

Some Experiences Utilizing **IP-XACT** within the Verification Environment of a Real Project

28th, Apr 2008



freescaleTM
semiconductor

Authors from two companies



Bridget Hooser
Frank Donner
Michael Rohleder

Freescale Semiconductor
Microcontroller Systems Group
New Product Development Center
Munich, Germany



Bhawna Chopra
Mukesh Chopra
ST Microelectronics
Central CAD & Design Systems
Greater Noida, India

Giorgio Mastrorocco
ST Microelectronics
Automotive Products Group
Agrate, Italy



Contents

- Introduction
- Project Overview
 - Project Setup: JDP Overview
 - The Problem: Islands of Data
 - Single Source Approach: A Good Place to Start
 - What is IP-XACT?
 - IP-XACT: The Standard of Choice
- IP-XACT: Flow
- IP-XACT: Environment Development
- How do we describe IP data with IP-XACT?
 - What do we generate with IP-XACT?
 - How do we build SoC register description ?
- What is done different ...
- Findings: Benefits & Issues
- Summary & Conclusion
- Outlook

Introduction

This presentation describes our experiences made when:

- Developing a metadata structure for intellectual property (IP) utilizing the IP-XACT standard.
- Targeting this metadata as the basis for the verification of three real microcontroller product families developed within a Joint Development Program (JDP) between ST Microelectronics and Freescale Semiconductor.

Project Setup: JDP overview

The JDP (Joint Development Program) is a collaborative effort of *Freescale Semiconductor* and *ST-Microelectronics* on a global basis

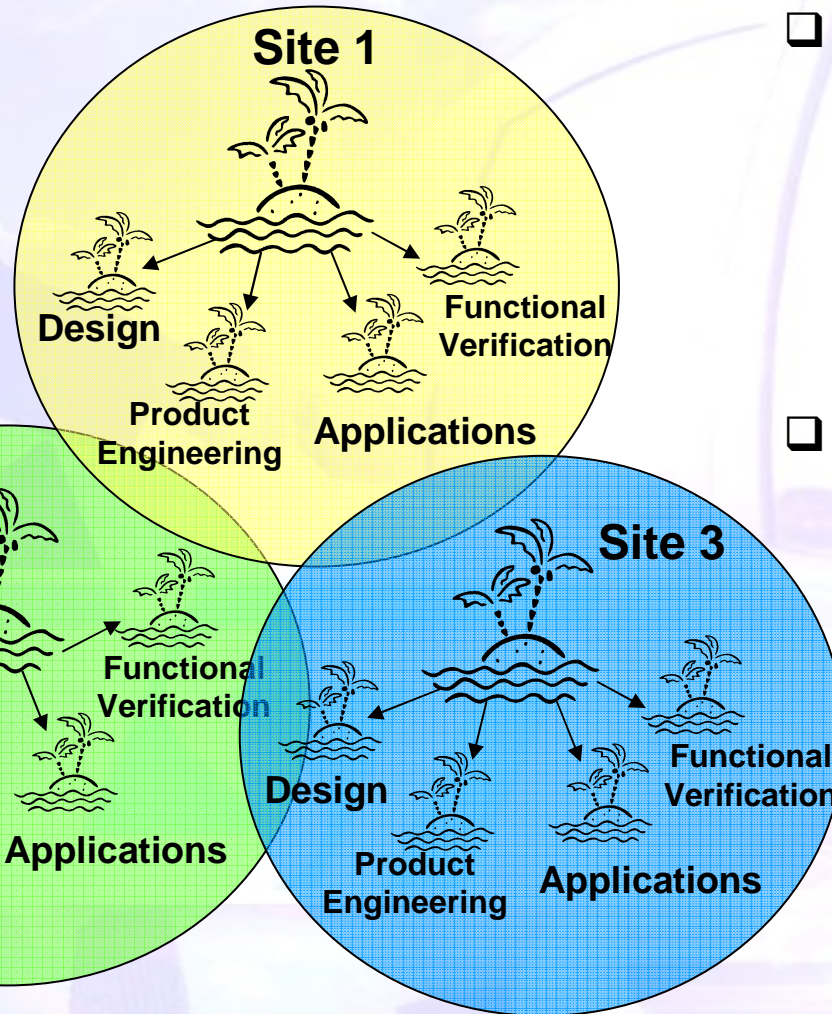
Headquarters
in Munich,
with several
development
sites
worldwide

