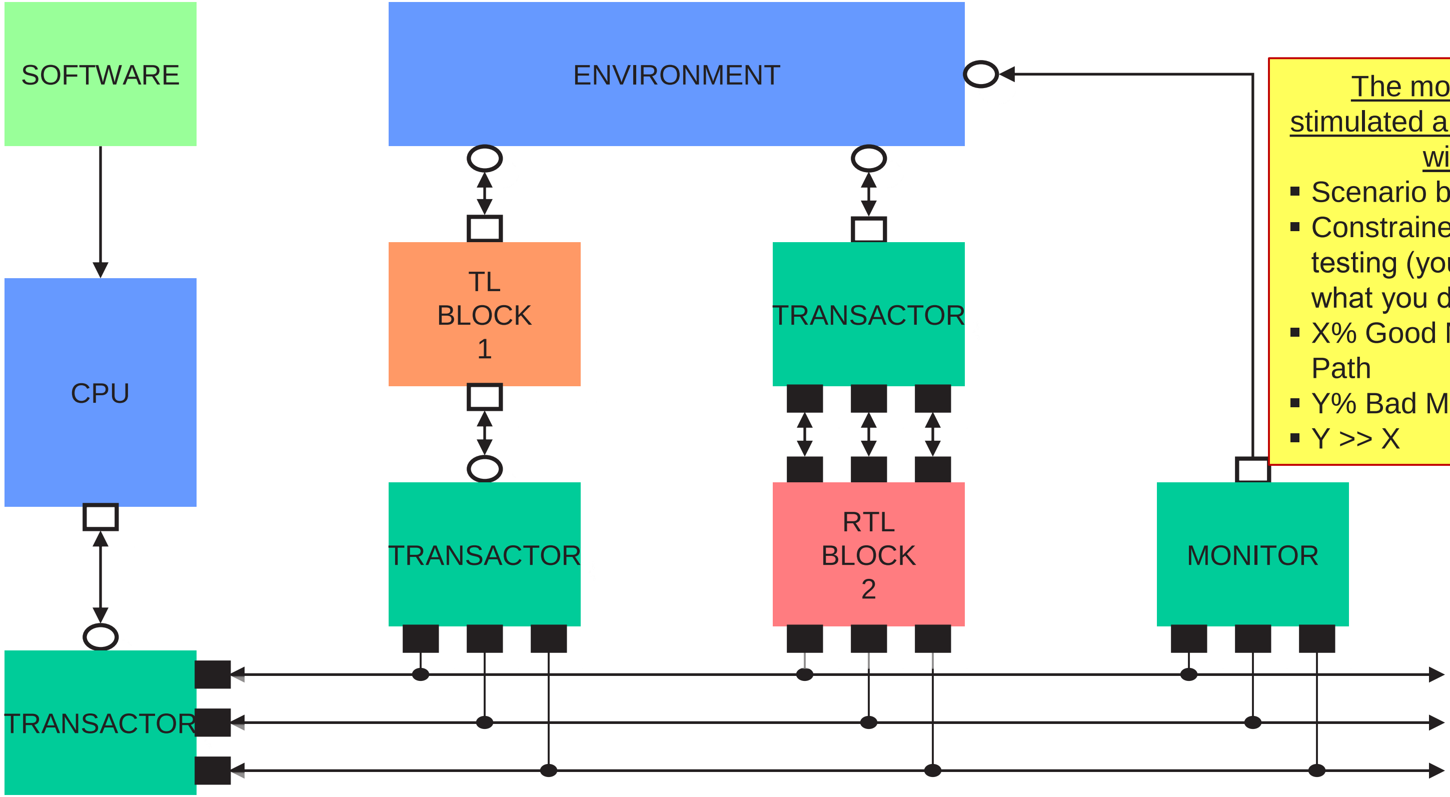
Refinement – TL Bus



<u>The models are stimulated and evaluated with:</u>

- Scenario based tests
- Constrained Random testing (you don't know what you don't know)
- X% Good Machine Path
- Y% Bad Machine Path
- Y >> X

Refinement – Pin Level Bus



<u>The models are stimulated and evaluated with:</u>

- Scenario based tests
- Constrained Random testing (you don’t know what you don’t know)
- X% Good Machine Path
- Y% Bad Machine Path
- Y >> X

Outline of the Talk & Progress Toward the Goals

Purpose: Convince you of the new capabilities of standards and standard-based tools for assembling, simulating, and validating (heterogeneous) components and interconnections

- □ Structure and Resources: IP-XACT
- □ Verification (+ coverage): UVM
- □ Power Management: UPF
- □ Quality Assessment: QIP
- □ Traceability: Soft & Hard Tags
- □ Additional Semantics and constraints
- □ Conclusions and Questions