Formed in 2006,
targets to
accelerate
innovation in the
automotive market

Devices will be
fabricated in
90nm technology,
using advanced
PowerPC cores

Headcount for both companies
is anticipated to reach 120 engineers

The Problem: Islands of Data



❑ Inconsistency of Data

- o Legacy IP: metadata not matching specification
- o Created by hand – room for mistakes
- o Lack of ability to convert and interchange data formats

❑ Metadata needs to be shared across a device family:

- o 2 companies
- o Several design locations
- o Different functional groups:
 - Design
 - Product Engineering
 - Applications
 - Functional Verification
 - Customers

Single Source Approach: A Good Place to Start

- **Single Source Approach can minimize previously described issues**
 - Relies on standard format.
 - Ensures uniformity of IP descriptions across the project.
 - Provides single interchangeable format for current and upcoming applications.
 - Provides a common flow for converting data and therefore allowing repeatability and comparability of the different data formats.
- **Single Source Approach can also create new risks**
 - Update of tools and scripts might be more difficult than editing data directly.
 - Flow concept must be understood and followed by everyone in the team.

What is **IP-XACT** ?

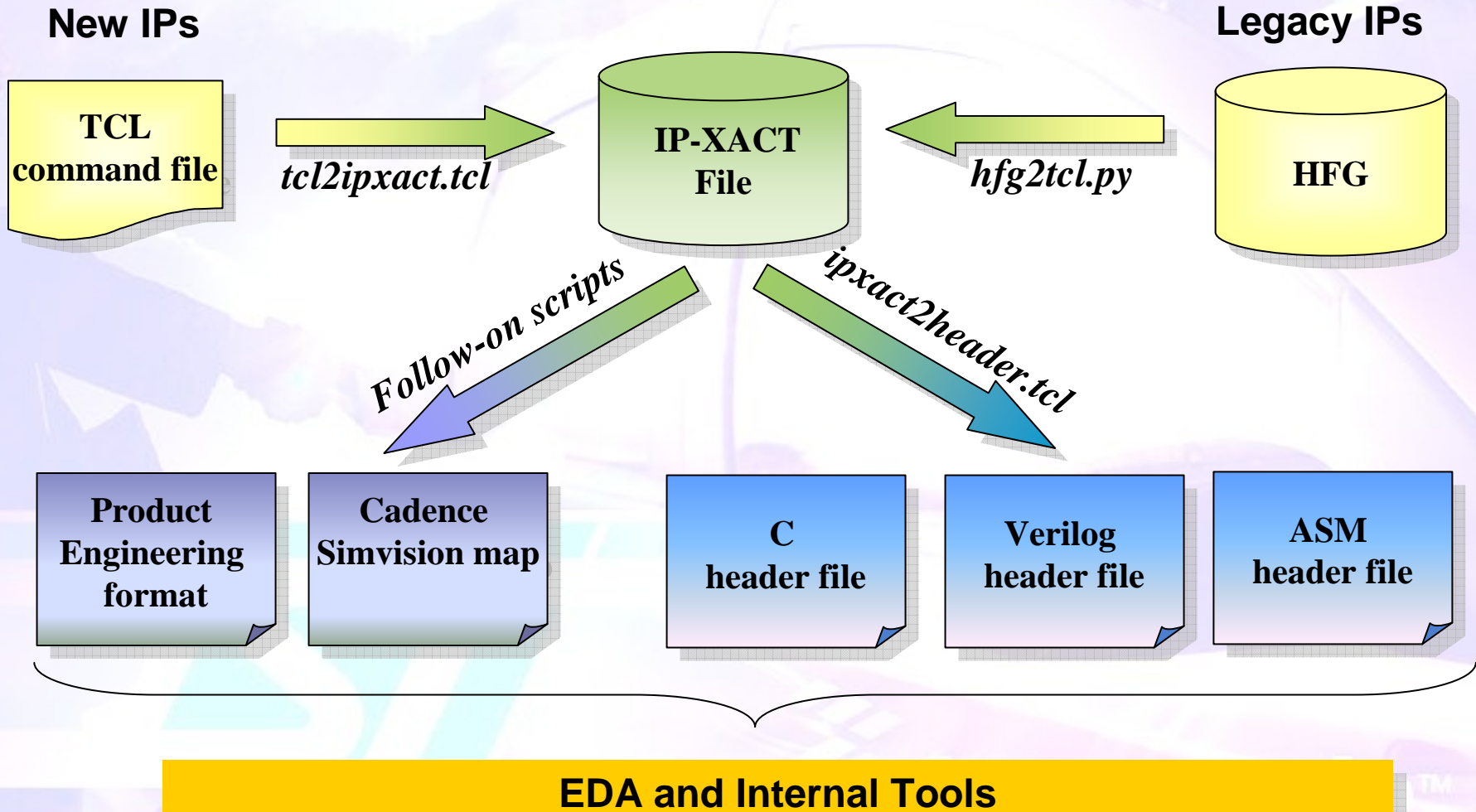
- **IP-XACT is a set of specifications created by The SPIRIT Consortium (www.spiritconsortium.org)**
 - SPIRIT stands for "*Structure for Packaging, Integrating and Re-using IP within Tool flows*".
 - Intended to be standardized as **IEEE Standard P1685**
- **IP-XACT specifies metadata intended to facilitate the exchange of EDA related information about intellectual property (IP)**
 - Metadata is “data about data”
 - In the context of IP-XACT metadata is “a tool interpretable way of describing the: design-history, locality, object association, configuration options, constraints against, and integration requirements of an IP object”
- **IP-XACT uses a XML Schema as Metadata specification**
 - The Extensible Markup Language (XML) is a simple, very flexible text format derived from SGML (standardized in ISO 8879)
 - Further information about XML can be found at <http://www.w3.org/XML>