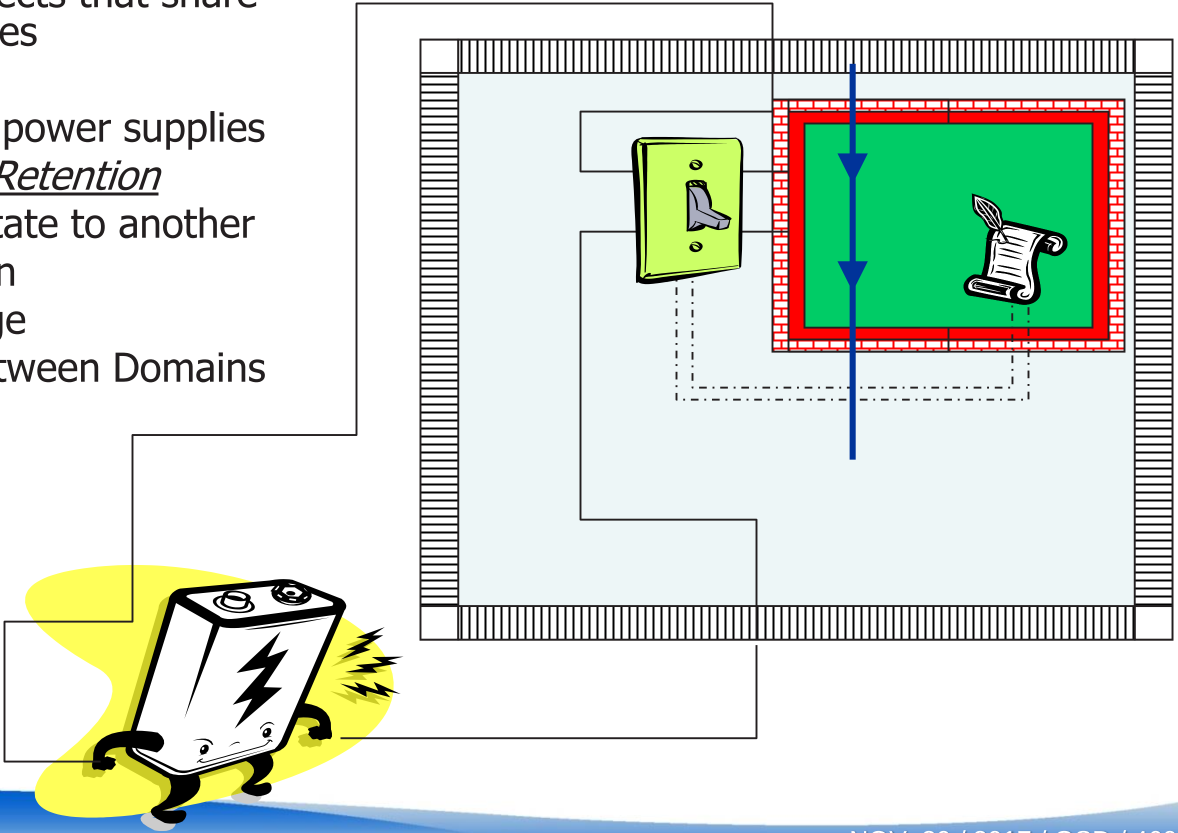
Goals of Structured Interconnections

Useful
Functional
Logical
Physical
Correct
Consistent
Easier

Verify-able
Build-able
Test-able
Abstract-able
Reuse-able
Transport-able
Implement-able

1801/UPF Supports Power Management Capabilities

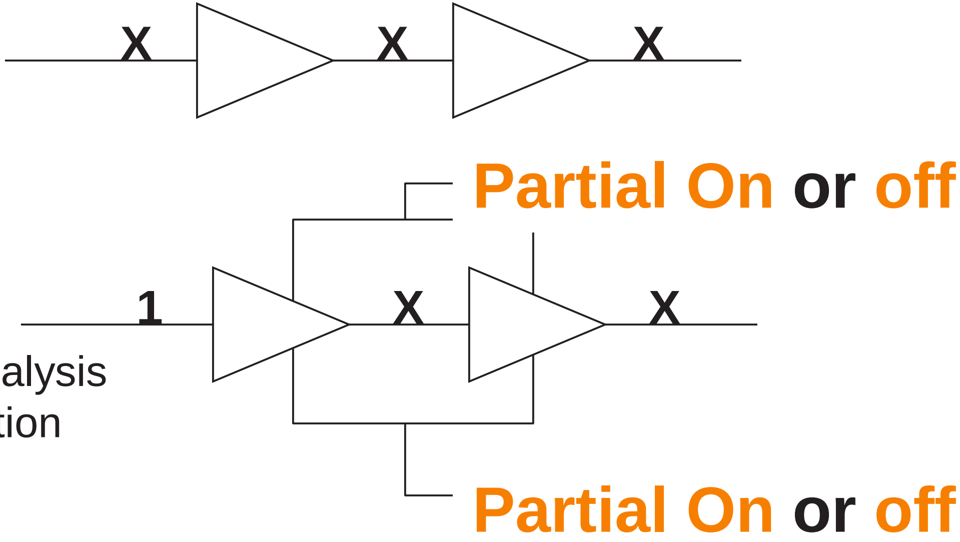
- *Power Domain*
  - Collection of design objects that share common power attributes
- Power *States*
  - Determined by state of power supplies
  - Memories may require *Retention*
  - Sequencing from one state to another
  - Semantics for simulation
  - DVFS, Bias, Multi-voltage
- Relations & Connections between Domains
  - *Level shifters*
  - *Isolation* logic
  - "Gas Stations" alternate supply
- Consistent implementation & verification semantics



Gary Delp, Erich Marschner and Kenneth Bakalar, February 2010 "Understanding the Low Power Abstraction", DVCON

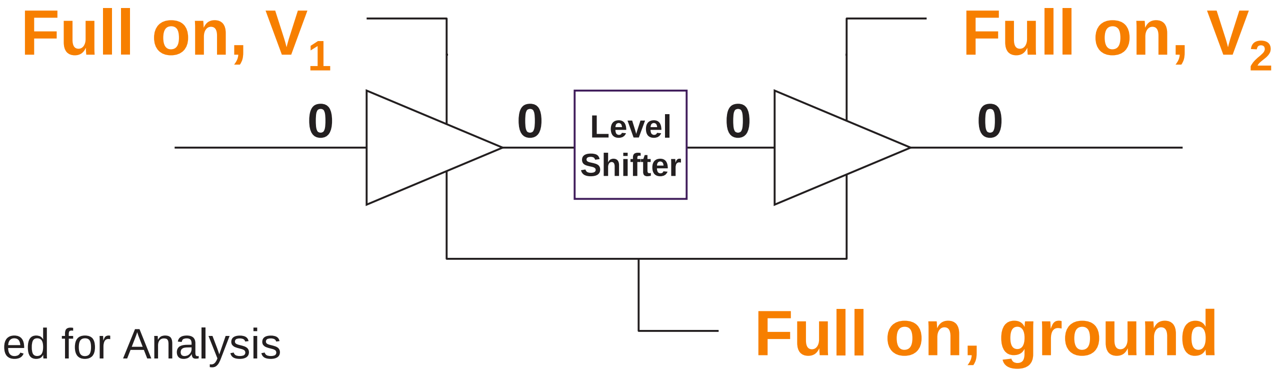
The concept of corruption – supply *Off* (or *Partially On*)