IP-XACT: The Standard of Choice

- **IP-XACT is a mechanism for describing semiconductor IP enabling the automation of IP integration through the use of plug-in tools**
 - In the EDA industry there is no other standard intended to help with the integration of multi-vendor flows of tool and IP
- **In the software industry, the usage of XML for the exchange of metadata is already a widely accepted practice**
 - ASCII format (human readable) similar to HTML (SGML is common root)
 - A wealth of related tools and scripts is available
 - Many “open-source” implementations ease adoption
- **Required infrastructure is easy to obtain**
 - XML schema (IP-XACT specification)
 - XML parser (many open-source implementations)
 - Standard access functions which can be integrated in a common script languages to use them (many candidates, e.g. Tcl/Tk, PERL, Python)

IP-XACT: Flow



semiconductor

IP-XACT: Environment Development

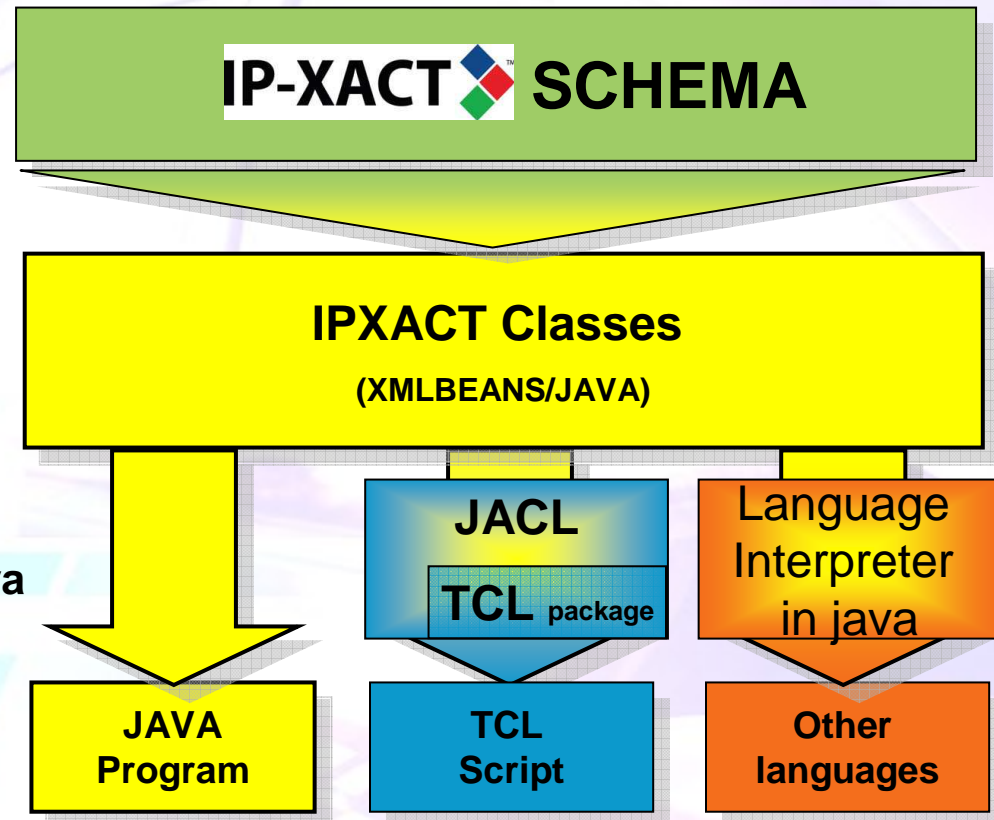
Released by SPIRIT Consortium

Compiled using xmlbeans

Common database

Use Java
or any language interpreter in java
add packages for ease

User written programs/scripts



- ipxactsh automatically built once
- User writes TCL scripts for using IP-XACT files

IP-XACT: Environment Development

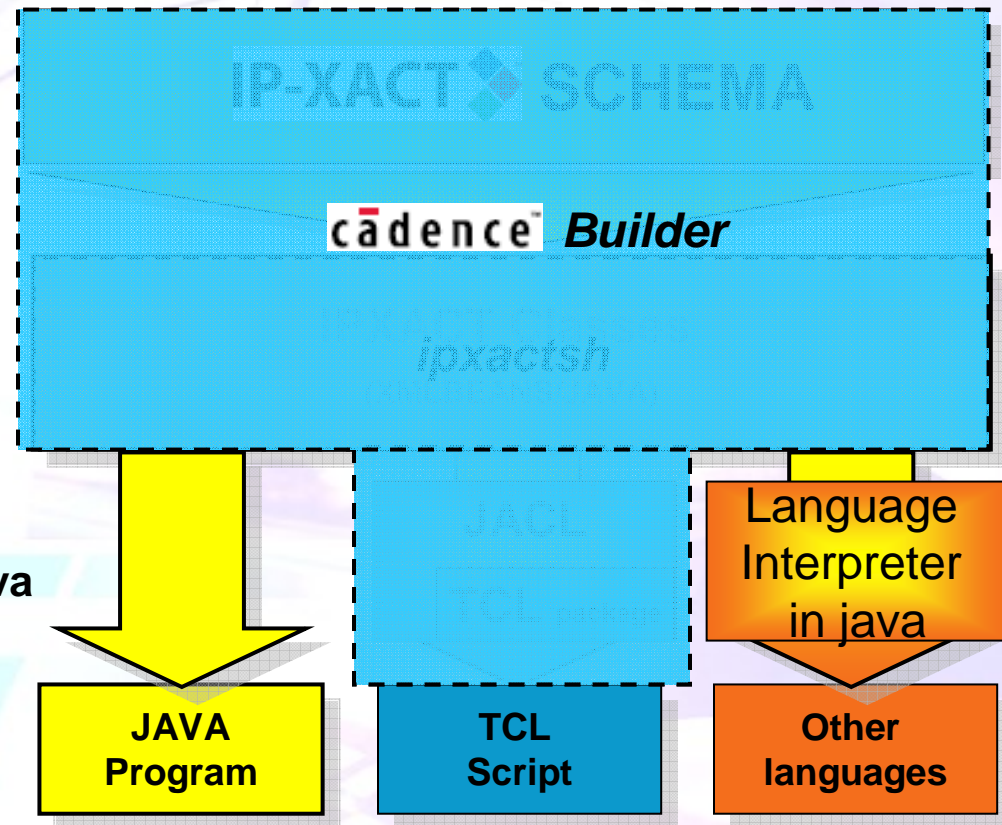
Released by SPIRIT Consortium

Compiled using xmlbeans

Common database

Use Java
or any language interpreter in java
add packages for ease

User written programs/scripts



~~ipxactsh automatically built once~~

User writes TCL scripts for using IP-XACT files

How do we describe IP data using IP-XACT ?

- Tcl based input format can be used at module and/or SoC level
 - Tcl-to-XML script is converting this input into central XML database (IP-XACT 1.2).
 - Input format and XML data are packaged along with IP deliverables on module level.
 - IP owners are responsible for data input.
- Once in XML – conversion of data into different formats is easy
- Database is part of verification environment, which is packaged and released to all sites in the JDP projects (Skeleton approach)
- Project specific data was added at the site owning a product

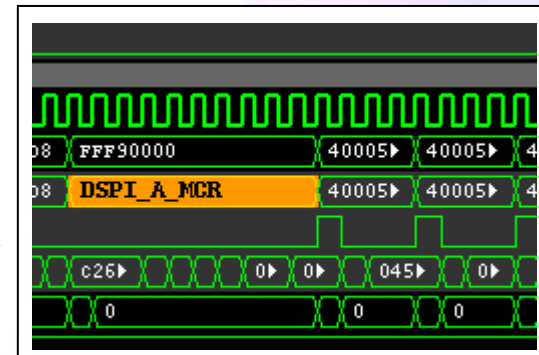
What do we generate with IP-XACT?

TCL Input format.

```
# DSPI Module Configuration Register (MCR)
add_register MCR 0x0000 32 rw 0x00004001 \
{MCR - Module Configuration Register} {