- Supply Nets
  - Net_State
    - Full_on
    - Partial_on
    - Off
  - Voltage
    - May be specified for Analysis and Time based corruption

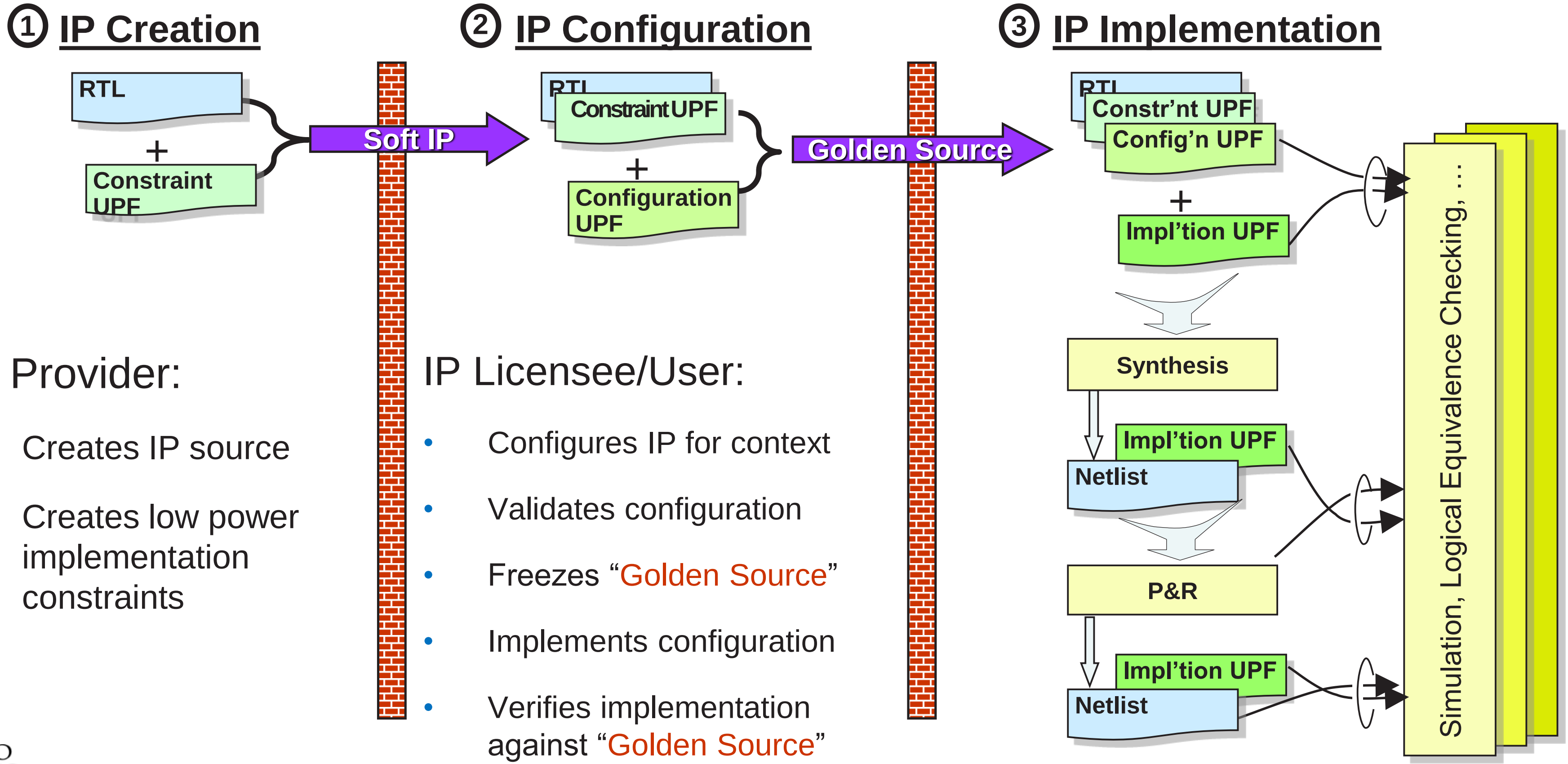


Protection from corruption Electrical – Level Shifting

- Supply Nets
  - Net_State
    - Full_on
    - Partial_on
    - Off
  - Voltage
    - May be specified for Analysis and Time based corruption



Successive Refinement of Low Power Intent



IP Provider:

- Creates IP source
- Creates low power implementation constraints

IP Licensee/User:

- Configures IP for context
- Validates configuration
- Freezes "Golden Source"
- Implements configuration
- Verifies implementation against "Golden Source"