  add_field MSTR 31 1 rw \
  {Master/Slave mode select}
  ...
}
```

```
struct DSPI_A_tag {
  union {
    vuint32_t R;
    struct {
      vuint32_t DSPI_A_MSTR:1;
      ...
      vuint32_t:9;
      vuint32_t DSPI_A_HALT:1;
    } B;
  } MCR;
  /* MCR - Module Configuration Register */
}
```

Product Appl. Support



Single Source (IP-XACT XML)

```
<spirit:name>MCR</spirit:name>
<spirit:addressOffset>0x0000</spirit:addressOffset>
<spirit:size>32</spirit:size>
<spirit:access>read-write</spirit:access>
<spirit:reset>
  <spirit:value>0x00004001</spirit:value>
</spirit:reset>
<spirit:field>
  <spirit:name>MSTR</spirit:name>
  <spirit:bitOffset>31</spirit:bitOffset>
  <spirit:bitWidth>1</spirit:bitWidth>
  <spirit:access>read-write</spirit:access>
  ...
</spirit:field>
```

```
# MCR - Module Configuration Register
.equ DSPI_A_MCR (DSPI_A_REGS_BASE+0x00000000)
```

ASM

```
/* MCR - Module Configuration Register */
#define DSPI_A_MCR (DSPI_A_REGS_BASE + 32h0000_0000)
```

SV

```
/* MCR - Module Configuration Register */
#define DSPI_A_MCR (*(vuint32_t *) (DSPI_A_BASEADDRESS+0x00000000))
```

C

How do we build SoC register description ?

- Once XML data are existing on module level, it is easy to build up SoC top-level description.
- Each product can setup its own SoC configuration (“prototype”)
- Module base addresses (“0xC3F8C000”)
- Multiple instances of a module type (“4”)
- Size of the module space (“0x4000”)
- Add additional information/documentation (“yes”)
- Define numbering scheme for multiple instances (“number” vs. “letter”)

Top-level header file description

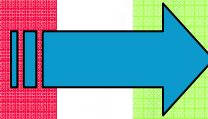
```
# Set Device name for header file
set dev_name prototype

# Module
set modules {
  {flash_shadow.xml 0x00200000 1 0x0000 yes }
  {flash_test.xml   0x00400000 1 0x0000 yes }
  {flash_internal_test.xml 0x00400000 1 0x0000 yes}
  {dflash_test.xml  0x00C00000 1 0x0000 yes}
  {dflash_internal_test.xml 0x00C00000 1 0x0000 yes}
  {flash.xml        0xC3F88000 1 0x0000 yes}
  {dflash.xml       0xC3F8C000 1 0x0000 yes}
  ...
  {linflex.xml      0xFFE40000 4 0x4000 yes number}
  {aips.xml         0xFFF00000 1 0x0000 yes}
  {swt_ips.xml      0xFFF38000 1 0x0000 yes}
  {stm_ips.xml      0xFFF3C000 1 0x0000 yes}
  {spp_mcm.xml      0xFFF40000 1 0x0000 yes}
  {spp_dma2x.xml    0xFFF44000 1 0x0000 yes}
  {intc_ipi.xml     0xFFF48000 1 0x0000 yes}
  {dsp_i.xml        0xFFF90000 3 0x4000 yes letter}
  {flexcan2_ipi.xml 0xFFFC0000 3 0x4000 yes letter}
  {dma_ch_mux.xml   0xFFFD0000 1 0x0000 yes}
}
```

What is done different ...

• How we used to do things

- Data entered multiple times by hand
 - > Not repeatable
 - > Error prone
 - > No commonality of data
 - > Disconnect between documentation & actual design



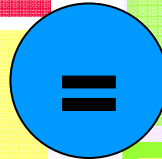
• Using IP-XACT now

- > Repeatable & more accurate...
- > Reuse across product families
- > Data is available earlier
- > Less impact when a problem is found downstream
- > Increased reusability

- Data generation by simple scripts
 - > Simple to create & maintain

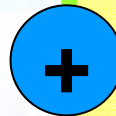
• Saved effort

- Maintenance of former scripts
 - > Usage of standard XML software
- Generation of control data needed for software driven verification
 - > C, ASM, Verilog related info
- Generation of an application header file for 3 product (families)
- Reuse of data by tool enablement



• New opportunities

- Enhanced capabilities enabled by easier available information
 - > E.g. Debugging support