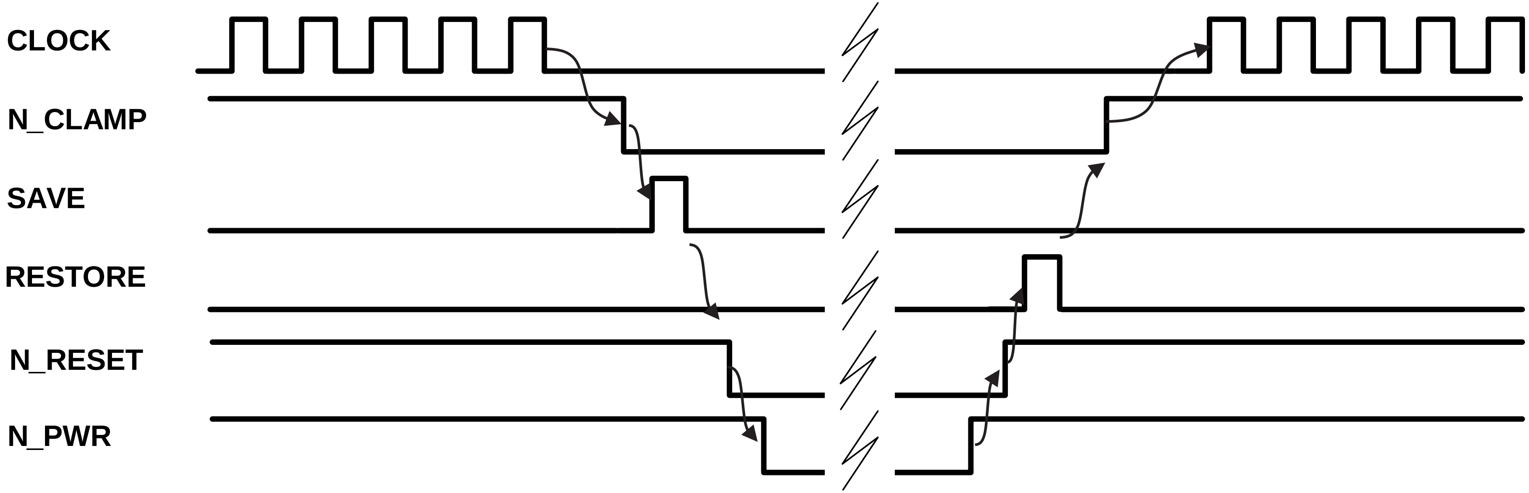
Sequencing Clocks, Resets and Power

Power down sequence:

1. Stop the clocks
2. Apply isolation
3. Optionally save state
4. Assert reset
5. Remove power

Power up sequence:

1. Apply power
2. Remove reset
3. Optionally restore state
4. Remove isolation
5. Start the clocks



Retention Power State Table For Balloon-style Latches

VDD	VDD RET	SS & SC	RS & RC	RTC	Retained value	Register value	Register state	Valid next states	Comments
ON	ON	FALSE	FALSE	FALSE	Previous saved data	Previous state value	NORMAL	SAVE, RESTORE	—
ON	ON	FALSE	FALSE	TRUE	Previous saved data	Previous state value	RETAIN_ON	NORMAL, RETAIN_OFF, RESTORE	—
ON	ON	FALSE	TRUE	X	Previous saved data	Retention value	RESTORE	NORMAL, RETAIN_ON	—
ON	ON	TRUE	FALSE	X	Register value	Previous state value	SAVE	NORMAL, RETAIN_ON	—
ON	ON	TRUE	TRUE	X	CORRUPT	CORRUPT	CORRUPT	NA	SAV_RES_COR is set
ON	OFF	X	X	TRUE	CORRUPT	CORRUPT	CORRUPT	NA	—
ON	OFF	X	TRUE	FALSE	CORRUPT	CORRUPT	CORRUPT	NA	RET_SUP_COR is set
ON	OFF	X	FALSE	FALSE	CORRUPT	Previous state value	PARTIAL_ CORRUPT	NORMAL	RET_SUP_COR is set
OFF	OFF	X	X	X	CORRUPT	CORRUPT	CORRUPT	NA	RET_SUP_COR is set
OFF	ON	FALSE	FALSE	FALSE	CORRUPT	CORRUPT	CORRUPT	NA	!RTC
OFF	ON	FALSE	FALSE	TRUE	Previous saved data	CORRUPT	RETAIN_OFF	RETAIN_ON	—
OFF	ON	FALSE	TRUE	X	CORRUPT	CORRUPT	CORRUPT	NA	Restore during power-down
OFF	ON	TRUE	X	X	CORRUPT	CORRUPT	CORRUPT	NA	Save during power-up

VDD – Primary Supply

VDD RET – Retention Supply

SS & SC – Save Signal & Save Condition

RS & RC – Restore Signal & Restore Condition

RTC – Retention Condition

NORMAL: Functional/active mode, all supplies ON.

SAVE: The time snapshot where the save action occurs (for balloon-latch style registers).

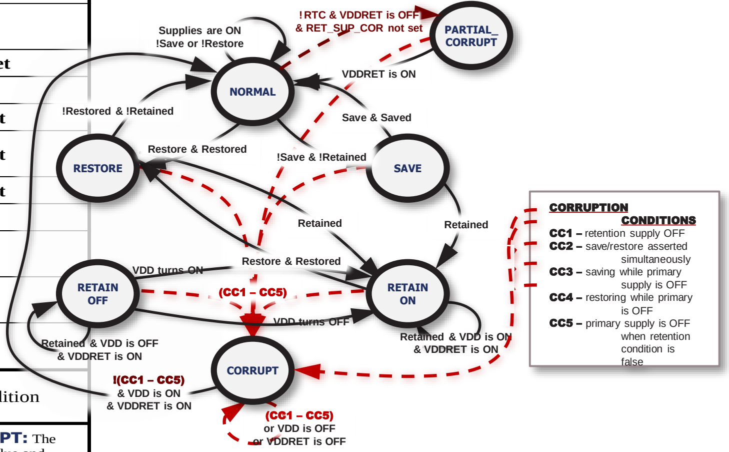
RESTORE: The time snapshot where the restore action occurs (for balloon-latch style registers).

RETAIN_ON: The time snapshot where the primary supply is ON and the register is in retention state (retention_condition == True).

RETAIN_OFF: The time snapshot where the primary supply is OFF and the register is in retention state (retention_condition == True).

PARTIAL_CORRUPT: The retained value is corrupted, but the register value is not corrupted.

CORRUPT: The register value and retained value are both corrupted.