• Combination with related data

- IP assembly flow – generated by forthcoming Cadence Builder tool



Findings: Benefits & Issues

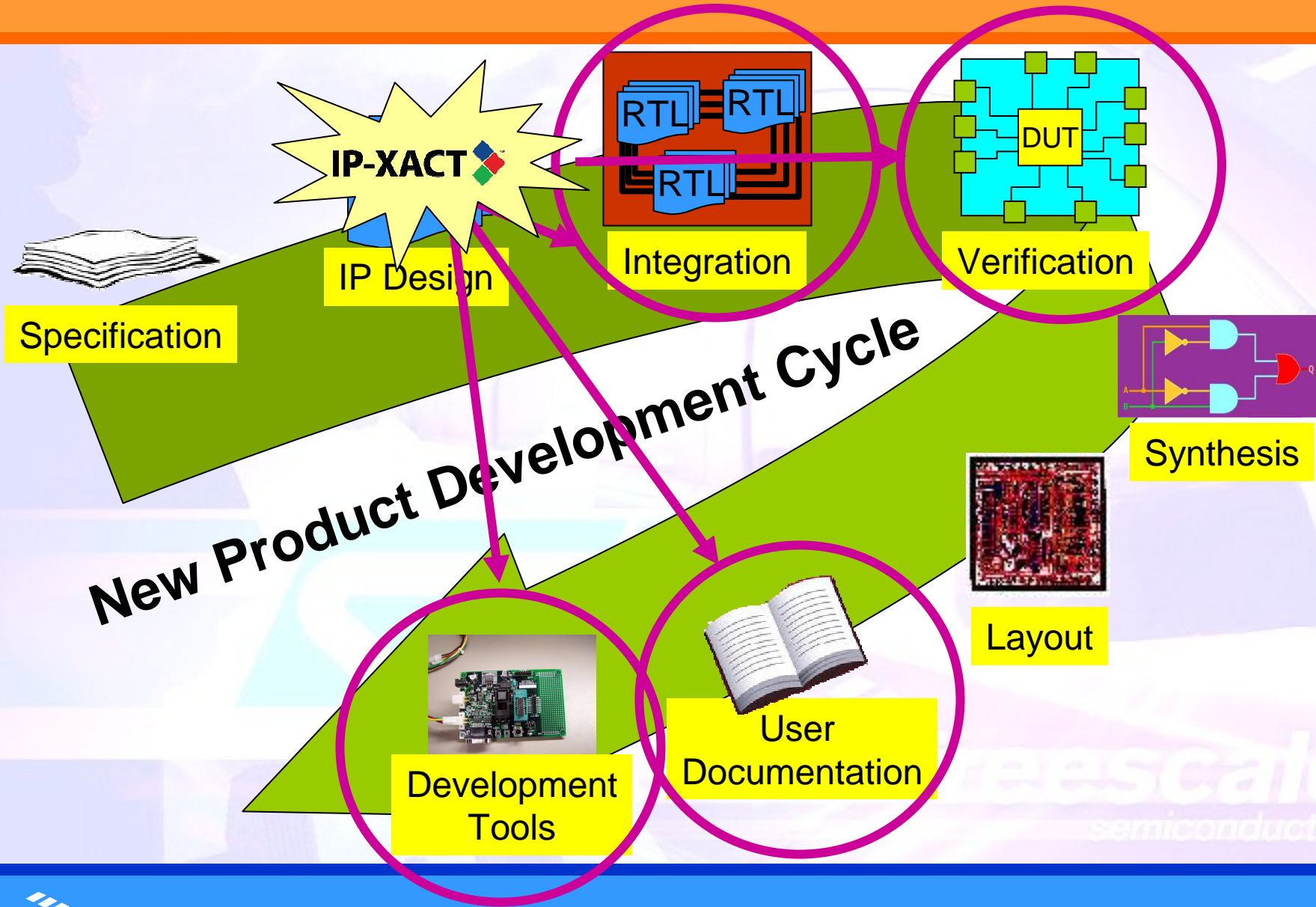
- ⊕ Significant information reuse
 - Ownership of data content ?
 - > Integration Team ?
 - > Documentation Team ?
 - > Systems or Applications ?
 - > IP Owner !!!
 - Data usage by experts coming from multiple application domains
 - ⊕ Consistency of information
 - ⊕ Quality of data content
 - ⊕ Different viewpoints can identify problems otherwise not found
- ⊖ Database management
 - > Collection of feedback, defects
 - > Synchronization of updates

- XML database
 - ⊕ Structured data, data coherency
 - ⊕ Standardized format permits further reuse by other groups
 - > E.g. Debugger enablement
 - ⊕ Available tool & script support
 - ⊖ Data larger, more complicated
 - ⊕ Overcomes disconnect between documentation and real world
 - > Reduces need for reviews
 - > Spec vs. Design vs. Docu
 - > Verification against ...
- ⊖ Distribution of related effort
 - > Bulk of creation effort at IP owner
 - > Bulk of benefit at IP consumer
 - > Mitigation by simpler input format

Summary & Conclusion

- We have shown the power of a database based on IP-XACT within the verification environment and its potential for further improvements.
- IP-XACT is a standard that can really change the industry and how we share data.
- In this particular project with its global scale & two different companies, having a common structure for the IP metadata was critical.
- Data shared throughout the world in different formats based on a single source (XML) severely helps to reduce errors.

Outlook



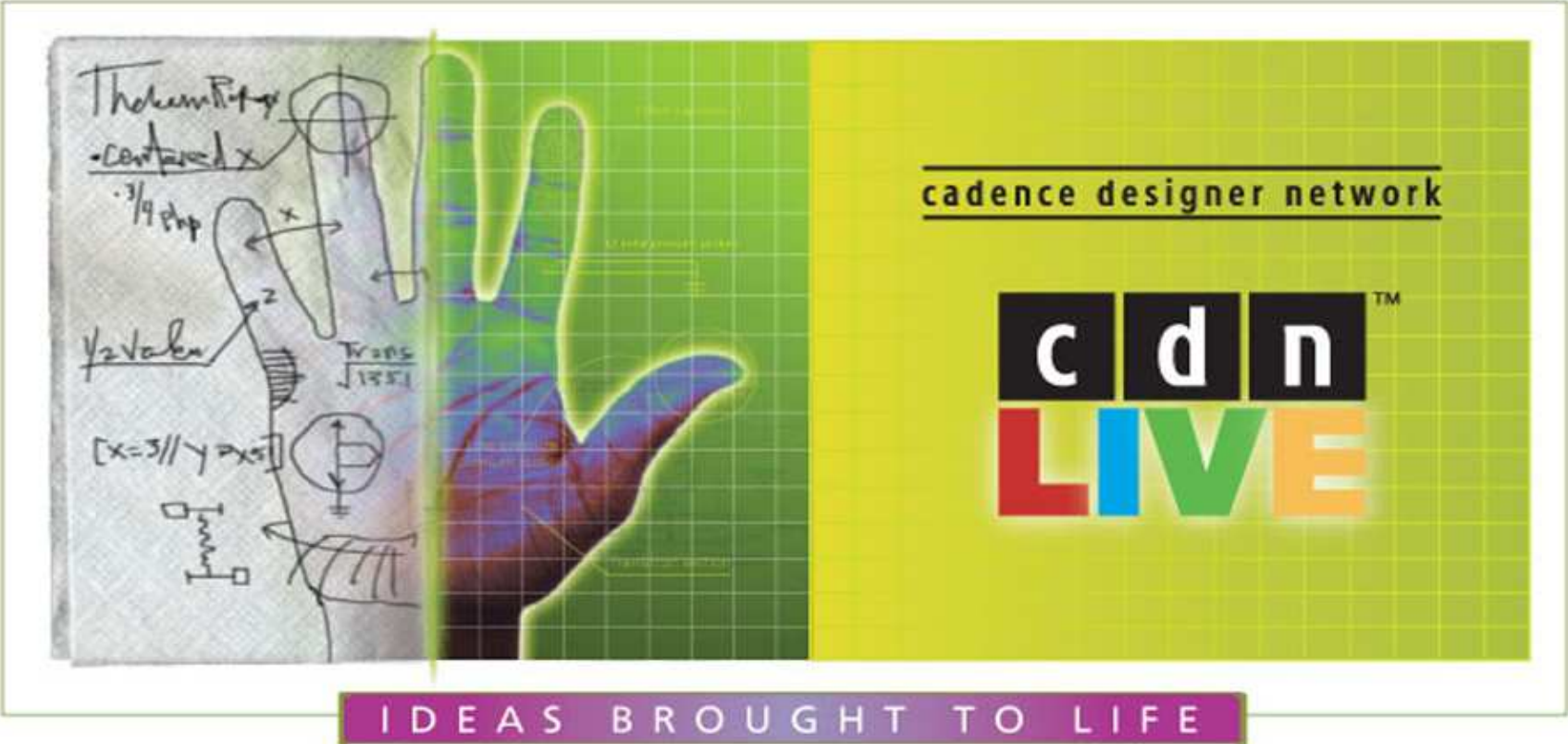
Outlook

IP-XACT  is a standard having the potential to be used throughout the whole product development cycle!

Thank you for your attention!



freescale™
semiconductor



The sketch on the left includes handwritten notes: "The center of gravity", "-center of mass", " $\frac{3}{4}$ step", " $\frac{1}{2}$ value", "[x=3//y=xs]", and "Trans 1981". It also contains a small circuit diagram with a resistor and a capacitor.

cadence designer network

cdn™
LIVE

IDEAS BROUGHT TO LIFE