CORRUPTION CONDITIONS

CC1 – retention supply OFF

CC2 – save/restore asserted simultaneously

CC3 – saving while primary supply is OFF

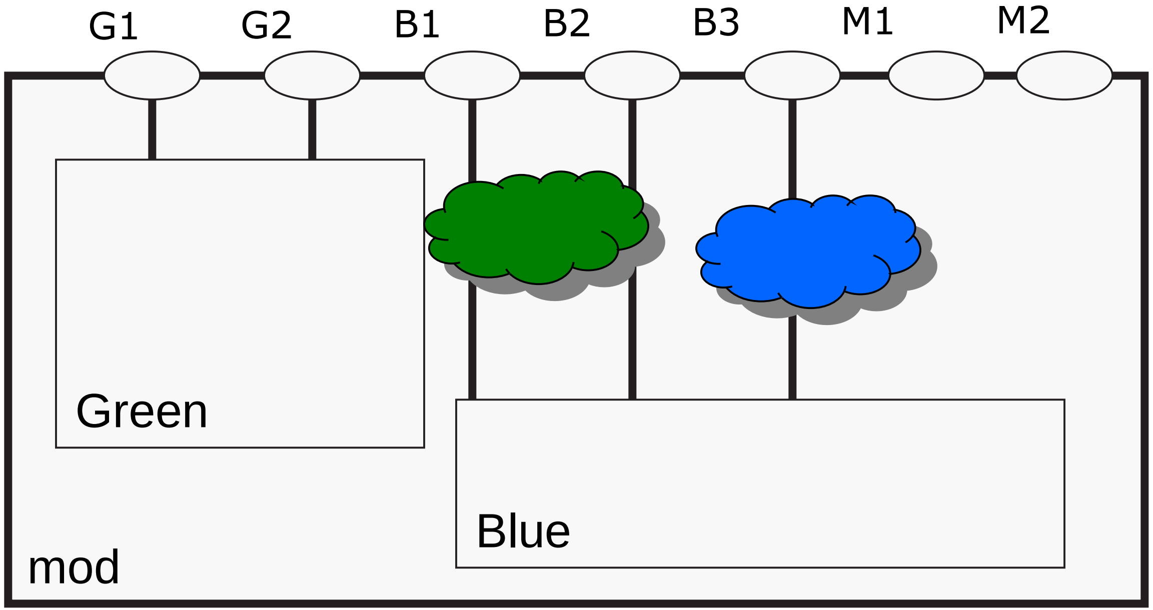
CC4 – restoring while primary is OFF

CC5 – primary supply is OFF when retention condition is false

Design Automation Standards Committee of the IEEE Computer Society: P1801 working group, (2015, 5 December). IEEE Standard for Design and Verification of Low-Power, Energy-Aware Electronic Systems, IEEE Std 1801-2015, IEEE Standards Association, Approved, Available: http://standards.ieee.org/getieee/1801/download/1801-2015.pdf p.201

The Power Portion of Robust Interfaces:

Designing within a context

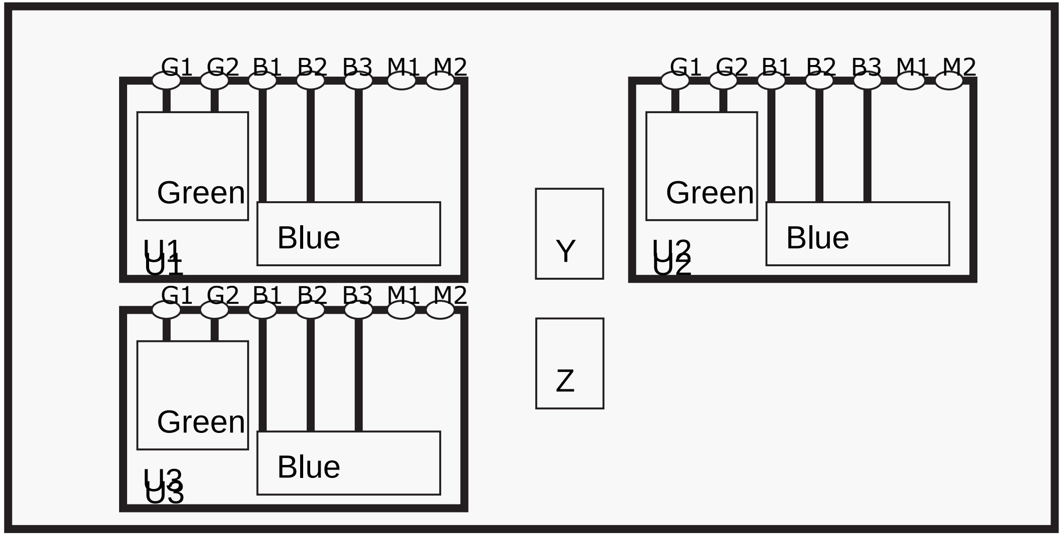


- UPF Commands: mod_if
- Create_power_domain mod_PD -include_scope
- Create_power_domain G
- Create_power_domain B
- Set_port_attributes
  -ports {G1, G2}
  -supply_set G.primary
- Set_port_attributes
  -ports {B1, B2, B3}
  -supply_set B.primary
- <Set_port_attributes
  -ports {M1, M2}
  -supply_set mod_pd.primary>
- mod_details:
- Create_power_domain G -update
  -elements {Green}
- Create_power_domain B-update
  -elements {Blue}

OK to Click Next

UPF 2.0 Supply sets and set_port_attributes

Using UPF 2.0 Components in a Design



UPF Commands: top

```
create_power_domain top_PD
    -include_scope
create_power_domain pd_G
    -elements {Y}
create_power_domain pd_B
    -elements {Z}
set base set_scope
foreach {el} {U1 U2 U3} {
    set_scope $base/$el
    load_upf “mod_if.upf”
}
set_scope $base

create_composite_domain topc_PD –subdomains
    {top_PD U1/mod_PD U2/mod_PD U3/mod_PD}
create_composite_domain pdc_G –subdomains
    {pd_G U1/G U2/G U3/B}
create_composite_domain pdc_B –subdomains
    {pd_B U1/B U2/B U3/G}
```

OK to Click Next

Outline of the Talk & Progress Toward the Goals

Purpose: Convince you of the new capabilities of standards and standard-based tools for assembling, simulating, and validating (heterogeneous) components and interconnections

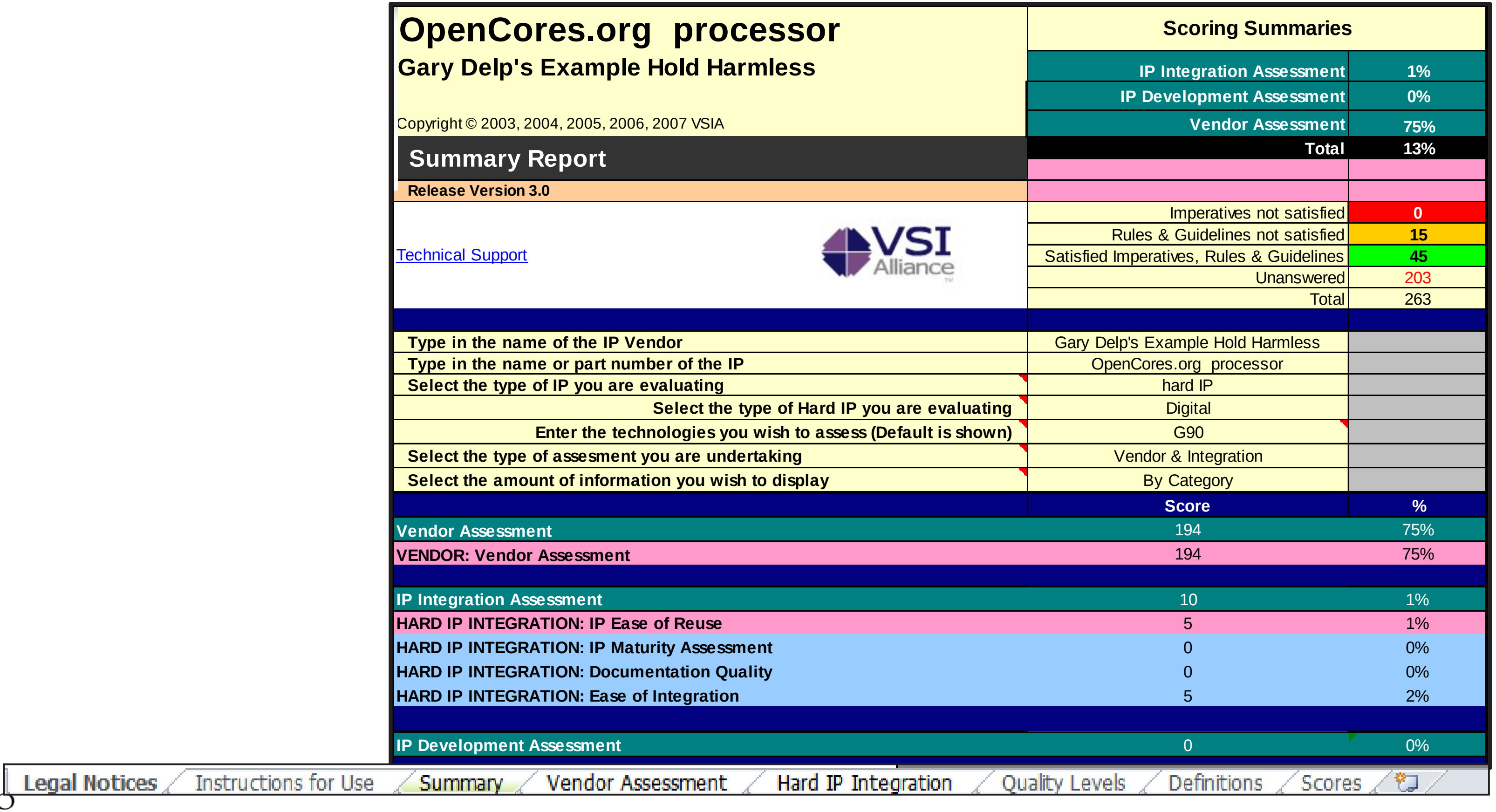
- □ Structure and Resources: IP-XACT
- □ Verification (+ coverage): UVM
- □ Power Management: UPF
- □ Quality Assessment: QIP
- □ Traceability: Soft & Hard Tags
- □ Additional Semantics and constraints
- □ Conclusions and Questions

Goals of Structured Interconnections

Useful
Functional
Logical
Physical
Correct
Consistent
Easier

Verify-able
Build-able
Test-able
Abstract-able
Reuse-able
Transport-able
Implement-able

An Example of a QIP Spreadsheet



OpenCores.org processor

Gary Delp's Example Hold Harmless

Summary Report

Release Version 3.0

Technical Support

Scoring Summaries	
IP Integration Assessment	1%
IP Development Assessment	0%
Vendor Assessment	75%
Total	13%
Imperatives not satisfied	0
Rules & Guidelines not satisfied	15
Satisfied Imperatives, Rules & Guidelines	45
Unanswered	203
Total	263

Type in the name of the IP Vendor	Gary Delp's Example Hold Harmless
Type in the name or part number of the IP	OpenCores.org processor
Select the type of IP you are evaluating	hard IP
Select the type of Hard IP you are evaluating	Digital
Enter the technologies you wish to assess (Default is shown)	G90
Select the type of assesment you are undertaking	Vendor & Integration
Select the amount of information you wish to display	By Category

	Score	%
Vendor Assessment	194	75%
VENDOR: Vendor Assessment	194	75%
IP Integration Assessment	10	1%
HARD IP INTEGRATION: IP Ease of Reuse	5	1%
HARD IP INTEGRATION: IP Maturity Assessment	0	0%
HARD IP INTEGRATION: Documentation Quality	0	0%
HARD IP INTEGRATION: Ease of Integration	5	2%
IP Development Assessment	0	0%

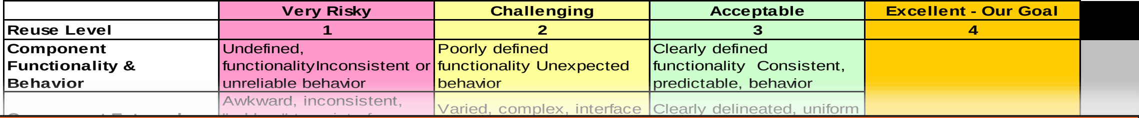
VSIA
Quality Metrics and Semantics

IEEE Std 1734

VSI Alliance™ QIP METRIC™

	Very Risky	Challenging	Acceptable	Excellent - Our Goal	
Reuse Level	**1**	**2**	**3**	**4**	
Component Functionality & Behavior	Undefined, functionalityInconsistent or unreliable behavior	Poorly defined functionality Unexpected behavior	Clearly defined functionality Consistent, predictable, behavior	Same as Level-3	Reusability - Characteristics
Component External Structure	Awkward, inconsistent, "ad hoc" type interface. May have multiple source baselines	Varied, complex, interface May have multiple source baselines	Clearly delineated, uniform interface Single source baseline		
Component Adaptability	Complex, non-existent or "improvised" configuration capabilities	Complex configuration capabilities with redundant or conflicting mechanisms	Structured, consistent configuration capabilities		
Component Packaging	Contains some items in Submission Checklist	Contains most items in Submission Checklist	Contains all items in Submission Checklist		
Documentation	Does not adhere to Reuse guidelines and templates. No consistency	May adhere to Reuse guidelines and templates Although may not adhere to Reuse guidelines and templates, docs. are consistent	Follows all Reuse guidelines and templates Consistent		
Knowledge of Component's Internal Logic	Expert, thorough knowledge of component's internal logic required in order to integrate/use component	Substantial knowledge of component's internal logic required in order to integrate/use component	Knowledge of component's internal logic not required in order to integrate/use component		
Support	Thorough, on-going training and discovery necessary. Constant (full-time) access to component developers necessary or full-time developer participation necessary	Detailed training necessary Access to component developers necessary	Minimal or no training necessary Access to component developers not necessary		
Requirements	Poorly focused requirements. Reusability not considered	Fundamental requirements identified Reusability may be a requirement	Fundamental requirements defined Reusability a requirement	Level-3 with the additional requirements that: a. All reuse procedures rigorously followed b. All deliverables adhere to reuse standards c. IP component can be rapidly extended and supported by other than the original development team	Component-Characteristics
Architecture & Design	Developed "ad hoc". No recognizable structured or complex, flat structure. Requirements not addressed	Developed with limited rigor or attention to process Minimal structure with mediocre documentation Unknown Requirements Adherence	Rigorously developed using defined process Documented, self-contained, layered structure Requirements satisfied		
Source code characteristics	Ad hoc structure. No apparent style. Sporadic comments	Minimal structure Inconsistent style and comments Sporadic comments	Well structured, follows design Standard style, well commented Consistent style adhering to a documented set of guidelines		
Test Plan & Verification software	Developed & implemented "ad hoc". Not documented. Test coverage is unknown	Developed & implemented with limited rigor or attention to process Insufficiently documented Test coverage is incomplete or questionable	Rigorously developed & implemented using same process and standards as component. Well documented Test coverage is complete		
Integrator's Score	**< 60%**	**60%**	**85%**	**95%**	
	Daunting / Very High Risk Excessive Reuse Costs	Challenging / Risky, High to Acceptable Reuse Cost	Efficient / Negligible RiskMinimal Reuse Cost		

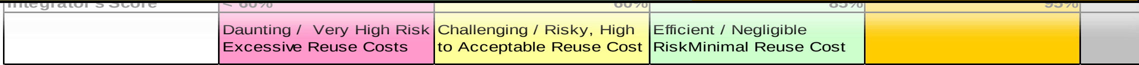
VSIA



	Very Risky	Challenging	Acceptable	Excellent - Our Goal
Reuse Level	1	2	3	4
Component Functionality & Behavior	Undefined, functionalityInconsistent or unreliable behavior	Poorly defined functionality Unexpected behavior	Clearly defined functionality Consistent, predictable, behavior	
	Awkward, inconsistent,	Varied, complex, interface	Clearly delineated, uniform	

	Very Risky	Challenging	Acceptable	Excellent Our Goal
Reuse Level	1	2	3	4
• • •	• • •	• • •	• • •	• • •
Architecture & Design	Developed "ad hoc". No recognizable structured or complex, flat structure. Requirements not addressed	Developed with limited rigor or attention to process Minimal structure with mediocre documentation Unknown Requirements Adherence	Rigorously developed using defined process Documented, self-contained, layered structure Requirements satisfied	**Level-3 with the additional requirements that:** **a. All reuse procedures rigorously followed** **b. All deliverables adhere to reuse standards** **c. IP component can be rapidly extended and supported by other than the original development team**
• • •	• • •	• • •	• • •	• • •

VSI Alliance™ QIP METRIC



Integrator's Score	< 60%	60%	85%	95%
	Daunting / Very High Risk Excessive Reuse Costs	Challenging / Risky, High to Acceptable Reuse Cost	Efficient / Negligible RiskMinimal Reuse Cost	

Outline of the Talk & Progress Toward the Goals

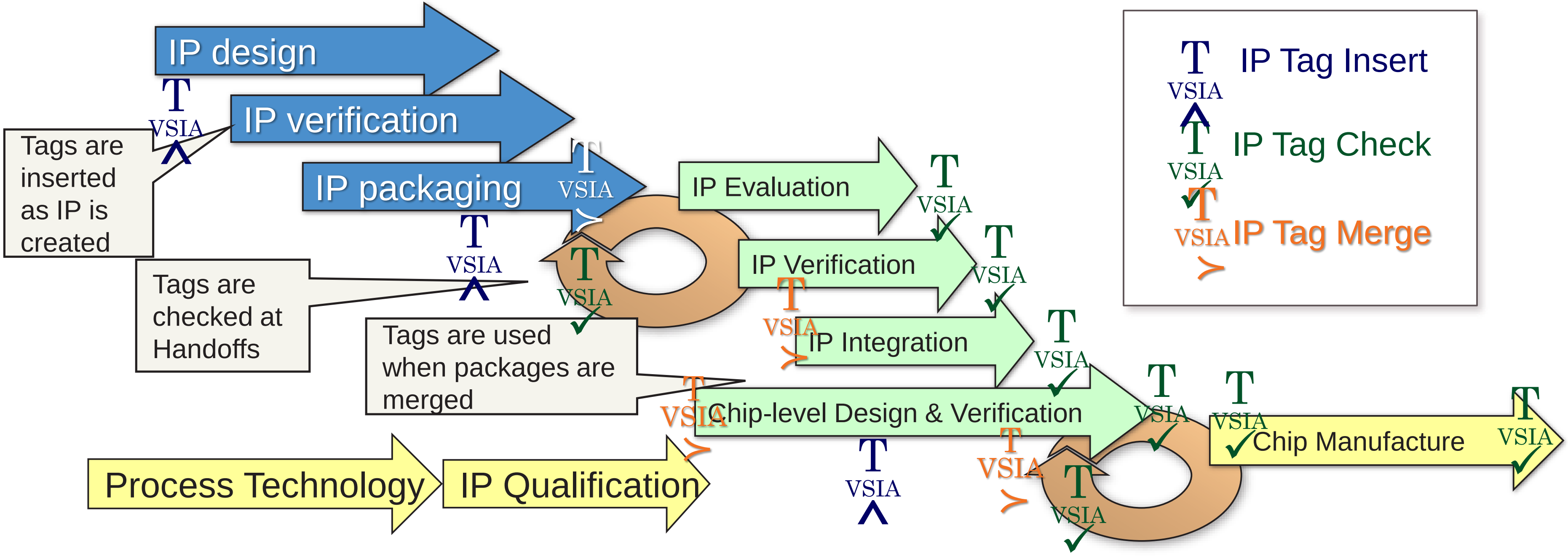
Purpose: Convince you of the new capabilities of standards and standard-based tools for assembling, simulating, and validating (heterogeneous) components and interconnections

- ❑ Structure and Resources: IP-XACT
- ❑ Verification (+ coverage): UVM
- ❑ Power Management: UPF
- ❑ Quality Assessment: QIP
- ❑ Traceability: Soft & Hard Tags
- ❑ Additional Semantics and constraints
- ❑ Conclusions and Questions

Goals of Structured Interconnections

Useful	Verify-able
Functional	Build-able
Logical	Test-able
Physical	Abstract-able
Correct	Reuse-able
Consistent	Transport-able
Easier	Implement-able

The VSIA IP ECO System – Tagging flow
Track the versions, licensing, and quality checks



2005 Illustration by Gary Delp, CTO VSIA

IEEE Standard for Quality of Electronic and Software Intellectual Property Used in System and System on Chip (SoC) Designs," in *IEEE Std 1734-2011*, Sept. 30 2011 doi: 10.1109/IEEESTD.2011.6035731

Outline of the Talk & Progress Toward the Goals

Purpose: Convince you of the new capabilities of standards and standard-based tools for assembling, simulating, and validating (heterogeneous) components and interconnections

- □ Structure and Resources: IP-XACT
- □ Verification (+ coverage): UVM
- □ Power Management: UPF
- □ Quality Assessment: QIP
- □ Traceability: Soft & Hard Tags
- □ Additional Semantics and constraints
- □ Conclusions and Questions

Goals of Structured Interconnections

Useful
Functional
Logical
Physical
Correct
Consistent
Easier

Verify-able
Build-able
Test-able
Abstract-able
Reuse-able
Transport-able
Implement-able

Standard Semantics, Constraints and Properties

- **IEEE Std 1685 IP-XACT**
  Sematic Consistency Rules (SCRs)
- B.1.1 Compatibility of busDefinitions
- A set of busDefinitions are considered compatible if they are related by the extends relationship as described in 5.11.1. A busDefinition is always compatible with itself.
- B.1.2 Interface mode of a bus interface
- Specifies which of the following exclusive subelements of the busInterface is present: master, slave, system, mirroredMaster, mirroredSlave, mirroredSystem, or monitor. For instance, a busInterface containing a system subelement is considered to have an interface mode of system.

- SCF Standard Constraint File

  VERB OBJECT OPTIONS SCOPE

  This form also used for UPF
- PSL Property Specification Language
  - Consistent semantics of property expression across VHDL, Verilog, TCL, SystemC, UVM

Outline of the Talk & Progress Toward the Goals

Purpose: Convince you of the new capabilities of standards and standard-based tools for assembling, simulating, and validating (heterogeneous) components and interconnections

- ❑ Structure and Resources: IP-XACT
- ❑ Verification (+ coverage): UVM
- ❑ Power Management: UPF
- ❑ Quality Assessment: QIP
- ❑ Traceability: Soft & Hard Tags
- ❑ Additional Semantics and constraints
- ❑ Conclusions and Questions

Goals of Structured Interconnections

Useful
Functional
Logical
Physical
Correct
Consistent
Easier

Verify-able
Build-able
Test-able
Abstract-able
Reuse-able
Transport-able
Implement-able

Conclusions

- The addition of the Low power Abstraction has enhanced the current abstraction hierarchy to treat with new design issues
- Corruption is a good thing
  - as long as you recognize it and use it for good
- Transactional abstractions still require the “reduction to simplicity”
- Initialization/Reset is not a “one-time-thing”
- Abstractions provide:
  - The ability to not think about things that you don’t want to think about
  - but not be in denial
- Standards are only useful if the enable new capabilties

Key Messages & Goals for the talk

Goals of Structured Interconnections

Useful
Functional
Logical
Physical
Correct
Consistent
Easy
Verify-able
Build-able
Test-able
Abstract-able
Reuse-able
Transport-able
Implement-able

Goals of the Talk

After the past 25 minutes do you now:

- Know about electronic design automation (EDA) standards as *providers of capability*?
  - IP-XACT IEEE Std 1685-2014
  - Unified verification methodology (UVM)
    - IEEE Std 1800.2
  - Power intent IEEE Std1801-2015
  - IP Quality IEEE Std 1734-2011
  - IP Tagging - VSIA => Accellera Standard
- Know where to find tools to successfully interconnect heterogeneous is systems?
- Know why you might care?

FIN